Unipro UGENE Manual

Version 35

June 17, 2020

Unipro UGENE User Manual v. 35

- Unipro UGENE User Manual

Unipro UGENE User Manual

- About Unipro
- About UGENE
  - Key Features
  - User Interface
  - High Performance Computing
  - Cooperation
- Download and Installation
  - System Requirements
  - Installation on Windows
  - Installation on macOS
  - Installation on Linux
  - Data Analysis Tools
  - Reference Data for NGS
    - Configure Data for Metagenomics Classification
    - Configure ChIP-Seq Analysis Data
- Basic Functions
  - UGENE Terminology
  - UGENE Window Components
    - Start Page
    - Project View
    - Task View
    - Log View
    - Notifications
  - Main Menu Overview
  - Creating New Project
  - Creating Document
  - Renaming Object
  - Opening Document
    - Opening for the First Time
      - Advanced Dialog Options
    - Opening Document Present in Project
    - Opening Several Documents
    - Opening Unloaded Documents
    - Opening from Clipboard
  - Opening Containing Folder
  - Exporting Documents
  - Locked Documents
  - Using Objects and Object Views
  - Exporting Objects
    - Exporting Sequences to Sequence Format
    - Exporting Sequences as Alignment
    - Exporting Alignment to Sequence Format
    - Exporting Nucleic Alignment to Amino Translation
    - Export Sequences Associated with Annotation
  - Using Bookmarks
  - Exporting Project
  - Search in Project
  - Options Panel
  - Plugins Viewer
  - Searching NCBI Genbank
  - Fetching Data from Remote Database
  - UGENE Application Settings
    - General
    - Resources
    - Network
    - File Format
    - Directories
    - Logging
    - Alignment Color Scheme
    - External Tools
      - Supported External Tools
      - Custom External Tools
    - Workflow Designer Settings
    - OpenCL
- Sequence View
  - Sequence View Components
  - Global Actions
  - Sequence Toolbars
  - Sequence Overview
  - Zoom View

- Details View
- Information about Sequence
- Manipulating Sequence
  - Show/Hide Sequence View Components
  - Showing Sequence in Multiple Lines
  - Translating Nucleotide Sequence
  - Zooming Sequence
  - Creating New Ruler
  - Going To Position
  - Selecting Sequence Region
  - Copying and Pasting Sequence
  - Editing Sequence
  - Searching in Sequence
  - Exporting Selected Sequence Region
  - Exporting Sequence of Selected Annotations
  - Exporting Sequence Image
  - Locking and Synchronize Ranges of Several Sequences
  - Multiple Sequence Opening
- Annotations Editor
  - "db_xref" Qualifier
  - Automatic Annotations Highlighting
  - The "comment" Annotation
- Manipulating Annotations
  - Creating Annotation
  - Selecting Annotations
  - Editing Annotation
  - Highlighting Annotations
    - Annotations Color
    - Annotations Visability
    - Show on Translation
    - Captions on Annotations
  - Creating and Editing Qualifier
  - Adding Column for Qualifier
  - Copying Qualifier Text
  - Finding Qualifier
  - Deleting Annotations and Qualifiers
  - Importing Annotations from CSV
  - Exporting Annotations
- Sequence View Extensions
  - Circular Viewer
    - Circular View Settings
  - 3D Structure Viewer
    - Opening 3D Structure Viewer
    - Changing 3D Structure Appearance
      - Selecting Render Style
      - Selecting Coloring Scheme
      - Calculating Molecular Surface
      - Selecting Background Color
      - Selecting Detail Level
      - Enabling Anaglyph View
    - Moving, Zooming and Spinning 3D Structure
    - Highlight Region on 3D Structure
    - Selecting Models to Display
    - Structural Alignment
    - Exporting 3D Structure Image
    - Working with Several 3D Structures Views
  - Chromatogram Viewer
    - Exporting Chromatogram Data
    - Viewing Two Chromatograms Simultaneously
  - Graphs Package
    - Description of Graphs
    - Graph Settings
    - Saving Graph Cuttoffs as Annotations
  - Dotplot
    - Creating Dotplot
    - Navigating in Dotplot
    - Zooming to Selected Region
    - Selecting Repeat
    - Interpreting Dotplot: Identifying Matches, Mutations, Invertions, etc.
    - Editing Parameters
    - Filtering Results
    - Saving Dotplot as Image
    - Saving and Loading Dotplot
    - Building Dotplot for Currently Opened Sequence
    - Comparing Several Dotplots
- Alignment Editor
  - Overview
    - Alignment Editor Features
    - Alignment Editor Components

- Navigation
- Coloring Schemes
  - Creating Custom Color Scheme
- Highlighting Alignment
- Zooming and Fonts
- Consensus
  - Export Consensus
- Alignment Overview
- Working with Alignment
- Selecting Alignment Region
- Moving Subalignment
- Copying and Pasting Subalignment
- Searching in Alignment
- Editing Alignment
  - Removing Selection
  - Filling Selection with Gaps
  - Replacing Selected Character
  - Replacing with Reverse-Complement
  - Replacing with Reverse
  - Replacing with Complement
  - Removing Columns of Gaps
  - Removing Sequence
  - Removing All Gaps
  - Undo and Redo Framework
- Saving Alignment
- Aligning Sequences
- Aligning Sequence to this Alignment
- Pairwise Alignment
- Working with Sequences List
  - Adding New Sequences
  - Renaming Sequences
  - Sorting Sequences
  - Shifting Sequences
  - Collapsing Rows
  - Copying Sequences
- Exporting in Alignment
  - Extracting Selected as MSA
  - Exporting Sequence from Alignment
  - Exporting Alignment as Image
- Importing APR and ACE Files
- Statistics
- Distance Matrix
- Grid Profile
- Advanced Functions
- Building HMM Profile
- Building Phylogenetic Tree
- PHYLIP Neighbor-Joining
- MrBayes
- PhyML Maximum Likelihood

- Phylogenetic Tree Viewer
- Tree Settings
- Selecting Tree Layout and View
- Modifying Labels Appearance
  - Showing and Hiding Labels
  - Aligning Labels
  - Changing Labels Formatting
- Adjusting Branch Settings
- Zooming Tree
- Working with Clade
- Selecting Clade
- Collapsing and Expanding Branches
- Swapping Siblings
- Zooming Clade
- Adjusting Clade Settings
- Changing Root
- Exporting Tree Image
- Printing Tree

- Sanger Reads Editor
- Sanger Reads Editor Overview
- Sanger Reads Editor Features
- Sanger Reads Editor Components
- Working with Chromatogram
- Mapping Reads to Reference
- Alignment Statistics
- Alignment Appearance
- Overview and Show and Hide Chromatogram
- Sanger Reads Consensus
  - Export Chromatogram Consensus
- Navigation in Sanger Reads Alignment

- Editing Sanger Reads
  - Inserting Character
  - Replacing Character and Gap
  - Removing Character and Gap
  - Inserting Gap
  - Removing Gap at the Left
  - Removing All Columns of Gaps
  - Trimming Left End
  - Trimming Right End
  - Renaming Read
  - Removing Read
  - Undo and Redo
- Exporting Alignment without Chromatograms

- Assembly Browser
  - Import BAM and SAM Files
  - Browsing and Zooming Assembly
    - Opening Assembler Browser Window
    - Assembly Browser Window
    - Assembly Browser Window Components
    - Reads Area Description
    - Assembly Overview Description
    - Ruler and Coverage Graph Description
    - Go to Position in Assembly
    - Using Bookmarks for Navigation in Assembly Data
  - Getting Information About Read
  - Short Reads Vizualization
    - Reads Highlighting
    - Reads Shadowing
  - Associating Reference Sequence
  - Associating Variations
  - Consensus Sequence
  - Exporting
    - Exporting Reads
    - Exporting Visible Reads
    - Exporting Coverage
    - Exporting Consensus
    - Exporting Consensus Variations
    - Exporting Assembly as Image
    - Exporting Assembly Region
  - Options Panel in Assembly Browser
    - Navigation in Assembly Browser
    - Assembly Statistics
    - Assembly Browser Settings
  - Assembly Browser Hotkeys
    - Assembly Overview Hotkeys
    - Reads Area Hotkeys
- Workflow Designer
  - About the Workflow Designer
  - Introduction
    - Launching Workflow Designer
    - Workflow Designer Window Components
    - Workflow Elements and Connections
    - Managing Parameters
    - UGENE Components and Workflow Designer
      - Task View, Notifications and Log View
      - Actions Menu
      - Toolbar
      - Context Menus
      - Application Settings
    - How to Create and Run Workflow
    - How to Use Sample Workflows
  - Manipulating Element
    - Adding Element
    - Copying Element
    - Pasting Element
    - Cutting Element
    - Deleting Element
    - Selecting All Elements on Scene
  - Manipulating Workflow
    - Creating New Workflow
    - Loading Workflow
    - Saving Workflow
    - Exporting Workflow as Image
    - Validating Workflow
    - Running Workflow
    - Dashboard
      - Dashboard Window Components
      - Using Dashboard
    - Stopping and Pausing Workflow

- Changing Appearance
- Custom Elements with Scripts
  - Functions Supported for Multiple Alignment Data
  - Functions Supported for Sequence Data
  - Functions Supported for Set of Annotations Data
  - Functions Supported for Files
  - Common Function
- Custom Elements with External Tools
  - Creating Element
  - Editing Element
  - Adding Existent Element
  - Removing Element
- Using Script to Set Parameter Value
- Running Workflow from the Command Line
- Running Workflow in Debugging Mode
  - Creating Breakpoints
  - Manipulating Breakpoints
- Workflow File Format
  - Header
  - Body
    - Elements
    - Dataflow
    - Metainformation
- Workflow Elements
  - Data Readers
    - Read Alignment Element
    - Read Annotations Element
    - Read FASTQ File with SE Reads Element
    - Read FASTQ Files with PE Reads Element
    - Read File URL(s) Element
    - Read NGS Reads Assembly Element
    - Read Plain Text Element
    - Read Sequence Element
    - Read Sequence from Remote Database Element
    - Read Variants Element
  - Data Writers
    - Write Alignment Element
    - Write Annotations Element
    - Write FASTA Element
    - Write NGS Reads Assembly Element
    - Write Plain Text Element
    - Write Sequence Element
    - Write Variants Element
  - Data Flow
    - Filter Element
    - Grouper Element
    - Multiplexer Element
    - Sequence Marker Element
  - Basic Analysis
    - Amino Translations Element
    - Annotate with UQL Element
    - CD-Search Element
    - Collocation Search Element
    - Export PHRED Qualities Element
    - Fetch Sequences by ID From Annotation Element
    - Filter Annotation by Name Element
    - Filter Annotations by Qualifier
    - Find Correct Primer Pairs Element
    - Find Pattern Element
    - Find Repeats Element
    - Gene-by-gene approach report
    - Get Sequences by Annotations Element
    - Group Primer Pairs Element
    - Import PHRED Qualities Element
    - Intersect Annotations Element
    - Local BLAST+ Search Element
    - Merge Annotations Element
    - ORF Marker Element
    - Remote BLAST Element
    - Sequence Quality Trimmer Element
    - Smith-Waterman Search Element
  - Data Converters
    - Convert bedGraph Files to bigWig Element
    - Convert Text to Sequence Element
    - File Format Conversion Element
    - Reverse Complement Element
    - Split Assembly into Sequences Element
  - DNA Assembly
    - Assembly Sequences with CAP3

- HMMER2 Tools
  - HMM2 Build Element
  - HMM2 Search Element
  - Read HMM2 Profile Element
  - Write HMM2 Profile Element
- HMMER3 Tools
  - HMM3 Build Element
  - HMM3 Search Element
  - Read HMM3 Profile
  - Write HMM3 Profile
- Multiple Sequence Alignment
  - Align Profile to Profile with MUSCLE Element
  - Align with ClustalO Element
  - Align with ClustalW Element
  - Align with Kalign Element
  - Align with MAFFT Element
  - Align with MUSCLE Element
  - Align with T-Coffee Element
  - Extract Consensus from Alignment as Sequence
  - Extract Consensus from Alignment as Text
  - In Silico PCR Element
  - Join Sequences into Alignment Element
  - Map to Reference Element
  - Split Alignment into Sequences Element
- NGS: Basic Functions
  - CASAVA FASTQ Filter Element
  - Cut Adapter Element
  - Extract Consensus from Assembly Element
  - Extract Coverage from Assembly Element
  - FASTQ Merger Element
  - FASTQ Quality Trimmer Element
  - FastQC Quality Control Element
  - Filter BAM/SAM Files Element
  - Genome Coverage Element
  - Improve Reads with Trimmomatic Element
  - Merge BAM Files Element
  - Remove Duplicates in BAM Files Element
  - Slopbed Element
  - Sort BAM Files Element
- NGS: ChIP-Seq Analysis
  - Annotate Peaks with peak2gene Element
  - Build Conservation Plot Element
  - Collect Motifs with SeqPos Element
  - Conduct GO Element
  - Create CEAS Report Element
  - Find Peaks with MACS Element
- NGS: Map/Assemble Reads
  - Assemble Reads with SPAdes Element
  - Map Reads with Bowtie Element
  - Map Reads with Bowtie2 Element
  - Map Reads with BWA Element
  - Map Reads with BWA-MEM Element
  - Map Reads with UGENE Genome Aligner Element
  - Map RNA-Seq Reads with TopHat Element
- NGS: Metagenomics Classification
  - Build CLARK Database
  - Build DIAMOND Database Element
  - Build Kraken Database Element
  - Classification Report Element
  - Classify Sequences with CLARK Element
  - Classify Sequences with DIAMOND Element
  - Classify Sequences with Kraken Element
  - Classify Sequences with MetaPhlAn2 Element
  - Ensemble Classification Data Element
  - Filter by Classification Element
  - Improve Classification with WEVOTE Element
- NGS: RNA-Seq Analysis
  - Assemble Transcripts with StringTie Element
  - Assembly Transcripts with Cufflinks Element
  - Extract Transcript Sequences with gffread Element
  - Merge Assemblies with Cuffmerge Element
  - StringTie Gene Abudance Report Element
  - Test for Diff. Expression with Cuffdiff Element
- NGS: Variant Analysis
  - Call Variants with SAMtools Element
  - Change Chromosome Notation for VCF Element
  - Convert SnpEff Variations to Annotations Element
  - Create VCF Consensus Element
  - SnpEff Annotation and Filtration Element

- Transcription Factor
  - Build Frequency Matrix Element
  - Build SITECON Model Element
  - Build Weight Matrix Element
  - Convert Frequency Matrix Element
  - Read Frequency Matrix Element
  - Read SITECON Model Element
  - Read Weight Matrix Element
  - Search for TFBS with SITECON Element
  - Search for TFBS with Weight Matrix Element
  - Write Frequency Matrix Element
  - Write SITECON Model Element
  - Write Weight Matrix Element
- Utils
  - DNA Statistics Element
  - Generate DNA Element

- Workflow Samples
  - Alignment
    - Align Sequences with MUSCLE
    - Extract Consensus as Sequence
    - Extract Consensus as Text
  - Conversions
    - Convert "seq/qual" Pair to FASTQ
    - Convert Alignments to ClustalW
    - Convert UQL Schema Results to Alignment
    - Convert Sequence to Genbank
  - Custom Elements
    - CASAVA FASTQ Filter
    - FASTQ Trimmer
    - Dump Sequence Info
    - LinkData Fetch
    - Quality Filter
  - Data Marking
    - Marking by Annotation Number
    - Marking by Length
  - Data Merging
    - Find Substrings in Sequences
    - Merge Sequences and Shift Corresponding Annotations
    - Search for TFBS
  - HMMER
    - Build HMM from Alignment and test it
    - Search Sequences with Profile HMM
  - NGS
    - ChIP-Seq Coverage
    - ChIP-seq Analysis with Cistrome Tools
    - Extract Consensus from Assembly
    - Extract Coverage from Assembly
    - Extract Transcript Sequences
    - Quality Control by FastQC
    - De novo Assemble Illumina PE Reads
    - De novo Assemble Illumina PE and Nanopore Reads
    - De novo Assemble Illumina SE Reads
    - De Novo Assembly and Contigs Classification
    - Parallel NGS Reads Classification
    - Serial NGS Reads Classification
    - RNA-Seq Analysis with TopHat and StringTie
    - RNA-seq Analysis with Tuxedo Tools
    - Variation Annotation with SnpEff
    - Call Variants with SAMtools
    - Variant Calling and Effect Prediction
    - Raw ChIP-Seq Data Processing
    - Raw DNA-Seq Data Processing
    - Raw RNA-Seq Data Processing
    - Get Unmappet Reads
  - Sanger Sequencing
    - Trim and Align Sanger Reads
  - Scenarios
    - Filter Sequence That Match a Pattern
    - Search for Inverted Repeats
    - Find Patterns
    - Gene-by-gene Approach for Characterization of Genomes
    - Group Primer Pairs
    - Intersect Annotations
    - Filter out Short Sequences
    - Merge Sequences and Annotations
    - In Silico PCR Sample
    - Remote BLASTing
    - Get Amino Translations of a Sequence
  - Transcriptomics

- Search for Transcription Factor Binding Sites (TFBS) in Genomic Sequences
- Query Designer
  - About the Query Designer
  - Query Designer Introduction
    - Launching the Query Designer
    - Terminology
      - Query Designer Window Components
      - Schema Elements
    - How to Create and Run Schema
  - Manipulating Query Designer Element
    - Adding Algorithm Element
    - Adding Constraint Element
    - Renaming Algorithm Element
    - Resizing and Moving Algorithm Element
    - Managing of Elements Parameters
    - Changing Algorithm Element Appearance
    - Deleting Query Designer Element
  - Manipulating Schema
    - Creating New Schema
    - Loading Schema
    - Saving Schema
    - Changing Schema Appearance
    - Setting Order of Algorithms Execution
    - Managing Strands
      - Element Direction in Schema
      - Querying Sequence Strands
    - Running Schema from the Query Designer
  - Running Schema from the Sequence View
  - Query Designer Schema File Format
    - Header Query Designer Element
    - Body Query Designer Element
      - Element Description
        - Algorithm Element Description
        - Constraint Element Description
      - Metainformation Query Designer Element
  - Query Elements
    - Algorithm Elements
      - CDD Algorithm Element
      - Base Content Algorithm Element
      - Gc Content Algorithm Element
      - HMM3 Algorithm Element
      - ORF Algorithm Element
      - Primer Algorithm Element
      - Repeats Algorithm Element
      - Restriction Sites Algorithm Element
      - Pattern Algorithm Element
      - SITECON Algorithm Element
      - Smith-Waterman Algorithm Element
      - Tandem Repeats Algorithm Element
      - Weight Matrix Algorithm Element
      - HMM2 Algorithm Element
    - Constraint Elements
      - End-Start Constraint Element
      - Start-End Constraint Element
      - End-End Constraint Element
      - Start-Start Constraint Element
- Extensions
  - DNA Annotator
  - DNA Flexibility
    - Configuring Dialog Settings
    - Result Annotations
  - DNA Statistics
  - DNA Generator
  - ORF Marker
  - Remote BLAST
    - Exporting BLAST Results to Alignment
    - Fetching Sequences from Remote Database
  - BLAST+
    - Creating Database
    - Making Request to Database
    - Fetching Sequences from Local BLAST+ Database
  - Repeat Finder
    - Repeats Finding
    - Tandem Repeats Finding
      - Tandem Repeats Search Result
  - Restriction Analysis
    - Selecting Restriction Enzymes
    - Using Custom File with Enzymes
    - Filtering by Number of Hits

- Excluding Region
- Circular Molecule
- Results
- Molecular Cloning in silico
  - Digesting into Fragments
  - Creating Fragment
  - Constructing Molecule
    - Available Fragments
    - Fragments of the New Molecule
    - Changing Fragments Order in the New Molecule
    - Removing Fragment from the New Molecule
    - Editing Fragment Overhangs
    - Reverse Complement a Fragment
    - Other Constuction Options
    - Output
  - Creating PCR Product
- In Silico PCR
  - Primers Details
  - Primer Library
- Secondary Structure Prediction
- SITECON
  - SITECON Searching Transcription Factors Binding Sites
  - Types of SITECON Models
    - Eukaryotic
    - Prokaryotic
  - Building SITECON Model
- Smith-Waterman Search
- HMM2
  - Building HMM2 Model
  - Calibrating HMM2 Model
  - Searching Sequence Using HMM2 Profile
- HMM3
  - Building HMM Model
  - Searching Sequence Using HMM Profile
  - Searching Sequence Against Sequence Database
- uMUSCLE
  - MUSCLE Aligning
  - Aligning Profile to Profile with MUSCLE
  - Aligning Sequences to Profile with MUSCLE
- ClustalW
- MAFFT
- T-Coffee
- Bowtie
  - Bowtie Aligning Short Reads
  - Building Index for Bowtie
- Bowtie 2
  - Bowtie 2 Aligning Short Reads
  - Building Index for Bowtie 2
- BWA
  - Aligning Short Reads with BWA
  - Building Index for BWA
- BWA-SW
  - Aligning Short Reads with BWA-SW
  - Building Index for BWA-SW
- BWA-MEM
  - Aligning Short Reads with BWA-MEM
  - Building Index for BWA-MEM
- UGENE Genome Aligner
  - Aligning Short Reads with UGENE Genome Aligner
  - Building Index for UGENE Genome Aligner
  - Converting UGENE Assembly Database to SAM Format
- CAP3
- SPAdes
- Weight Matrix
  - Searching JASPAR Database
  - Building New Matrix
- Primer3
  - RTPCR Primer Design
- Spliced Alignment mRNA and cDNA
- External Tools Plugin
  - Configuring External Tool
- Plasmid Auto Annotation
- ClustalO
- Kalign Aligning

- Shared Database
  - Configuring Database
  - Connecting to a Shared Database
  - Adding Data to the Database
  - Database in the Project

        - Deleting Data
        - Drag'n'drop in the Database
        - Exporting Objects from the Database
- UGENE Public Storage
- UGENE Command Line Interface
    - CLI Options
    - CLI Predefined Tasks
        - Format Converting Sequences
        - Converting MSA
        - Extracting Sequence
        - Finding ORFs
        - Finding Repeats
        - Finding Pattern Using Smith-Waterman Algorithm
        - Adding Phred Quality Scores to Sequence
        - Local BLAST+ Search
        - Remote NCBI BLAST and CDD Requests
        - Annotating Sequence with UQL Schema
        - Building Profile HMM Using HMMER2
        - Searching HMM Signals Using HMMER2
        - Aligning with MUSCLE
        - Aligning with ClustalW
        - Aligning with ClustalO
        - Aligning with Kalign
        - Aligning with MAFFT
        - Aligning with T-Coffee
        - Building PFM
        - Searching for TFBS with PFM
        - Building PWM
        - Searching for TFBS with Weight Matrices
        - Building Statistical Profile for SITECON
        - Searching for TFBS with SITECON
        - Fetching Sequence from Remote Database
        - Gene-by-Gene Report
        - Reverse-Complement Converting Sequences
        - Variants Calling
        - Generating DNA Sequence
    - Creating Custom CLI Tasks
- APPENDIXES
    - Appendix A. Supported File Formats
        - Specific File Formats
        - UGENE Native File Formats
        - Other File Formats

About Unipro

Established in 1992 Unipro company has its headquarters located in Novosibirsk Akademgorodok (the home of Siberian Branch of Russian Academy of Sciences). The company's primary activity is IT outsourcing solutions. To learn more about the company, please, visit the company website.

About UGENE

Unipro UGENE is a free cross-platform genome analysis suite. It is distributed under the terms of the GNU General Public License.

To learn more about UGENE visit UGENE website.

It works on Windows, macOS or Linux and requires only a few clicks to install.

- Key Features
- User Interface
- High Performance Computing
- Cooperation

Key Features

- Creating, editing and annotating **nucleic** acid and **protein** sequences
- Search through online databases: **NCBI**, **ENSEMBL**, **PDB**, **SWISS-PROT**, **UniProtKB/Swiss-Prot**, **UniProtKB/TrEMBL**, **UniProt(DAS)**, **Ensembl Human Genes (DAS)**
- Multiple sequence alignment: **ClustalW**, **ClustalO**, **MUSCLE**, **Kalign**, **MAFFT**, **T-Coffee**
- Online and local **BLAST+** search
- Restriction analysis with integrated **REBASE** restriction enzyme database
- Integrated **Primer3** package for **PCR** primers design
- Search for direct, inverted and **tandem repeats** in DNA sequences
- Constructing **dotplots** for nucleic acid sequences
- Search for transcription factor binding sites **(TFBS)** with **weight matrix** and **SITECON** algorithms
- Aligning short reads with **Bowtie**, **Bowtie 2**, **BWA, BWA-SW** and **UGENE Genome Aligner**

- Contig assembly with **CAP3**
- Search for **ORF**s
- **Cloning in silico**
- **3D structure viewer** for files in **PDB** and **MMDB** formats, anaglyph view support
- Protein secondary structure prediction with **GOR IV** and **PSIPRED** algorithms
- **HMMER2** and **HMMER3** packages integration
- Building (using integrated **PHYLIP** and **MrBayes** packages) and viewing **phylogenetic trees**
- Local sequence alignment with optimized **Smith-Waterman** algorithm
- Combining various algorithms into custom workflows with **UGENE Workflow Designer**
- Search for a pattern of various algorithms' results in a nucleic acid sequence with **UGENE Query Designer**
- Visualization of **next generation sequencing data** (**BAM** files) using **UGENE Assembly Browser**
- PCR in silico
- Spade de novo assembler

User Interface

- Visual and interactive genome browsing including circular plasmid view
- Multiple alignment editor
- Chromatograms visualization
- 3D viewer for files in PDB and MMDB formats with anaglyph stereo mode support
- Phylogenetic tree viewer
- Easy to use Workflow Designer for custom computational workflows
- Easy to use Query Designer for analyze a nucleotide sequence using different algorithms at the same time
- Assembly Browser for visualize and efficiently browsing large next generation sequence assemblies

High Performance Computing

- Complete support of modern multicore processors and SSE instructions
- Out of the box support of modern GPUs using NVIDIA CUDA and ATI Stream
- Integrated solutions for Cell Broadband Engine

Cooperation

- Can be used for education purposes in schools and universities
- Features to be included into the next release are initiated by users
- UGENE team is ready for collaboration in related projects, both free and commercial

Download and Installation

UGENE is compatible with the three most common operating systems: Windows, macOS, and Linux. It has some minimum system requirements. If your system fits these requirements, you're welcome to download UGENE from http://ugene.net/download. The program can be used and distributed under the terms of GPLv2.

Below you can also find links to the guides on UGENE installation on different operating systems.

- System Requirements
- Installation on Windows
- Installation on macOS
- Installation on Linux
- Data Analysis Tools
- Reference Data for NGS
  - Configure Data for Metagenomics Classification
  - Configure ChIP-Seq Analysis Data

System Requirements

Operating system

Since v. 34, 64-bit versions of the operating systems are supported only:

- Windows 8+
- macOS 10.7+
- Linux

Note that some external tools may be available only on particular operating systems, see details in the "Data Analysis Tools" chapter of the manual.

For older packages, if required, see http://ugene.net/download-all.html.

RAM

512 Mb RAM is required. At least 2 Gb RAM is recommended.

Disk space

The minimum required disk space depends on the installed UGENE components. At least 1 Gb is recommended. For installing all components, including NGS modules for ChIP-seq and metagenomic data processing, about 250Gb is required.

Display

It is recommended to set the screen resolution to 1280 x 720 or greater.

Internet

Internet connection is required for a few tasks like loading data from online databases.

UGENE takes care to use the capabilities of your system: the more RAM and cores you have, the more quickly you'll get results of your calculations.

Also, if you have an OpenCL-capable video card, you can use GPU-optimized versions of the following tools:

- Smith-Waterman Search
- UGENE Genome Aligner

Installation on Windows

The following UGENE package delivery types are available on Windows:

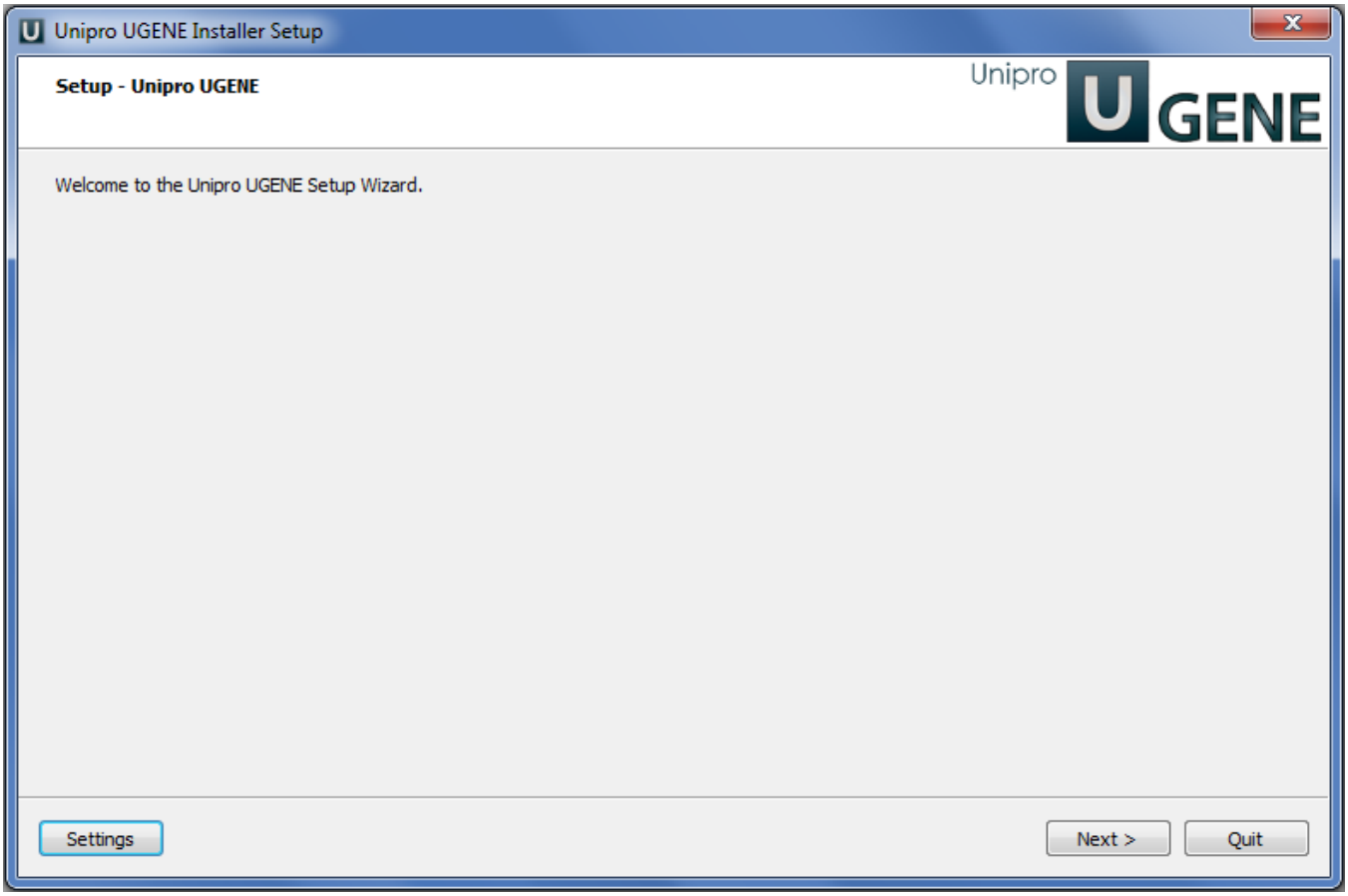
- **Online installer:** recommended, updates to new versions are supported.
- **Offline installer:** does not require Internet connection to be installed, an extra configuration is required for additional NGS components for ChIP-Seq and metagenomic data analysis.

Current UGENE version is available for Windows 8 and higher on 64-bit platform.

For other Windows versions older UGENE versions are available on the web page http://ugene.net/download-all.html.

Installation using online installer

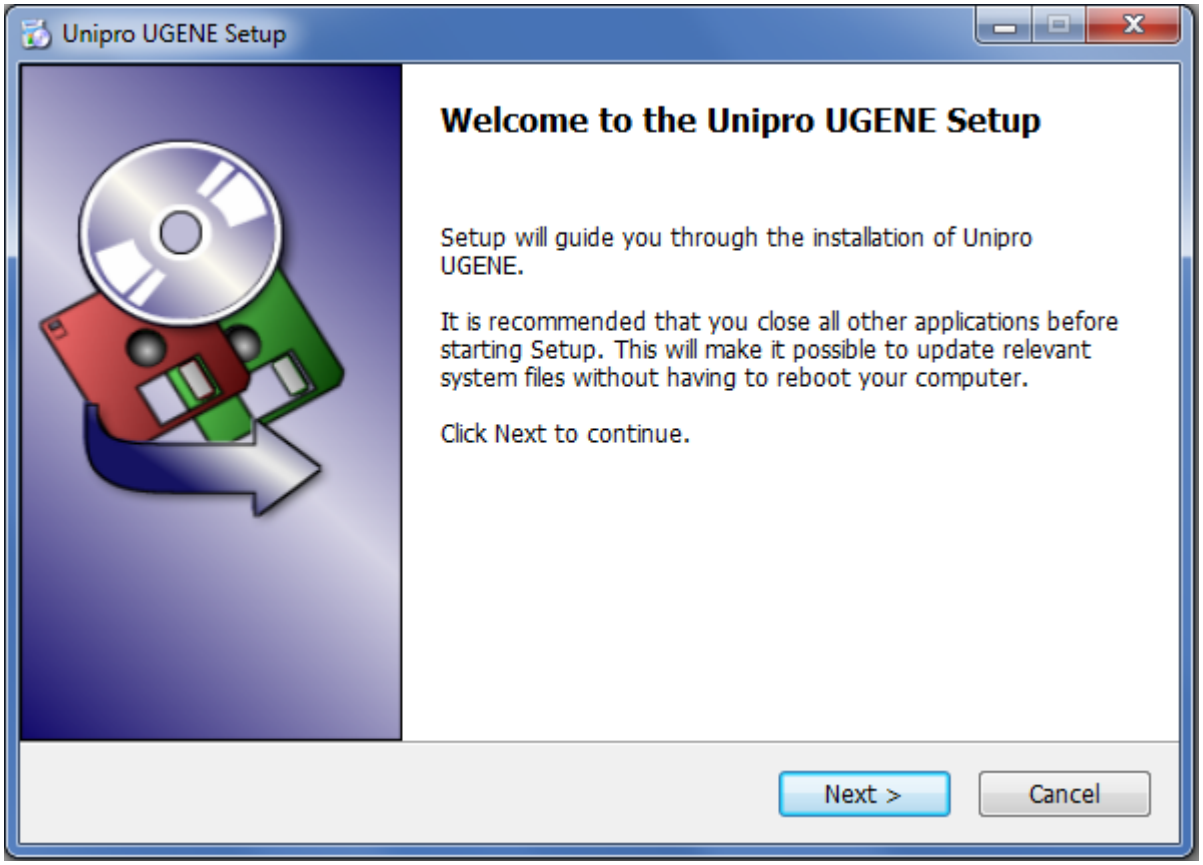
- Download the online installer executable file.
- When installing the package, make sure the Internet connection is available.
- Run the executable file with administrative rights.
- Follow the installation wizard.



On a new UGENE version release, a notification will appear with an option to update the package.

Installation using offline installer

- Download the offline installer executable file.
- Run the executable file with administrative rights.
- Follow the installation wizard.



On a new UGENE version release, a notification will appear. However, to install the new version, it is required to manually download the package again.

Installation on macOS

UGENE delivery type on macOS is a DMG installer. There are two options:

- **Online installer:** recommended, updates to new versions are supported
- **Offline installer:** does not require Internet connection to be installed

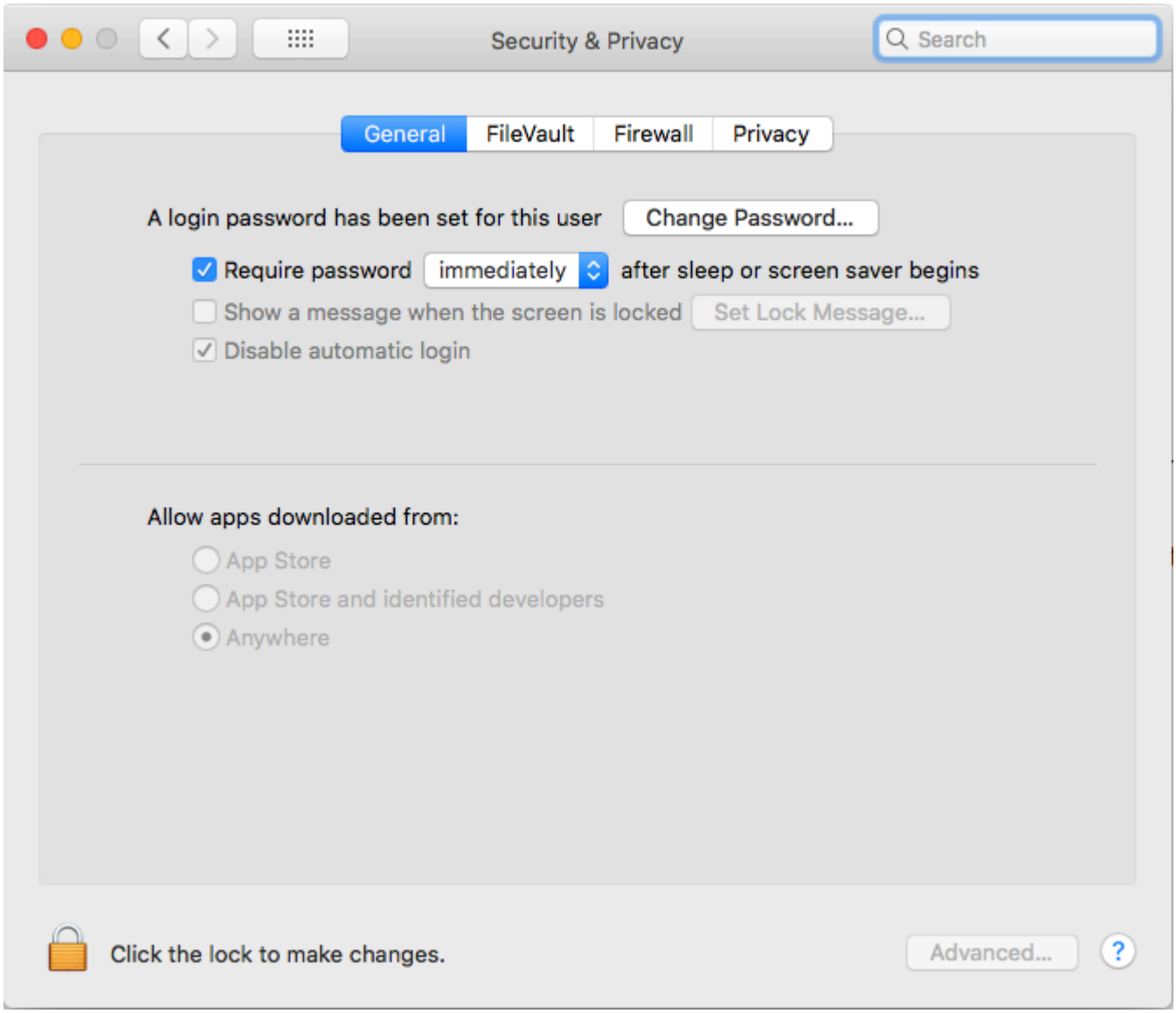
Current UGENE version is available for operating system version 10.7 or higher. Old UGENE packages are also provided for earlier OS versions.

All download links are available on the web page http://ugene.net/download-all.html.

Configure system preferences

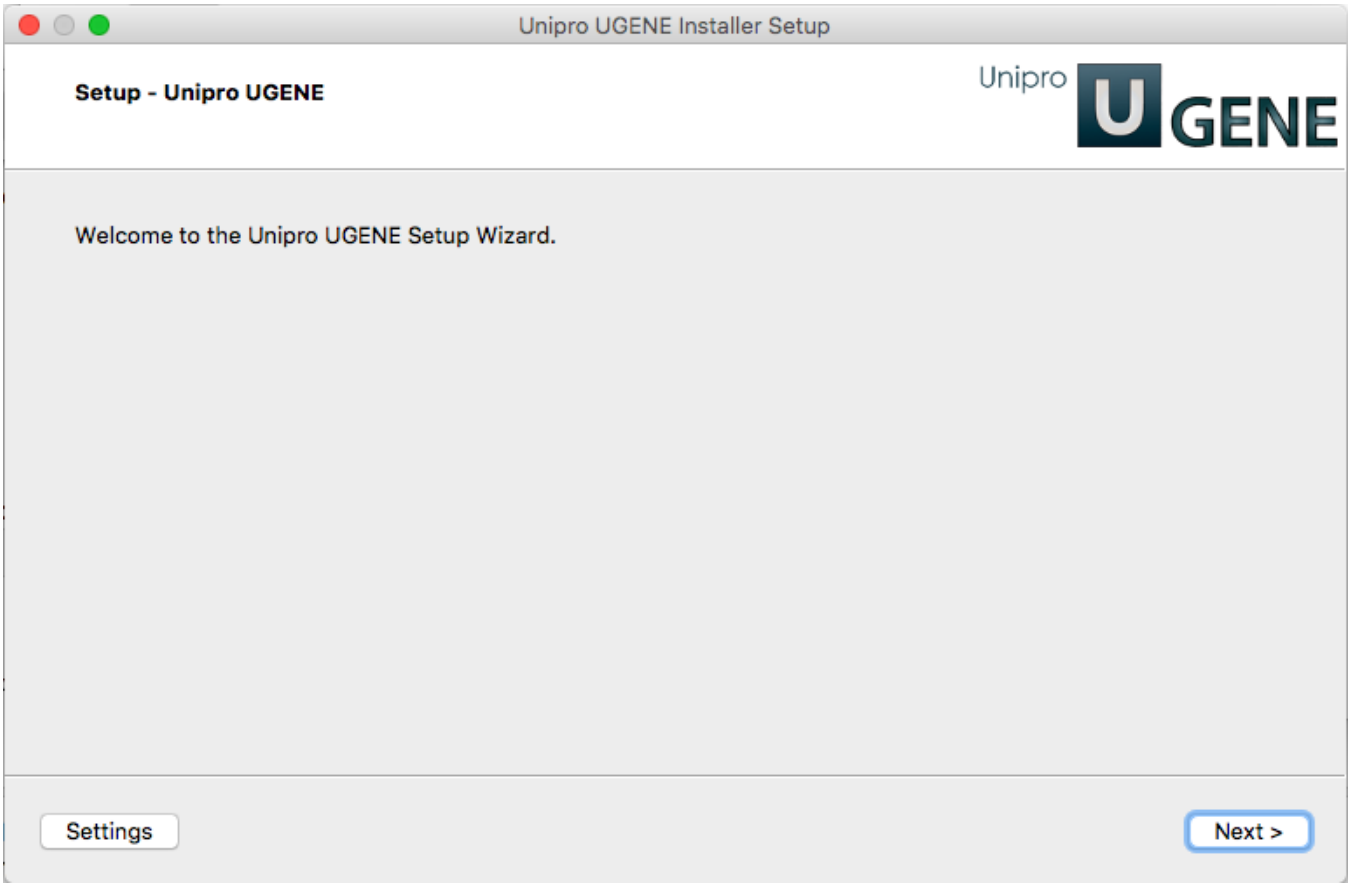
It may be required to configure macOS settings to install UGENE:

- Select "System Preferences -> Security & Privacy".
- Unlock the settings to make changes.
- Under "Allow apps downloaded from" select "Anywhere".



Installation using online installer

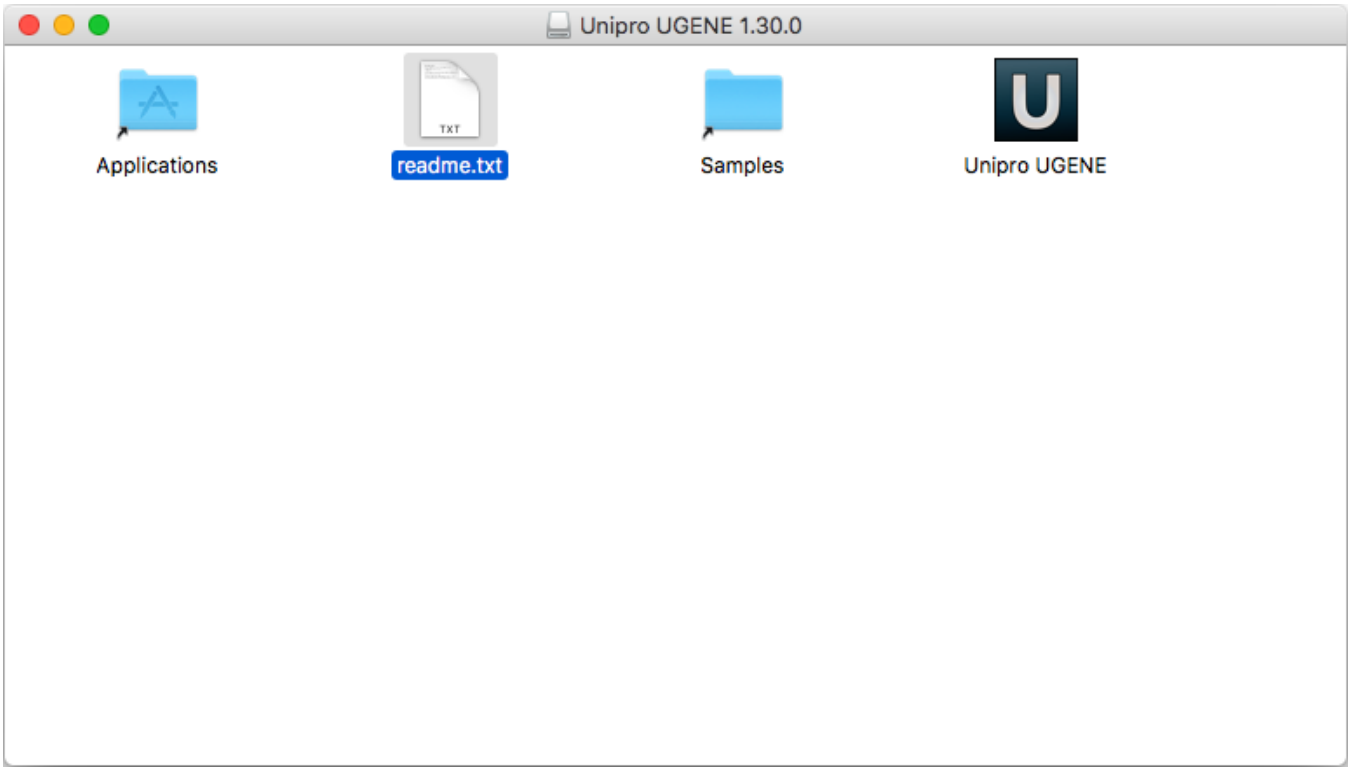
- Download the online installer DMG archive.
- When installing the package, make sure the Internet connection is available.
- Double-click on the archive to unpack it.
- Run the unpacked executable file.
- Follow the installation wizard.



On a new UGENE version release, a notification will appear with an option to update the package.

Installation using offline installer

- Download the offline installer DMG archive.
- Double-click on the archive.
- UGENE is provided under free GNU GPL v2 license. Accept the license terms to install the package.
- Move "Unipro UGENE" application bundle to the "Applications" folder on macOS or to another appropriate location.



On a new UGENE version release, a notification will appear. However, to install the new version, it is required to manually download the package again.

Installation on Linux

The following UGENE package delivery types are available on 64-bit Linux:

- **Online installer:** updates to new versions are supported
- **TAR.GZ archive:** does not require Internet connection to be installed

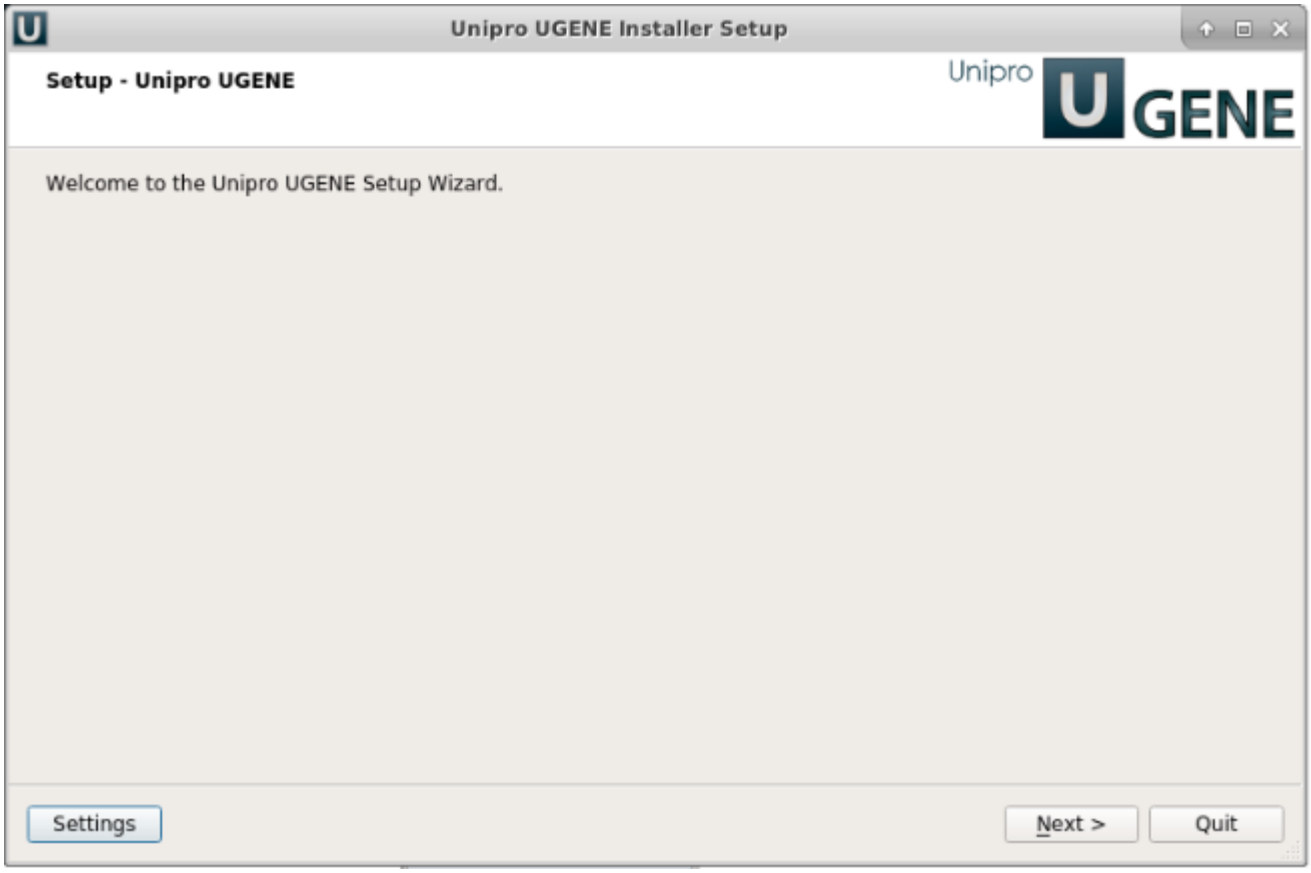
Other packages for older UGENE version are available on the web page http://ugene.net/download-all.html.

Installation using online installer

- Download an archive with the online installer file ("ugeneInstaller_64bit.tar").
- Unpack the archive.
- Run the extracted "ugeneInstaller_64bit" executable file.

```
cd ~/Downloads
tar -xf ugeneInstaller_64bit.tar
./ugeneInstaller_64bit
```

- Follow the installation wizard.



On a new UGENE version release, a notification will appear with an option to update the package.

Installation using TAR.GZ archive

- Download the TAR.GZ archive.
- Unpack the archive - in the Terminal run "tar -xzf *archive_file*".
- From the unpacked folder run "./ugeneui" (here "ui" states for "User Interface"). For example:

```
cd ~/Downloads
tar -xzf ugene-current-version-x86-64.tar.gz
cd ugene-current-version
./ugeneui
```

On a new UGENE version release, a notification will appear. However, to install the new version, it is required to manually download the package again.

Data Analysis Tools

UGENE provides GUI for a lot of different bioinformatics tools. A tool is supplied either as embedded or external:

- **Embedded tool:** the tool is integrated into the UGENE source code. Such tools are always available in UGENE independently on the UGENE package type and operating system.
- **External tool:**
  - **Integrated (supported) external tool:** the tool executable file (or files) is provided with UGENE. It is launched when one uses the tool in the UGENE GUI. The result is also shown in the UGENE GUI. Note that the tool may be available on some operating systems only.
  - **Custom external tool:** it is possible to add any custom external tool as a workflow element in the Workflow Designer. See "Custom External Tools" documentation chapter for details.

Specify integrated external tools in the installed package

Most of the external tools (if available on the target operating system) are installed by default in the Online Installer.

One can specify the tools manually:

1. Download the tools executables, see links on the web page http://ugene.net/download-all.html:

- **External Tools Package:** the package contains all external tools except JRE (i.e. "java") and RScript.
- **Java Runtime Environment (JRE):** although UGENE is not a Java application, a few tools (e.g. FastQC, Trimmomatic) require JRE. Make sure JRE 8 or higher is installed.
- **RScript:** R is currently used in UGENE for ChIP-Seq data analysis only.

2. Configure the tools in UGENE: open the UGENE Application Settings dialog, select the "External Tools" page and specify the external tools executables.

Integrated external tools

See the list of the currently integrated external tools, information about their availability on different operating systems and version included into the corresponding External Tools Package in the table below.

External tool	Windows (64-bit)	macOS (64-bit)	Linux (64-bit)
bedGraphToBigWig	4	4	4
bedtools	2.20.1	2.20.1	2.20.1
BLAST+	2.2.30	2.2.30	2.2.30
Bowtie	0.12.7	1.0.0	1.0.0
Bowtie2	2.3.4.3	2.3.4.3	2.3.4.3
BWA	0.7.12	0.7.12	0.7.12
CAP3	12/21/07	12/21/07	12/21/07
CEAS	1.0.2	1.0.2	1.0.2
CLARK	N/A	1.2.4 (UGENE-customized)	1.2.4 (UGENE-customized)
ClustalO	1.2.2	1.2.3	1.2.4
ClustalW	2.1	2.1	2.1
ConductGO	Version not specified	Version not specified	Version not specified
conservation_plot	0.1.0	0.1.0	0.1.0
Cufflinks tools	N/A	2.0.2	2.0.2
Cutadapt	1.7.1	1.7.1	1.7.1
DIAMOND	N/A	0.9.22	0.9.22
FastQC	0.11.2	0.11.2	0.11.2
JRE (java)	1.8.0	N/A	N/A
HMMER	3.1b2	3.1b2	3.1b2
Kraken	N/A	1.0	1.0
MACS	1.4.2	1.4.2	1.4.2
MAFFT	7.450	7.450	7.450
MetaPhlAn2	N/A	2.0.0	2.0.0
MrBayes	3.2.3	3.2.3	3.2.3
peak2gene	1.02	1.02	1.02
perl	5.24.3	N/A	N/A
PhyML	20120412	20120412	20120412
python	2.7.5	2.7.5	2.7.5
Rscript	R is not included into the External Tools Packages. However, it may be installed using the UGENE Online Installer package.		
samtools	0.1.19	0.1.19	0.1.19
SeqPos	1.01	1.01	1.01
SnpEff	4.3i	4.3i	4.3i
SPAdes	N/A	3.12.0	3.12.0
Spidey	Version not specified	N/A	Version not specified
StringTie	N/A	1.3.4d	1.3.4d
T-Coffee	8.99	11.00	11.00
TopHat	N/A	2.1.1	2.1.1

Trimmomatic	0.38	0.38	0.38
vcftools	0.1.12b	0.1.12b	0.1.12b
WEVOTE	N/A	Version not specified	Version not specified

Reference Data for NGS

To run some NGS data analysis (metagenomics classification or ChIP-Seq) it is required to provide reference data (e.g. RefSeq databases, NCBI taxonomy files, etc.).

Both taxonomy classification and ChIP-Seq tools are available on macOS and Linux operating systems only. Also, make sure there is enough disk space on the target computer.

It is recommended to use the Online Installer to download these data, however, one can also download them directly (see links on the web page http://ugene.net/download-all.html). In the latter case it is required to additionally manually configure an installed UGENE package, see instructions below.

Configure Data for Metagenomics Classification

UGENE integrates a set of tools for taxonomy classification of microorganisms using whole-genome shotgun sequencing data. The tools are Kraken, CLARK, DIAMOND, etc. See, for example, "Parallel NGS Reads Classification" sample workflow that allows one to classify input FASTQ files with these tools working in parallel and then join their output with another tool WEVOTE.

The tools are available on 64-bit macOS or Linux operating systems only. Also, as the tools are quite resource-consuming, it is recommended to have at least 16 Gb of RAM available.

To use these tools one should provide appropriate taxonomy data and reference data, specific for a tool. Some reference databases are provided for each of the tool. One can use these data or build a custom database (see, for example, workflow element "Build Kraken Database").

It is recommended to use the UGENE Online Installer package to install and automatically configure the data. However, if the Internet is not available on the target computer, or it is required to use another UGENE package for some other reason, follow the instructions below on how to download and configure the data.

Download data for metagenomics classification

Use links in the "Data for NGS metagenomics classification" section on the web page http://ugene.net/download-all.html to download the data.

Make sure to have enough disk space on the target computer.

See the list of the available downloads in the table below.

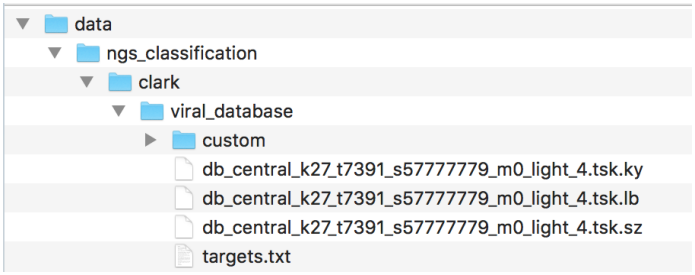
Data	Archive size	Unpacked data size	Description	Data source
NCBI taxonomy classification	2.5 Gb	31 Gb	This includes a set of taxonomy data files from NCBI. These data should be present for any type the NGS classification analysis.	Original data were downloaded from the NCBI FTP (ftp://ftp.ncbi.nlm.nih.gov/pub/taxonomy/).
NCBI RefSeq bacterial genomes	130 Gb	132 Gb	The data can be used to build a database for CLARK-l (light version of CLARK), CLARK, or Kraken. As UGENE integrates modified version of CLARK/CLARK-l, it is possible to provide *.gz archives as input for building the database. In particular, "CLARK-l DB: RefSeq bacterial+viral genomes" (see below) was generated using the archived data. Also, keep in mind that changing of some parameters of the "Classify Sequences with CLARK" element may cause re-building of the reference database. The reference data should be present in this case! For building a Kraken database usage of *.gz archives is not supported, it is required to unpack each *.gz file, so even more disk space will be required.	Original data were downloaded from the NCBI FTP (ftp://ftp.ncbi.nlm.nih.gov/refseq/release/bacteria/bacteria*.genomic.fna.gz)
NCBI RefSeq viral genomes	77 Mb	77 Mb	Similarly to "NCBI RefSeq bacterial genomes", although the size of the data is rather small. The reference data are included into "CLARK-l DB: RefSeq bacterial+viral genomes" and "CLARK-l DB: RefSeq viral genomes" databases.	Original data were downloaded from the NCBI FTP (ftp://ftp.ncbi.nlm.nih.gov/refseq/release/viral/viral*.genomic.fna.gz).

NCBI RefSeq GRCh38 human genome	837 Mb	838 Mb	Similarly to "NCBI RefSeq bacterial genomes". The data are not included into any database, but provided in case one would like to use them when building a custom database.	Original data were downloaded from the NCBI FTP (ftp://ftp.ncbi.nih.gov/genomes/H_sapiens/CHR_*/hs_ref_GRC*chr*.fa.gz).
Kraken DB: MiniKraken 4Gb database	2.5 Gb	4.3 Gb	A sample reference database provided in UGENE for Kraken. It is a pre-built 4 GB database constructed from complete bacterial, archaeal, and viral genomes in RefSeq (as of Oct. 18, 2017). This can be used by users without the computational resources needed to build a Kraken database. However this contains only 2.7% of kmers from the original database.	Original data were downloaded using a link on the Kraken web site (https://ccb.jhu.edu/software/kraken/).
CLARK-l DB: RefSeq bacterial+viral genomes	7.4 Gb	11 Gb	One of the reference databases provided in UGENE for CLARK-l. The database was build using archived RefSeq bacterial and viral genomes.	See above.
CLARK-l DB: RefSeq viral genomes	16 Mb	72 Mb	One of the reference databases provided in UGENE for CLARK-l. The database was build using archived RefSeq viral genomes.	See above.
DIAMOND DB: UniRef50	5.2 Gb	13 Gb	One of the reference databases provided in UGENE for DIAMOND. Note that unlike Kraken and CLARK, DIAMOND requires protein reference sequences as input.	Original data were downloaded from the Uniprot FTP (ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/uniref/uniref50/uniref50.fasta.gz). Then a DIAMOND database was built.
DIAMOND DB: UniRef90	13 Gb	34 Gb	One of the reference databases provided in UGENE for DIAMOND.	Original data were downloaded from the Uniprot FTP (ftp://ftp.uniprot.org/pub/databases/uniprot/current_release/uniref/uniref90/uniref90.fasta.gz). Then a DIAMOND database was built.
MetaPhlAn 2 embedded DB: mpa_v20_m200	1 Gb	1.2 Gb	This database is provided with the MetaPhlAn2 tool. It was built on ~1M unique clade-specific marker genes identified from ~17,000 reference genomes (~13,500 bacterial and archaeal, ~3,500 viral, and ~110 eukaryotic).	The database was downloaded here: https://bitbucket.org/biobakery/metaphlan2/downloads/.
Total:	**162 Gb**	**227 Gb**		

Configure data

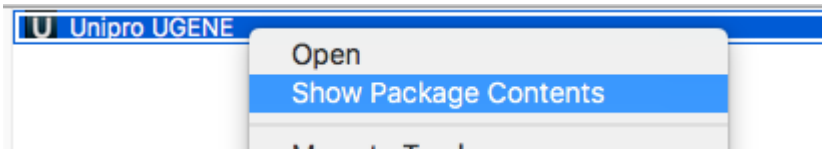
Data described above are stored as 7zip archives. After a file download, unpack it using an appropriate file archiver (for example, Keka on macOS).

The unpacked data are stored in a folders structure with the root folder called "data". For example, for "NCBI RefSeq viral genomes" the archive is called "ngs_classification.clark.viral_database.7z" and the unpacked data look as follows:



It is required to move these data to the UGENE data folder, following the hierarchical data structure:

- On Linux it is "data" folder, located in the UGENE installation folder.
- On macOS the "data" folder is located inside the "Unipro UGENE.app" bundle. Right-click on the bundle, select "Show Package Contents", select "Contents -> MacOS -> data" folder.



Thus, all required data will be placed to the "ngs_classification" sub-folder of the UGENE data folder.

Kraken, CLARK, DIAMOND and WEVOTE are integrated as external tools. So, also make sure the tools executables are set in the UGENE Application Settings.

Configure ChIP-Seq Analysis Data

UGENE integrates the Cistrome workflow for analysis of ChIP-Seq data. The workflow requires reference data to be present in the UGENE data folder. Reference data for human can be installed via the Online Installer. To configure them manually download the data from the web page http://ugene.net/download-all.html and unpack to the "cistrome" folder inside the UGENE data folder.

Note that RScript external tool should also be provided.

Basic Functions

- UGENE Terminology
- UGENE Window Components
  - Start Page
  - Project View
  - Task View
  - Log View
  - Notifications
- Main Menu Overview
- Creating New Project
- Creating Document
- Renaming Object
- Opening Document
  - Opening for the First Time
    - Advanced Dialog Options
  - Opening Document Present in Project
  - Opening Several Documents
  - Opening Unloaded Documents
  - Opening from Clipboard
- Opening Containing Folder
- Exporting Documents
- Locked Documents
- Using Objects and Object Views
- Exporting Objects
  - Exporting Sequences to Sequence Format
  - Exporting Sequences as Alignment
  - Exporting Alignment to Sequence Format
  - Exporting Nucleic Alignment to Amino Translation
  - Export Sequences Associated with Annotation
- Using Bookmarks
- Exporting Project
- Search in Project
- Options Panel
- Plugins Viewer
- Searching NCBI Genbank
- Fetching Data from Remote Database
- UGENE Application Settings
  - General
  - Resources
  - Network
  - File Format
  - Directories
  - Logging
  - Alignment Color Scheme
  - External Tools
    - Supported External Tools
    - Custom External Tools
  - Workflow Designer Settings
  - OpenCL

UGENE Terminology

Project

Storage for a set of data files and visualization options.

Document

A single file (can be stored on a local hard drive or be a remote web page). Each *document* contains a set of *objects*.

Object

A minimal and complete model of biological data. For example: a single sequence, a set of annotations, a multiple sequence alignment.

Task

A process, usually asynchronous, that works in background. For example: some computations, loading and writing files.

Plugin

A dynamically loaded module that adds new functionality to UGENE.

Object View

A graphical view for a single or a set of *objects*.

Project View

A visual component used to manage active *project*.

Task View

A visual component used to manage active *tasks*.

Log View

A visual component used to show logs.

Notifications

A visual component used to show notifications. Generally it is used to open tasks reports.

Plugin Viewer

A visual component used to manage *plugins*.

Sequence View

An *Object View* aimed to visualize DNA, RNA or protein sequences along with their properties like annotations, chromatograms, 3D models, statistical data, etc.

Annotation

Additional information about a sequence, identified by its name and the sequence region.

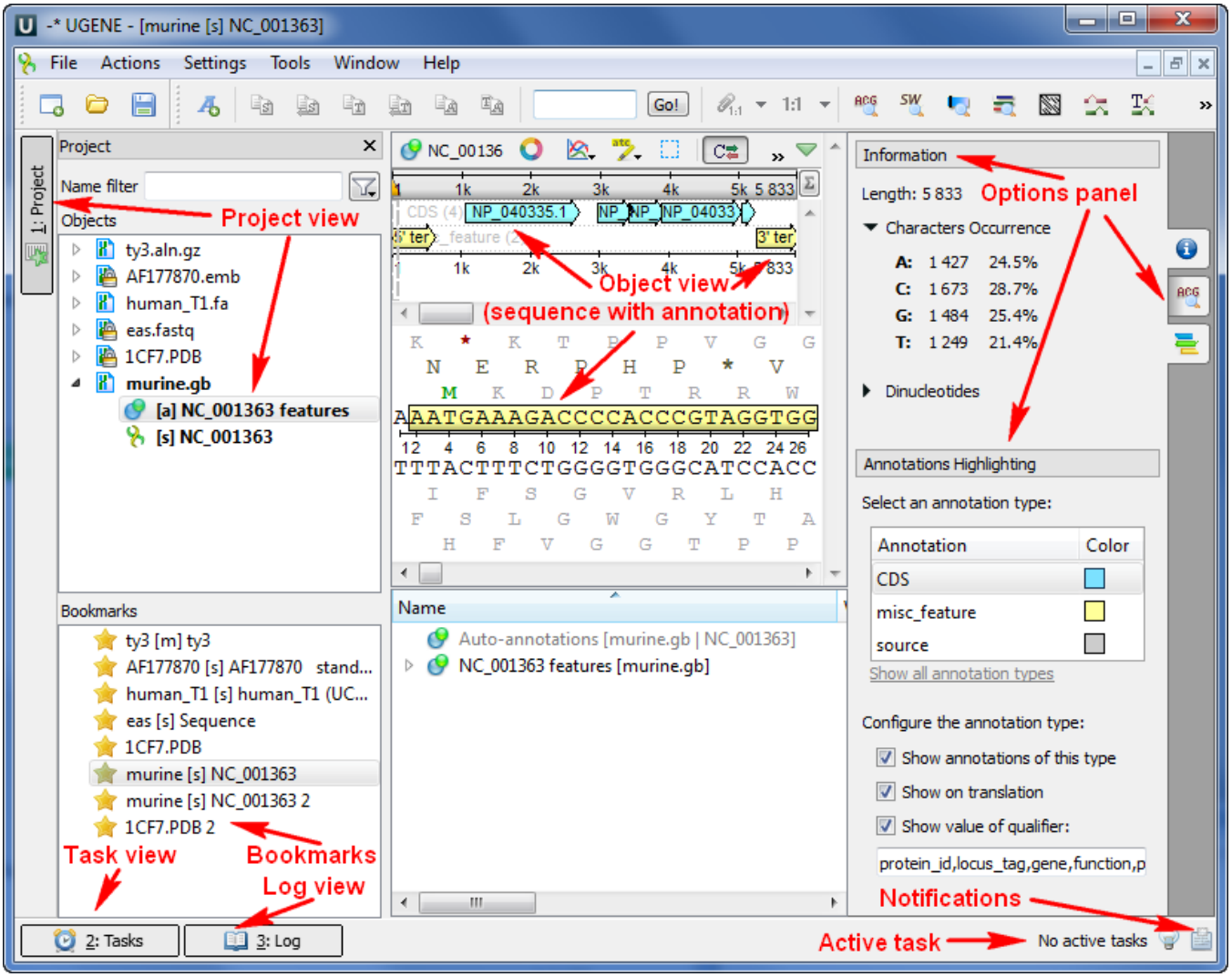
Alignment Editor

An *Object View* used to visualize and edit DNA, RNA or protein multiple sequence alignments.

Options Panel

An *Options Panel* it is the panel with different information tabs and tabs with settings for *Sequence View* and *Assembly Browser*.

In the image below you can see a typical UGENE window with a *Project View* and a single *Object View* window opened:



UGENE Window Components

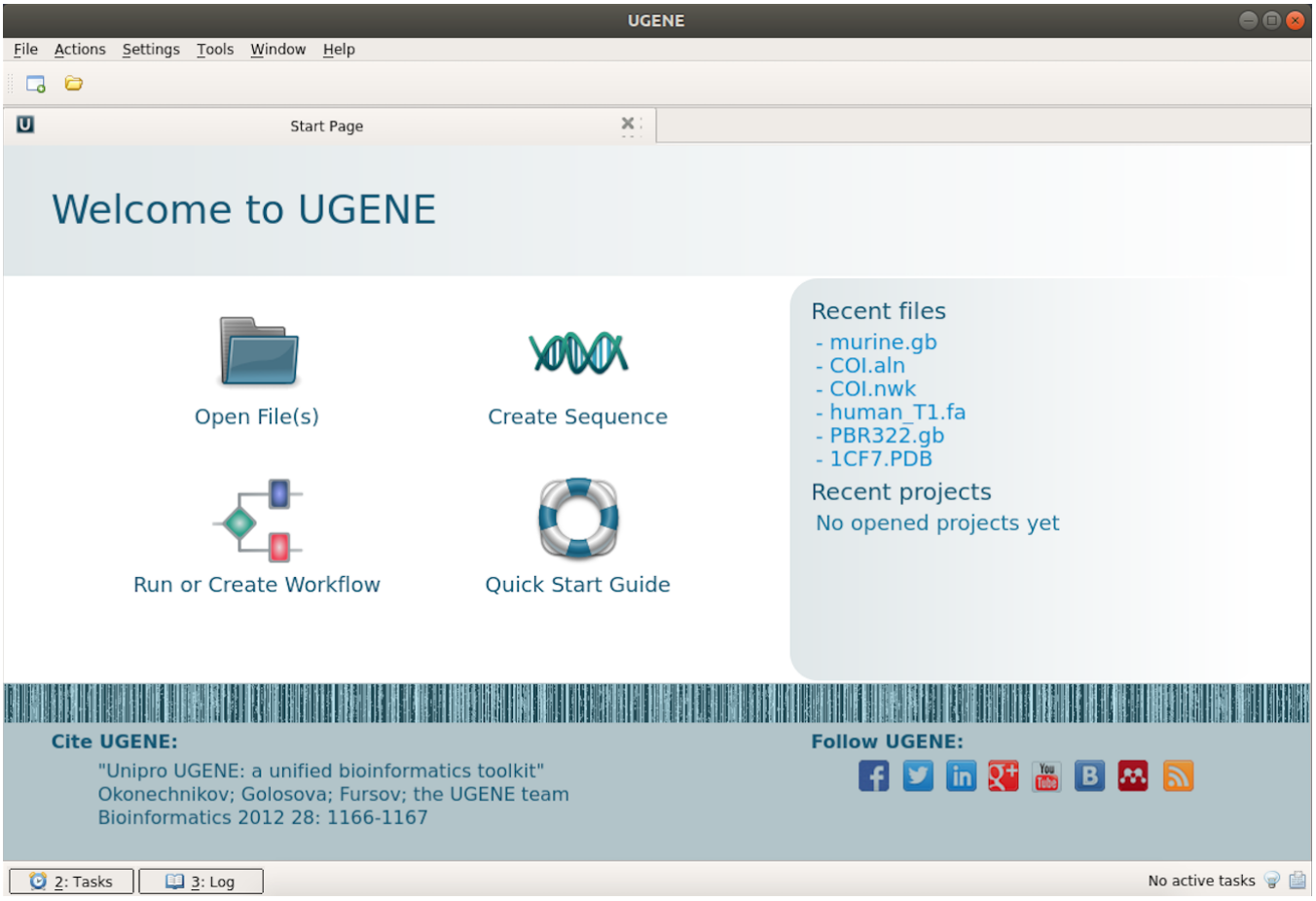
This chapter describes UGENE main window components *Project View*, *Task View*, *Log View* and the *Notifications* popup window.

- Start Page
- Project View
- Task View
- Log View
- Notifications

Start Page

The *Start Page* is the first page that appear when UGENE is launched. From this page you can open a file, create a sequence, create a workflow (use a sample workflow or create a new one), open the Quick Start Guide, and open recent files directly.

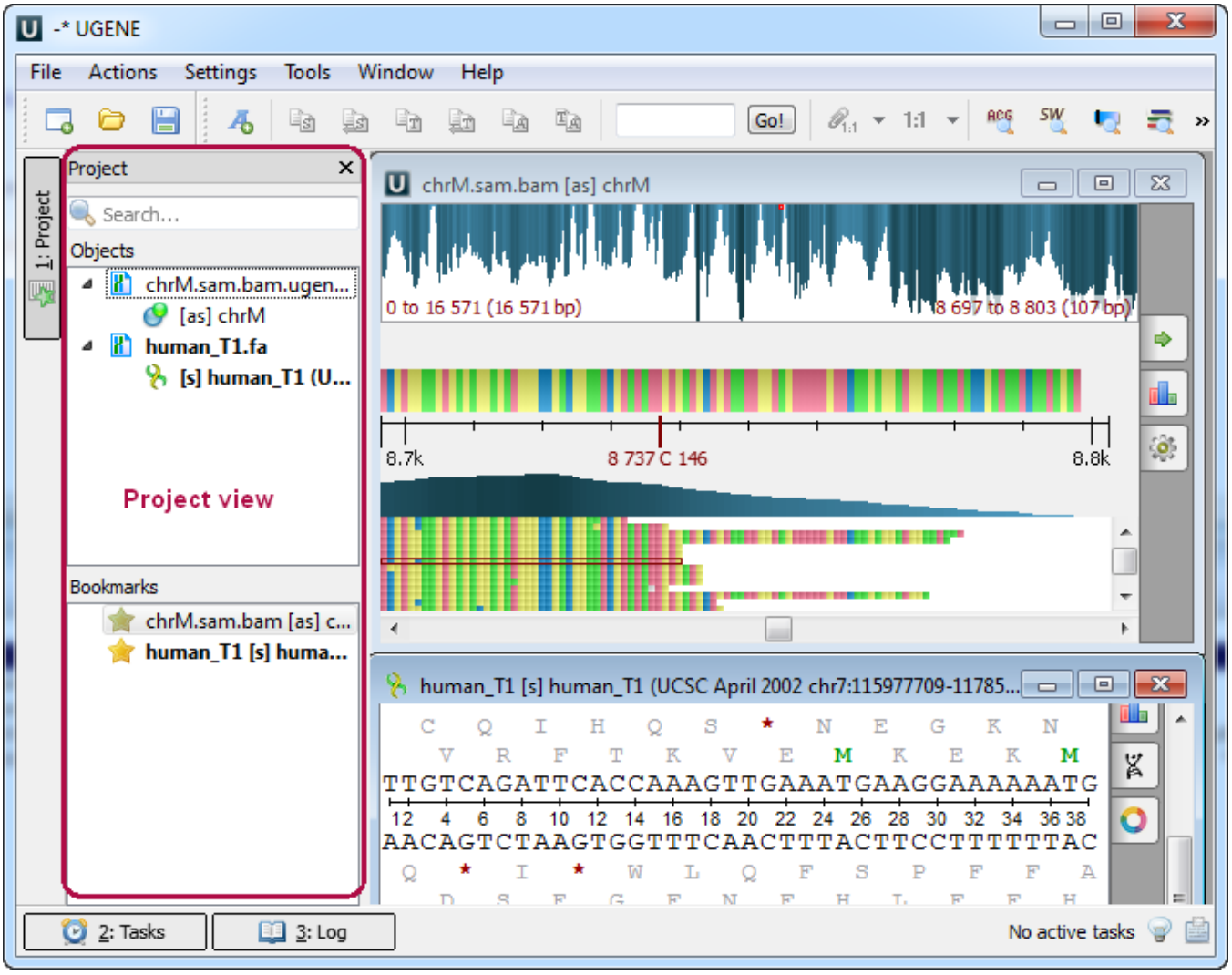
To return to the *Start Page* at any time select *Help->Open Start Page* in the main menu.



Project View

The *Project View* shows documents and bookmarks of the current *project*. The documents are files added to the project. And the bookmarks are visual view states of the documents. Read *Using Bookmarks* to learn more about bookmarks.

To show/hide the *Project View*, click the *Project* button in the main UGENE window:



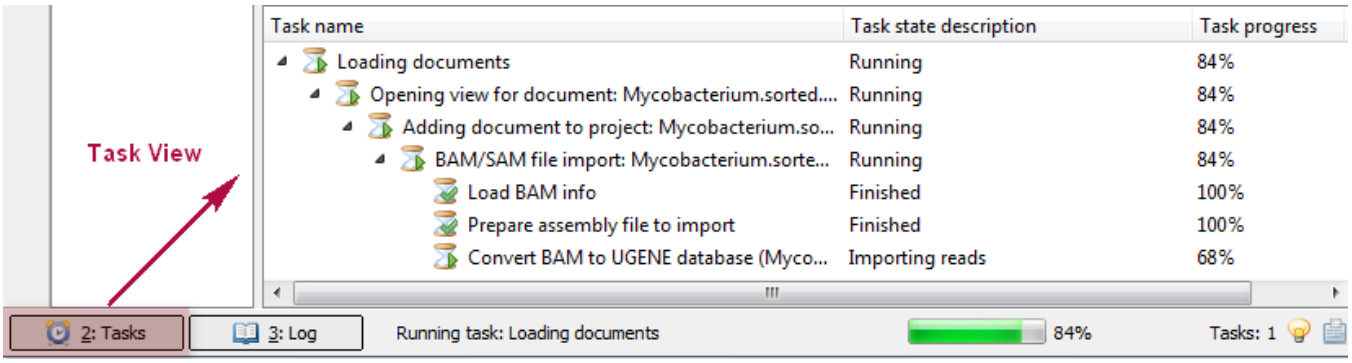
You can also use the Alt+1 hotkey to show/hide the *Project View*.

To create a new project, refer to *Creating New Project*. Note that if you have no project created when opening file with a sequence, an alignment or any other biological data, a new anonymous project is created automatically.

Task View

The *Task View* shows active tasks, for example, algorithms computations.

To show/hide the *Task View*, click the *Tasks* button in the main UGENE window:



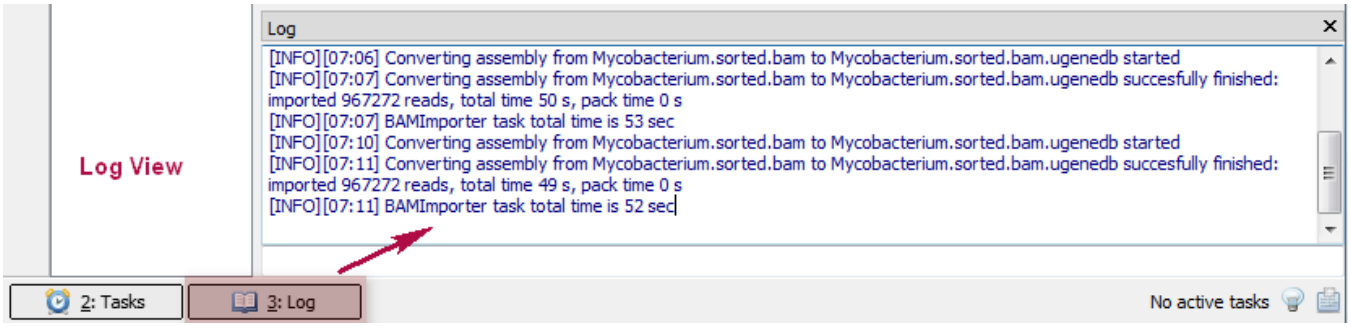
The hotkey for showing/hiding the *Task View* is Alt+2.

The *Task name* column of the *Task View* shows the tasks names. *Task state description* shows the status of the active tasks: Started, Running, Finished and so on. The *Task progress* column shows the percentage of the tasks progress. If you want to cancel a task, click the red cross button in the *Actions* col umn for the task.

Log View

The *Log View* shows the program log information.

To show/hide the *Log View* click the *Log* button in the main UGENE window:

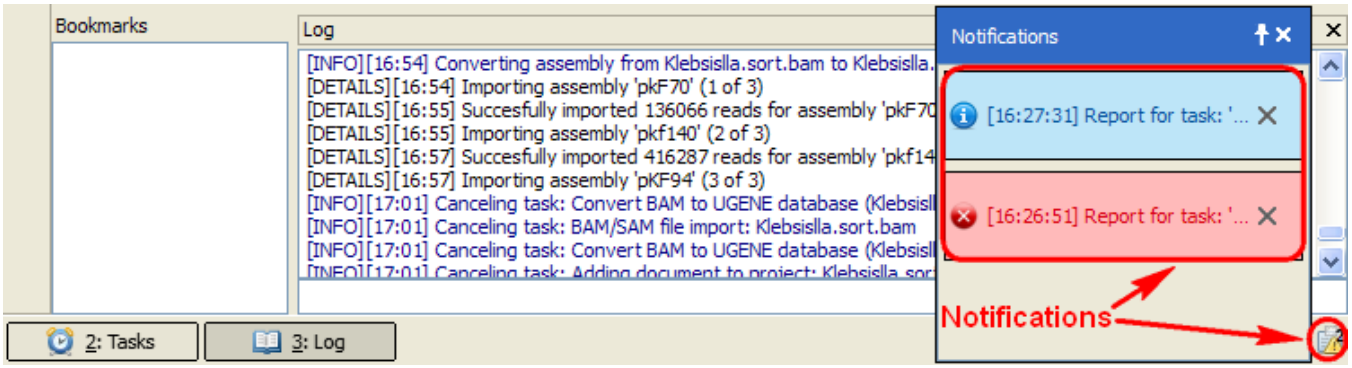


The hotkey for this action is Alt+3.

It is possible to configure the *Log View* settings: the level of the log to show (ERROR, INFO, DETAILS, TRACE) , the category (Algrorithms, Tasks, etc.), and the format of the log messages (format of the dates, etc.). This settings can be configured in the UGENE *Application Settings*.

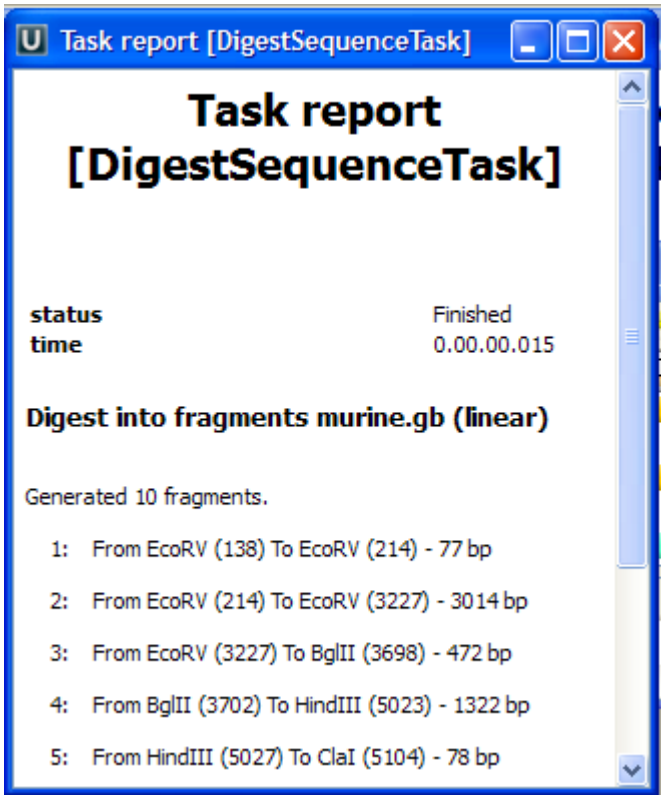
Notifications

The *Notifications* component shows notifications for tasks reports.



If a task has finished without errors, the notification is blue. If an error has occured during the task execution, the notification is red. If a warning has occured during the task execution, the notification is yellow.

To open a task report, click on the corresponding notification. See an example of a task report below:



To remove a notification from the *Notifications* popup window, click the notification cross button.

Note that you can click on the clip button of the *Notifications* popup window to show the window always on top.

Main Menu Overview

Menu	Description
File	A set of project level operations. **List of operations:** new project, new document from text, new workflow, open, open as, open from clipboard, access remote database, search NCBI genbank, connect to UGENE shared database, save all, save project as, export project, close project, recent files, recent projects, exit.
Actions	Various actions associated with the active window. **List of operations:** go to position, add, copy/paste, analyze, align, cloning, export, remove, edit (for the *Sequence View*); go to position, add, copy/paste, colors, highlighting, edit, align, tree, statistics, view, export, advanced, consensus mode, close active window (for the *Alignment Editor*).
Settings	Preferences and plugin settings.
Tools	Various tools. This menu is extended by different plugins. **List of operations:** sanger data analysis, NGS data analysis, BLAST, multiple sequence alignment, cloning, primer, search for TFBS, HMMER tools, build dotplot, random sequence generator, show counters, expert discovery, query designer, workflow designer.
Window	A list of active windows and basic manipulations with the windows. **List of operations:** window layout (multiple or tubbed), close active view, close all windows, tile windows, cascade windows, next window, previous window.
Help	Application help and check for updates. **List of operations:** open UGENE user manual, view UGENE documentation online, visit UGENE website, check for updates, open start page, about.
Unipro UGENE (macOS	**List of operations:** about Unipro UGENE, preferences, services, hide Unipro UGENE, hide others, show all, quit Unipro UGENE.

only)	

The menus can be dynamically populated with new actions added by plugins. Check the *Plugins* documentation to learn how each plugin affects global and context menus.

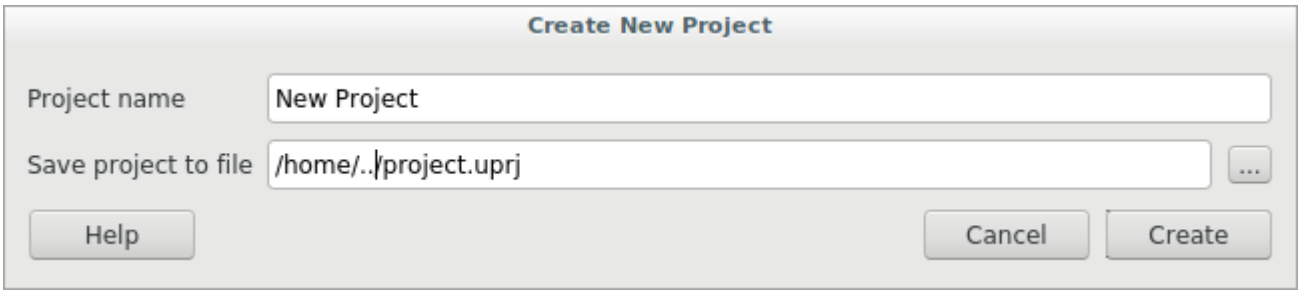
Creating New Project

A *project* store links to the data files, cross-file data associations, and visualization settings.

Below is a description of how to create a new project manually. Note that if you have no project created when opening a file with a sequence, an alignment or any other biological data, a new anonymous project is created automatically.

To create a new project select the *File New project* menu or click the *New project* button on the main toolbar.

The dialog will appear:



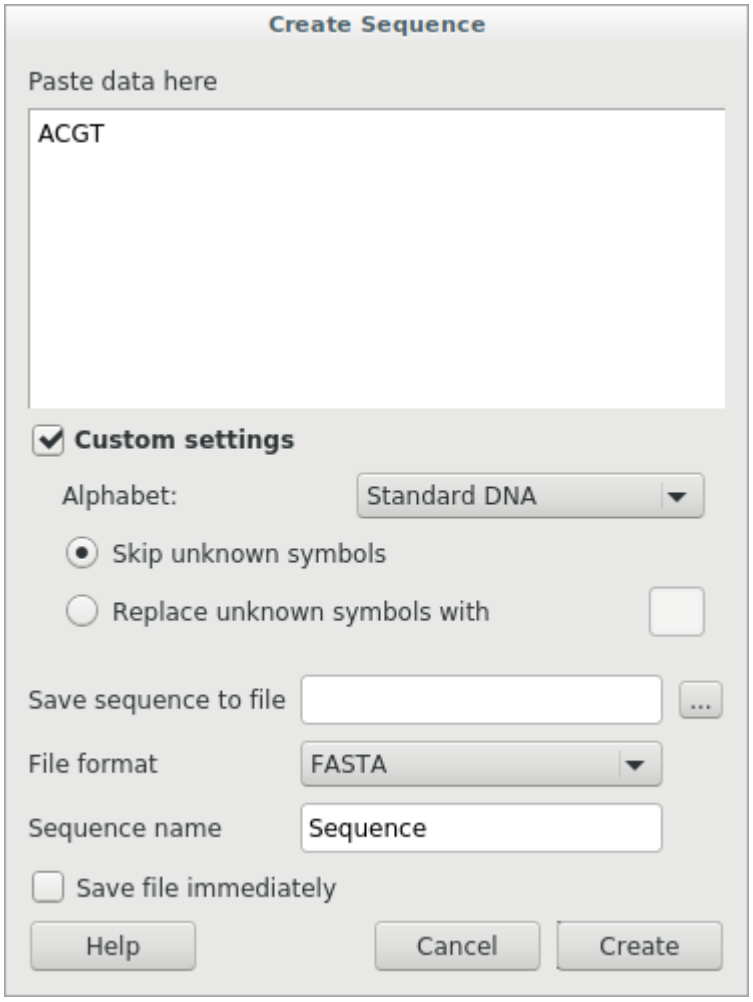
Here you need to specify the visual name for the project and the directory and file to store it.

After you click the *Create* button the *Project View* window is opened.

Creating Document

To create a new sequence file from a text, select the *File New document from text* main menu item.

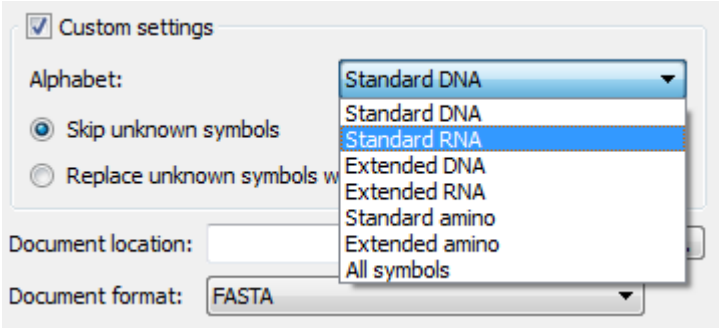
The *Create Sequence* dialog appears:



You can input the created sequence to the *Paste data here* field. You can type or paste sequences in FASTA or text format.

The following *Custom settings* are available:

Alphabet — here you can select the alphabet:



The following alphabets are available: Standard DNA, Standard RNA, Extended DNA, Extended RNA, Standard amino, Extended amino.

Skip unknown symbols / *Replace unknown symbols with* — you can select either to skip unknown input symbols or to replace them with the specified symbol.

Document location — a location of the created document.

Document format — a format of the created document. Currently available formats are FASTA and Genbank.

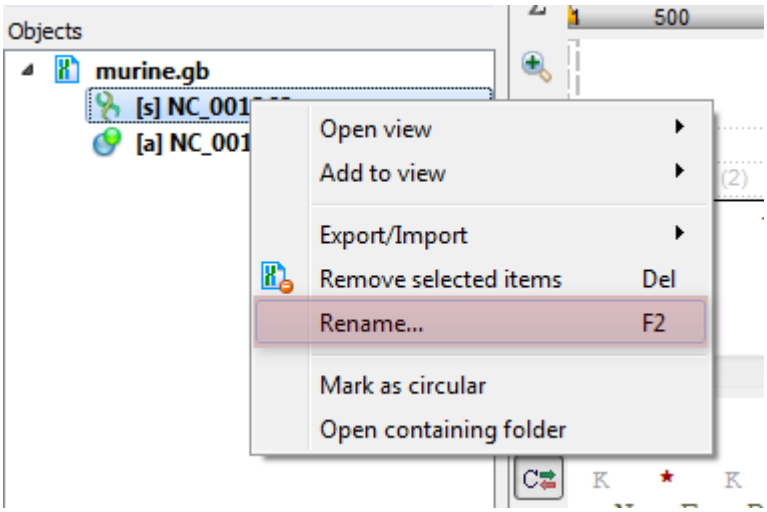
Sequence name — a name of the sequence in the created document.

Save file immediately — check this option if you want to save the document immediately after the *Create* button is pressed.

The created document will be added to the current project and opened in the *Sequence View*.

Renaming Object

To rename an object in the project tree view use the F2 shortcut or *Rename* context menu item:



Opening Document

UGENE stores information about *documents* you are working with in a *project*. Once a *document* has been opened, the information about it is saved in the current *project*.

- Opening for the First Time
  - Advanced Dialog Options
- Opening Document Present in Project
- Opening Several Documents
- Opening Unloaded Documents
- Opening from Clipboard

Opening for the First Time

To open a *document* that is not yet presented in the current *project* use either an advanced *Open* dialog, a simple open file dialog or just drag the document to the UGENE window.

UGENE automatically detects the *format* of the *document*, but if you use the advanced dialog you can choose the format manually.

To open the advanced dialog select one of the following:

- *Add Existing document* item in the *Project View* context menu
- *File Open As* item in the main menu

To simply open the document select one of the following:

- *Open* item in the main toolbar
- *File Open* item in the main menu

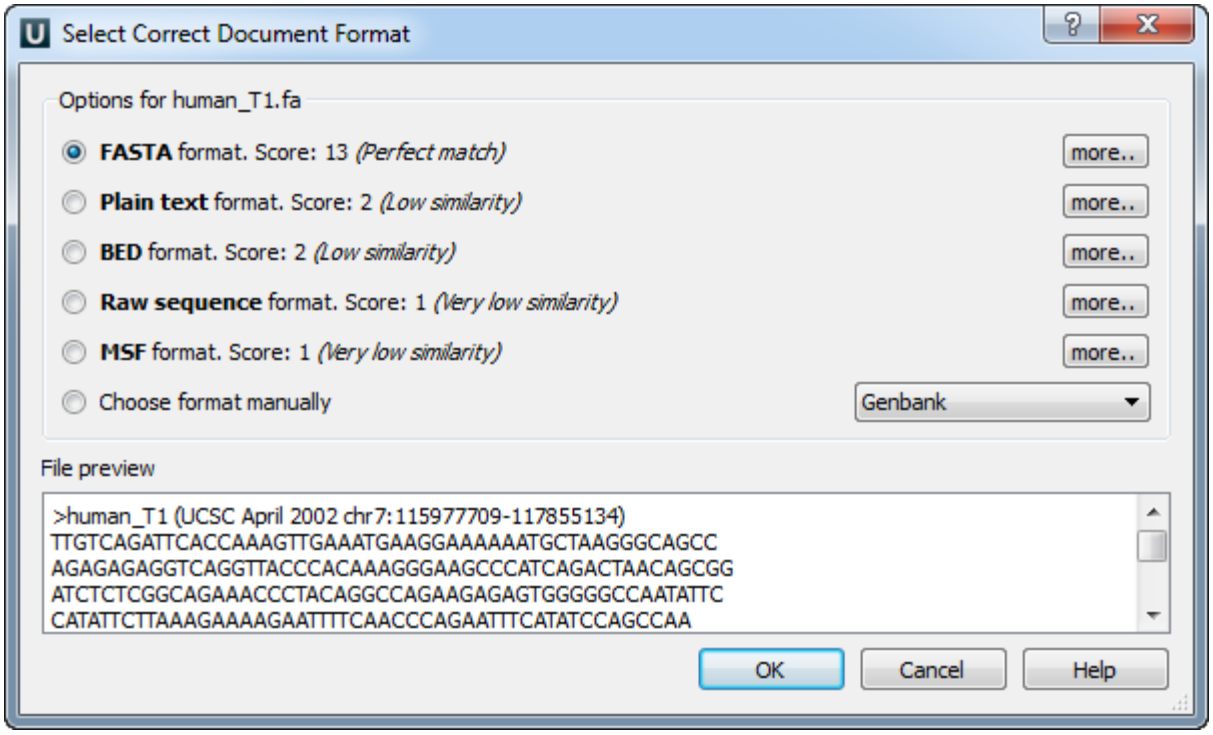
or drag the file to the UGENE window. Also it is able to drag and drop documents (not objects) between opened UGENEs.

Documents created not by UGENE are *locked*. To be able to edit the document you should save a copy of the document and continue working with the copy.

- Advanced Dialog Options

Advanced Dialog Options

Open the *Select Correct Document Format* dialog by *Add Existing document* item in the *Project View* context menu or by *File Open As* item in the main menu. The foolowing dialog will appear:



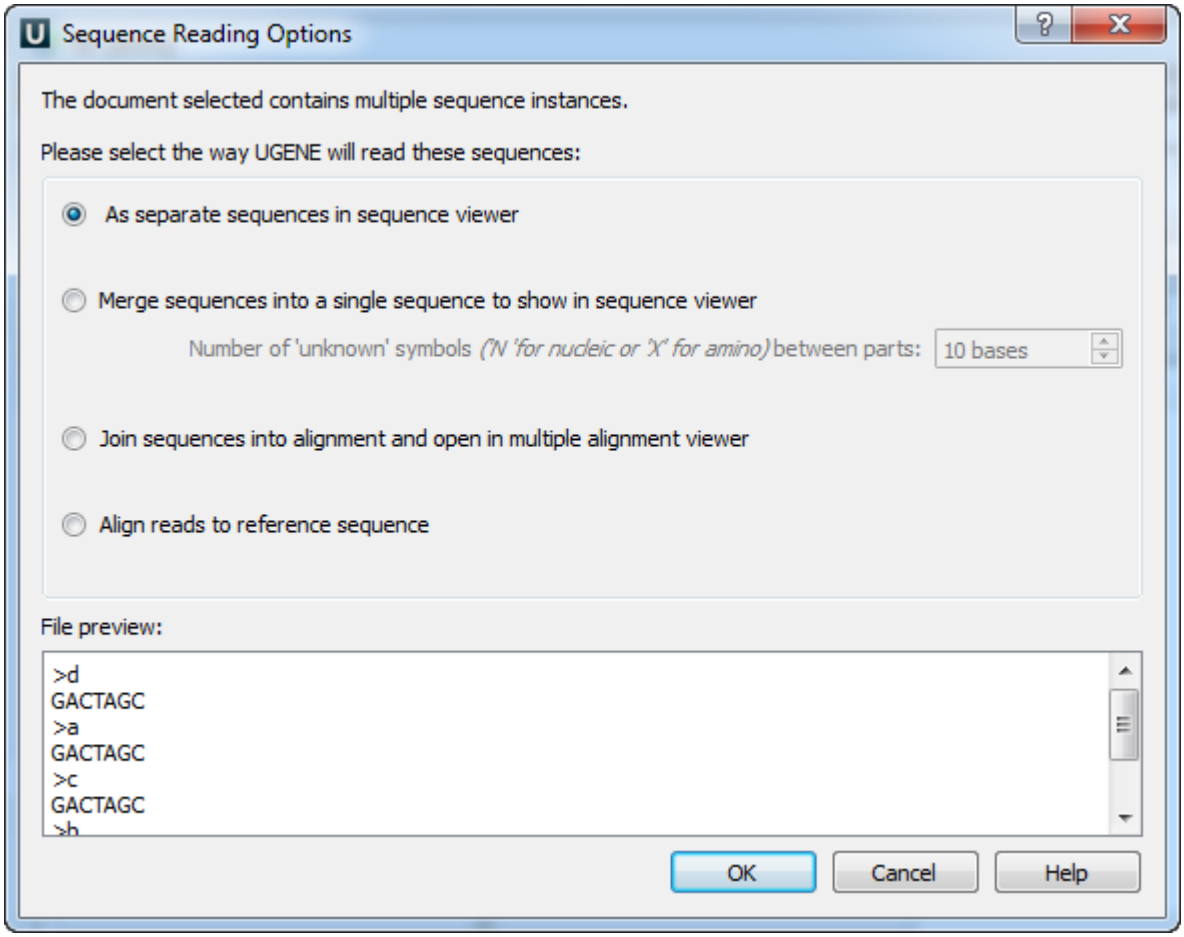
Here you can choose how to interpret the data stored in the file. The format is detected automatically, but you can select it manually.

Opening Document Present in Project

To open a *document* that is already present in the current *project* select it in the *Project View* and click Enter, double-click on it or drag it to an empty space of the UGENE window.

Opening Several Documents

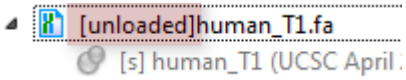
To open several documents that are not yet presented in the current *project* use the *File Open* item in the main menu. The *Select files to open* dialog wi appear. Select the documents with a help of the *Ctrl* button and click on the *Open* button. The following dialog will appear:



Select the reading options and click on the *OK* button.

Opening Unloaded Documents

The [unloaded] prefix indicates that document is unloaded:



To open unloaded document use the *Load selected document* context menu, *Enter* key or double click on the document. To unload the document use the corresponding context menu item.

Opening from Clipboard

It is possible in UGENE opening text data and files by pasting them into the *Project View* from the clipboard.

Select text data in a browser or in a text editor or file(s) from your system and press *Ctrl+C (Cmd+C for macOS).*

Switch to UGENE with an opened project, click the project area and press *Ctrl+V (Cmd+V)*.

The copied data or file(s) will be opened in UGENE.

Another way for the file(s) is drag-drop. Using your mouse, drag and drop the files from the file system to UGENE, and they will be opened by the program.

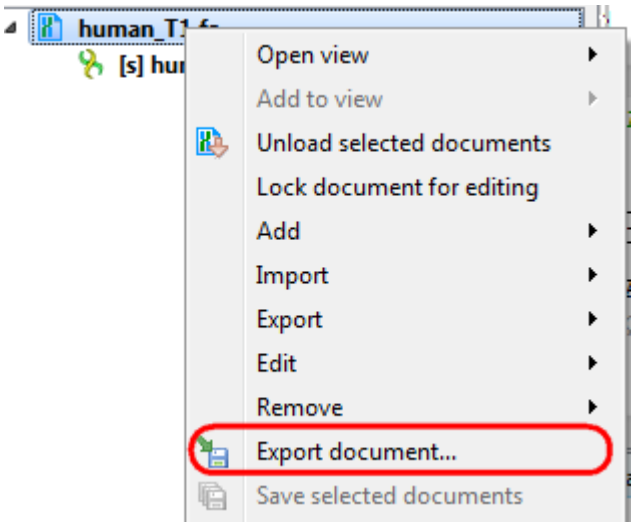
Opening Containing Folder

To open a containing folder of the *document* that is already present in the current *project* select it in the *Project View* and click on the *Open containing folder* context menu item.

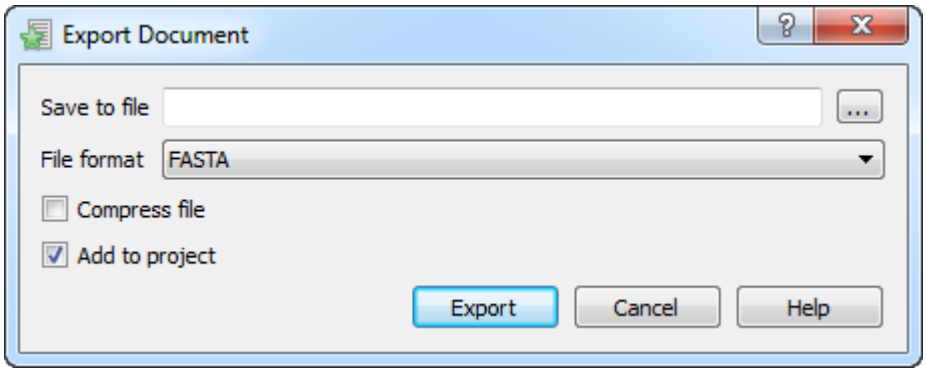
Exporting Documents

If a document has a format that supports writing in UGENE (see the *Supported File Formats* chapter), you can export the document to a new document in a required format.

To do it use the *Export document* item in the context menu:



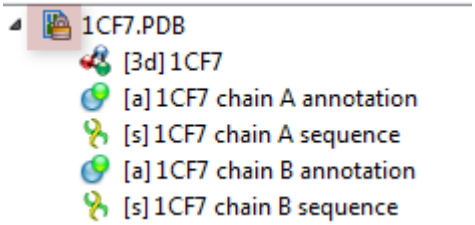
The following dialog appears:



Here you may select the name of the output file in the *Save to file* field and, optionally, choose the format of the output file in the *File format* field. Use the *C ompress file* checkbox to compress the file. The *Add to project* checkbox, checked by default, adds the output file to the current project. After choosing all parameters click the *Export* button.

Locked Documents

The lock icon in the document element indicates that the document can't be modified:



UGENE does not allow modification of some formats that were created not by UGENE.

If UGENE is able only to read a document (see the *Supported File Formats* chapter), you can export the document objects to a file. To do it use the *built-in export utilities*.

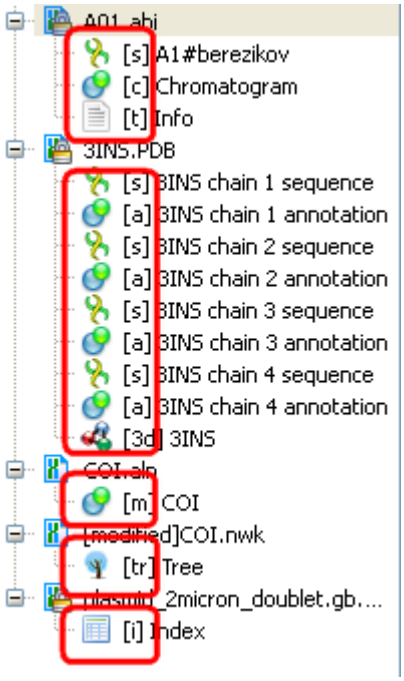
Also, you can *export* the document objects of unlocked documents.

Using Objects and Object Views

The document always contains one or more *objects*. An *object* is a structured biological data that can be visualized by different *Object Views*.

A single *Object View* can visualize one or several objects of different types. For example a single view can show a sequence, annotations for the sequence, 3D model for the part of the sequence or its chromatogram simultaneously.

The type of an object is indicated by the symbol in the square brackets and the icon near the object:



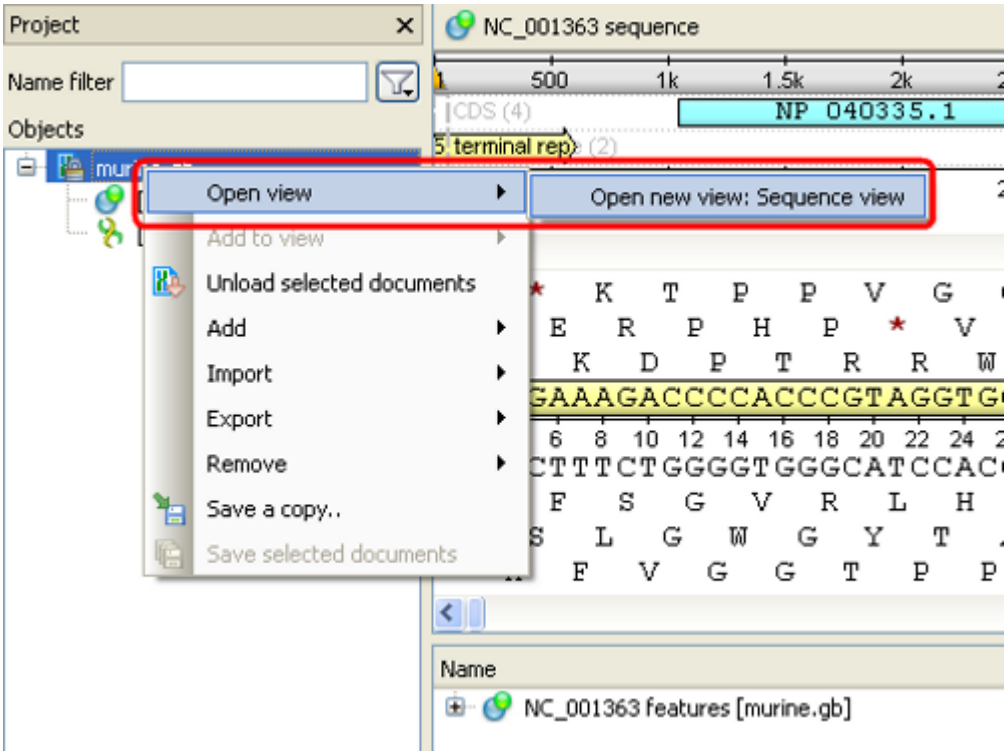
Below is the list of object types supported by the current version of UGENE.

Object types:

Symbol	Icon	Description
[3d]		A 3D model.
[a]		Annotations for DNA sequence regions.
[as]		An assembly.
[c]		Chromatogram data.
[i]		A file with index information for a set of other, usually large files.
[m]		A multiple sequence alignment.
[s]		A nucleic, protein or raw sequence.
[t]		A plain text.
[tr]		A phylogenetic tree.

You can edit names of particular objects, such as sequence objects, by selecting them in the *Project View* and then pressing F2. To be able to do so, the document containing the target object must be unlocked.

To see the list of all available views for a given object select the object and activate the context menu inside the *Project View* window and select the *Open view* submenu:



The picture above illustrates an option to visualize the selected DNA sequence object using the *Sequence View* — a complex and extensible *Object View* t hat focuses on visualization of sequence objects in combination with different kinds of related data: sequence annotations, graphs, chromatograms, sequence analysis algorithms. Note, that the *Sequence View* is described in more details in the separate *documentation section*.

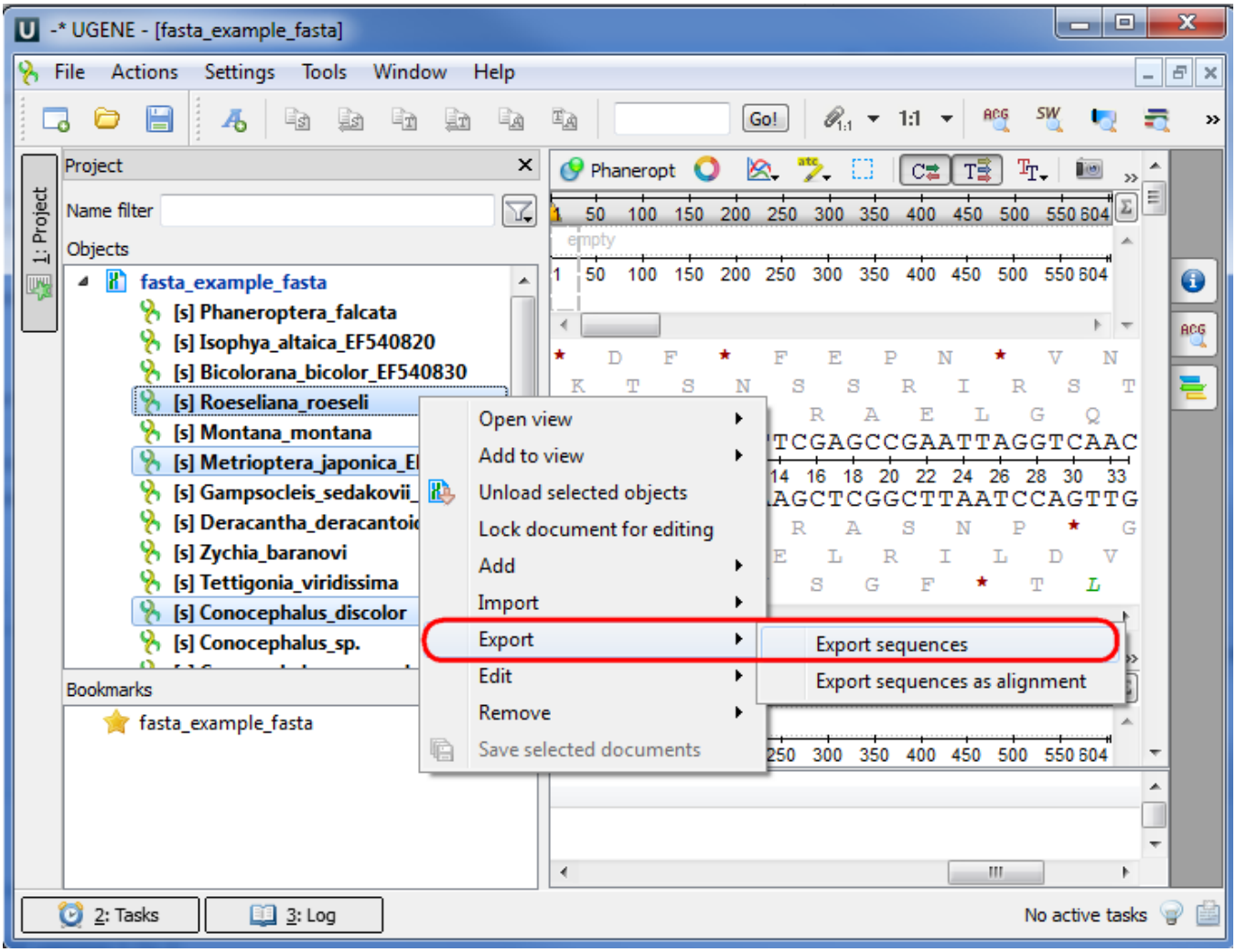
Exporting Objects

The document *objects* can be exported into a new document. For more details see the following chapters:

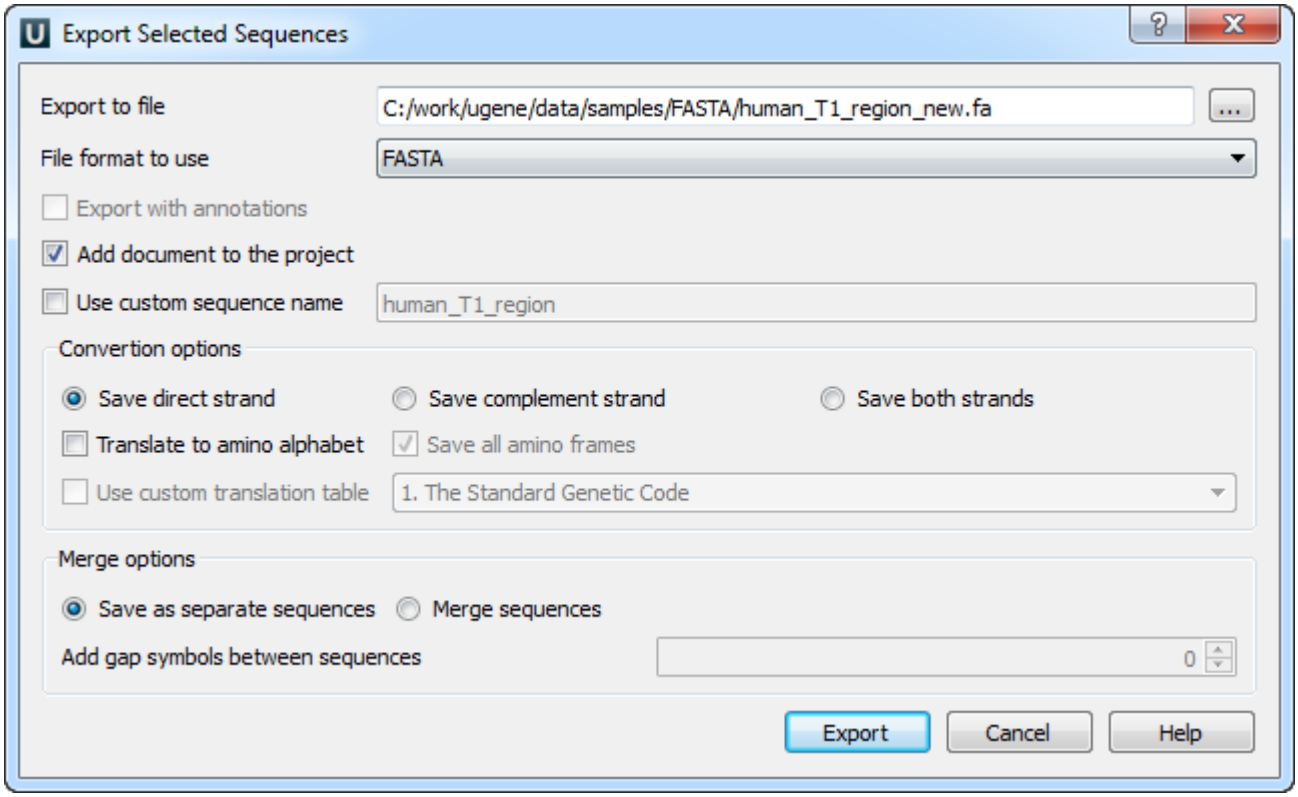
- Exporting Sequences to Sequence Format
- Exporting Sequences as Alignment
- Exporting Alignment to Sequence Format
- Exporting Nucleic Alignment to Amino Translation
- Export Sequences Associated with Annotation

Exporting Sequences to Sequence Format

Select a single or several sequence objects in the *Project View* window and click the *Export Export sequences* context menu item:



The *Export Selected Sequences* dialog will appear:



Here you can select the location of the result file and a sequence file format. You can choose to add newly created document to the current project and use custom sequence name. To do it check the corresponding checkboxes.

Use the *Conversion options* to choose a strand for saving sequence(s). Also you can translate sequence(s) to amino alphabet.

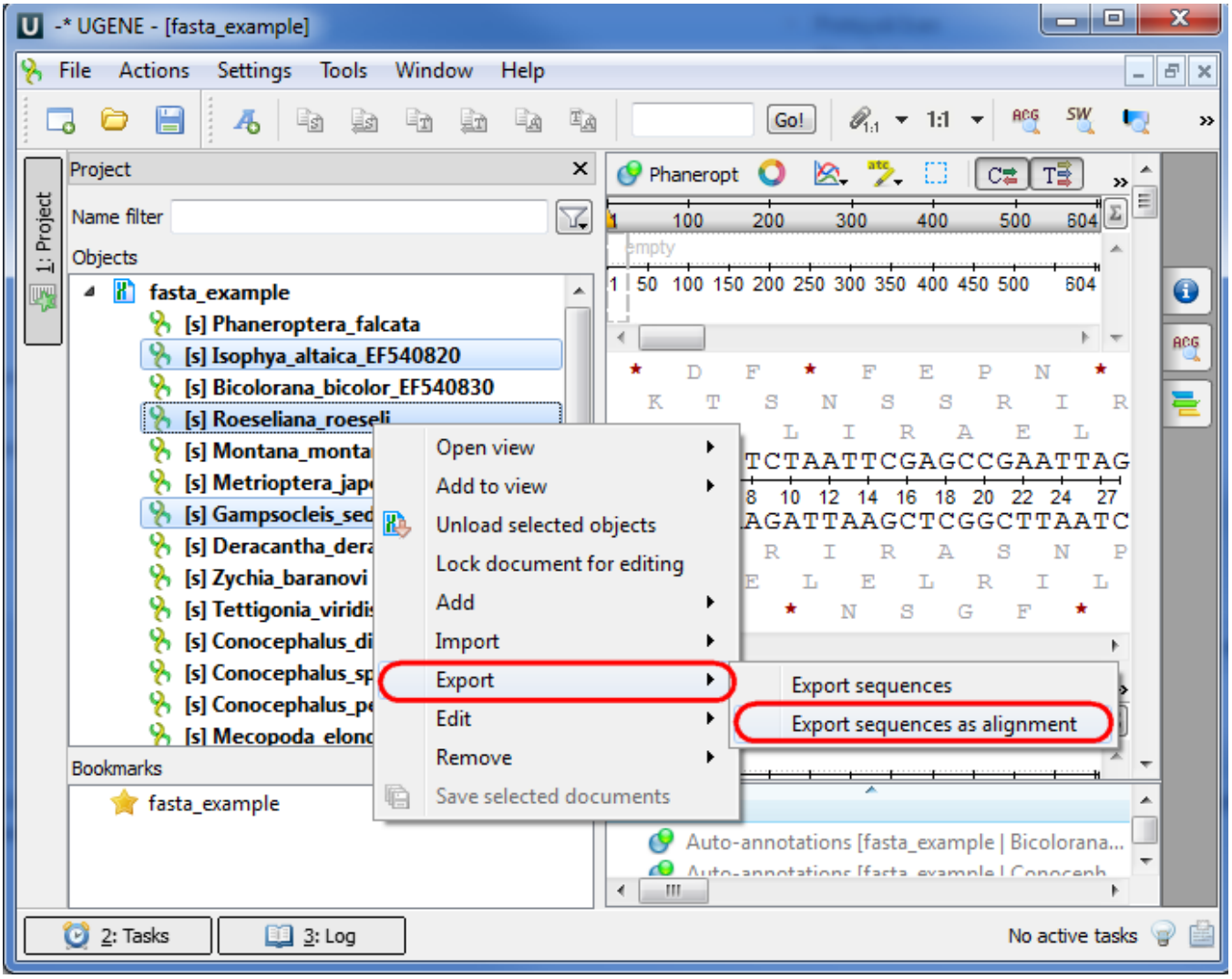
Also it is possible to specify whether to merge the exported sequences into a single sequence or store them as separate sequences. If you merge the sequences, you're allowed to select the gap symbols between sequences. This is the length of the insertion region between sequences that contain **N** sym bols for nucleic or **X** for protein sequences.

Export sequence with annotations

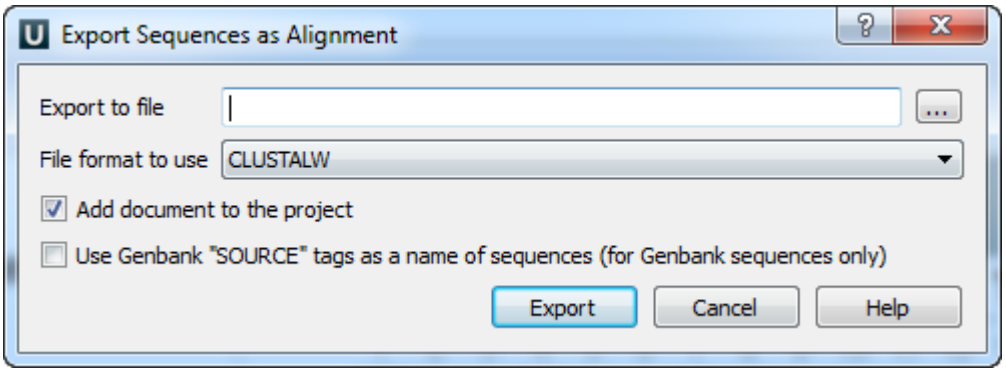
To export sequence with annotations choose Genbank or GFF format. The *Export with annotations* checkbox will ba available. Check the checkbox and sequence will be exported with annotations .

Exporting Sequences as Alignment

Suppose, we want to interpret FASTA file as multiple alignment. To do this, select a single or several sequence objects in the *Project View* window, click right mouse button to open the context menu and select the *Export Export sequences as alignment* item:

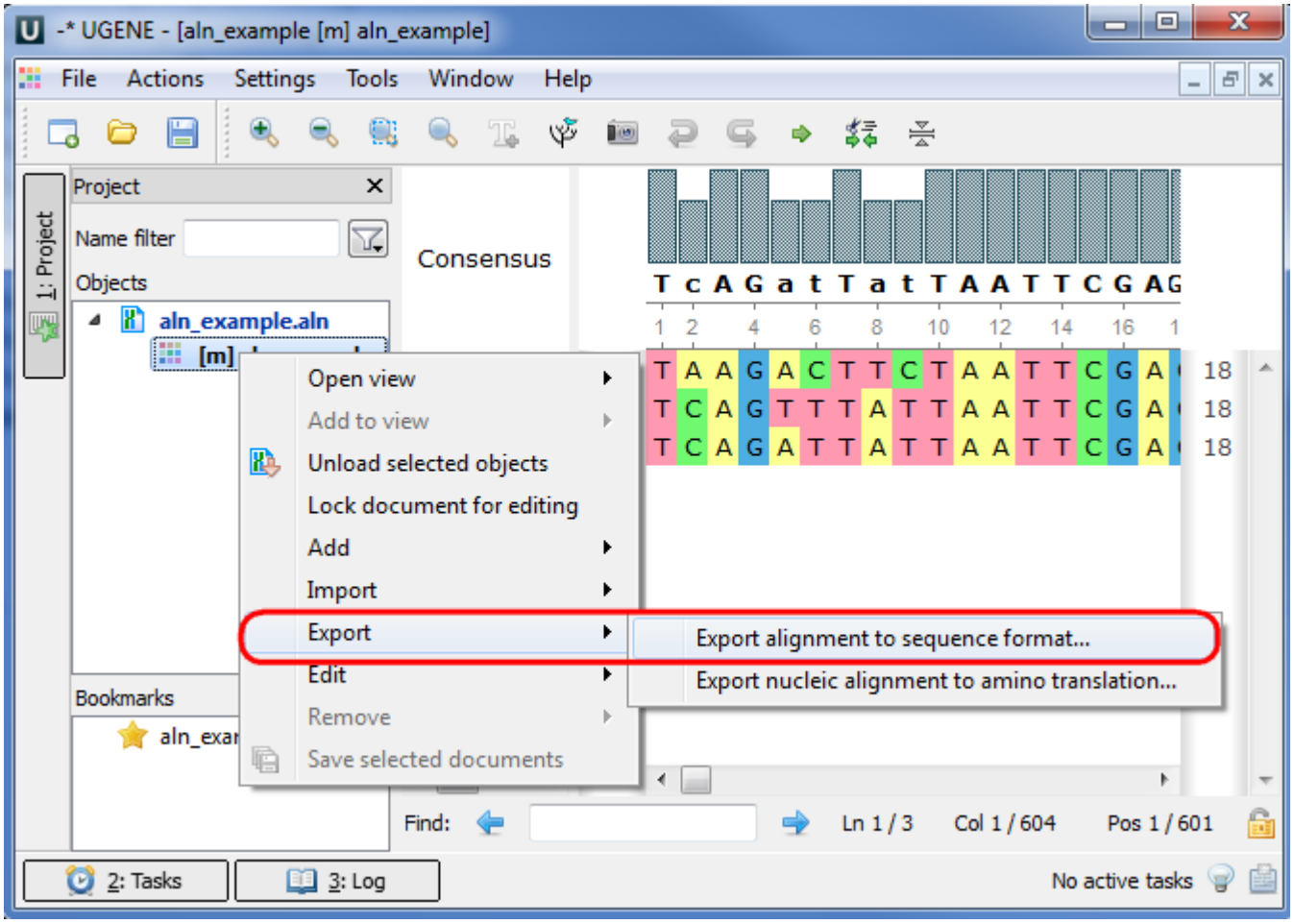


The *Export Sequences as Alignment* dialog will appear where you can point the result alignment file location, to select a multiple alignment file format, to use Genbank “SOURCE” tags as a name of sequences for Genbank sequences and optionally add the created document to the current project:

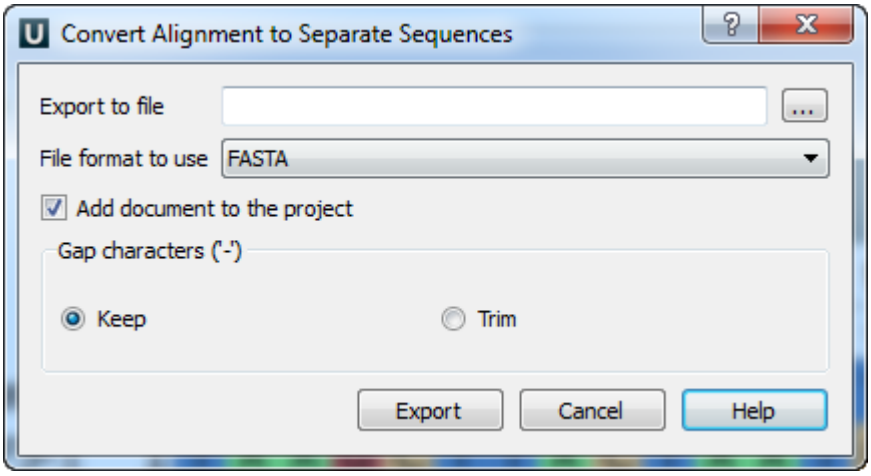


Exporting Alignment to Sequence Format

Select a single object with a sequence alignment in the *Project View* window and click the *Export Export alignment to sequence format* context menu item:



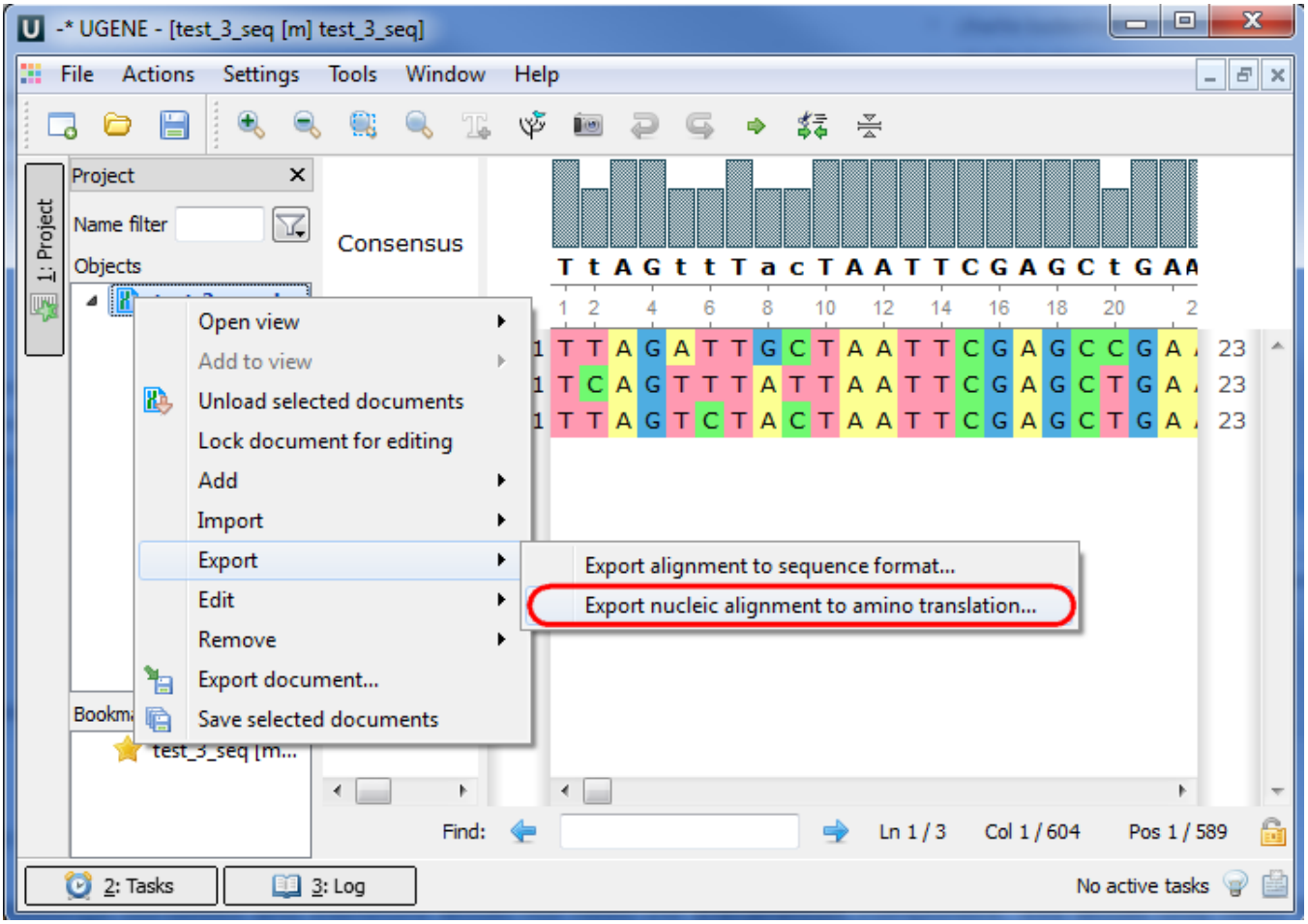
The *Convert Alignment to Separate Sequences* dialog will appear:



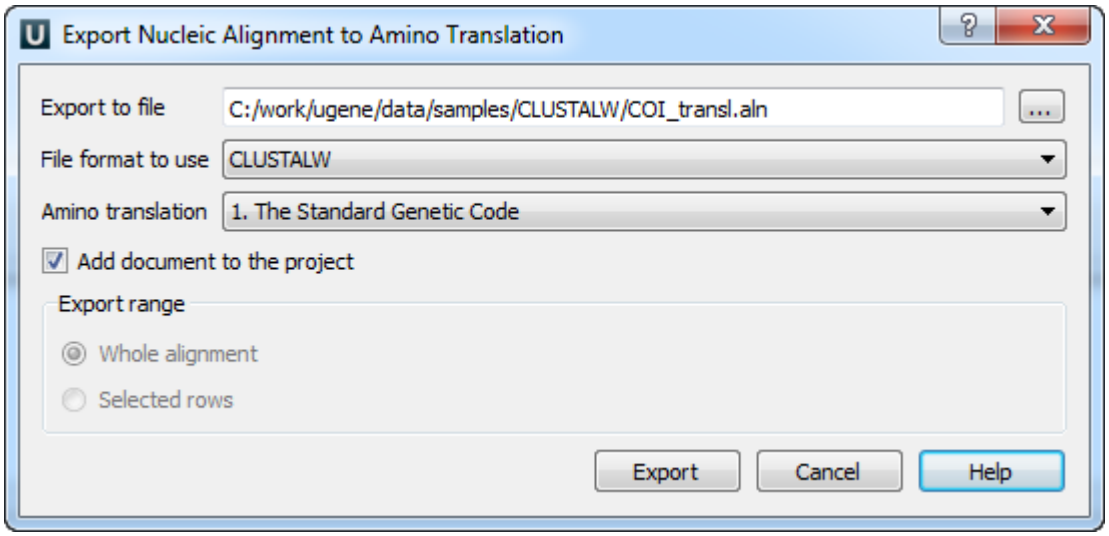
Here it is possible to specify the result file location, to select a sequence file format, to define whether to keep or remove gaps ('—' chars) in the aligned sequences and optionally add the created document to the current project.

Exporting Nucleic Alignment to Amino Translation

Select a single object with a nucleic sequence alignment in the *Project View* window and click the *Export Export nucleic alignment to amino translation* cont ext menu item:



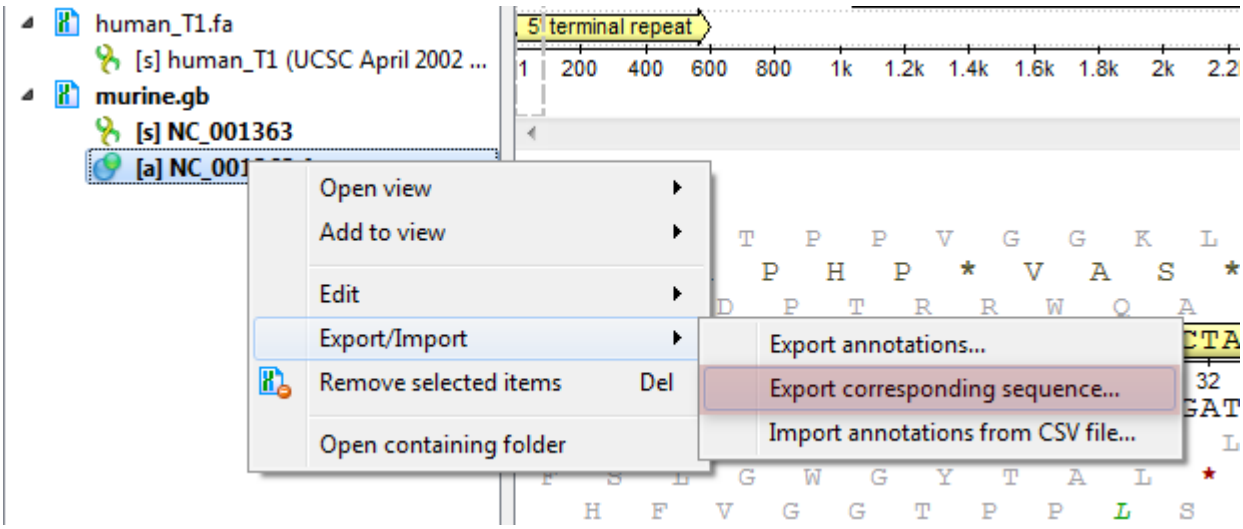
The *Export Nucleic Alignmemt to Amino Translation* dialog will appear:



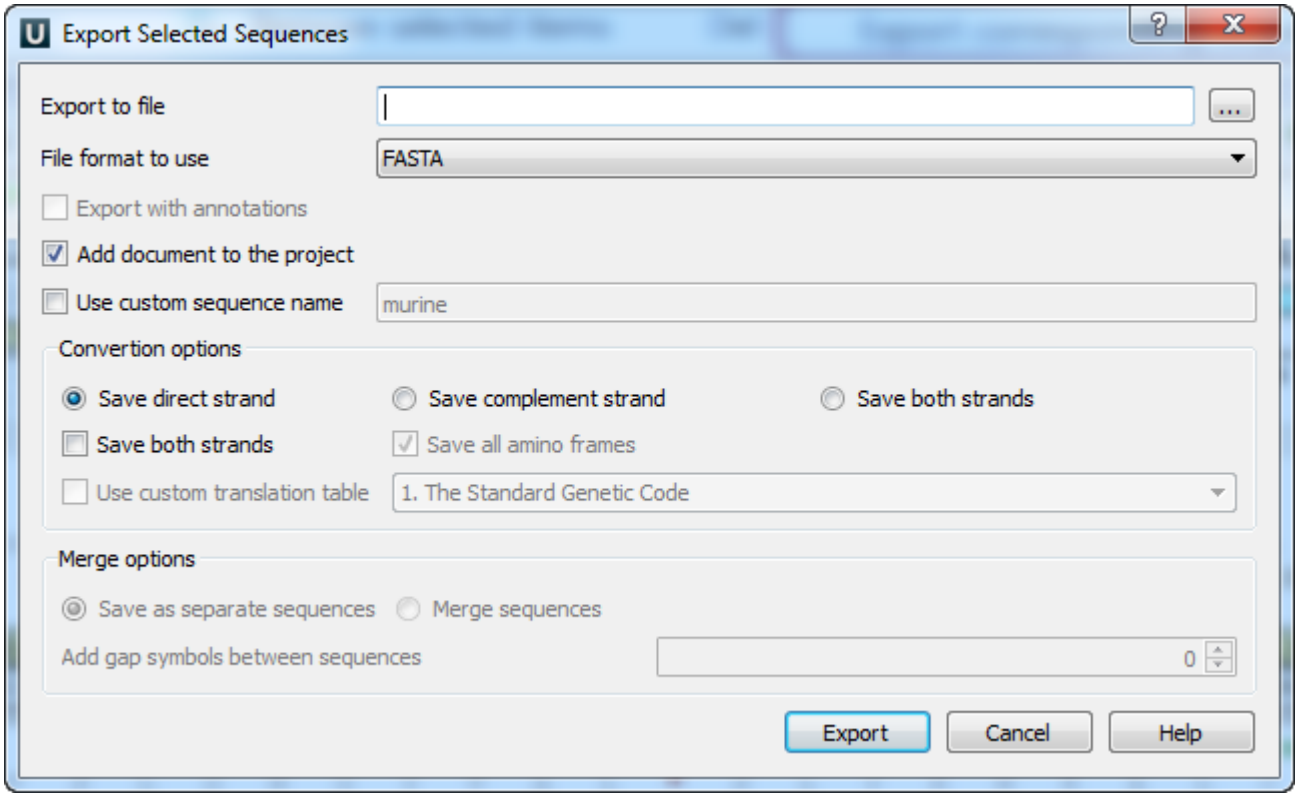
Here it is possible to specify the result file location, to select a file format and an amino translation, to export whole alignment or selected rows and optionally add the created document to the current project.

Export Sequences Associated with Annotation

In UGENE you can export a sequence associated with an annotation. To do it select the annotation in the *Project View* window and click the *Export/Import Export corresponding sequence* context menu item:



The *Export Selected Sequences* dialog will appear:



Here you can select the location of the result file and a sequence file format. You can choose to add newly created document to the current project and use custom sequence name. To do it check the corresponding checkboxes.

Use the *Conversion options* to choose a strand for saving sequence(s). Also you can translate sequence(s) to amino alphabet.

Also it is possible to specify whether to merge the exported sequences into a single sequence or store them as separate sequences. If you merge the sequences, you're allowed to select the gap symbols between sequences. This is the length of the insertion region between sequences that contain **N** sym bols for nucleic or **X** for protein sequences.

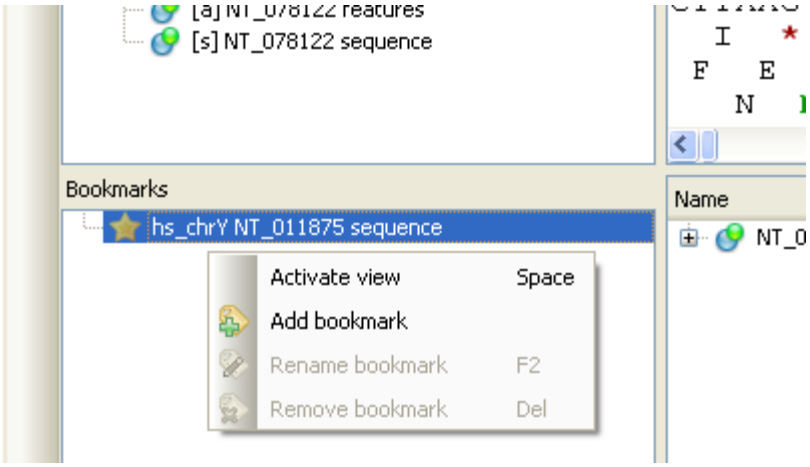
Export sequence with annotations

To export sequence with annotations choose Genbank or GFF format. The *Export with annotations* checkbox will ba available. Check the checkbox and sequence will be exported with annotations .

Using Bookmarks

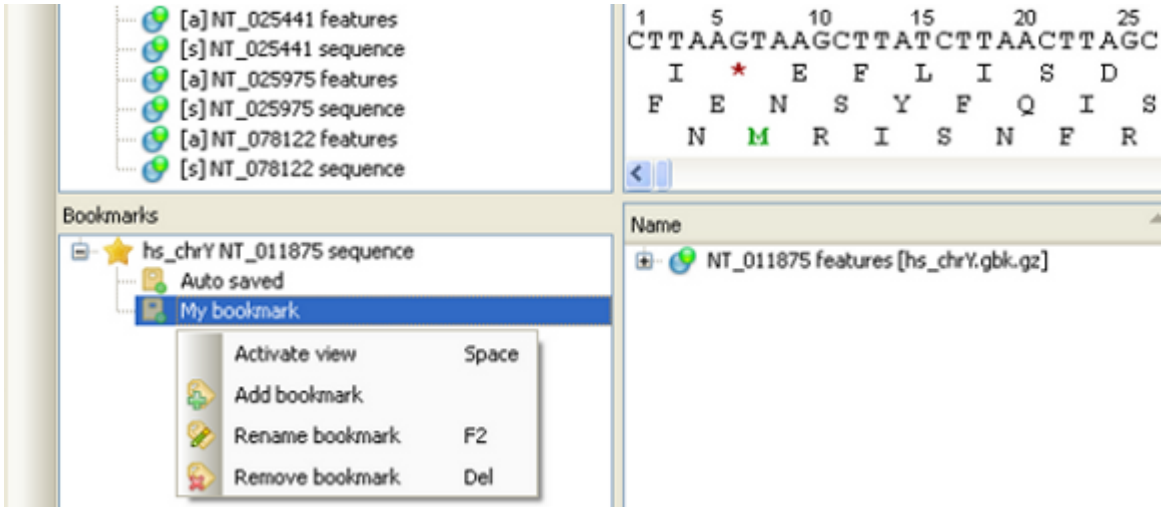
One of the most important features supported by most *Object Views* is an ability to save and restore visual view state. Saving and restoring visual state of an *Object View* enables rapid switching between different data regions and is similar to bookmarks used in Web browsers.

Initially an *Object View* is created as *transient*. It means that its state is not saved. To save current state of a view select an item with the view name in the *Bookmarks* part of the *Project View* windows and select the *Add bookmark* item in the context menu:



For every persistent view UGENE automatically saves the state of the view in the *Auto saved* bookmark when the view is closed.

Now, by activating bookmarks you can restore the original view state. For example for the *Sequence View* bookmarks you can store a visual position and zoom scale for the sequence region.

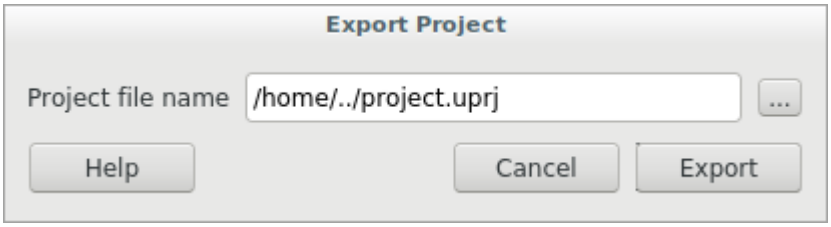


Use the F2 keyboard shortcut to rename a bookmark. To remove a bookmark press the Delete key.

UGENE has limited set of built-in *Object Views*. Extensions modules or plugins can be used to adjust the existing views or to add new views to the tool.

Exporting Project

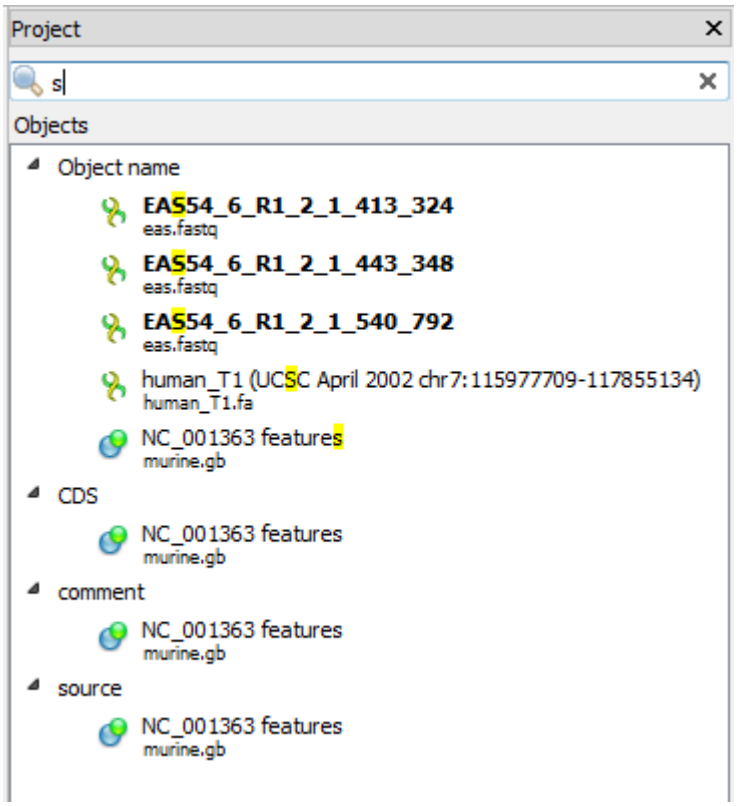
All the open documents and bookmarks (along with the corresponding views states) can be saved within a project file. To do so, select *File Export Project*. It will invoke the *Export project* dialog, where you can select the destination of the project file.



To load a saved project later, select *File Open* and specify the path to the project file.

Search in Project

Use the search field in the project view to search in the whole project:



Options Panel

The *Options Panel* is available in the *Sequence View* and in the *Assembly Browser*. By default, it is closed. To open a tab of the Options Panel click on the corresponding icon at the right side of a *Sequence View* or *Assembly Browser* window. To close the tab click again on the tab icon.

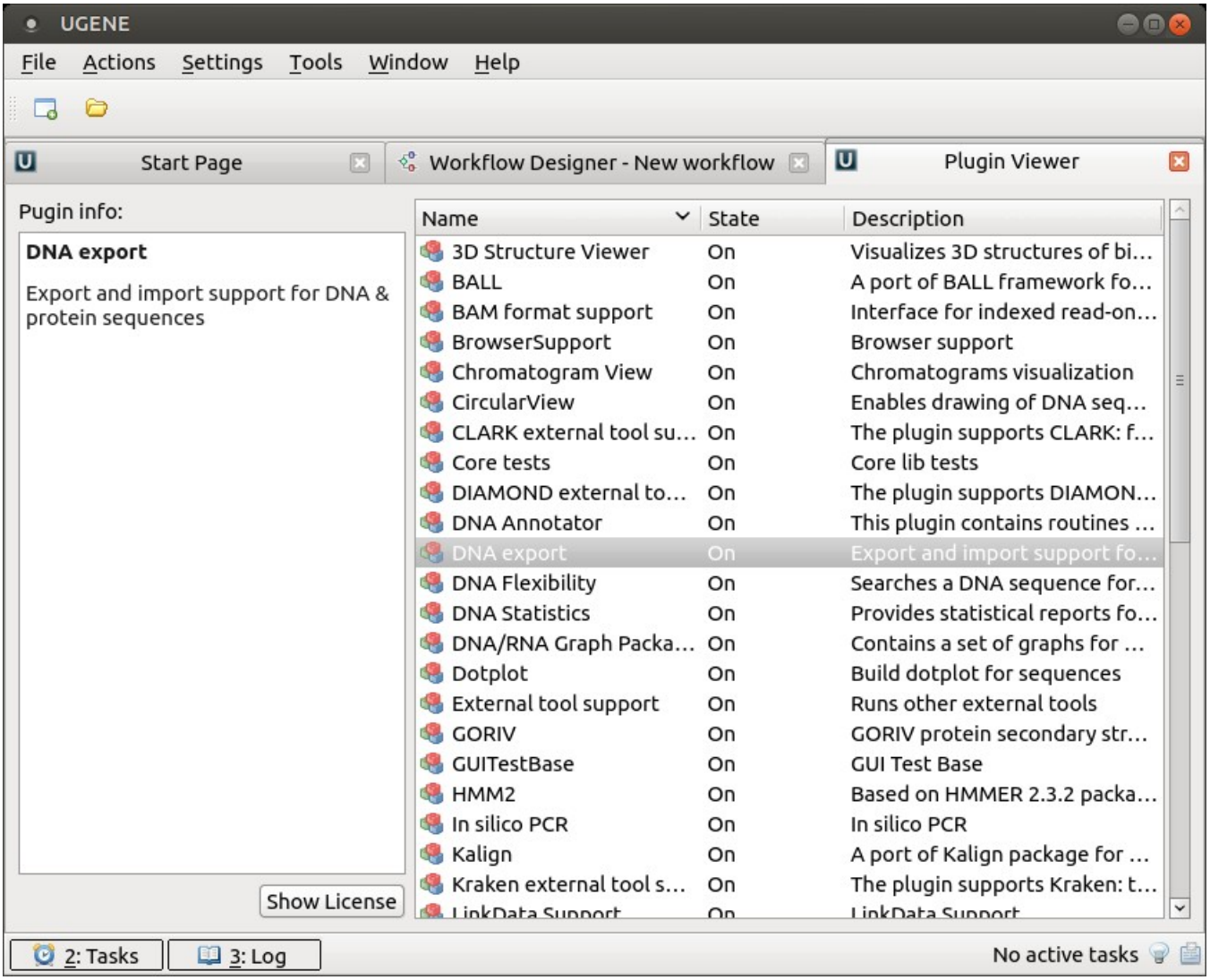
More detailed information about different *Options Panel* tabs can be found in the following chapters:

- *Options Panel in Sequence View*
  - *Information about Sequence*
  - *Search in Sequence*
  - *Highlighting Annotations*
- *Options Panel in Assembly Browser*
  - *Navigation in Assembly Browser*
  - *Assembly Browser Settings*
  - *Assembly Statistic*

Plugins Viewer

A *plugin* is a dynamically loaded module that adds new functionality to UGENE.

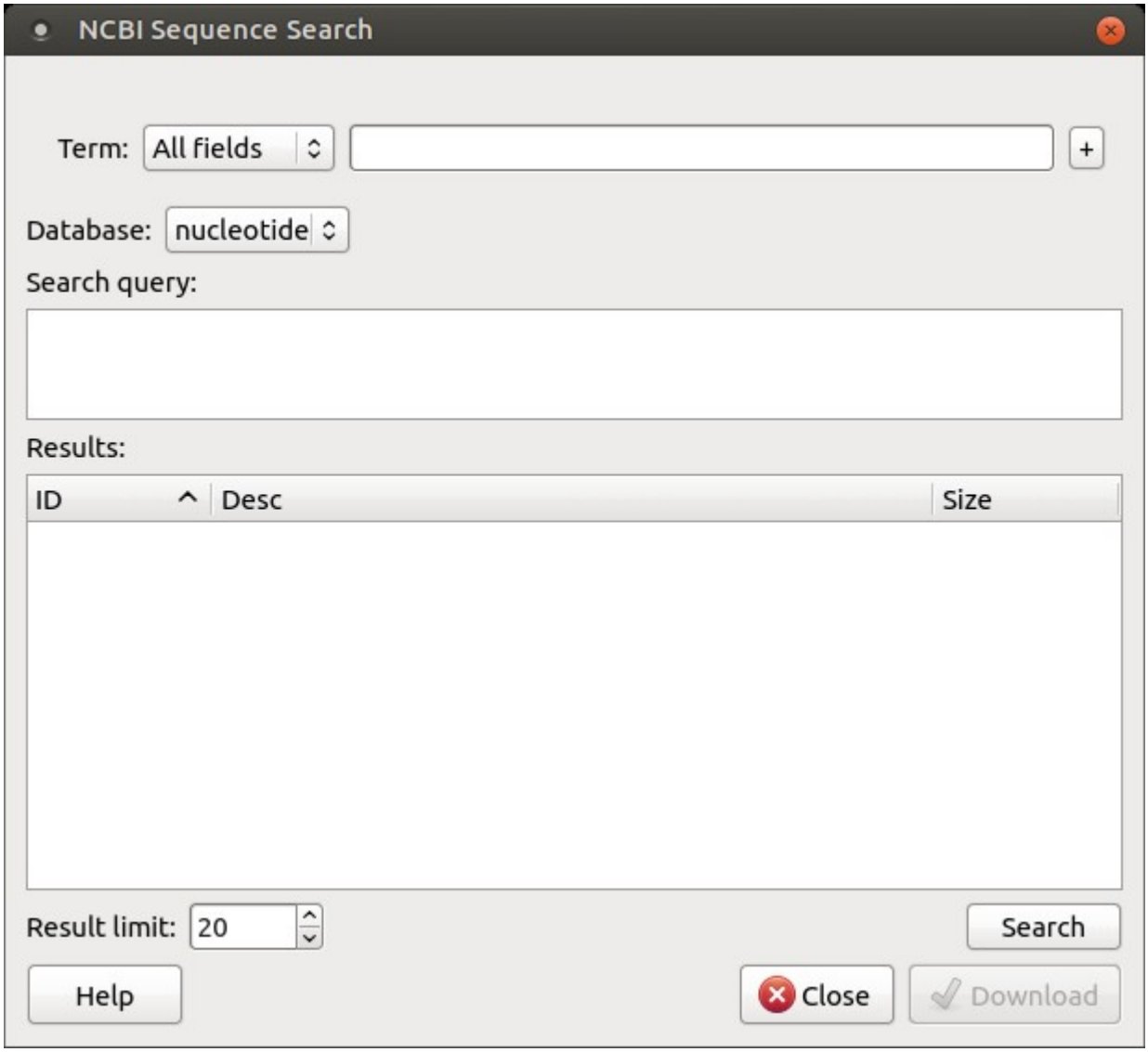
To manage plugins select the *Settings Plugins* main menu item. The *Plugin Viewer* window will appear:



The window shows the list of available plugins. Use the *Show License* button to show license each of them.

Searching NCBI Genbank

UGENE allows searching data in NCBI GenBank remote database. To do this open the following dialog by *File->Search NCBI Genbank* main menu:



To search data in the nucleotide or protein databases enter a general text query to the search field, select the database and click on the *Search* button. You can use a protein name, gene name, or gene symbol directly. Searching for a submitter or author name in the following format will produce the best results.

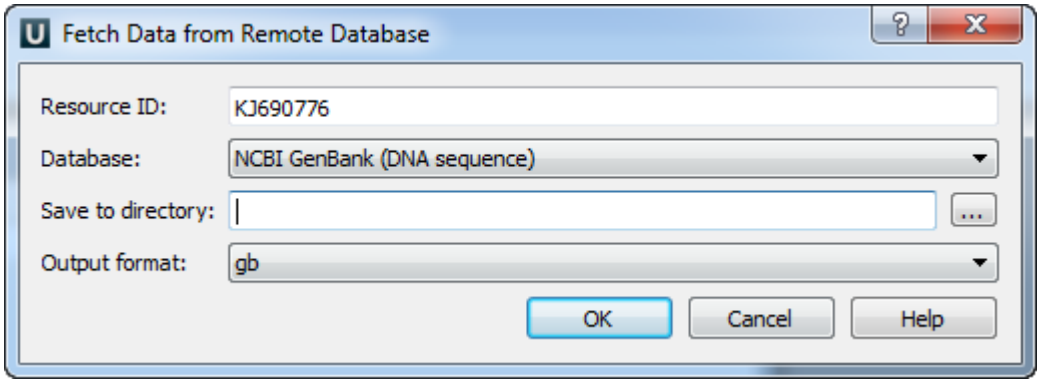
Use the boolean operator AND to find records that contain every one of your search terms, the intersection of search results.

Use the boolean operator OR to find records that include one of several search terms, the union of search results.

Use the boolean operator NOT to exclude records matching a search term.

To limit results use the *Result limit* field.

After you click the *Search* button, UGENE searches the biological objects and shows it in the *Results* field. You can download the object(s). Select one or several objects (for selecting several objects use the *Ctrl* button) and click the *Download* button. The dialog will appear:



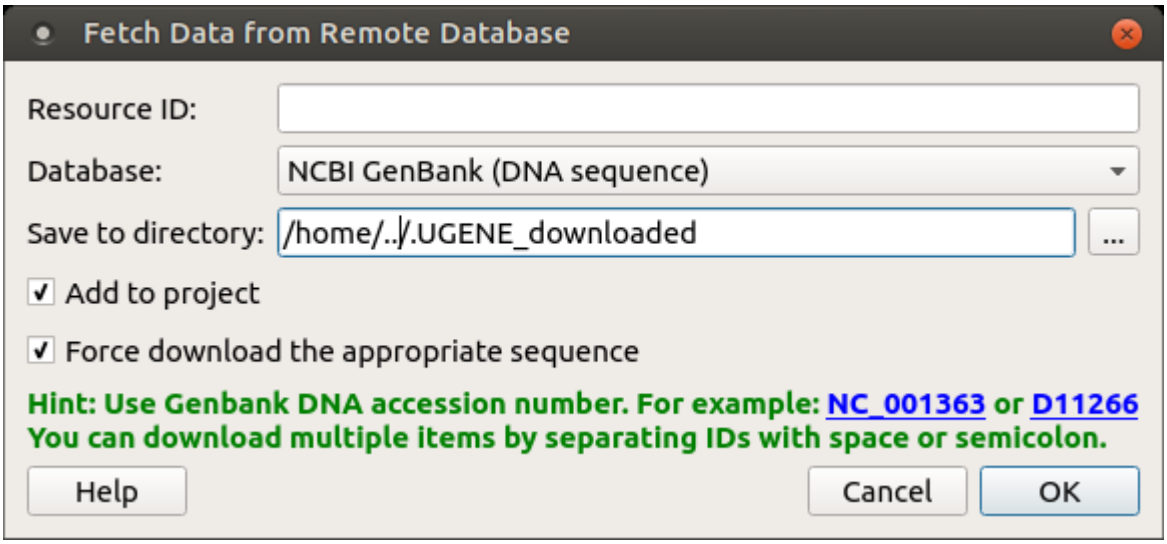
After you click the *OK* button, UGENE downloads the biological objects and adds it to the current *project*.

Fetching Data from Remote Database

UGENE allows fetching data from remote biological databases such as NCBI GenBank, NCBI protein sequence database, and some others.

To fetch data select the *File Access remote database...* item in the main menu.

The dialog will appear:



Here you need to enter the unique id of the biological object and choose a database. The following databases are available: NCBI Genbank (DNA sequence), NCBI protein sequence database, ENSEMBL, PDB, SWISS-PROT, UniProtKB/Swiss-Prot, UniProtKB/TrEMBL. Unique identifiers are different for various databases. For example, for NCBI GenBank such unique id could be Accession Number or NCBI GI number. Optionally, you can add the document to a project and browse for a directory to save the fetched file to.

After you click the *OK* button, UGENE downloads the biological object (DNA sequence, protein sequence, 3d model, etc.) and adds it to the current *project*.

If something goes wrong check the *Log View*, it will help you to diagnose the problem.

UGENE Application Settings

To open the UGENE *Application Settings* dialog:

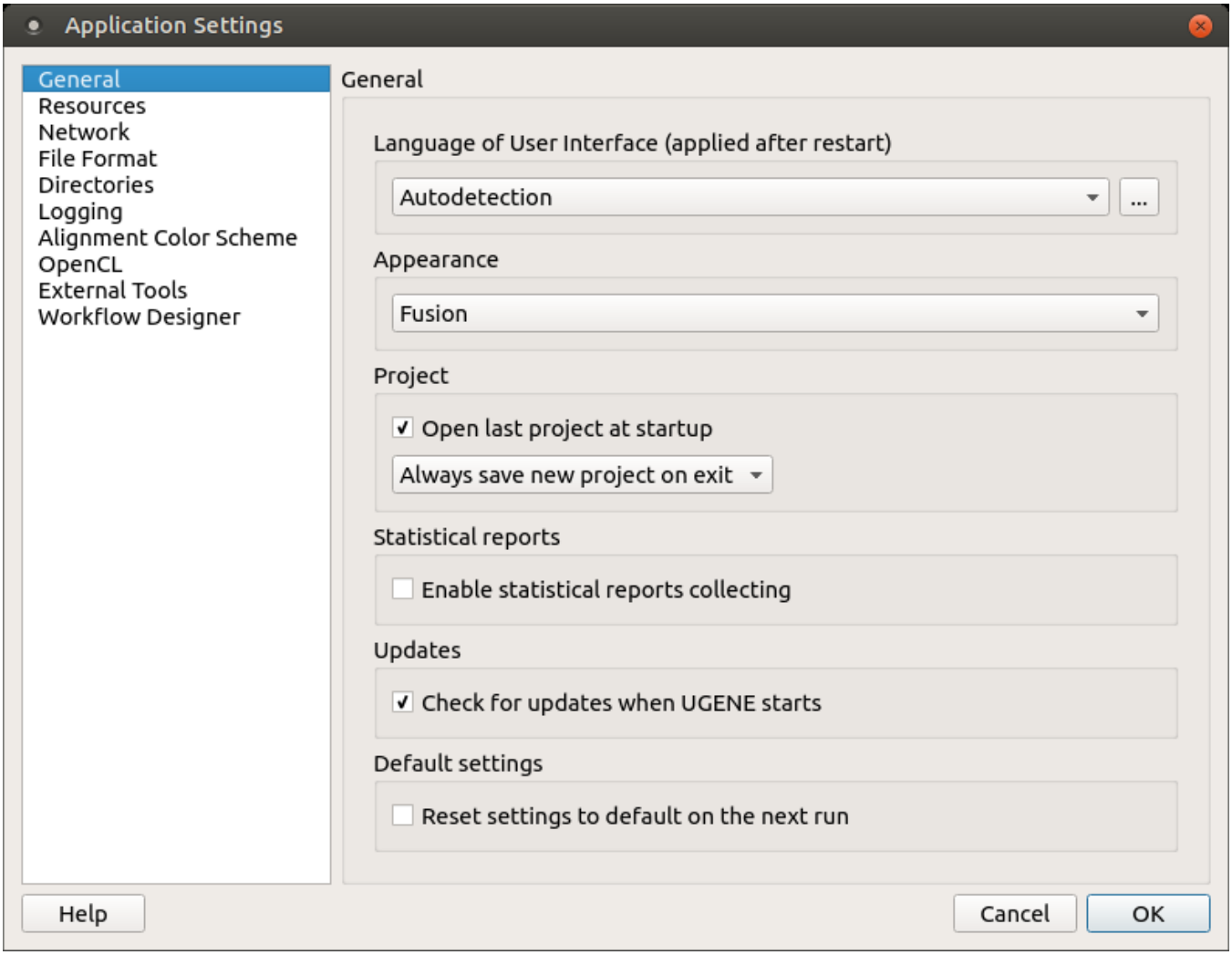
- On Windows or Linux: select "*Settings Preferences*" in the main menu.
- On macOS: select "*Unipro UGENE Preferences*" in the main menu

The following settings are available:

- General
- Resources
- Network
- File Format

- Directories
- Logging
- Alignment Color Scheme
- External Tools
  - Supported External Tools
  - Custom External Tools
- Workflow Designer Settings
- OpenCL

General



The following settings are available on the tab:

Language of User Interface (applied after restart) — here you can select UGENE localization. Currently, available localizations are EN and RU. The default value (*Autodetection*) specifies that UGENE should use the operating system regional options to select the localization. This setting is applied only after UGENE is reopened.

Appearance — defines the appearance of the application.

Open last project at startup — if the option is checked, the last project is opened when UGENE is started. Also, you can choose default settings for a saving project.

Enable statistical reports collecting — collects information about UGENE usage and sends it to the UGENE team to help improve the application.

The collected information includes:

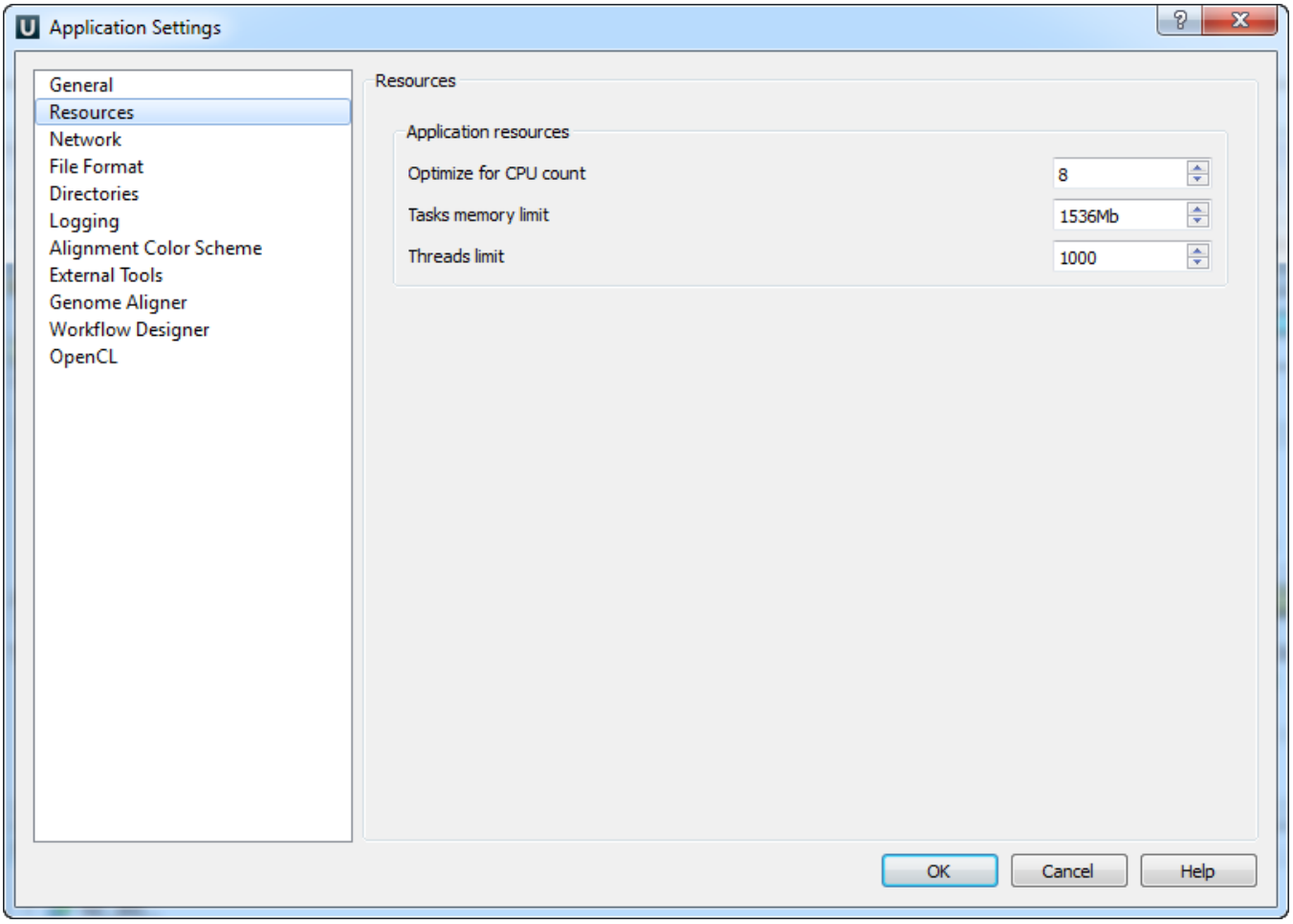
1. System info: UGENE version, OS name, Qt version, etc.
2. Counters info: number of launches of certain tasks (e.g. HMM search, MUSCLE align).

The collected information DOESN'T include any personal data.

Check for updates when UGENE starts - checks for updates when UGENE starts.

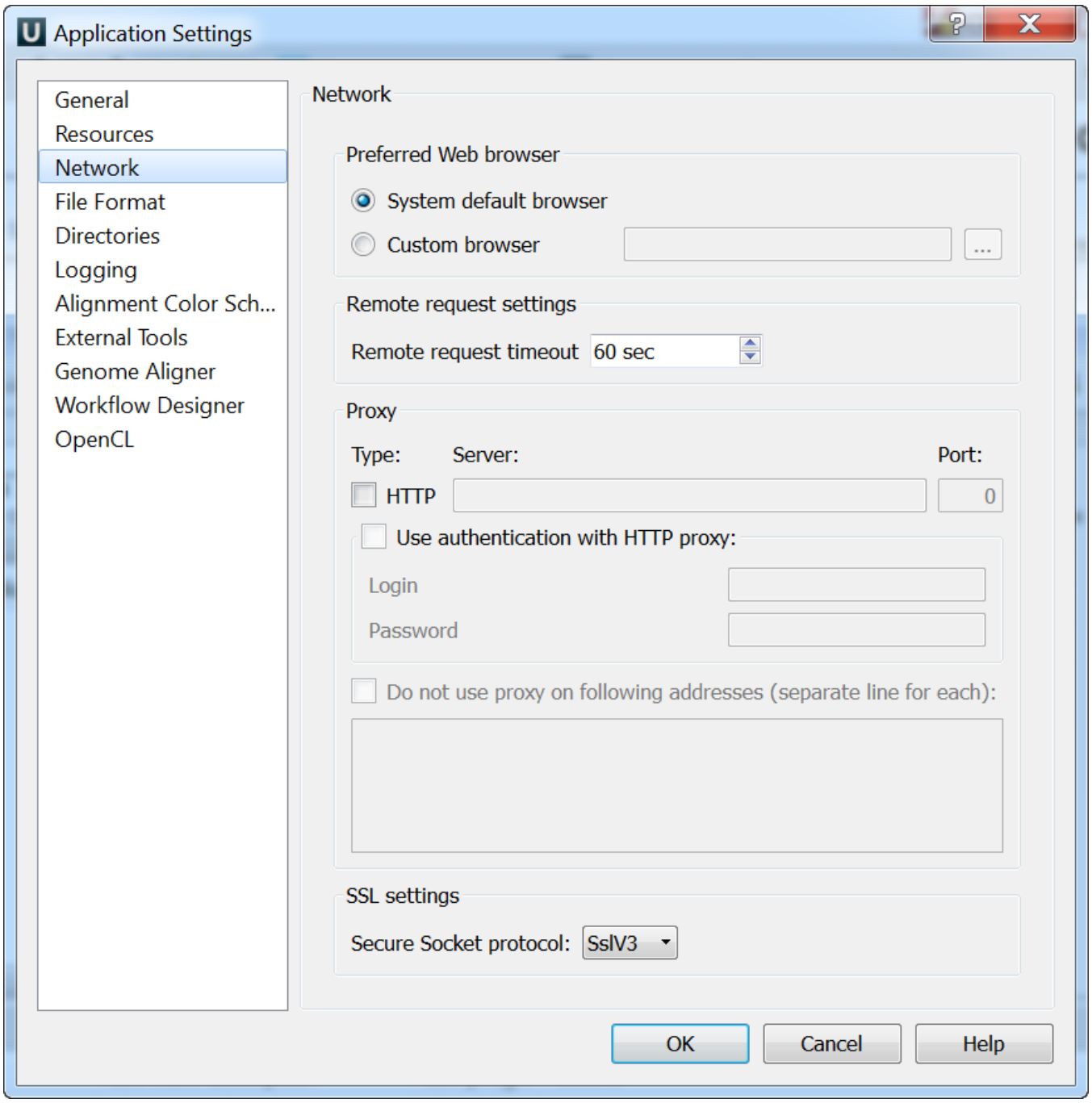
Default settings — this option resets the default settings on the next run.

Resources



On the *Resources* tab you can set resources that can be used by the application: *Optimize for CPU count*, *Tasks memory limit* and *Threads limit*.

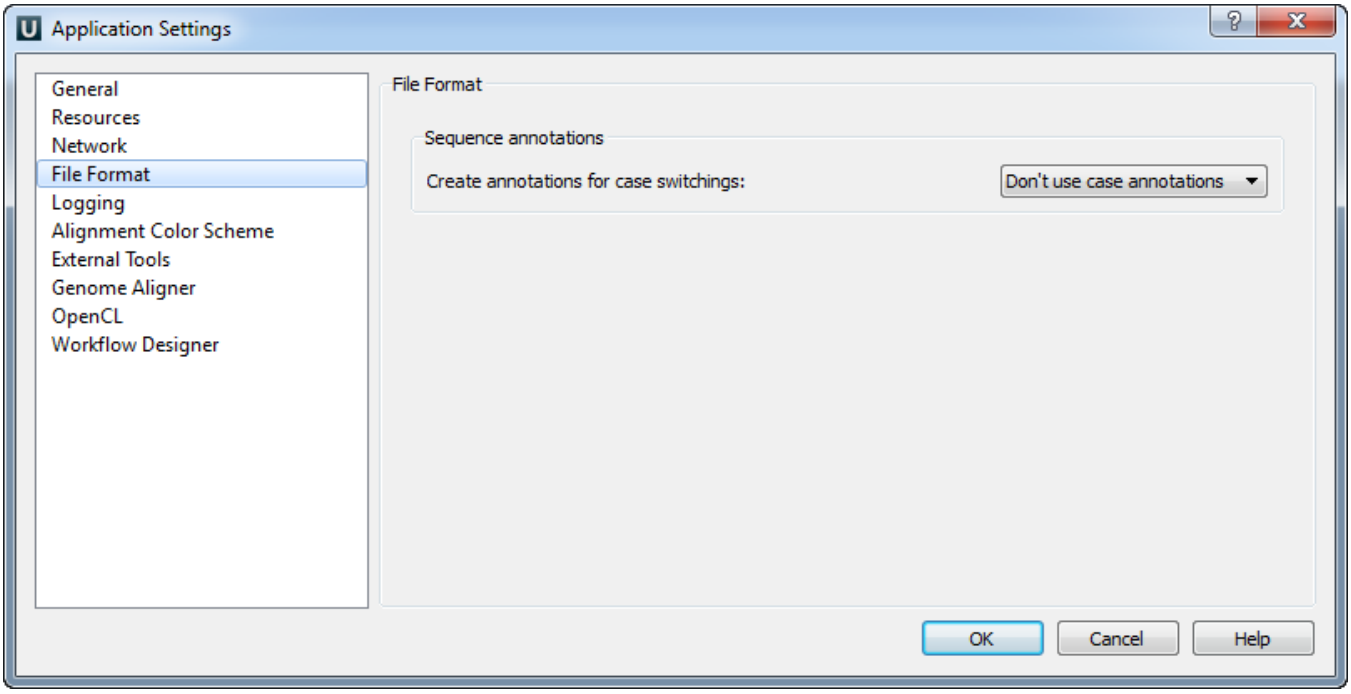
Network



On the *Network settings* tab of the dialog you can specify *Proxy* server parameters, select *SSL settings* and configure the *Remote request timeout*.

Preferred Web browser — you can use either *System default browser* or specify some other browser.

File Format

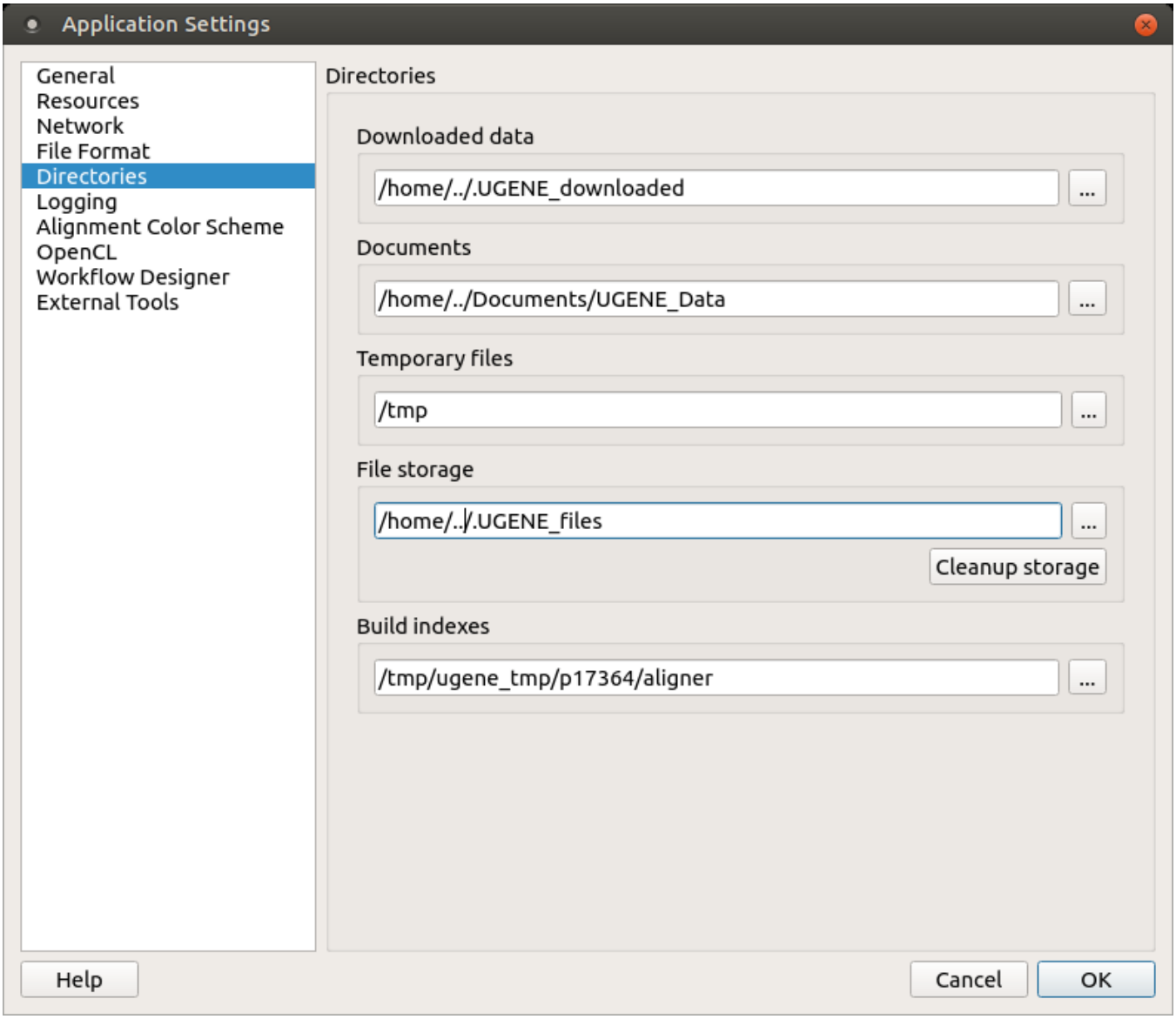


The *Sequence Annotations* settings allows to use upper/lower case annotations during the file reading process.

Format options:

1. *Don't use case annotations* (default mode) — usual sequence reading and writing.
2. *Use lower case annotation* — sequences are read and annotations with names *lower_case* are added. When these sequences are written to file then the case becomes like original the file case (the case is saved).
3. *Use upper case annotation* — there is a similar behavior but with "upper_case" annotations.

Directories



The following settings are available on the tab:

Downloaded data — specifies the path where files downloaded from the remote databases will be stored.

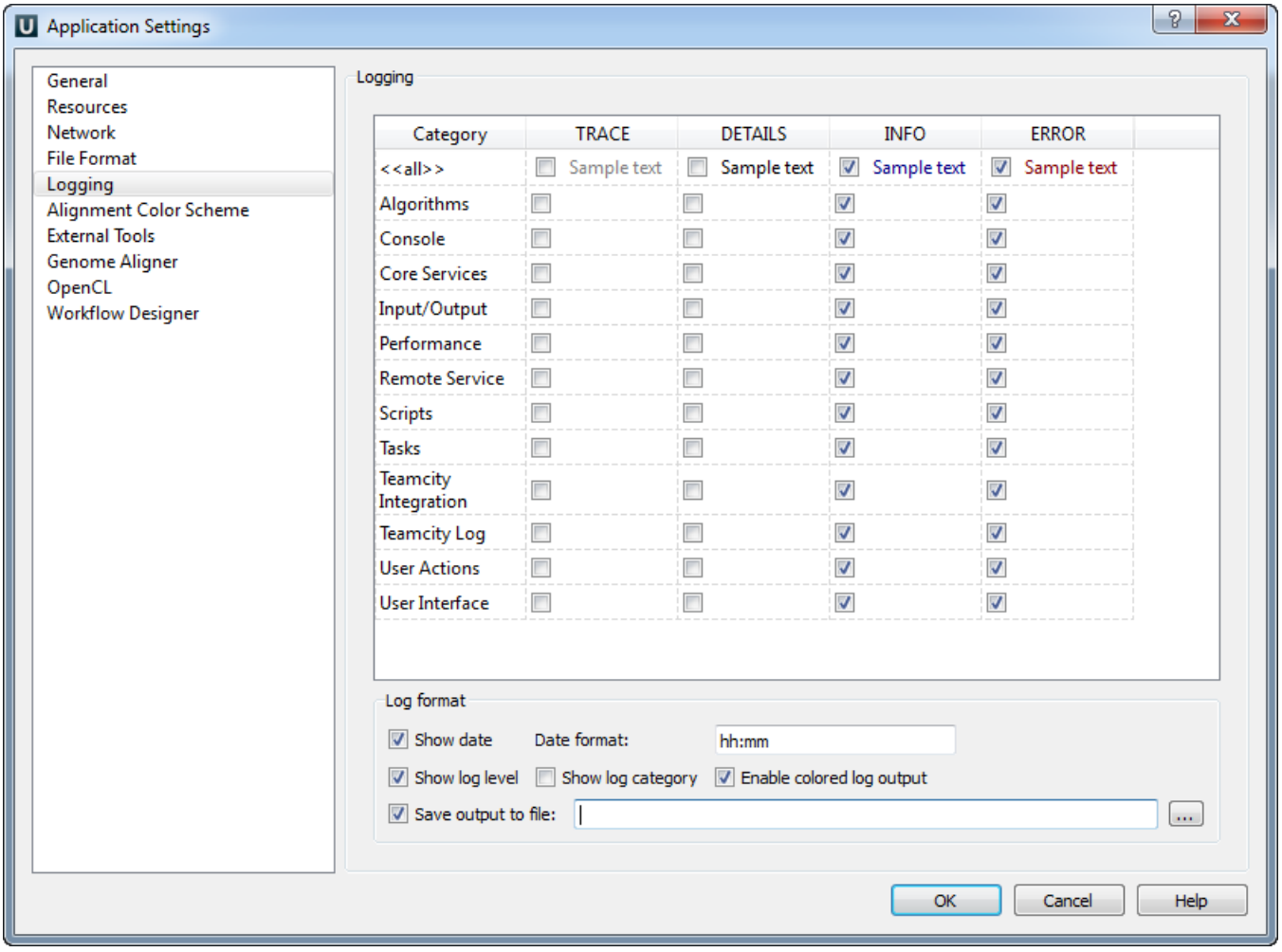
Documents - the path where will be stored UGENE data.

Temporary files — the path where will be stored temporary files.

File storage — the path where will be stored UGENE files.

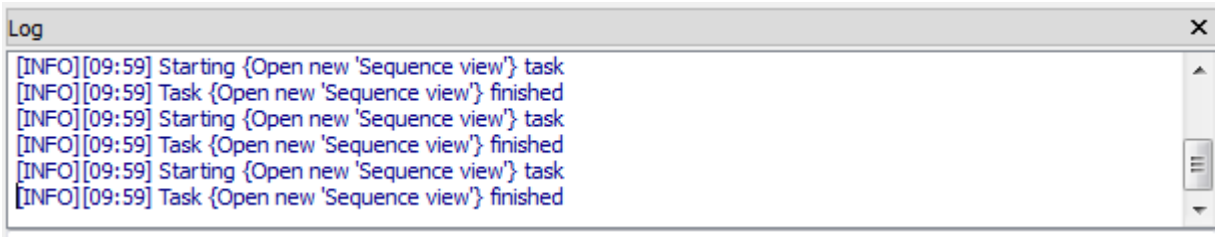
Build indexes - the path where will be stored UGENE Genome Aligner indexes.

Logging

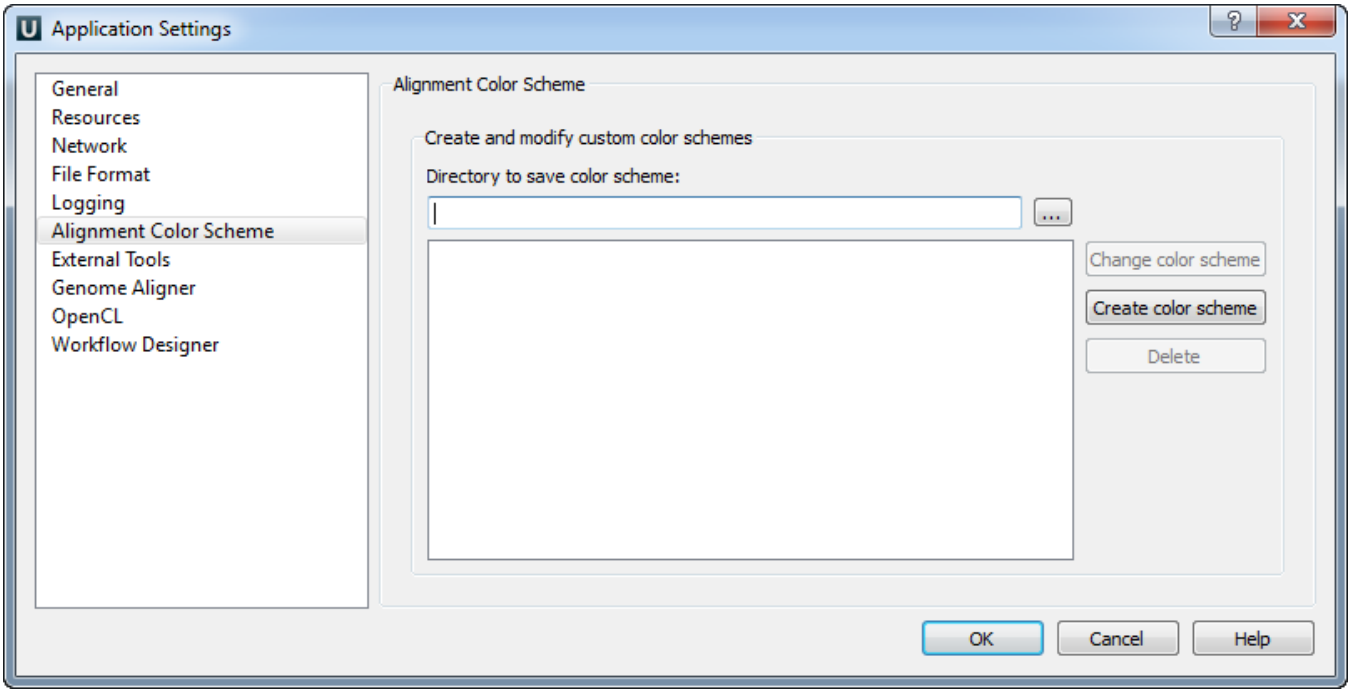


On the *Logging* tab you can select type of log information (*ERROR*, *INFO*, *DETAILS*, *TRACE*) for each *Category* that will be output to the *Log View*.

You can select format for each log message by checking the *Show date*, *Show log level* and *Show log category* options.



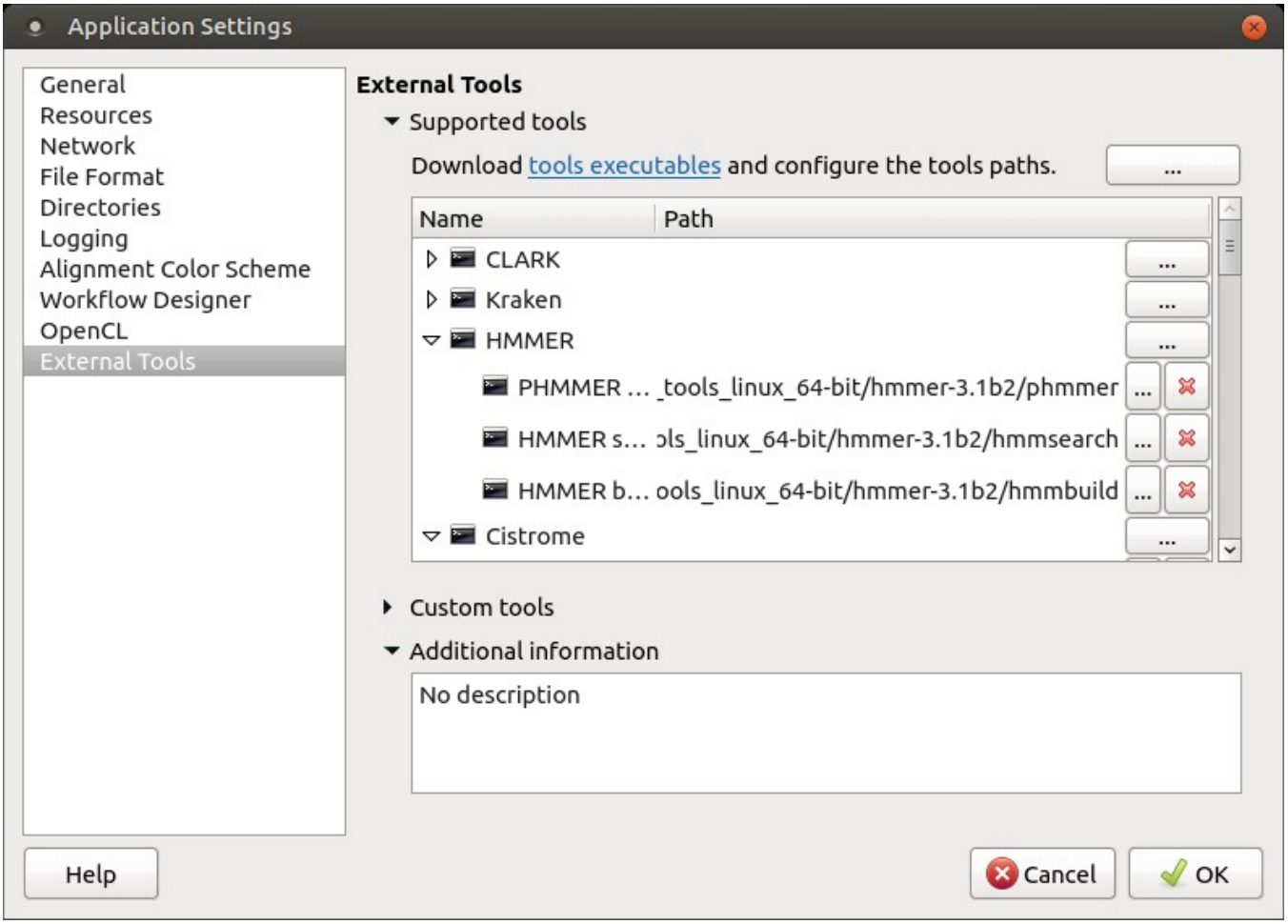
Alignment Color Scheme



On the *Alignment Color Scheme* tab you can create, change and delete custom color schemes.

External Tools

Set executable files for external tools:



Supported External Tools

Executable paths for external tools, already integrated into UGENE, can be configured in the "Supported tools" group of the "External tools" page of the "Application Settings" dialog. See the full list of the supported tools in the "Data Analysis Tools" chapter of the documentation.

Custom External Tools

A custom tool can be integrated into the UGENE GUI as a workflow element in the Workflow Designer - a UGENE component that allows joining tools into workflows.

As described in the "Custom Elements with External Tools" chapter of the Workflow Designer documentation, there are two options when one creates such workflow element:

- Set a local executable path
- Set an integrated external tool

The first option is intended for cases when a quick temporary integration of a tool into a workflow is enough.

The second option, although requires additional configuration steps, provides a more close integration with UGENE. It can be used by advanced UGENE users for providing a tool for their colleagues.

Below the preparatory configuration steps for the second option are described.

Create an XML configuration file for custom external tool

Create an XML file with the custom external tool description. The file should match the following XML schema:

```
<?xml version = "1.0" encoding = "UTF-8"?>
<xs:schema xmlns:xs = "http://www.w3.org/2001/XMLSchema">

<xs:element name = "ugeneExternalToolConfig">
```

```
    <xs:complexType>
        <xs:sequence>
            <xs:element name = "name" type = "xs:string" />
            <xs:element name = "id" type = "xs:string" />
            <xs:element name = "executableName" type = "xs:string" />
            <xs:element name = "executableFullPath" type = "xs:string" minOccurs ="0" />
            <xs:element name = "description" type = "xs:string" minOccurs ="0" />
            <xs:element name = "version" type = "xs:string" minOccurs ="0" />
            <xs:element name = "launcherId" type = "xs:string" minOccurs ="0" />
            <xs:element name = "dependencies" type = "xs:string" minOccurs ="0" />
        </xs:sequence>
        <xs:attribute name = "version" type = "xs:string" use="required" />
    </xs:complexType>
</xs:element>

</xs:schema>
```

There XML elements have the following meaning:

- **name** - the external tool name as it should be displayed in the UGENE GUI. This XML element is mandatory.
- **id** - the external tool unique identifier. The following characters are allowed: letters ('A'-'Z'), digits ('0'-'9), '_', '-'. This XML element is mandatory.
- **executableName** - the name of the tool executable file (with the file extension, if it is present). This XML element is mandatory.
- **executableFullPath** - a full path to the external tool executable. This XML element is optional. If the "executableFullPath" value is not specified, UGENE searches for a file with the specified "executableName" name in the same folder as the XML config file and sets the tool path accordingly. One can also specify either an absolute path to a file or a path, relative to the XML config. The path is displayed near the tool in the "Application Settings" dialog. It can be also manually set up by a user.
- **description** - a detailed description of the external tool. This XML element is optional.
- **version** - a version of the tool executable. This XML element is optional.
- **launcherId** - if the tool is supposed to be run with some script interpreter or it is a java application, then this element should contain an appropriate ID. Currently, UGENE supports the following launchers IDs: **USUPP_PYTHON2**, **USUPP_PERL**, **USUPP_RSCRIPT**, **USUPP_JAVA**. If the launcher is specified, it would be added to the launch command explicitly whenever you try to launch the external tool. The path to the launcher will be taken from the "Application Settings" (see the corresponding tools in the "Supported tools" group). If the element is not specified or it is empty, no additional launcher will be added to the command that will run the tool. This XML element is optional.
- **dependencies** - a comma-separated list of other external tool IDs the new tool is dependent from. If the dependency is not satisfied, then the dependent tool is marked as "not valid" in UGENE. This XML element is optional.

See also the following examples:

- A minimalistic config:

  ```
  <?xml version = "1.0" encoding = "UTF-8"?>
  <ugeneExternalToolConfig version = "1.0">
      <name>An imported custom tool</name>
      <id>CUSTOM_TOOL</id>
      <executableName>my_tool.py</executableName>
  </ugeneExternalToolConfig>
  ```

  This config describes only the tool name, the tool ID and the tool executable file path. If the XML config is imported from the same folder as the python script, the path will be set to the python script. The tool won't have any dependencies, its version will be "unknown".
- A config without dependencies and a launcher:

  ```
  <?xml version = "1.0" encoding = "UTF-8"?>
  <ugeneExternalToolConfig version = "1.0">
      <name>RAxML</name>
      <id>RAXML</id>
      <executableName>raxmlHPC</executableName>
      <executableFullPath>./my_folder/raxmlHPC</executableFullPath>
      <description>RAxML - Randomized Axelerated Maximum Likelihood</description>
      <version>8.2.12</version>
  </ugeneExternalToolConfig>
  ```

  This config describes an external tool with a description and a version. Path to the tool executable is relative: it is supposed that the folder structure is the following:

  ```
  my_config.xml
  my_folder/
     raxmlHPC
  ```

- A config with a dependency and a launcher:

```
<?xml version = "1.0" encoding = "UTF-8"?>
<ugeneExternalToolConfig version = "1.0">
    <name>HISAT2</name>
    <id>HISAT2</id>
    <executableName>hisat2</executableName>
    <executableFullPath>d:\work\hisat2.1\hisat2<executableFullPath>
    <description>HISAT2 is a fast and sensitive alignment program for
mapping next-generation sequencing reads (both DNA and RNA) to a
population of human genomes (as well as to a single reference genome).</description>
    <toolkitName>HISAT2</toolkitName>
    <version>2.1.0</version>
    <launcherId>UGENE_PERL</launcherId>
    <dependencies>UGENE_PERL</dependencies>
</ugeneExternalToolConfig>
```

This config describes a tool that is supposed to be run by the Perl interpreter. There is no Perl by default on Windows, but the UGENE installer provides a possibility to install Perl. The tool can notify UGENE that it should be launched by Perl, so UGENE will add a path to Perl to the command that runs the tool. The XML config also specifies an absolute full path to the executable file.

Prepare a package with the XML configuration file, the tool executable and, optionally, other files

The easiest way to "pack" the custom external tool is to:

- create a new folder with all files required for running the tool,
- specify a relative path in the XML configuration file described above,
- put the XML configuration file into the same folder as the main tool executable.

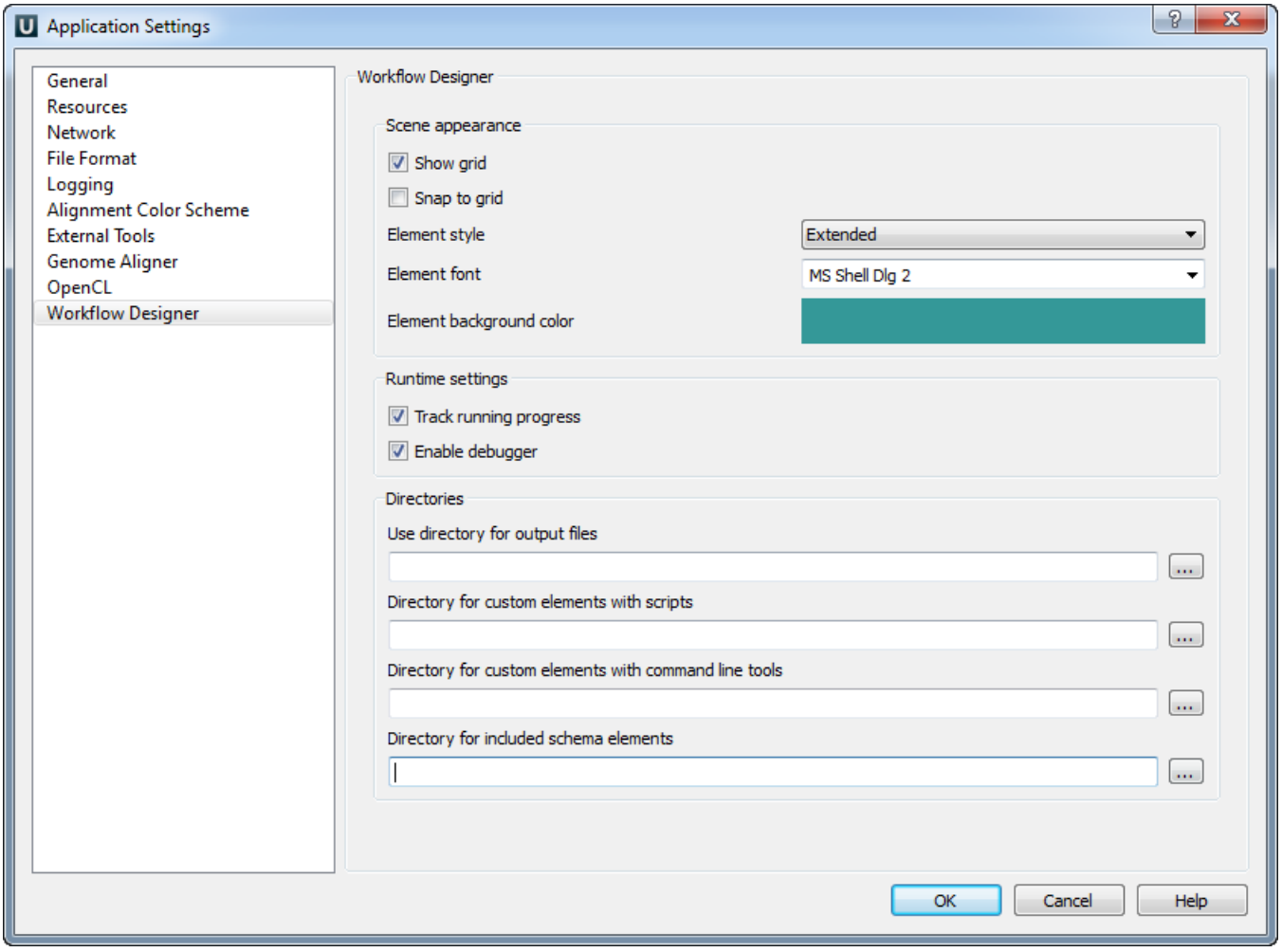
This "package" can be shared with other users, if required.

Import the tool into UGENE

To add the custom external tool one needs to import its XML configuration file in the "Custom tools" group on the "External Tools" page of the "Application Settings" dialog.

Workflow Designer Settings

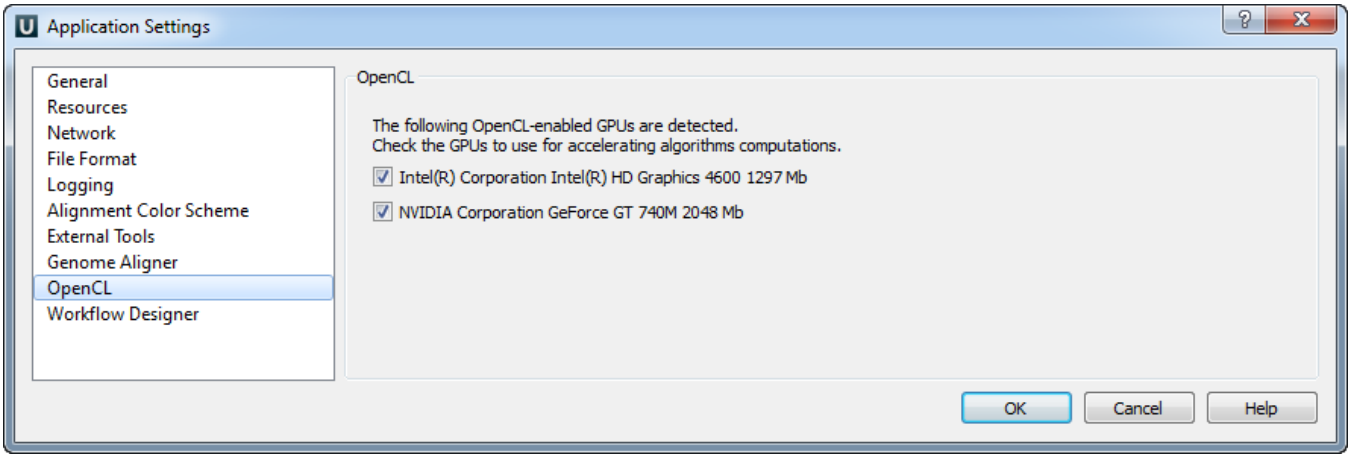
Use this tab to configure the Workflow Designer settings:



OpenCL

If you have a video card that supports OpenCL you can use it to speed up some calculations in UGENE.

To do it install the latest video card driver and check the corresponding check box:



Now you can, for example, use OpenCL optimization for the *Smith-Waterman algorithm*.

Sequence View

- Sequence View Components
- Global Actions
- Sequence Toolbars
- Sequence Overview
- Zoom View
- Details View
- Information about Sequence
- Manipulating Sequence
  - Show/Hide Sequence View Components
  - Showing Sequence in Multiple Lines
  - Translating Nucleotide Sequence
  - Zooming Sequence
  - Creating New Ruler
  - Going To Position
  - Selecting Sequence Region
  - Copying and Pasting Sequence
  - Editing Sequence
  - Searching in Sequence
  - Exporting Selected Sequence Region
  - Exporting Sequence of Selected Annotations
  - Exporting Sequence Image
  - Locking and Synchronize Ranges of Several Sequences
  - Multiple Sequence Opening
- Annotations Editor
  - "db_xref" Qualifier
  - Automatic Annotations Highlighting
  - The "comment" Annotation
- Manipulating Annotations
  - Creating Annotation
  - Selecting Annotations
  - Editing Annotation
  - Highlighting Annotations
    - Annotations Color
    - Annotations Visability
    - Show on Translation
    - Captions on Annotations
  - Creating and Editing Qualifier
  - Adding Column for Qualifier
  - Copying Qualifier Text
  - Finding Qualifier
  - Deleting Annotations and Qualifiers
  - Importing Annotations from CSV
  - Exporting Annotations

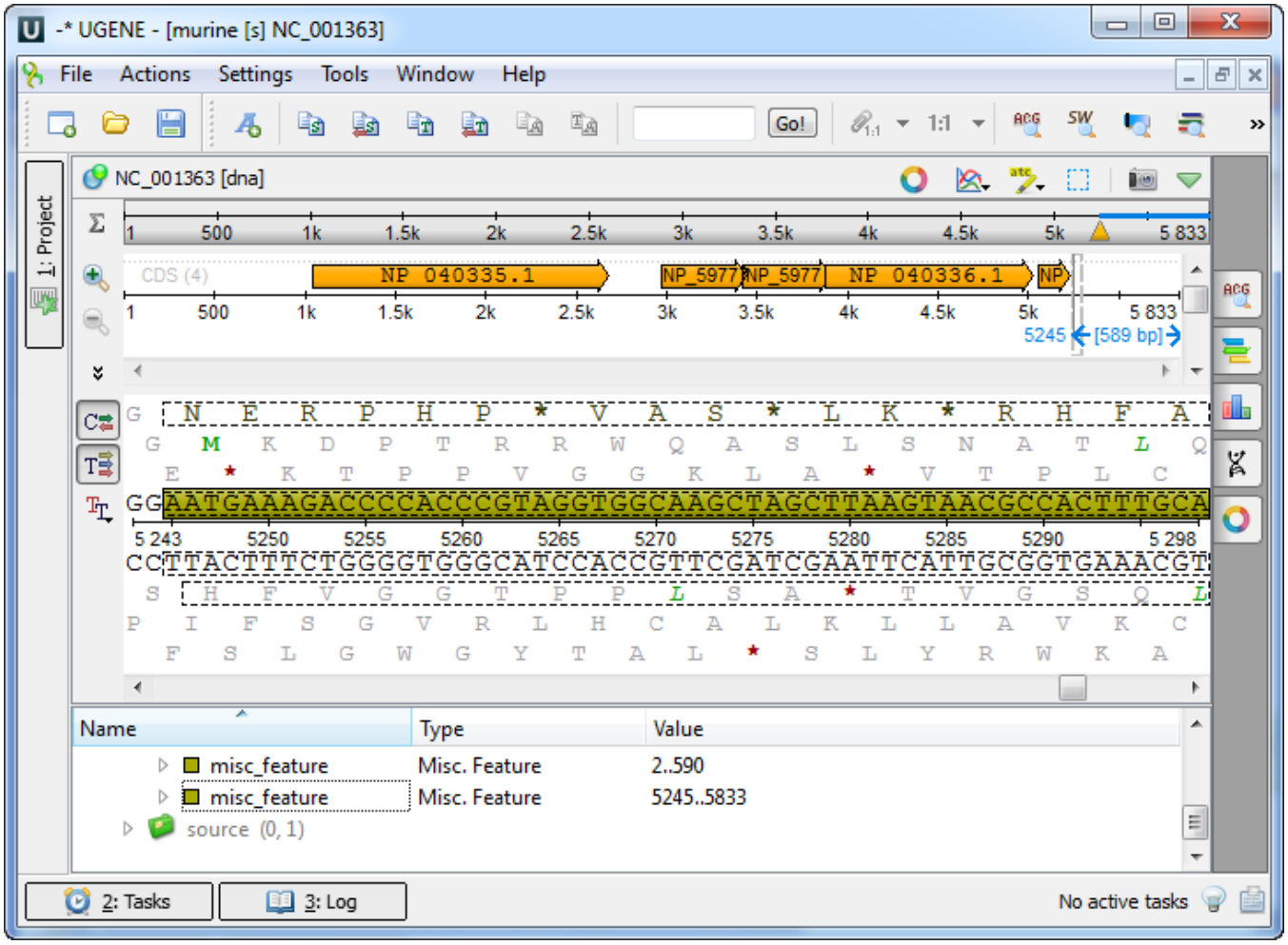
Sequence View Components

The *Sequence View* is one of the major *Object Views* in UGENE aimed to visualize and edit DNA, RNA or protein sequences along with their properties like annotations, chromatograms, 3D models, statistical data, etc.

For each file UGENE analyzes the file content and automatically opens the most appropriate view.

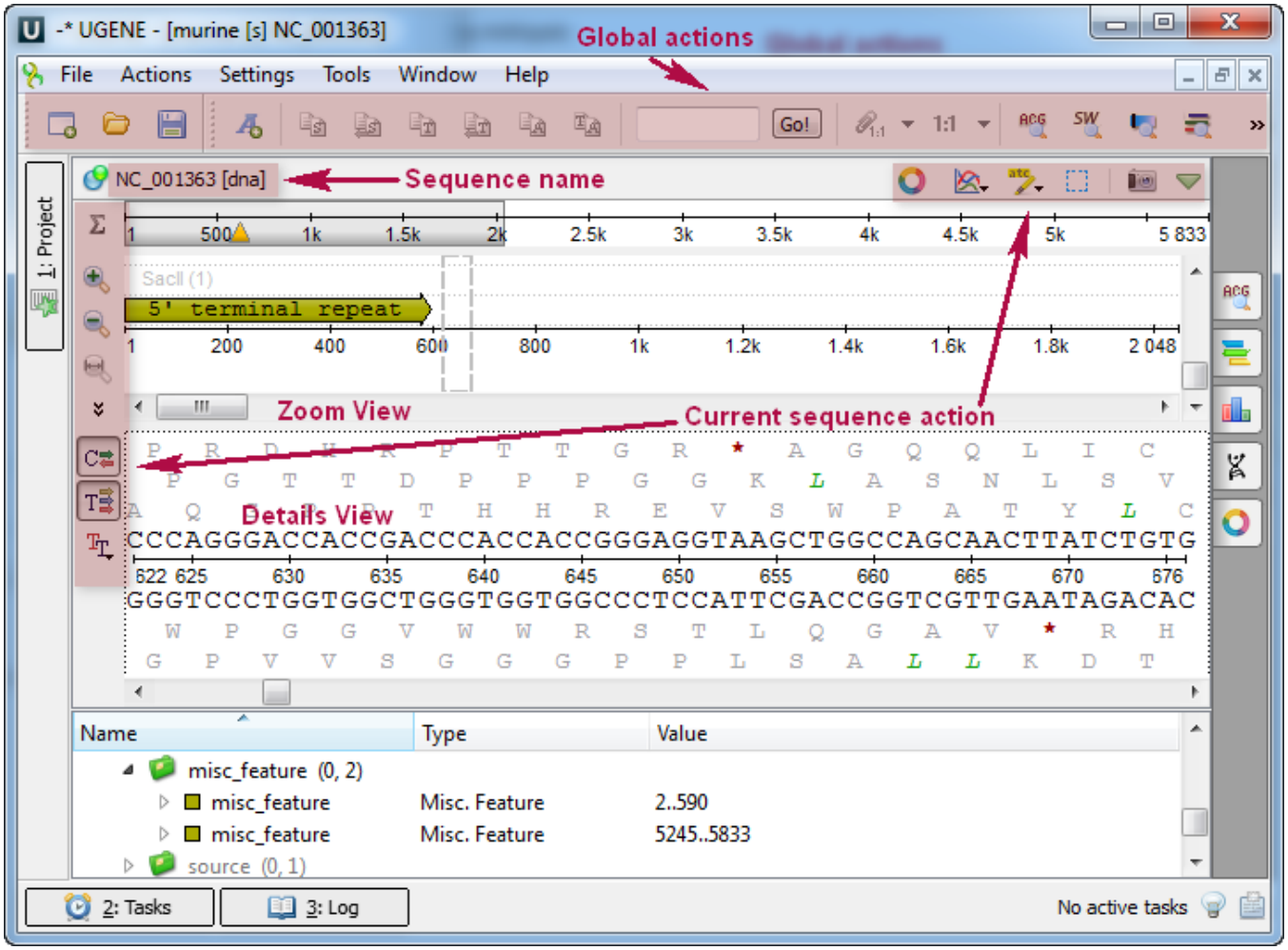
To activate the *Sequence View* open any file with at least one sequence. For example you can use the $UGENE/data/samples/EMBL/AF177870.emb file provided with UGENE.

After opening the file in UGENE the *Sequence View* window appears:



After the view is opened you can see a set of new buttons in the toolbar area. The actions provided by these buttons are available for all sequences opened in the view. In the picture below these buttons are pointed by the “Global actions” arrow.

Below the toolbar there is an area for a single or several sequences. For each sequence a smaller toolbar with actions for the sequence and the following areas are available:



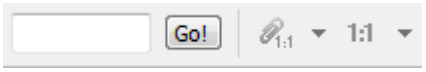
An example of the *Sequence View* with several sequences:



You can change the focus by clicking on the corresponding sequence area. All sequences that are not in focus have the sequence name and icon disabled.

The bottom area of the *Sequence View* is the *Annotations Editor*. It contains a tree-like structure of all annotations available for all sequences shown in the *Sequence View* and can be used to perform various actions on annotations: create a new annotation, modify the existing one, group, sort, etc.

Global Actions

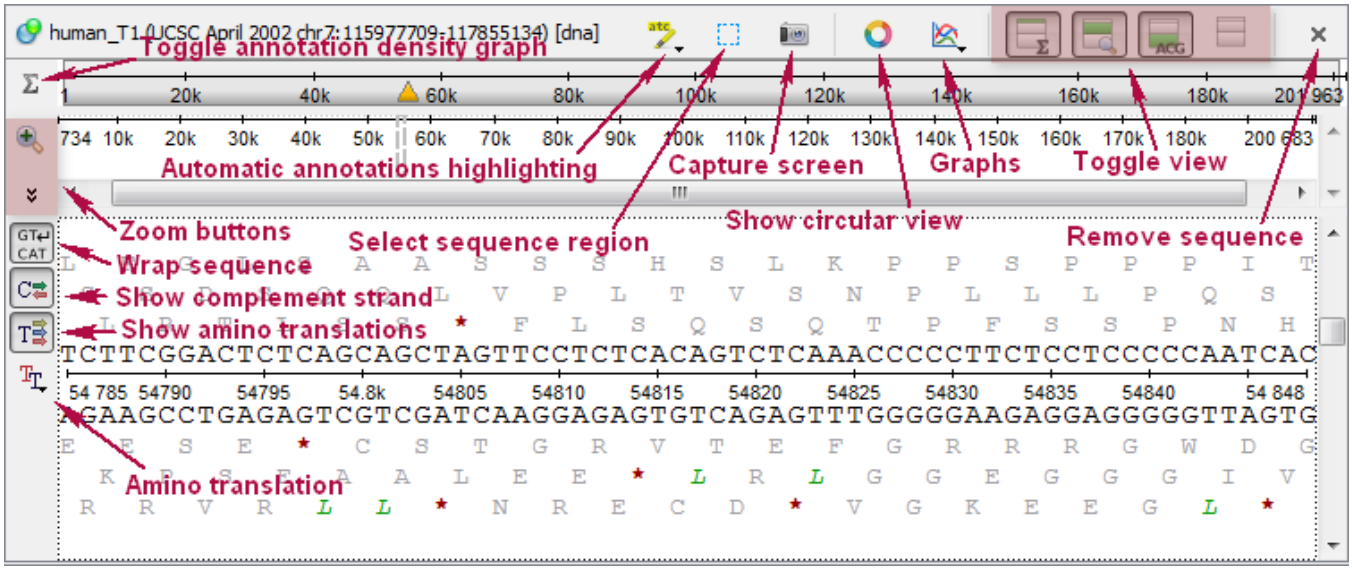


The global action toolbar provides possibility to go to the specified position (in all sequences at the same time).

Also it allows to lock or adjust ranges of sequences in the same *Sequence View*. See *this paragraph* for details.

Sequence Toolbars

A brief description of the sequence toolbars buttons is shown on the picture below:

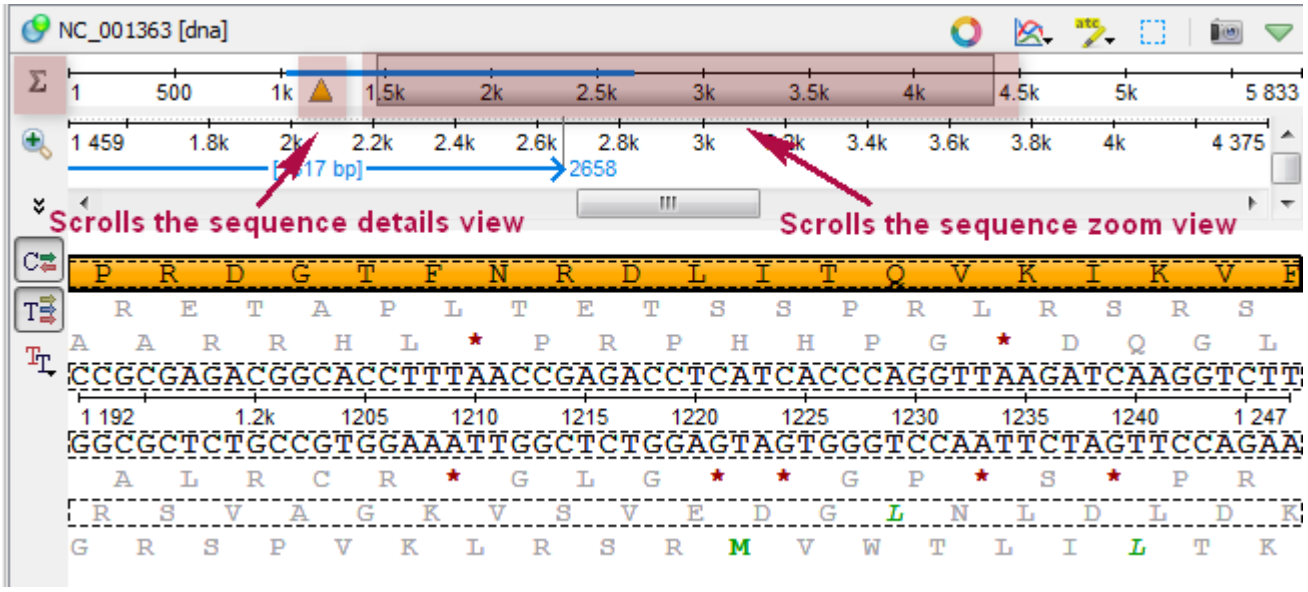


See also:

- *Toggling Views*
- *Capturing Screenshot*
- *Zooming Sequence*
- *Showing and Hiding Translations*
- *Selecting Sequence Region*
- *Showing Sequence in Multiple Lines*

Sequence Overview

The *Sequence overview* is an area of the *Sequence View* below the sequence toolbar. It shows the sequence in whole and provides handy navigation in the *Sequence zoom view* and the *Sequence details view*.



When the sigma button (in the left part of the *Sequence overview*) is pressed, density of annotations in the sequence is shown. For example in the picture below there are annotations in the parts of the sequence that are marked with dark grey color:



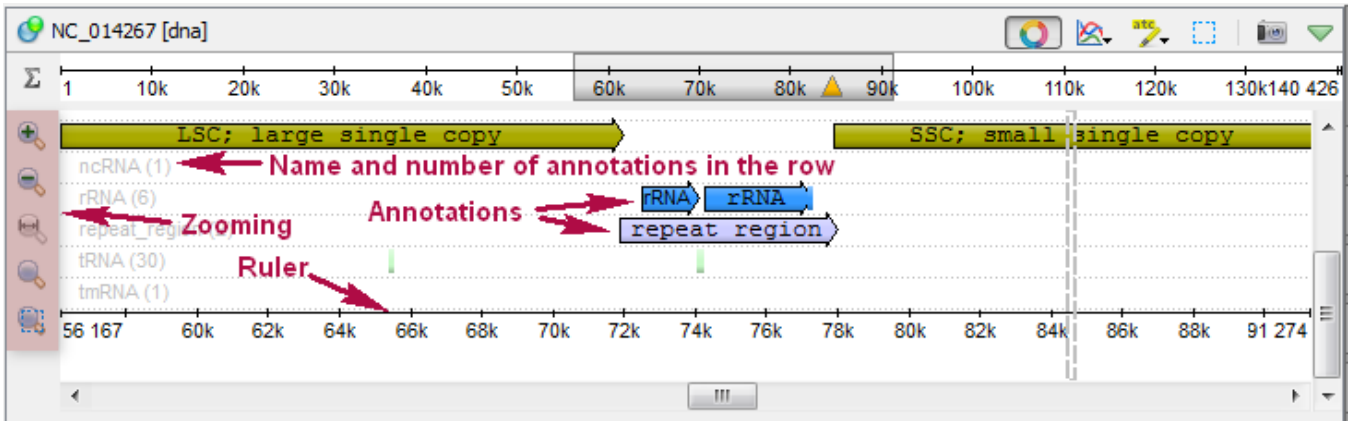
See also:

- *Sequence Zoom View*
- *Sequence Details View*

Zoom View

The *Sequence zoom view* is designed to provide flexible tools for navigation in large annotated sequence regions.

The most *Sequence zoom view* space is used to visualize annotations for the sequence. The annotations are organized in rows by their names. If two annotations with the same name overlap, an extra row is created. For every row the name and the total number of annotations in the row are shown with a light grey text at the left part of the area.

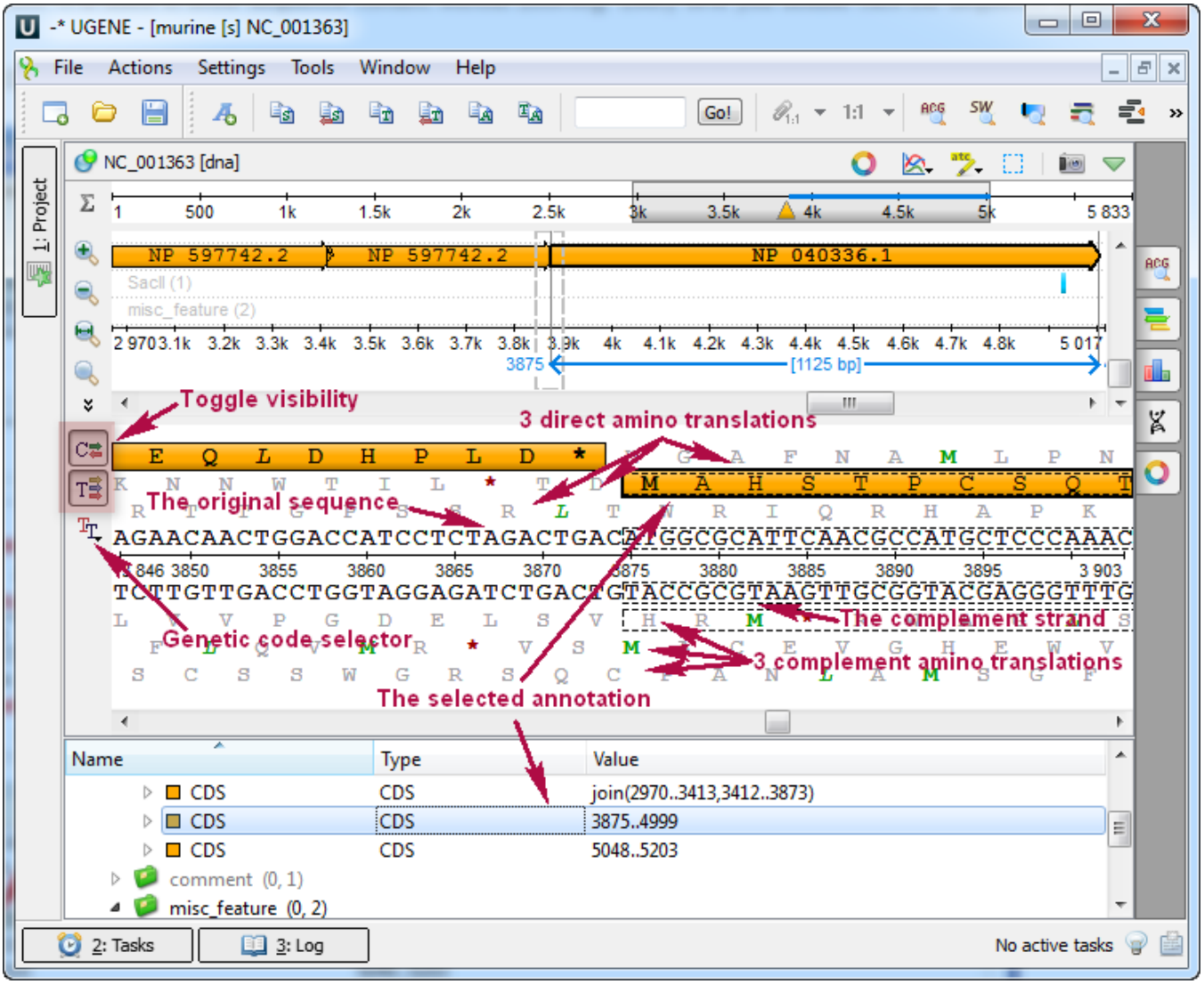


Below the annotation rows there is a ruler to show coordinates in the sequence.

Details View

The *Sequence details view* is a supplementary component of the *Sequence overview*. It is used to show sequence content without zooming. Every time you double click the sequence in the *Sequence overview* area or select an annotation, the corresponding sequence position is made visible in the *Sequence details view*.

For a DNA sequence the *Sequence details view* automatically shows complement DNA strand and 6 amino translation frames.



See also:

- *Navigating the Details View using the Sequence Overview*
- *Translating Nucleotide Sequence*

Information about Sequence

Statistics about an opened sequence can be found on the *Statistics* tab of the *Options Panel*. When a region is selected in the sequence, the statistics is interactively re-calculated for this region only. The following information is available:

- Common statistics (length, molecular weight, etc.) - see the detailed description below
- Characters occurrence
- Dinucleotides occurrence - available for nucleotide sequences only

Note that all data, displayed on the *Statistics* tab, can be selected with the mouse and copied. Use the copy item in the context menu or a shortcut - Ctrl+C on Windows or Linux, Cmd+C on macOS.

Nucleotide sequence common statistics

The following common statistical information is calculated for a nucleotide sequence:

- *Length*
- *GC content*
- *Molecular weight*
- *Extinction coefficient*
- *Melting temperature*
- *nmole/OD260*
- *g/OD260*

GC content

The percentage of guanine (G) and cytosine (C) bases within the sequence or its selected region, for example:

GC-content("ACGTAC") = ((0 + 1 + 1 + 0 + 0 + 1) / 6) * 100% = 50%

If the sequence contains degenerate base characters, average values are used, for example:

GC-content("ACGNBCT") = ((0 + 1 + 1 + 1/2 + 2/3 + 1 + 0) / 7) * 100% ~= 59.52%

In this example "1/2" is used for "N" (any nucleotide), "2/3" us used for "B" (that means "C", "G", or "T" according to the IUPAC nucleotide code).

Molecular weight

Molecular weight for a single-stranded molecule is calculated as a sum of the atomic masses of the molecule compounds:

DNA molecular weight = nA*251.24 + nT*242.23 + nC*227.22 + nG*267.24 + (n-1)*61.97

RNA molecular weight = nA*267.24 + nU*244.20 + nC*243.22 + nG*283.24 + (n-1)*61.97

Here "nA", "nT", "nC", "nG", "nU" denote the number of the corresponding nucleotide in the molecule, "n" is the number of all bases (61.97 is the weight of an internal phosphate).

Note that for degenerate base characters average value of nucleotide weight is used, for example, if the sequence also contain "Y" characters (that is "C" or "T"), the sum will include one more summand - "nY*(242.23 + 227.22)/2".

Molecular weight for a double-stranded molecule is calculated as the sum of the single strands molecular weights.

To calculate the *Extinction coefficient*, an approach proposed by Richard Owczarzy is used: http://www.owczarzy.net/extinctionDNA.htm. That is for a single-stranded molecule:

Extinction coefficient = sum(extinction coefficients of all dinucleotides) - sum(extinction coefficients of inner mononucleotides)

For example, let's calculate the molar extinction coefficient ("") for "ATGCA":

(ATGCA) = (AT) + (TG) + (GC) + (CA) - (T) - (G) - (C) =

= 22800 + 19000 + 17600 + 21200 - 8700 - 11500 - 7400 =

= 53000

As for the other statistics, average values are used in case of degenerate base characters.

*Extinction coefficient*for a double-stranded molecule is calculated as a sum of the extinction coefficients of the two single strands (e_{s1} + e_{s2}) multiplied by coefficient of (1 - hypochromicity h_{260nm}). The hypochromicity effect can be taken into account as follows:

$$h_{260nm} = (0.287f_{AT} + 0.059f_{GC})$$

where f_{AT} and f_{GC} are fractions of AT and GC base pairs, respectively.

Melting temperature

The melting temperature is calculated as follows. For sequences of length 15 or longer:

$$T_m = 64.9 + 41 * (nG + nC - 16.4) / (nA + nT + nG + nC)$$

For shorter sequences:

$$T_m = (nA + nT) * 2 + (nG + nC) * 4$$

Here "nA", "nT", "nC", "nG" denote the number of the corresponding nucleotide.

nmole/OD260

The amount of DNA or RNA represented in nanomoles per 1 unit of absorbance at 260 nm dissolved in 1 ml cuvette with 1 cm pathlength.

nmole/OD260 = 1000000 / molarExtCoef

g/OD_{260}

The amount of DNA or RNA represented in micrograms per 1 unit of absorbance at 260 nm dissolved in 1 ml cuvette with 1 cm pathlength.

g/OD260 = nmoleOD260 * molarWeight * 0.001

Amino acid sequence common statistics

The following common statistical information is calculated for an amino acid sequence:

- *Length*
- *Molecular weight*
- *Isoelectic point*

Manipulating Sequence

- Show/Hide Sequence View Components
- Showing Sequence in Multiple Lines
- Translating Nucleotide Sequence
- Zooming Sequence
- Creating New Ruler
- Going To Position
- Selecting Sequence Region
- Copying and Pasting Sequence
- Editing Sequence
- Searching in Sequence
- Exporting Selected Sequence Region
- Exporting Sequence of Selected Annotations
- Exporting Sequence Image
- Locking and Synchronize Ranges of Several Sequences
- Multiple Sequence Opening

Show/Hide Sequence View Components

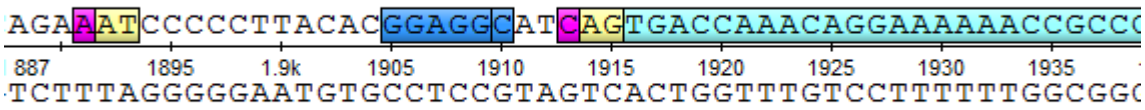
To show or hide a *Sequence View* component (*Sequence Overview*, *Zoom View*, or *Details View*) use the following buttons on a sequence toolbar. The fourth button allows one to show/hide all the components at once.



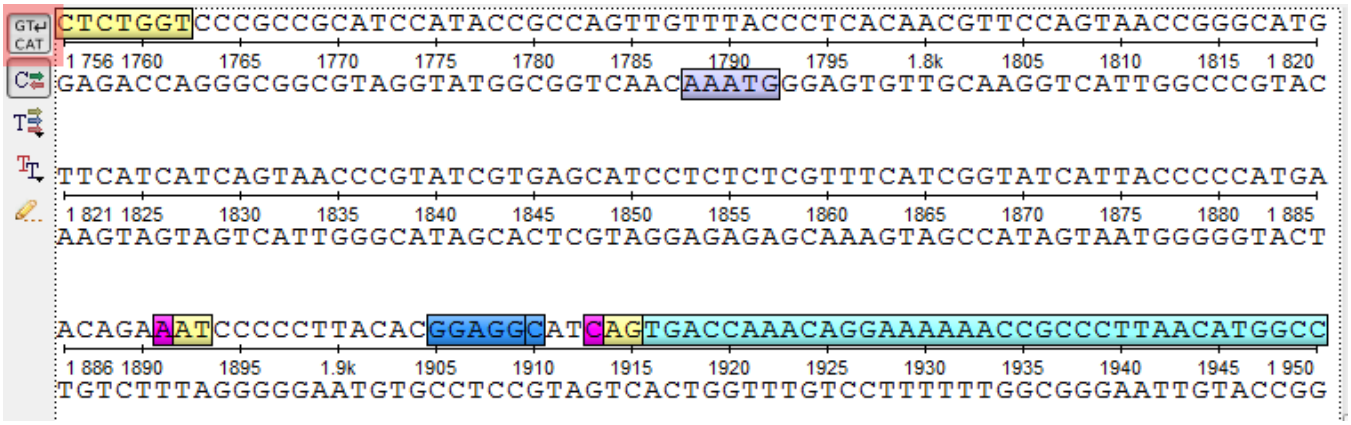
If several sequences are opened in the *Sequence View*, click the cross button  on a sequence toolbar to hide the sequence.

Showing Sequence in Multiple Lines

By default, a sequence in the *Details View* is shown in one line:



To show the sequence in multiple lines, click the *Wrap sequence* button on the left toolbar of the *Details View*:

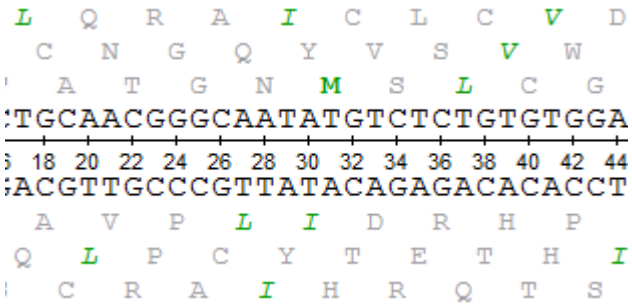


Translating Nucleotide Sequence

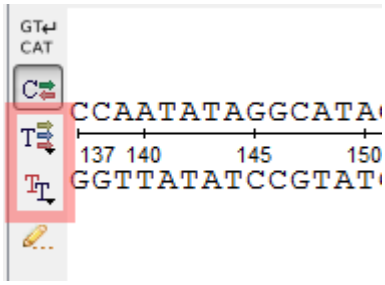
When a nucleotide sequence is opened in the *Sequence View*, the sequence and its complementary sequence are shown by default in the *Details View*.

Showing/hiding the amino acid sequences

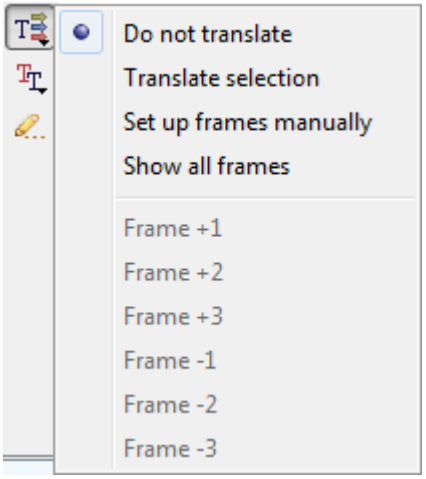
It is possible to translate them and show also the corresponding amino acid sequences:



The translation settings are available on the left toolbar of the *Details View*:

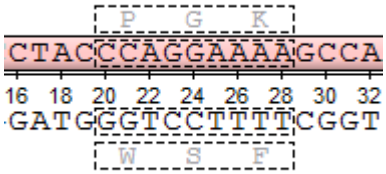


The *Show/hide amino acid translations* menu allows one to set up the mode of the amino acid sequences visualization:

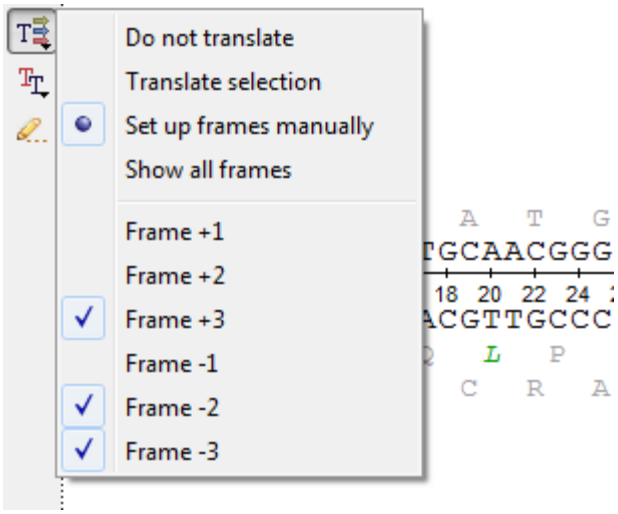


The following options are available:

- *Do not translate*— hide the amino acid sequences.
- *Translate selection* — translate only a selected region of the sequence and the complementary sequence.



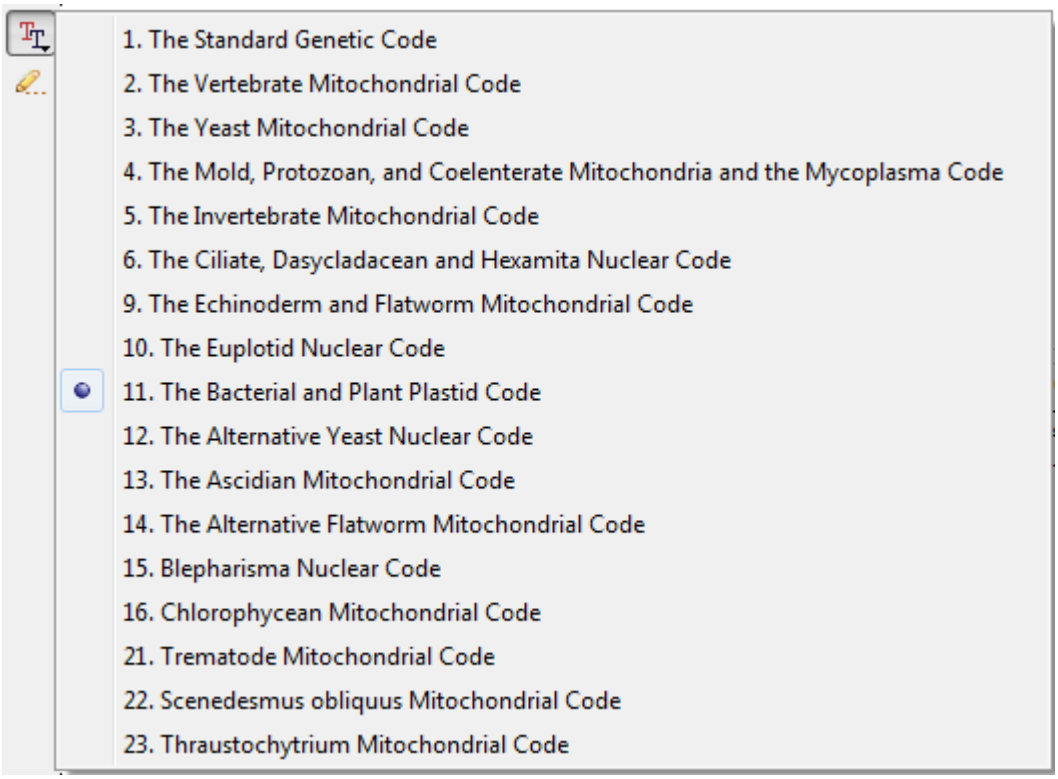
- *Set up frames manually* — select the reading frames to show. There are three frames for the sequence ("+1", "+2", "+3") and three frames for the complementary sequence ("-1", "-2", "-3"). Note that the complementary frame items are hidden in the menu, if the complementary sequence is hidden.



- *Show all frames* — show all amino acid sequences.

Setting the genetic code

The default value for the genetic code for a nucleotide sequence translation is read by UGENE from the sequence file when it is available. One can also set up the genetic code for the sequence using the *Select genetic code* menu:



All analysis routines (like HMMER, ORF finding, etc.) will use this code by default.

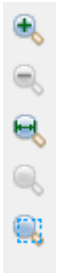
Codon table

To refresh the knowledge about the amino acid codes, select the *Show codon table* button  on the *Sequence View* global toolbar. The codon table appears at the upper part of the window:

1st base	2nd base U		2nd base C		2nd base A		2nd base G		3rd base
U	UUU	Phenylalanine (Phe, F)	UCU	Serine (Ser, S)	UAU	Tyrosine (Tyr, Y)	UGU	Cysteine (Cys, C)	U
	UUC		UCC		UAC		UGC		C
	UUA	Leucine (Leu, L)	UCA		UAA	Stop codon (*)	UGA	Stop codon (*)	A
	UUG		UCG		UAG		UGG	Tryptophan (Trp, W)	G
C	CUU		CCU	Proline (Pro, P)	CAU	Histidine (His, H)	CGU	Arginine (Arg, R)	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	Glutamine (Gln, Q)	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	Isoleucine (Ile, I)	ACU	Threonine (Thr, T)	AAU	Asparagine (Asn, N)	AGU	Serine (Ser, S)	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	Lysine (Lys, K)	AGA	Arginine (Arg, R)	A
	AUG	Methionine (Met, M)	ACG		AAG		AGG		G
G	GUU	Valine (Val, V)	GCU	Alanine (Ala, A)	GAU	Aspartic acid (Asp, D)	GGU	Glycine (Gly, G)	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	Glutamic acid (Glu, E)	GGA		A
	GUG		GCG		GAG		GGG		G

Zooming Sequence

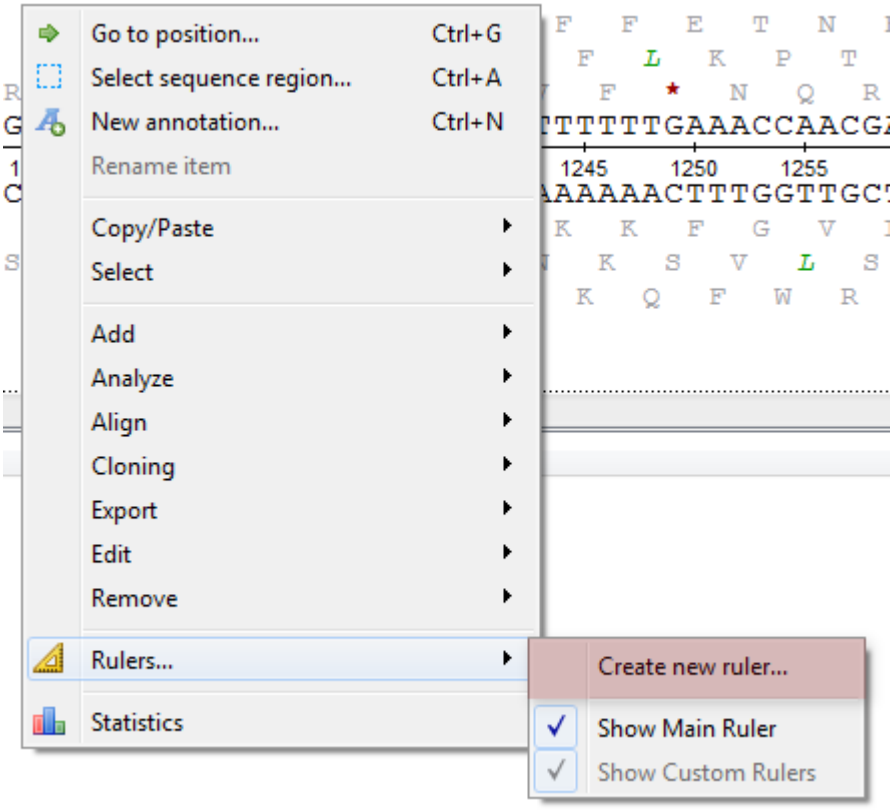
To zoom a sequence in the *Sequence zoom view* you can use one of the zoom button on the sequence toolbar:



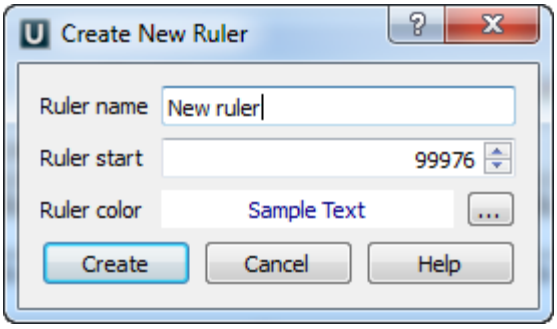
There are standard *Zoom In* and *Zoom Out* buttons. Additionally you can zoom to a selected region using the *Zoom to Selection* button. To restore the default view of the *Sequence zoom view* (when the sequence is not zoomed) use the *Zoom to Whole Sequence* button.

Creating New Ruler

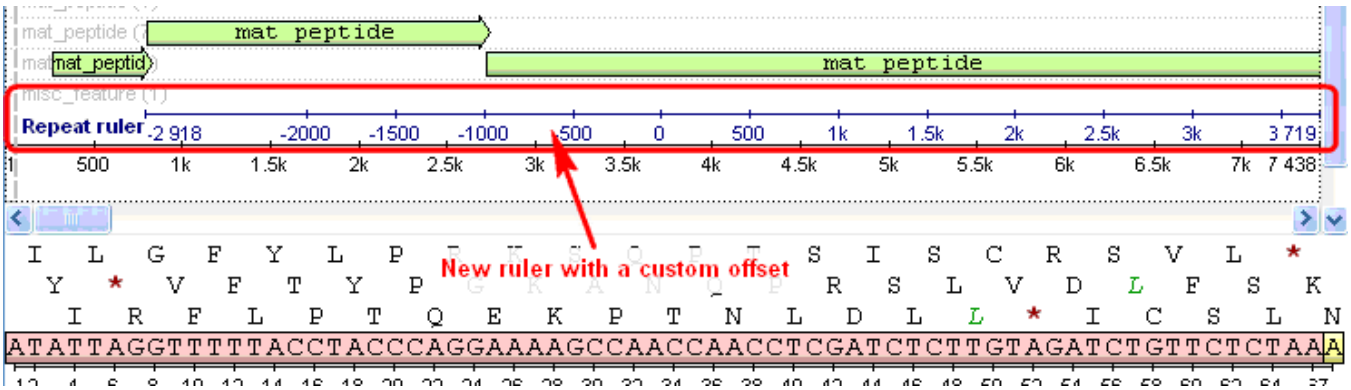
You can create any number of additional rulers by clicking the *Ruler Create new ruler* context menu item:



The following dialog will appear:



The new ruler will be shown right above the default one:

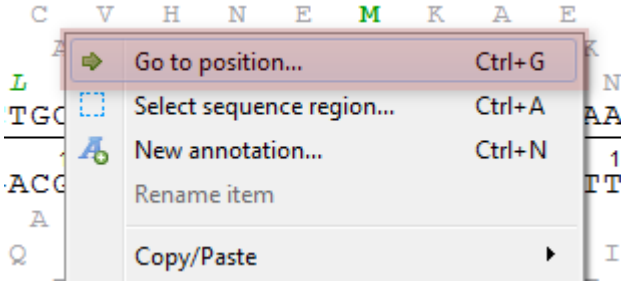


Going To Position

To go to a position, use the global actions toolbar:



Or use the *Go to position* context menu or the *Actions* main menu item.



Also you can use the shortcut Ctrl-G.

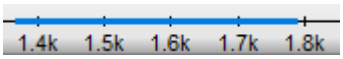
Selecting Sequence Region

Selection in the Sequence View components

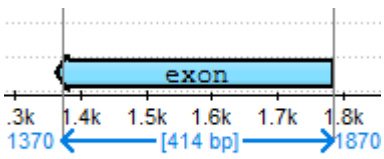
Holding the left mouse button down and moving the mouse, one can select a sequence region in the *Zoom View* or in the *Details View* components of the *Sequence View*. A sequence region that correspond to an annotation can also be selected by double-clicking on the annotation in the *Zoom View*, *Details View*, or *Annotations Editor*.

Selection of a sequence region is synchronized between different *Sequence View* components.

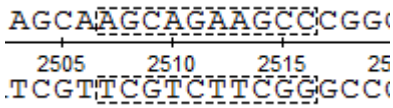
In the *Sequence Overview* the selection is displayed as a blue line:



In the *Zoom View* a blue line with coordinates and the length of the selected region is shown:



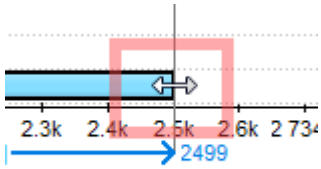
In the *Details View* a rectangle with a dashed border is shown around the selected region:



Selecting a region using the mouse

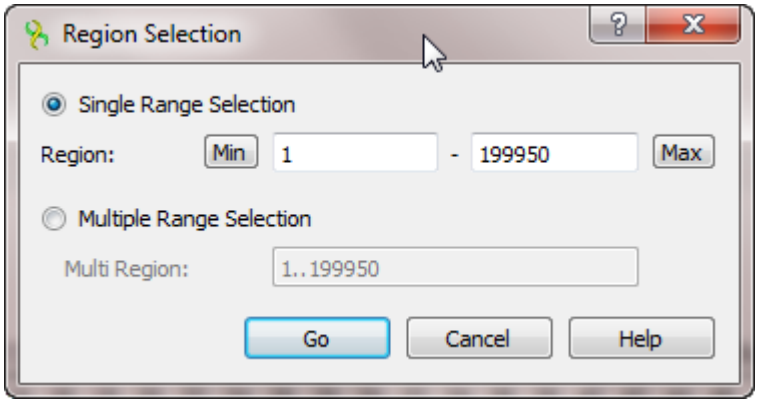
It is possible to select a region of a sequence using the mouse in each component of the *Sequence View*.

In the *Zoom View* and the *Details View* the selection can be adjusted by dragging its left or right border:



Selecting a region using the dialog

If information about exact coordinates of a region is available, click the *Select sequence region* button  on the toolbar or *Select–>Sequence region* in the context menu of the *Sequence View*. Input the coordinates in the *Region Selection* dialog:



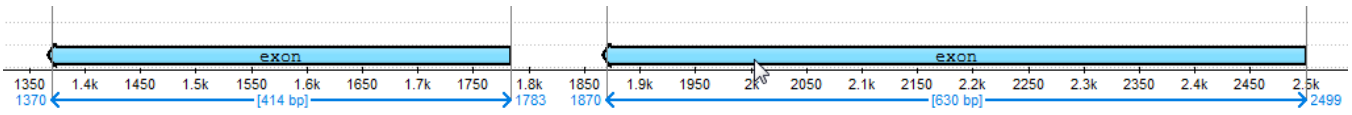
It is possible to input:

- *Single Range Selection* — a common region with a single range. Use the *Min* and *Max* buttons to automatically specify the beginning and the end of the sequence in the corresponding fields.
- *Multiple Range Selection* — a region that consists of several join elements, for example, exons of a gene. See "The DDBJ/ENA/GenBank Feature Table Definition" for details.

Selecting a region around or between annotations

To select a region between two annotations, for example, an intron region between two exons:

- Click on the annotations, holding the *Ctrl*/*Cmd* key:



- Click Select–>*Sequence between selected annotations* in the context menu of the *Sequence View*.

To select a region that contains the annotations in a single range, choose the *Sequence around selected annotations* item in the menu.

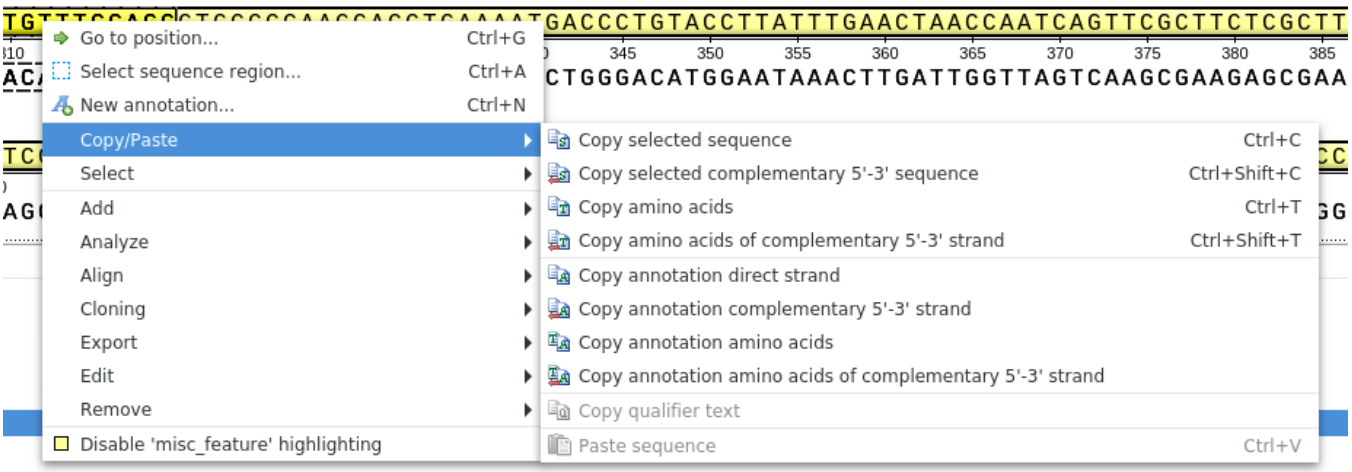
Copying and Pasting Sequence

The selected sequence region, an annotation sequence or their amino translations can be copied to clipboard:

- By pressing the corresponding buttons in the global toolbar.



- Using the following shortcuts:
  - Ctrl-C — copies direct sequence strand
  - Ctrl-T — copies direct amino translation
  - Ctrl-Shift-C — copies reverse-complement sequence
  - Ctrl-Shift-T — copies reverse-complement amino translation
- Using the *Copy* submenu of the context menu:



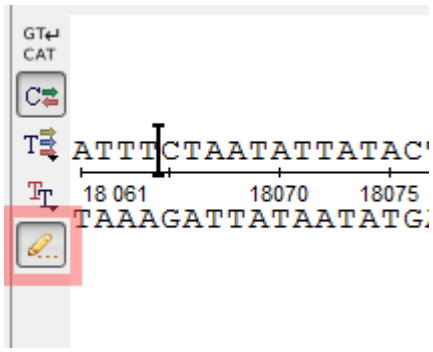
To paste sequence use the corresponding context menu item or *Ctrl+V* shortcut.

Editing Sequence

If the corresponding document is not locked, it is possible to edit a sequence.

Editing mode and annotations settings

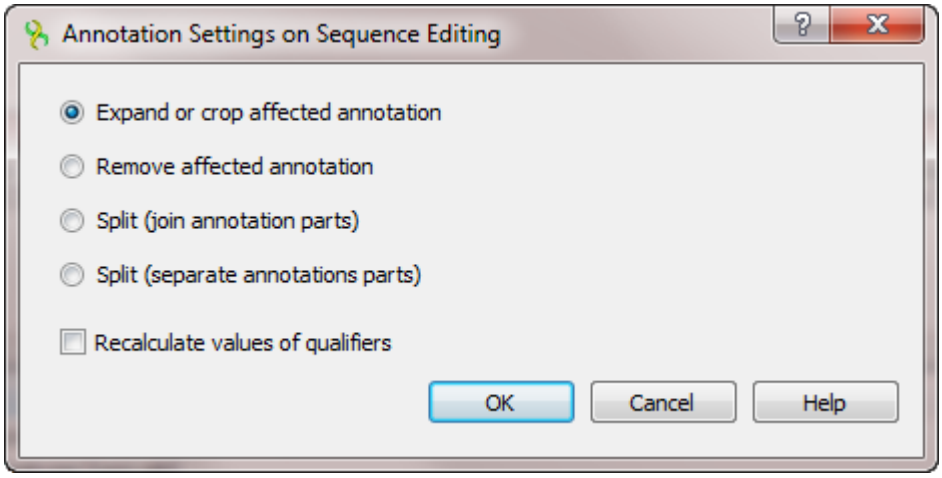
To switch on the editing mode for the sequence, select the *Edit sequence* button on the left toolbar of the *Details View*:



A special cursor appears in this case, and the sequence can be edited as in a text editor:

- To insert characters to the sequence, move the cursor to the required location and type the characters or paste them using *Ctrl + V* / *Cmd + V* key board shortcut.
- To replace a sequence region, select it and insert the new characters.
- To remove a character from the sequence, move the cursor to the required location and press *Backspace* or *Delete*. To remove a region, select it and press one of these shortcuts.

One can also configure the way how annotations located in an edited region should be modified. To open a dialog with the settings select *Edit–>Annotations settings on sequence editing* in the *Actions* main menu or in the context menu of the *Sequence View*.



Select one of the following:

- *Expand or crop affected annotation* — an annotation located in an edited region is expanded in case of characters insertion or cropped in case of characters deletion.
- *Remove affected annotation* — all annotations in an edited region are removed.
- *Split (join annotation parts)* — an annotation is split into two join elements (see "The DDBJ/ENA/GenBank Feature Table Definition" for details).
- *Split (separate annotation parts)* — an annotation is split into two annotations.

It is also possible to check the *Recalculate values of qualifiers* option in the dialog. If it is selected, qualifiers of all annotations are parsed on the sequence editing. Qualifiers values that specify coordinates (like "100..200") are re-calculated accordingly to the sequence modification. For example, the value might become "104..204", if four characters have been inserted before the corresponding annotation.

Getting reverse-complement, reverse, or complement sequence

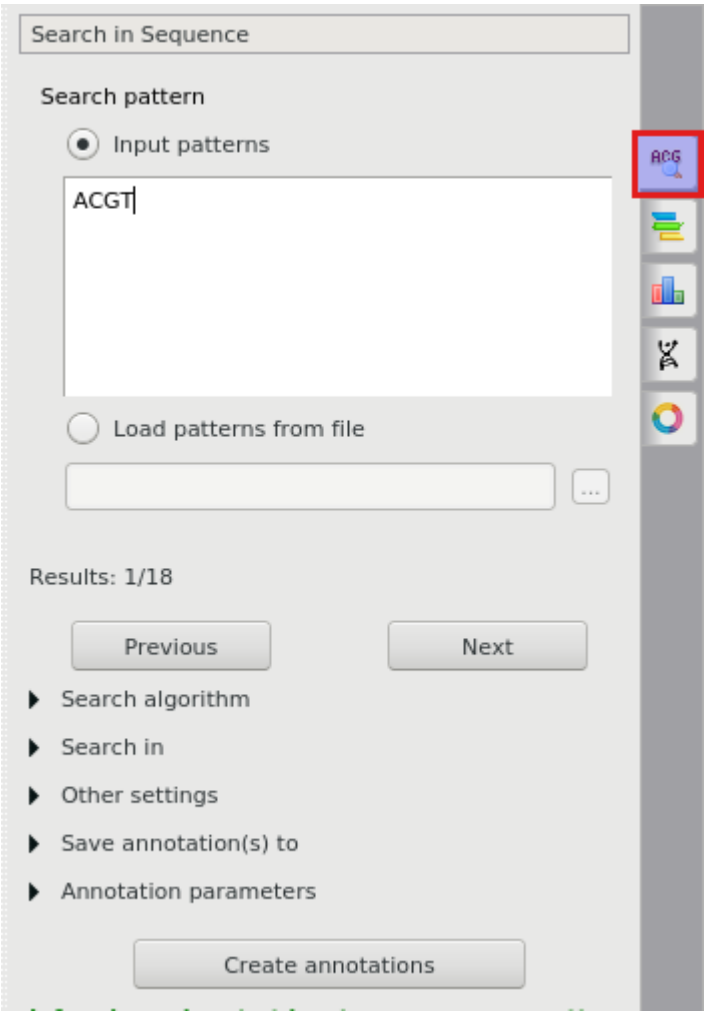
To replace a sequence with its reverse-complement sequence, select *Edit–>Reverse-complement sequence* in the *Actions* main menu or in the context menu of the Sequence View. The keyboard shortcut for this action is *Ctrl + Shift + R* / *Cmd + Shift + R*.

Use the *Reverse sequence* or *Complement sequence* items in the same menus to replace the sequence with its reverse or complement sequence.

Searching in Sequence

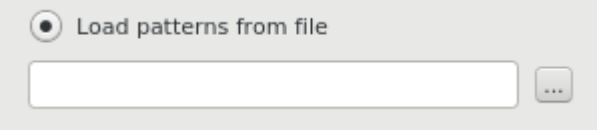
To search for a pattern(s) in a sequence go to the *Search in Sequence* tab of the *Options Panel* in the *Sequence View*.

Input the value you want to search in the text field. To search multiple patterns input the patterns separated by a new line in the pattern text field. To add a new line symbol *Ctrl+Enter* may be used. You can input the value as a sequence or name of the sequence in the FASTA format and sequence after that. Press the *Next* button to search in the direction “From left to right”. Press the *Previous* button to search in the direction “From right to left”. If the pattern is found, the result will be focused and highlighted in the *Sequence area*. You can continue the search in any direction from this position.



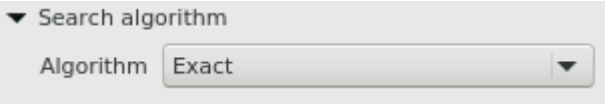
By default, **misc_feature** annotations are created for regions that exactly match the pattern. Find below the description of the available settings.

Load Patterns from File



Use this option to load patterns from a file. When this option is active the *Search for* field is disabled.

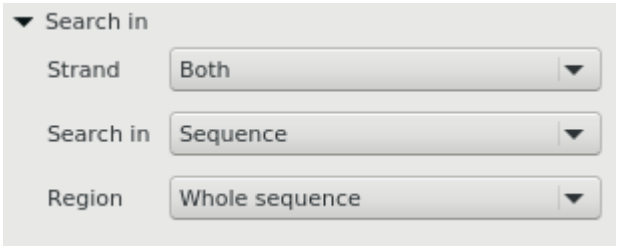
Search Algorithm



This group specifies the algorithm that should be used to search for a pattern. The algorithm can be one of the following:

- *InsDel* — there could be insertions and/or deletions, i.e. a pattern and the searched region can vary in their length. You can specify the percentage of the pattern and a searched region match in the field nearby. Note that this value also depends on the pattern length and is disabled when the pattern hasn't been specified.
- *Substitute* — a pattern may contain characters different from the characters in the searched region. When this algorithm has been selected you can also specify the match percentage and additionally it is possible to take into account ambiguous bases.
- *Regular expression* — a regular expression may be specified instead of a pattern. For example character '.' matches any character, '.*' matches zero or more of any characters. There is also the *Limit result length* option that specifies the maximum length of a result.
- *Exact* - find a place where one or several patterns are found within a larger pattern.

Search in



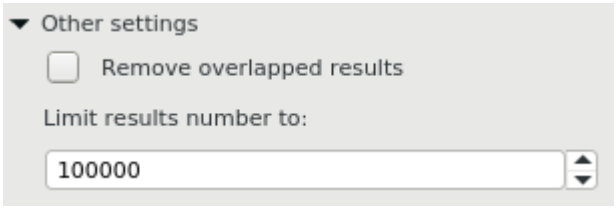
In this group you can specify where to search for a pattern: in what region and in which strand (for nucleotide sequences). Also for nucleotide sequences, it is possible to search for a pattern on the sequence translations.

Strand — for nucleotide sequences only. Specifies on which strand to search for a pattern: *Direct*, *Reverse-complementary,* or *Both* strands.

Search in — for nucleotide sequences you can select the *Translation* value for this option. In this case, the input pattern will be searched in the amino acid translations.

Region — specifies the sequence range where to search for a pattern. You can search in the whole sequence, specify a custom region or search in the selected region.

Other Settings

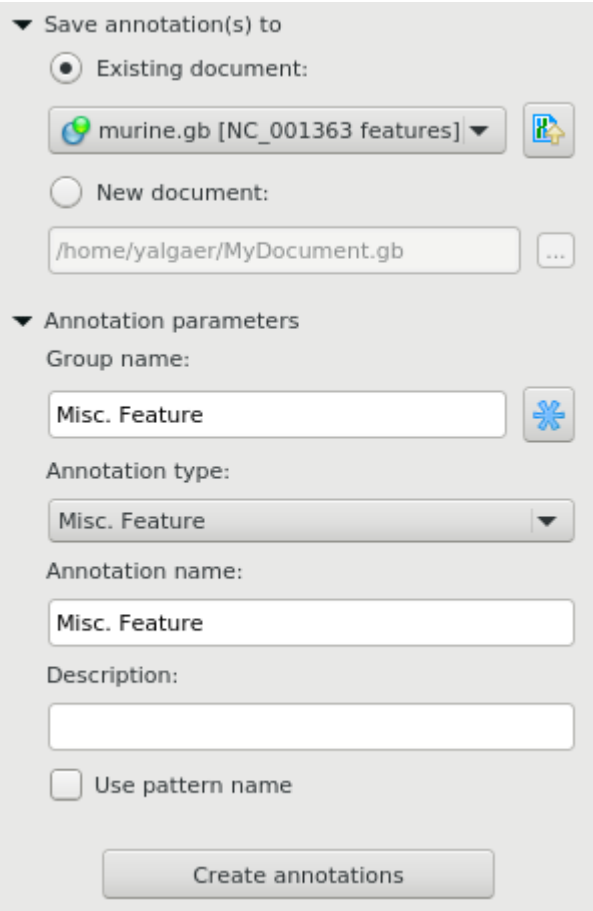


This group contains additional common settings:

Remove overlapped results — annotates only one of the overlapped results.

Limit results number to — limits the number of the searched results to the specified value.

Annotations Settings



In the *Save annotation(s) to* group, you can set up a file to store annotations. It could be either an existing annotation table object or a new annotation table.

In the *Annotation parameters* group, you can specify the name of the group and the name of the annotation. If the group name is set to <auto> UGENE will use the group name as the name for the group. You can use the '/' characters in this field as a group name separator to create subgroups. If the annotation name is set to *by type* UGENE will use the annotation type from the *Annotation type:* table as the name for the annotation. Also you can add a description in the corresponding text field. To use a pattern name for the annotations check the corresponding checkbox.

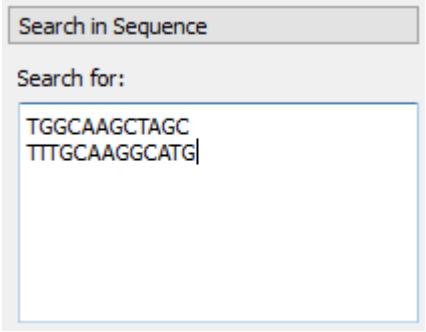
After that click the *Create annotations* button. The annotations will be created.

Searching for one or several patterns and names of the result annotations

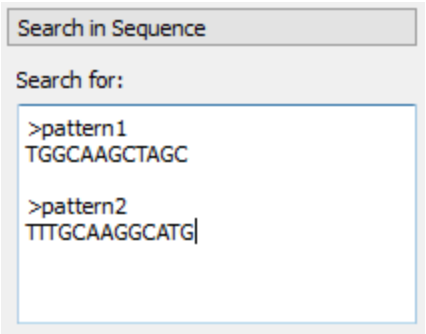
If you search for one pattern only, then input the required name into the *Annotation name* field and leave the *Use pattern name* checkbox unchecked.

You can also search for several patterns at a time by:

- Inputting several patterns into the search field (click <*Ctrl*> + <*Enter*> keys to insert to a new line):



- Inputting several patterns into the search filed in FASTA format:



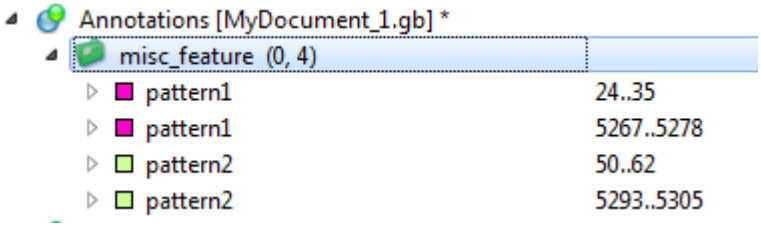
- Loading patterns from a FASTA file

Even when you search for several patterns, the names of the found annotations will be identical by default (the name is specified in the *Annotation name* field).

If you want to assign different names to annotations found for different patterns, then you should:

- Input the patterns in FASTA format (the latter two cases above)
- Check the *Use pattern name* checkbox in the *Annotation parameters* group

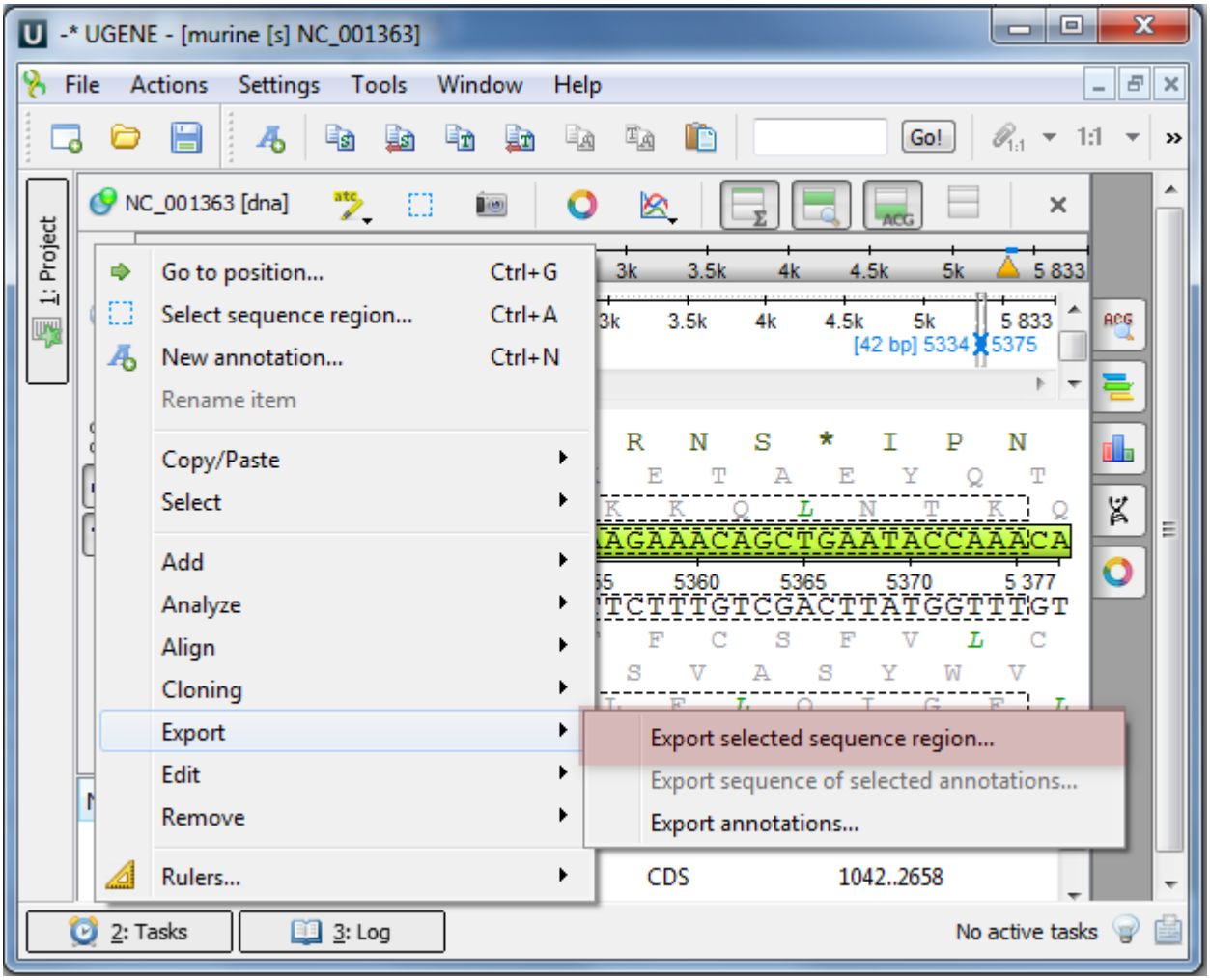
Here is an example of the found annotations in the Annotations Editor:



Exporting Selected Sequence Region

Open a sequence object in the *Sequence View* and select a region by pressing and moving the left mouse button over the sequence.

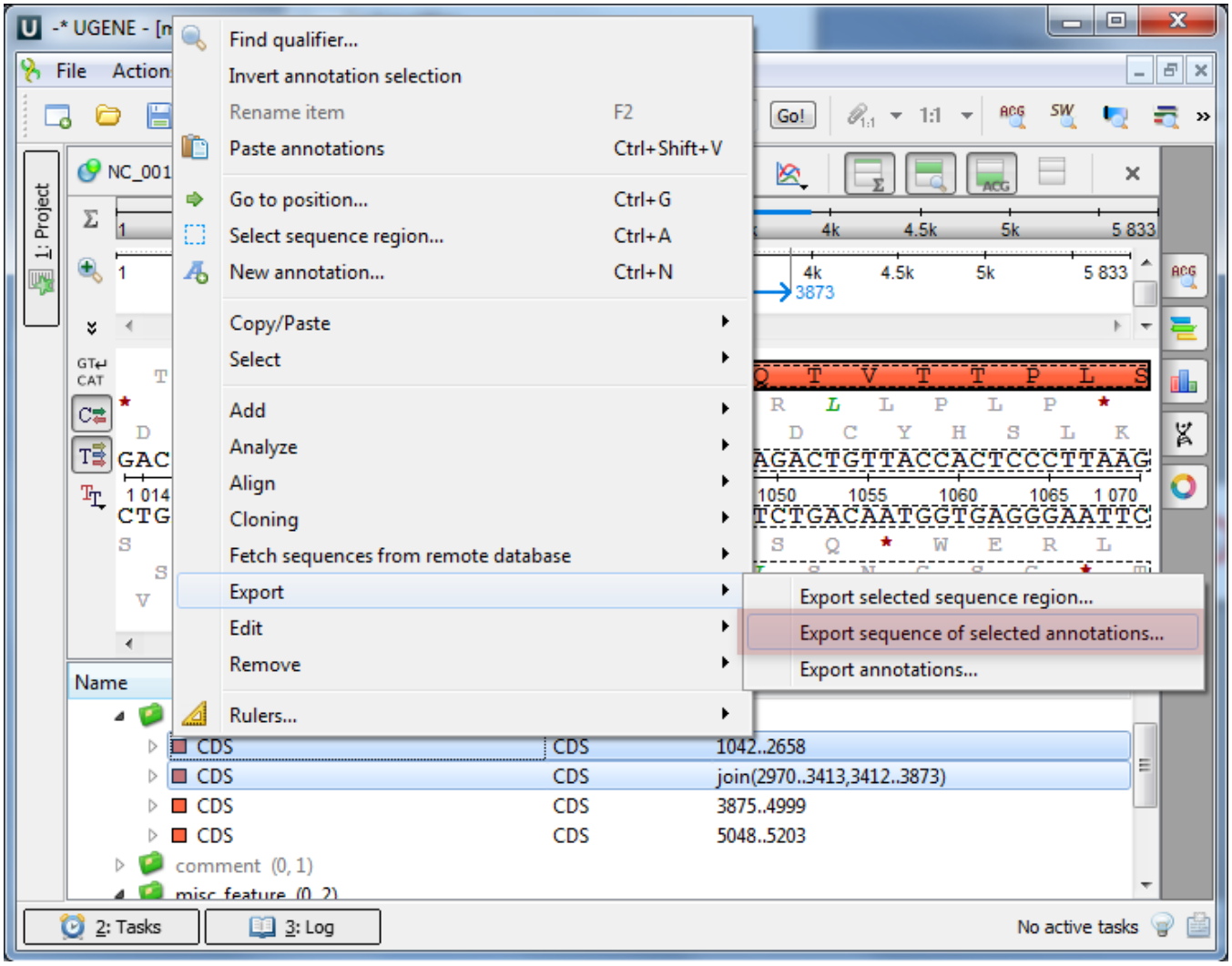
Use the *Export Export selected sequence region* context menu item to save selection into a file of a sequence format.



The *Export Selected Sequence Region* dialog will appear which is similar to the *Export Selected Sequences* dialog described *here*.

Exporting Sequence of Selected Annotations

Open the *Sequence View* with document that contains annotations. A good candidate here could be any file in Genbank format with both sequence and annotations. Select a single or several annotations or annotation groups in the *Annotation editor*, click the right mouse button to open the context menu and select the *Export Export sequence of selected annotations* item:



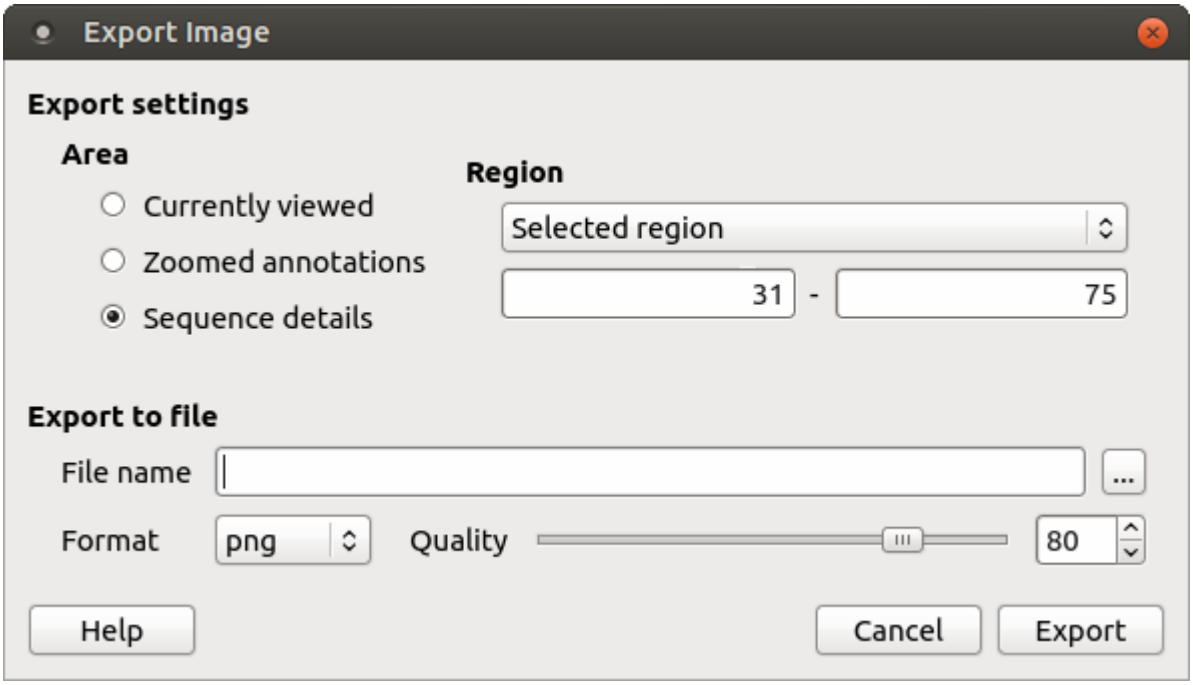
The *Export Sequence of Selected Annotations* dialog will appear which is similar to the *Export Selected Sequences* dialog described *here*.

Exporting Sequence Image

Use a sequence toolbar *Export image* button to save a screenshot of the sequence:



The Export Image dialog will appear where you should set name, location, export settings and format of the picture:



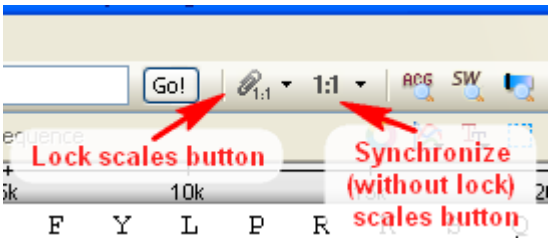
UGENE supports export to the BMP, JPEG, JPG, PNG, PPM, TIF, TIFF, XBM, XPM and SVG image formats. You can export currently viewed, zoomed annotations or sequence detailes areas. Also you can export whole sequence or custom region. Use the *Region* settings to do it.

Locking and Synchronize Ranges of Several Sequences

An important feature of the *Sequence zoom view* is the ability to synchronize and lock visual ranges of different sequences shown in the *Sequence View*.

This feature is available when there are two or more sequences opened in the same *Sequence View*.

If we click the *Lock scales* button the second sequence scale will be adjusted to be the same as the focused sequence scale and is locked. Now if we move a scrollbar or use zoom buttons for any of the sequence, visual ranges for the rest sequences will also be adjusted.



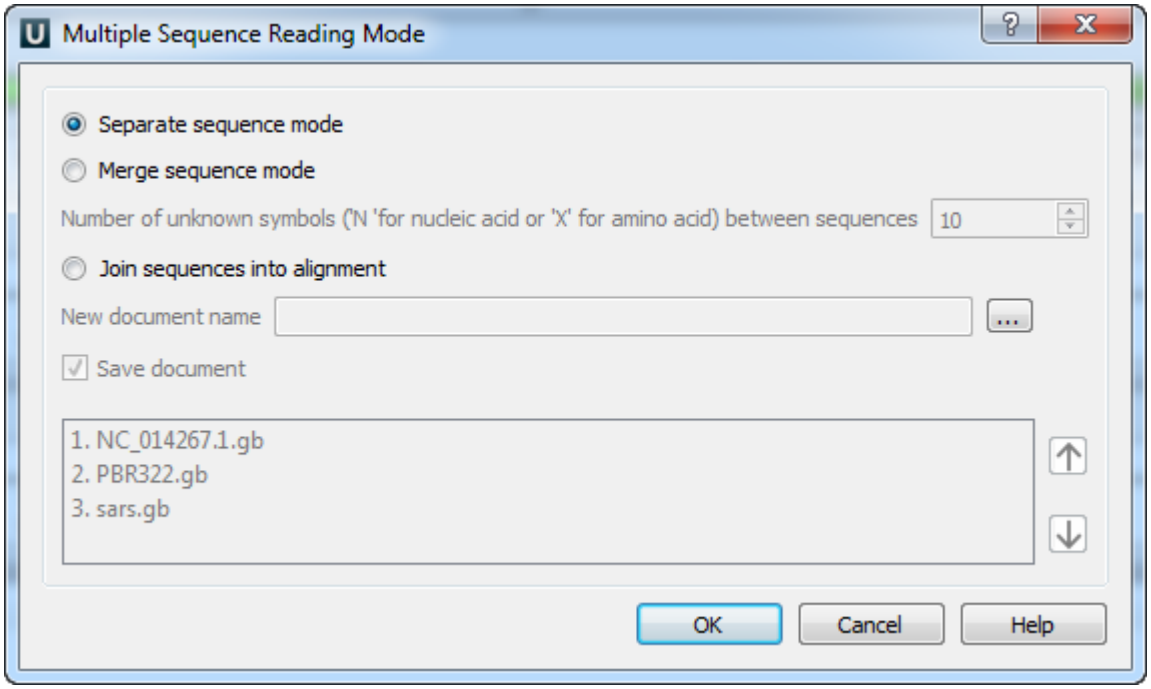
To unlock the scales click the same button again.

You may use the *Adjust scales* button to synchronize scales without locking them.

Note, that if you have a selected sequence region or a selected annotation the scales will be synchronized by the start position of the region or the annotation. If there are no active selection the regions are synchronized by the first visible sequence position on the screen.

Multiple Sequence Opening

To open several sequences use the *File->Open* menu item or *Open* toolbar button and using *Ctrl* select the several sequences and click the *Open* button. The following dialog will appear:



The following parameters are available:

Separate sequence mode - opens the sequences as separate sequences.

Merge sequence mode - merges sequences into one sequence with selected number of unknown symbols between sequences.

Join sequences into alignment - joins sequences into alignment.

Save document - save document to the selected document.

Also you can change the order of the sequences by up and down arrows.

Choose the parameters and click the *Open* button. Sequences will be opened in the selected mode.

In the *Separate sequence mode* sequences will be opened as separate sequences in selected order. You can change the sequences order by drag and drop in the sequence view.

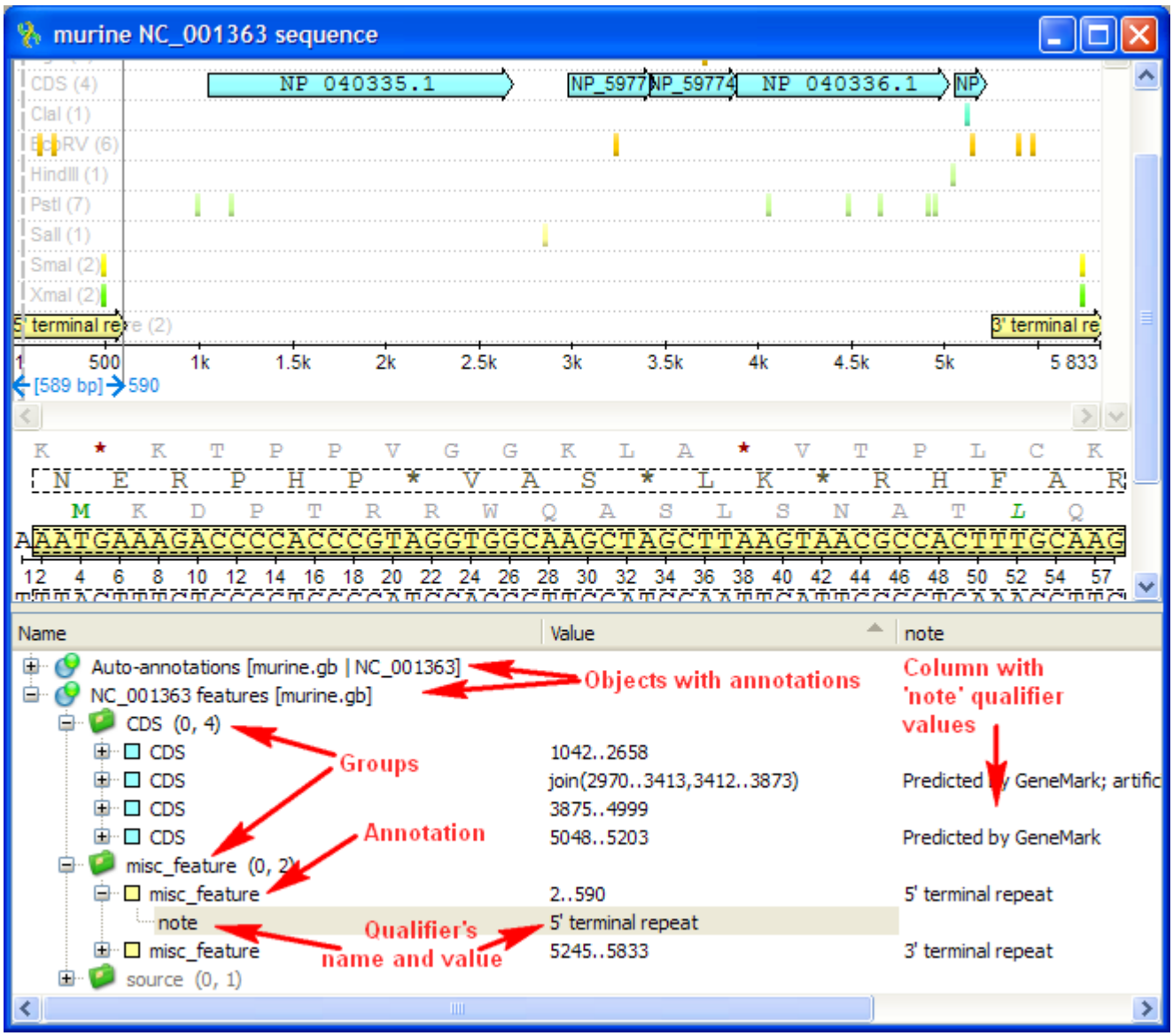
Annotations Editor

The *Annotations editor* contains tools to manipulate annotations for a sequence. It provides a convenient way to organize, view and modify a single annotation as well as annotation groups.

An annotation for a sequence consists of:

- *Name* (or key) — indicates the biological nature of the annotated feature.
- *Location* — coordinates in the sequence.
- *The list of qualifiers* — qualifiers are the general mechanism for supplying information about annotation. Qualifiers are stored as pairs of (name, value) strings.

Below is the default layout of the *Annotations editor* with an extra column for the “note” qualifier added:



There are usually several objects with annotations in the *Annotations editor*. A special *Auto-annotations* object is always presented for each sequnce opened. It contains annotations automatically calculated for the sequence (see *below* for details).

An object contains groups of annotations used by UGENE for logical organization of the annotations. An annotation must always belongs to some group.

For documents created not by UGENE annotations are grouped by their names. For annotations created in UGENE it is possible to use arbitrary group names.

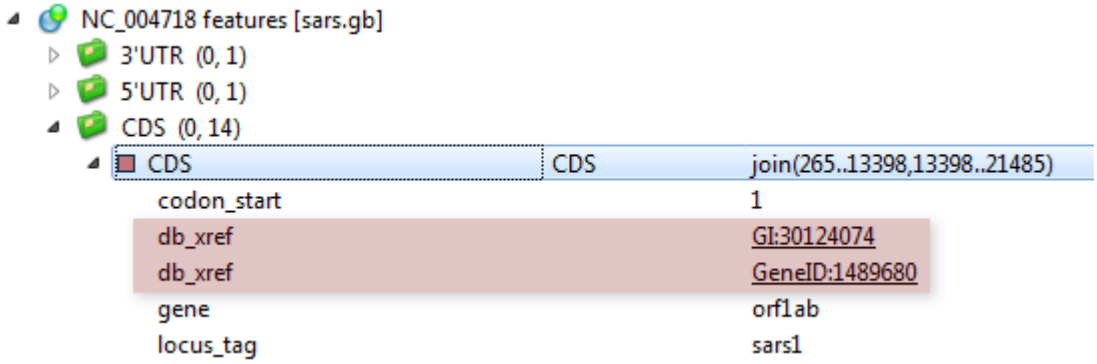
Groups can contain both annotations and other groups. The numbers in the brackets after a group name in the *Annotations editor* are the count of subgroups and annotations in the current group.

A single annotation is allowed to be presented in several groups simultaneously. An annotation is physically removed from the document when it does not belong to any group.

- "db_xref" Qualifier
- Automatic Annotations Highlighting
- The "comment" Annotation

"db_xref" Qualifier

Some files in Genbank format contain the *db_xref* qualifier. A value of this qualifier is a reference to a database.



When you click on the value a web page is opened or a file is loaded specified in the reference. The loaded file is added to the current *project*.

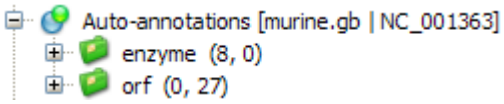
Automatic Annotations Highlighting

Enabling the automatic annotations highlighting allows you to automatically calculate and highlight annotations on each nucleotide sequence opened.

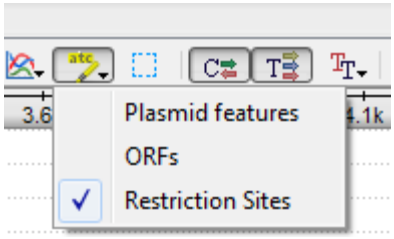
Currently, the following annotations types support the automatic highlighting:

- Open reading frames
- Restriction sites
- Plasmid features

The corresponding groups of annotations found are stored in the *Auto-annotations* object in the *Annotations editor*, for example:



To disable/enable the automatic annotations calculations use the *Automatic Annotations Highlighting* menu button on the *Sequence View* toolbar:

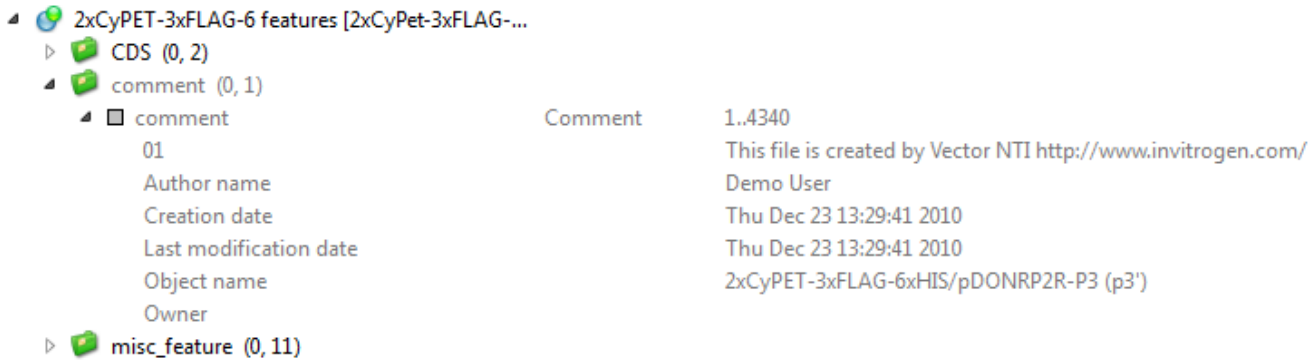


To create a permanent annotation click on the *Make auto-annotations persistent* context menu item and choose the annotation parameters in the *Create Permanent Annotation* dialog.

The "comment" Annotation

General information about a file in GenBank or Vector NTI Sequence format, stored in the COMMENT sections of the file, is shown in UGENE in a special *comment* annotation in the *Annotations Editor*.

The information, for example, may include the name of the file author, creation date and last modification date for the file, and so on:

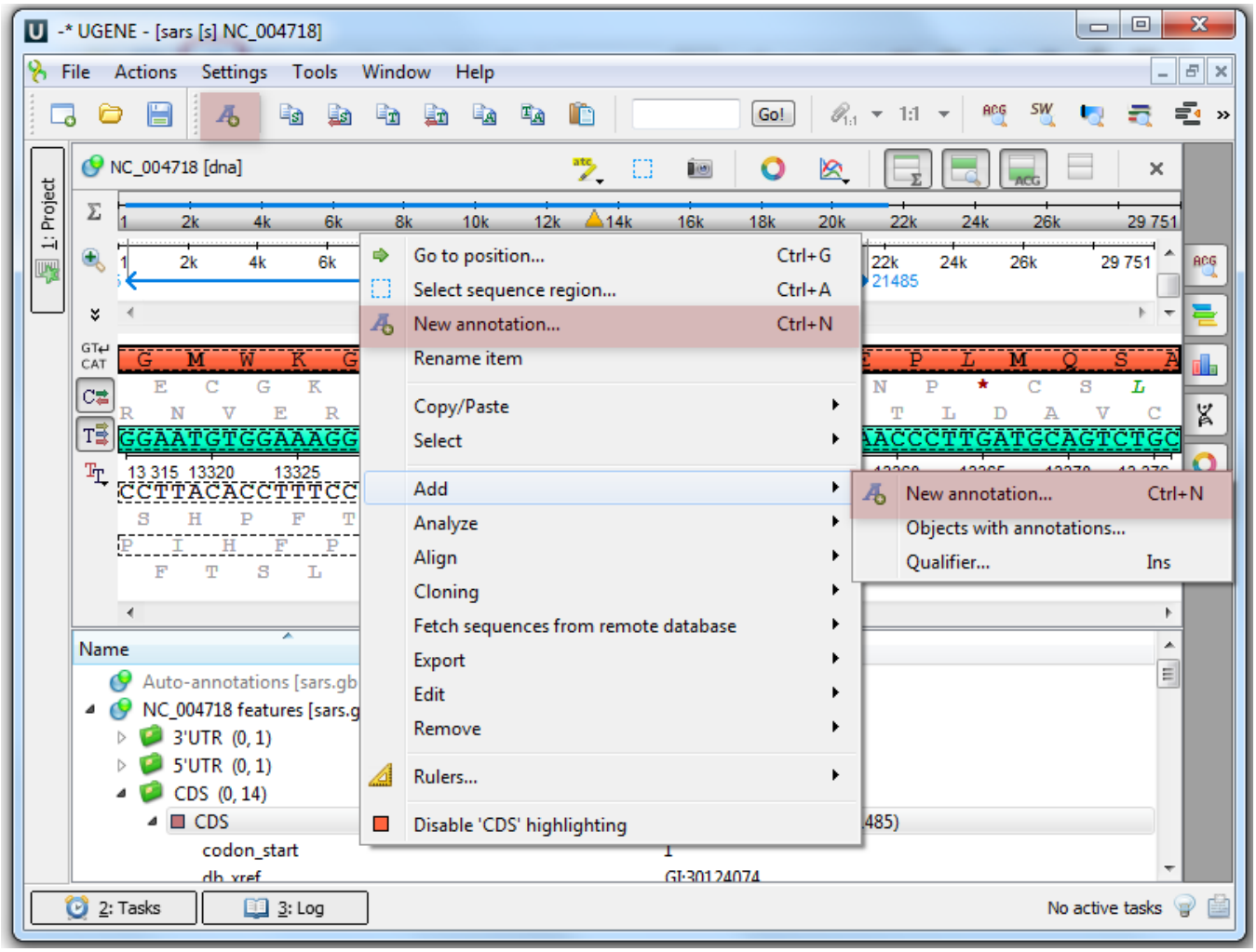


Manipulating Annotations

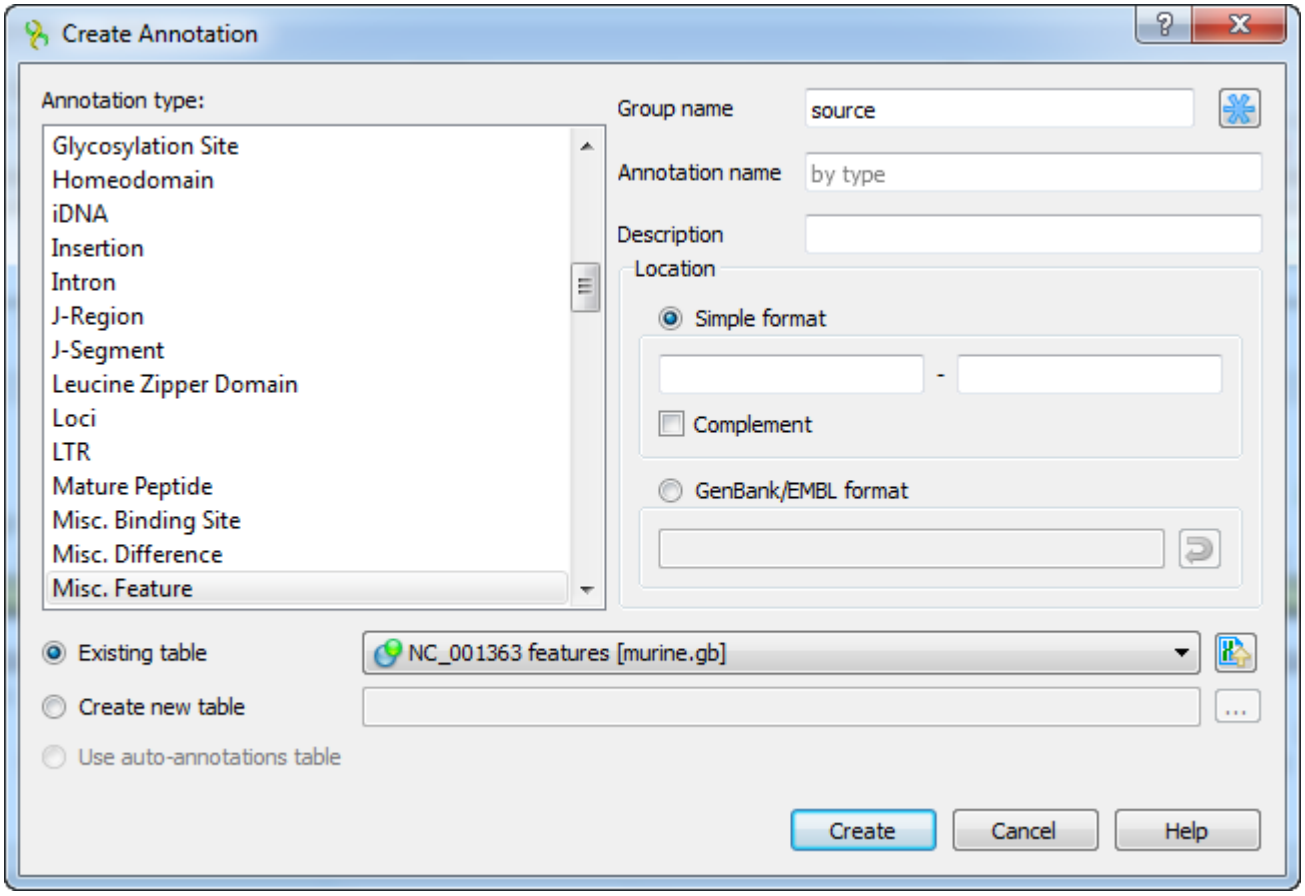
- Creating Annotation
- Selecting Annotations
- Editing Annotation
- Highlighting Annotations
  - Annotations Color
  - Annotations Visability
  - Show on Translation
  - Captions on Annotations
- Creating and Editing Qualifier
- Adding Column for Qualifier
- Copying Qualifier Text
- Finding Qualifier
- Deleting Annotations and Qualifiers
- Importing Annotations from CSV
- Exporting Annotations

Creating Annotation

To create a new annotation for the active sequence press the Ctrl-N key sequence, select the *New annotation* toolbar button or use the *Add New annotation* or *New annotation* context menu item:



This will activate a dialog where to set up annotation parameters:



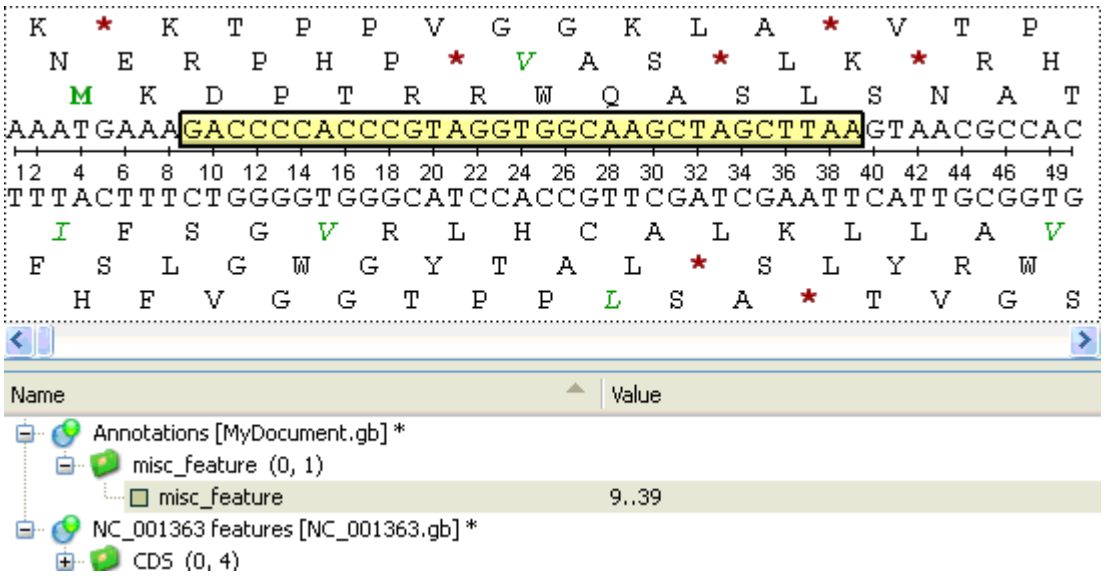
The dialog asks where to save the annotation. It could be either an existing annotation table object, a new annotation table or auto-annotations table (if it is available).

You can also specify the name of the group and the name of the annotation. If the group name is set to <auto> UGENE will use the group name as the name for the group. You can use the '/' characters in this field as a group name separator to create subgroups. If the annotation name is set to *by type* UGE NE will use the annotation type from the *Annotation type:* table as the name for the annotation. Also you can add a description in the corresponding text field.

The *Location* field contains annotation coordinates. The coordinates must be provided in the Genbank or EMBL file formats. If you want to annotate complement strand sequence check the corresponding checkbox for the simple format or surround the coordinates with the "complement()" word or press the last button in the corresponding row to do it automatically.

Note, that by default the *Location* field contains the coordinates of the selected sequence region.

Once the *Create* button is pressed the annotation is created and highlighted both in the *Sequence overview* and the *Sequence details view* areas:



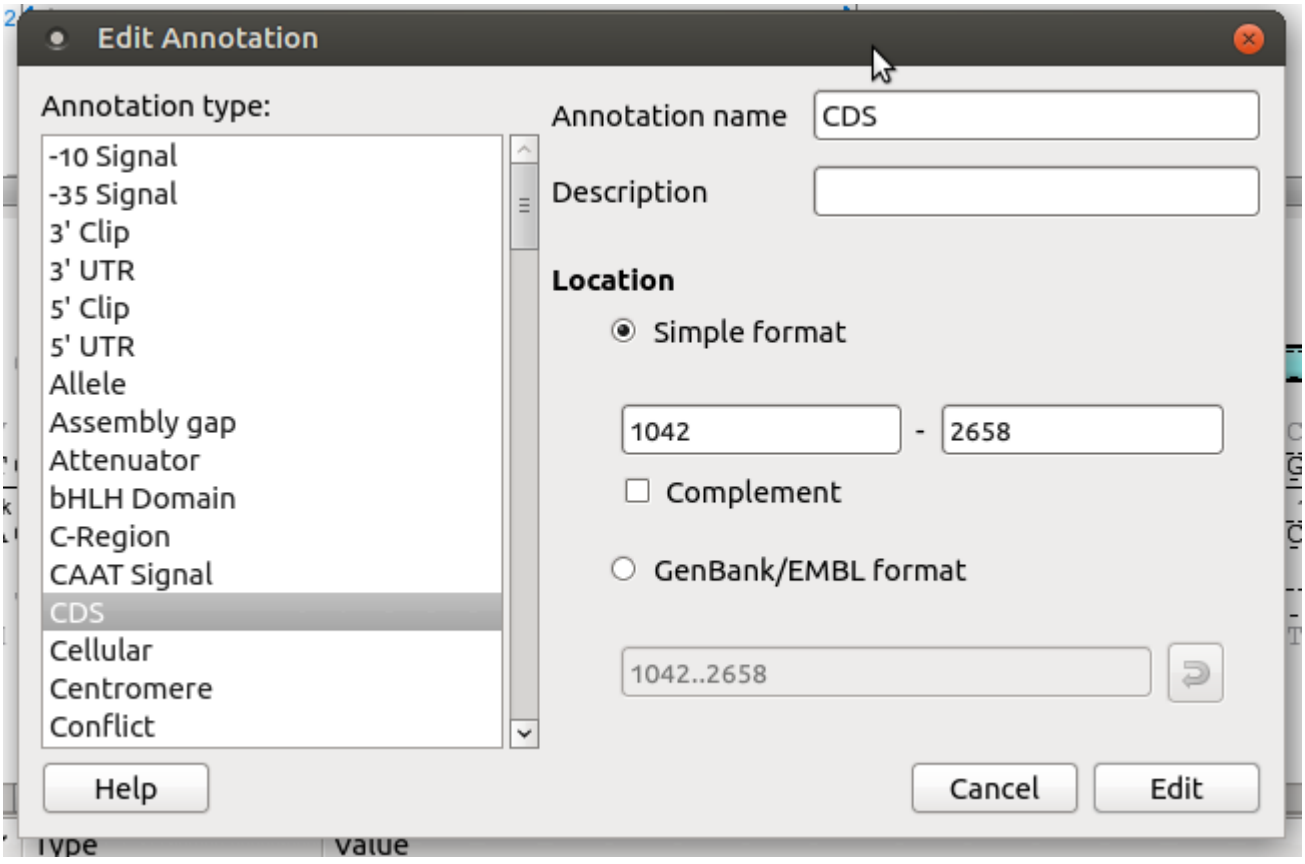
Selecting Annotations

To select one annotation click on it. To select several annotations hold *Ctrl* key while clicking on the annotations. To invert the selection use the *Invert annotation selection* item in the *Annotations editor* context menu.

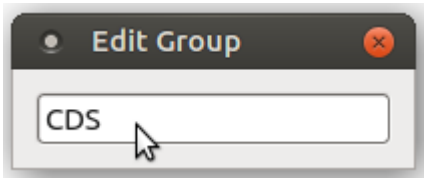
Double-clicking on an annotation also selects the corresponding sequence region.

Editing Annotation

If the document is not locked, it is possible to edit an annotation or an annotation group using the *Rename item* context menu from the *Annotation Editor* or from the *Sequence View* or with a help F2 key in the *Annotation Editor*. The result of pressing for an annotation:



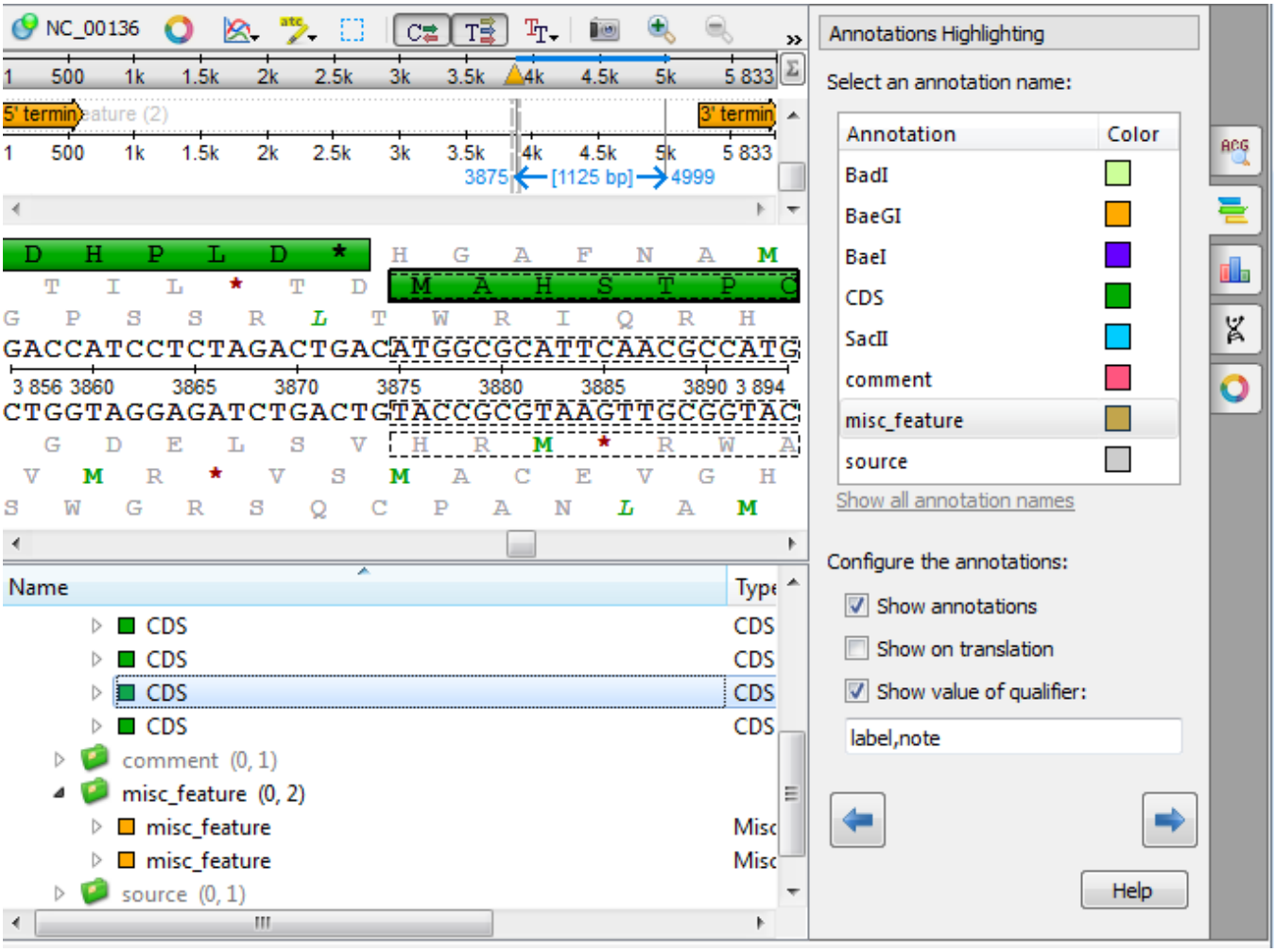
The result of pressing for an annotation group:



Highlighting Annotations

To configure settings of annotation names go to the *Annotation Highlighting* tab in the *Options Panel*.

By default the tab shows annotations names of the opened *Sequence View*.



If you want to see all annotation names, click the *Show all annotation names* link. The *Previous annotation* and *Next annotation* buttons seek to the previous or to the next annotation of the view correspondingly.

Find below information about annotations names' properties that you can configure.

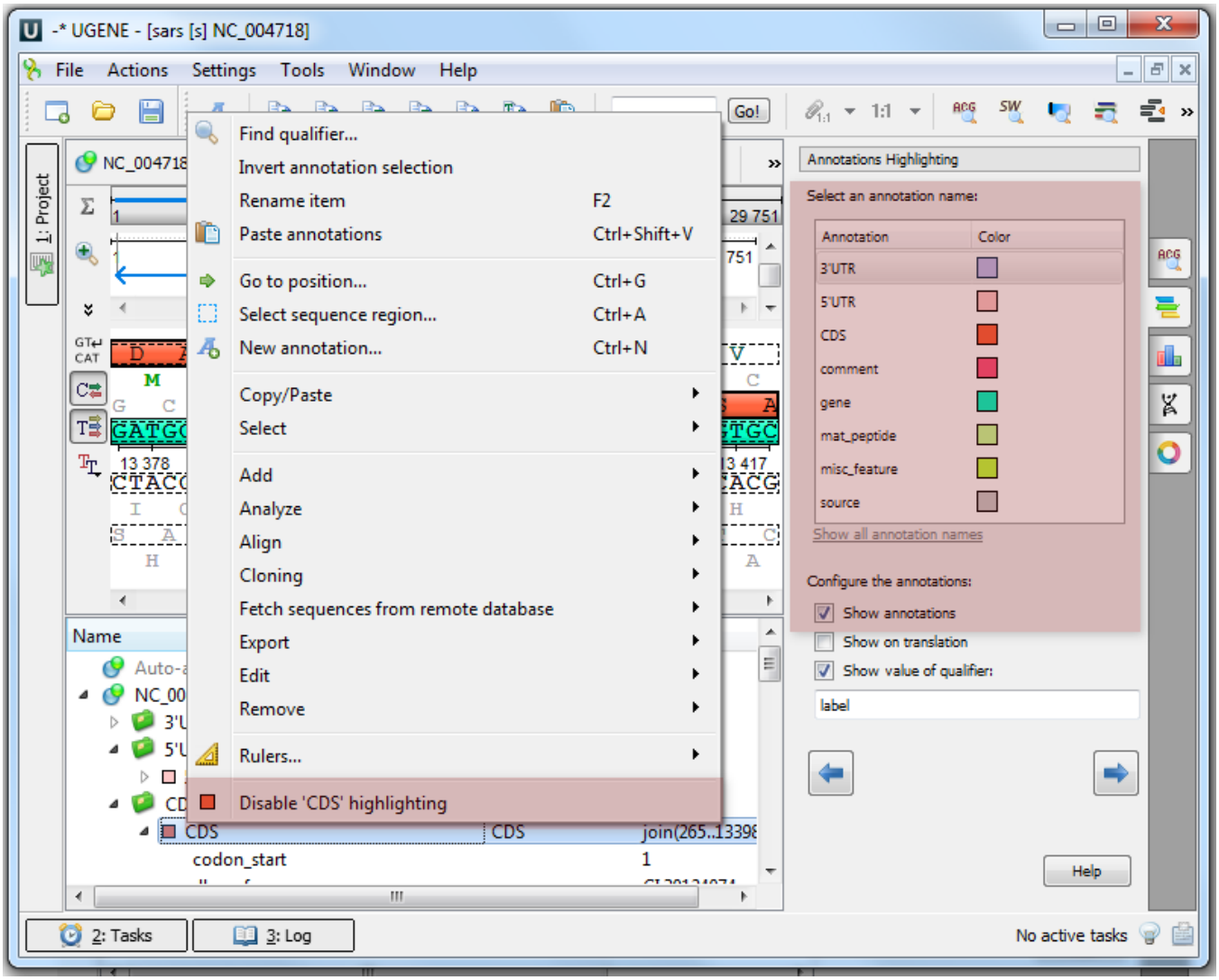
- Annotations Color
- Annotations Visability
- Show on Translation
- Captions on Annotations

Annotations Color

To change a color of all annotations of a certain type click on the corresponding color box in the annotations types table and select the required color in the appeared *Select Color* dialog.

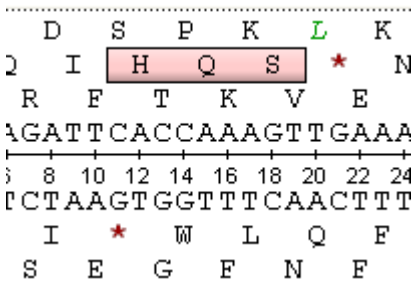
Annotations Visability

To show/hide annotations with a certain name, select this name in the annotations names table and check/uncheck the *Show annotations* check box below. Another way to show/hide the annotations is to select the *Enable/Disable highlighting* item in the context menu of an annotation.



Show on Translation

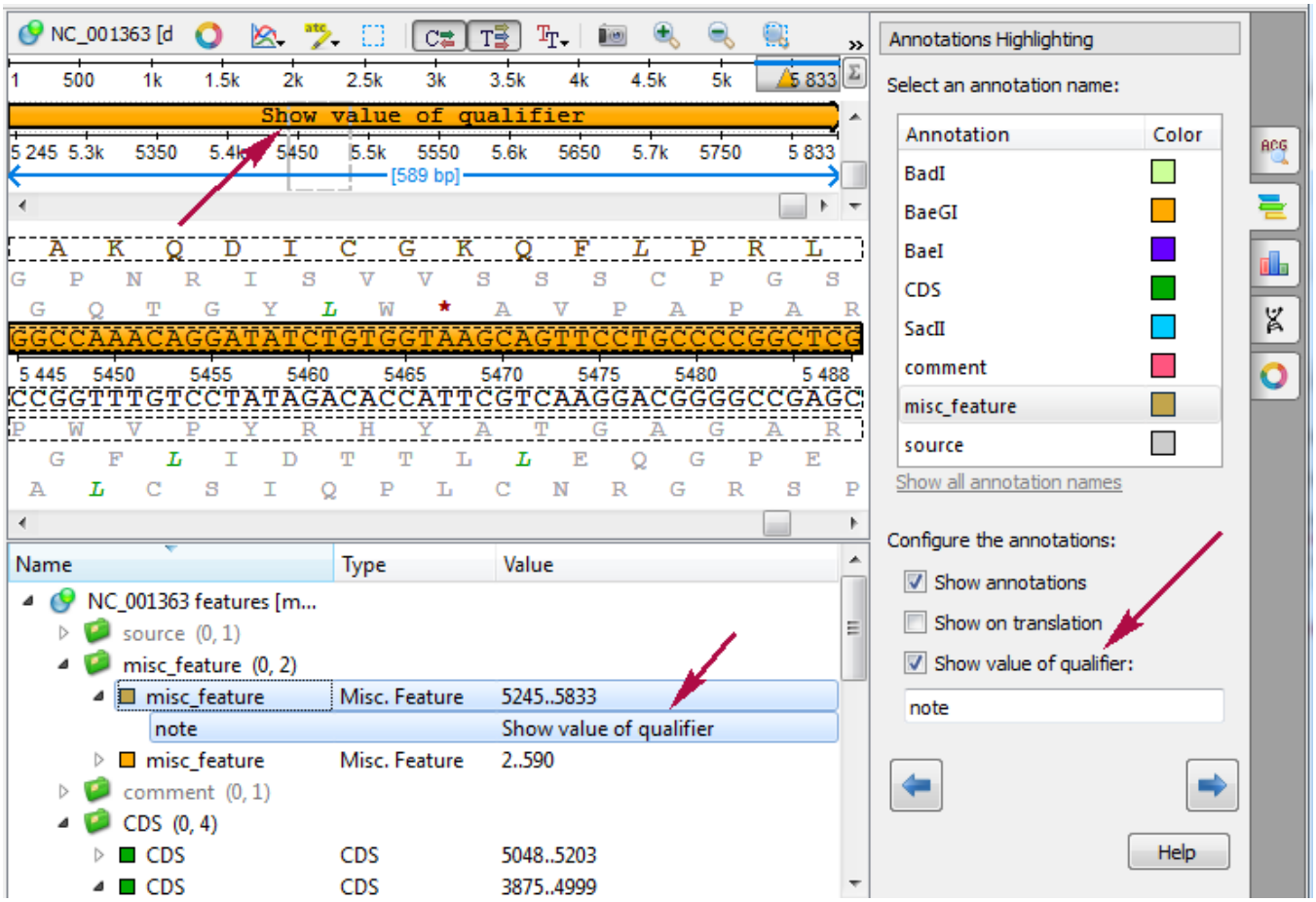
This option is available for nucleotide sequences only. It specifies to show the annotation on the corresponding amino sequence instead of the original nucleotide sequence in the *Sequence Detailed View*, for example:



You can enable/disable this option by checking/unchecking the *Show on translation* checkbox.

Captions on Annotations

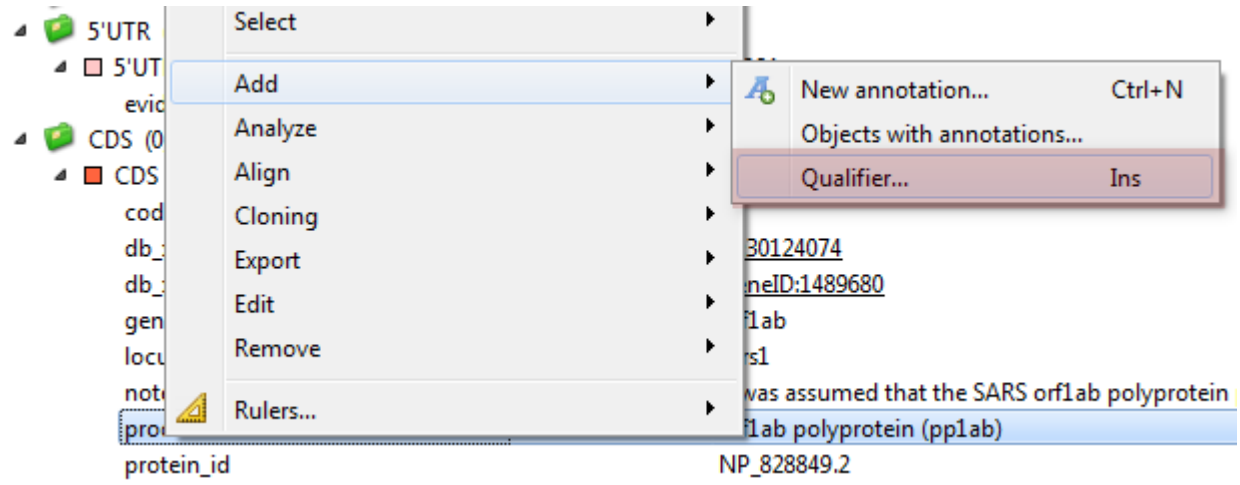
It is possible to show a value of a qualifier of an annotation instead of the annotation type name in the *Sequence Zoom View*. To enable this option for an annotation type check the *Show value of qualifier* check box and input the values of the required qualifiers in the text field nearby this check box. See the image below.



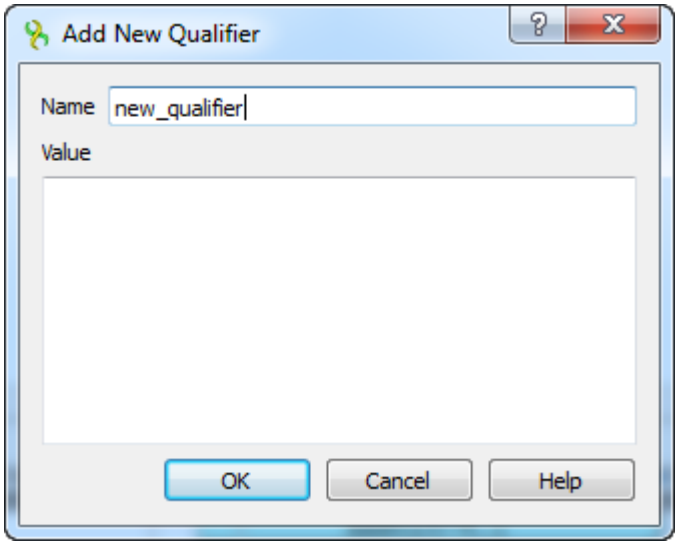
If you input several qualifiers names (separated by comma), then the first found qualifier is taken into account and shown on the annotation.

Creating and Editing Qualifier

To add a qualifier to an annotation select it in one of the *Sequence View* subviews and press the Insert key, or use the *Add Qualifier* context menu or the *A ctions* main menu item.

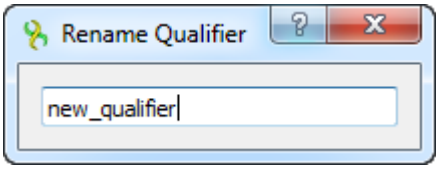


The dialog will appear:

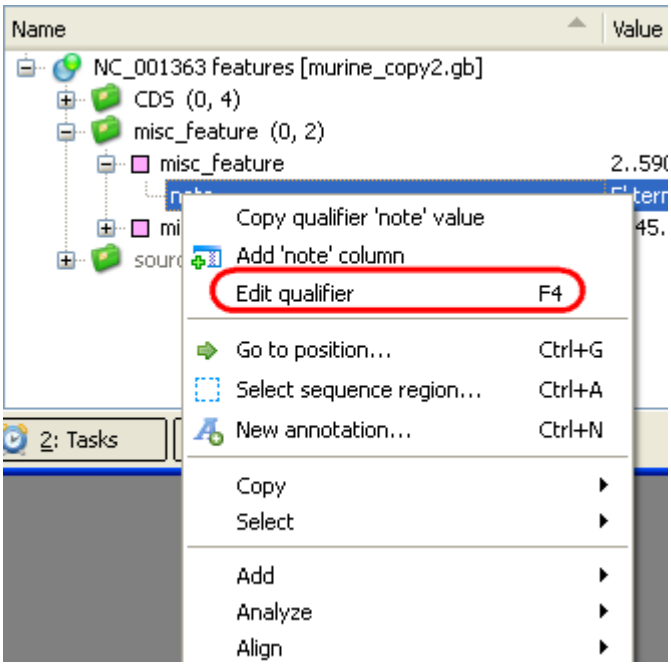


Here you can specify the name and the value of the qualifier.

You can use the F2 key to rename a qualifier:



To edit a qualifier, select the qualifier and press the F4 key or use the *Edit qualifier* context menu item:



Adding Column for Qualifier

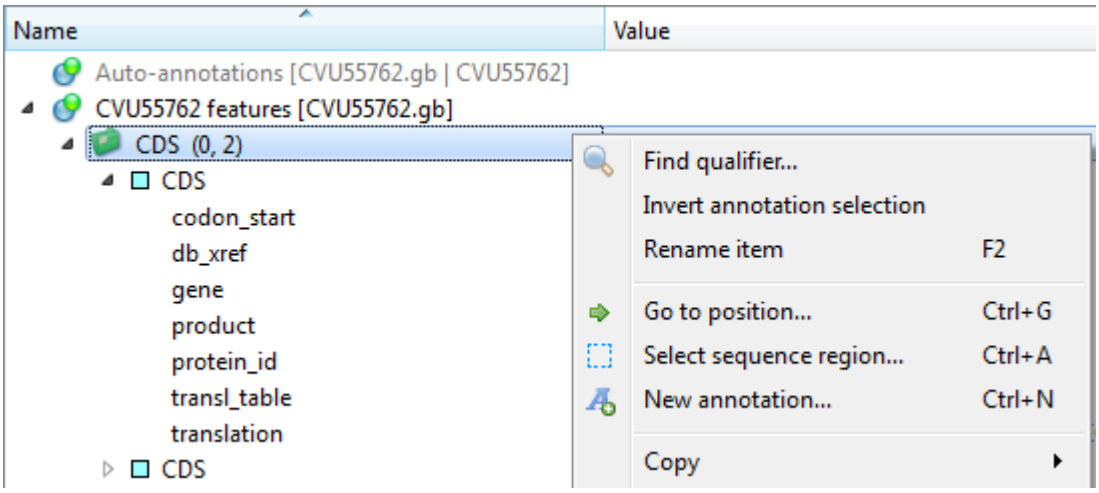
It is possible to add a column with the qualifier values to the *Annotations editor*. To add the column, select the *Add '[the qualifier name]' column* qualifier context menu item.

Copying Qualifier Text

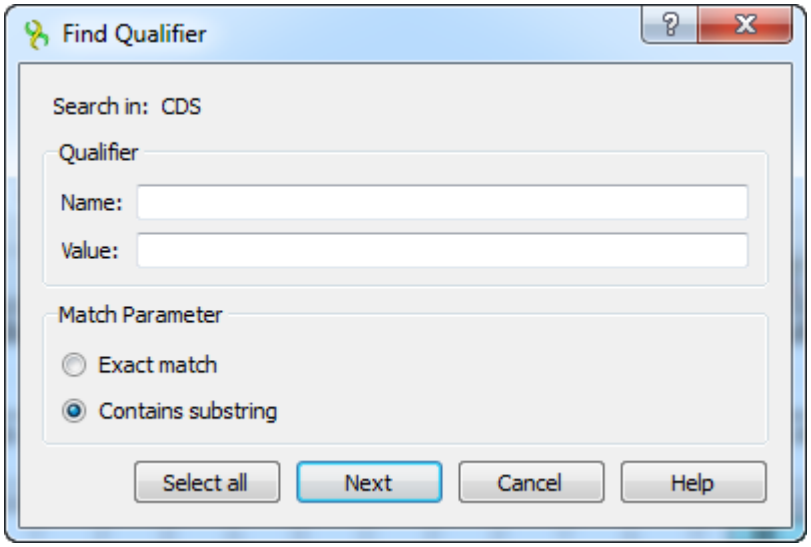
Use the *Copy qualifier '[the qualifier name]' text* qualifier context menu item to copy the qualifier value.

Finding Qualifier

To find a qualifier select annotation(s) or group(s) of annotations and use the *Find qualifier* context menu.



The dialog will appear:



Here you can specify the name and the value of the qualifier and select the searching parameter: *Exact match* or *Contains substring.*

Deleting Annotations and Qualifiers

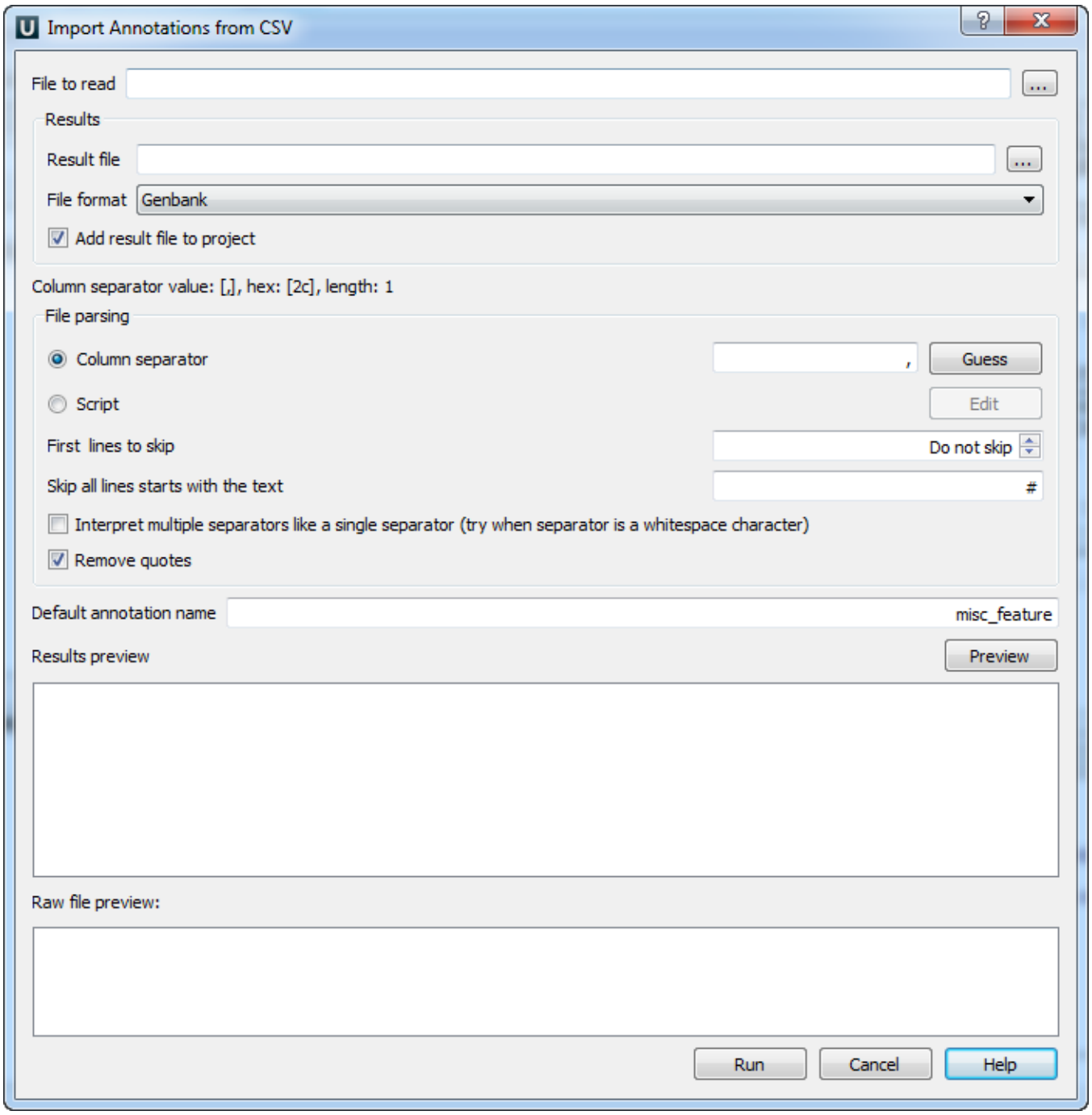
Selected annotations, groups and qualifiers can be deleted using the Delete key.

To remove an annotation object from the active view, select the object in the *Annotations editor* and press the Shift-Delete. Note that the object will not be removed from the project, but just from the active *Sequence View*. To add object again just drag and drop it to the *Sequence View*.

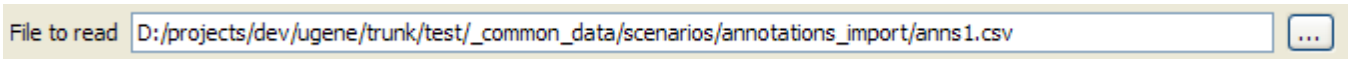
Importing Annotations from CSV

It is possible to import annotations for a sequence from an annotations table stored in the CSV format.

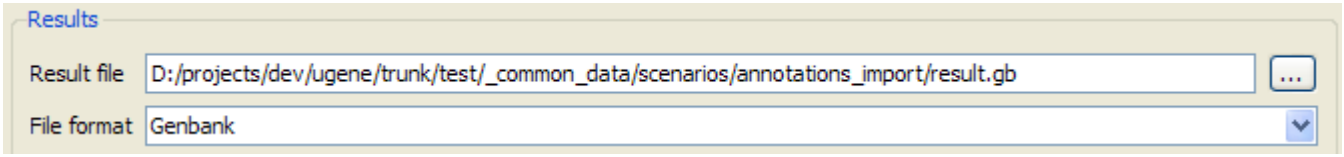
To import annotations from a CSV file, right-click on a *Project View* and select *Import* *Import annotations from CSV*. The following dialog box will appear:



Basically you need to specify the file to read annotations table from *(required)*:



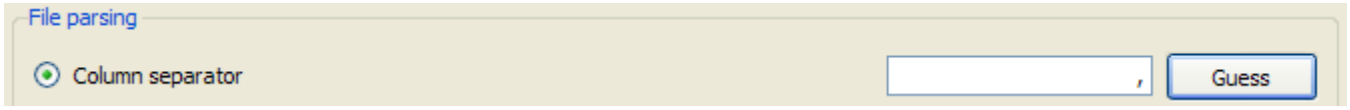
And the format of and the path to the file to write the annotations table into *(required)*:



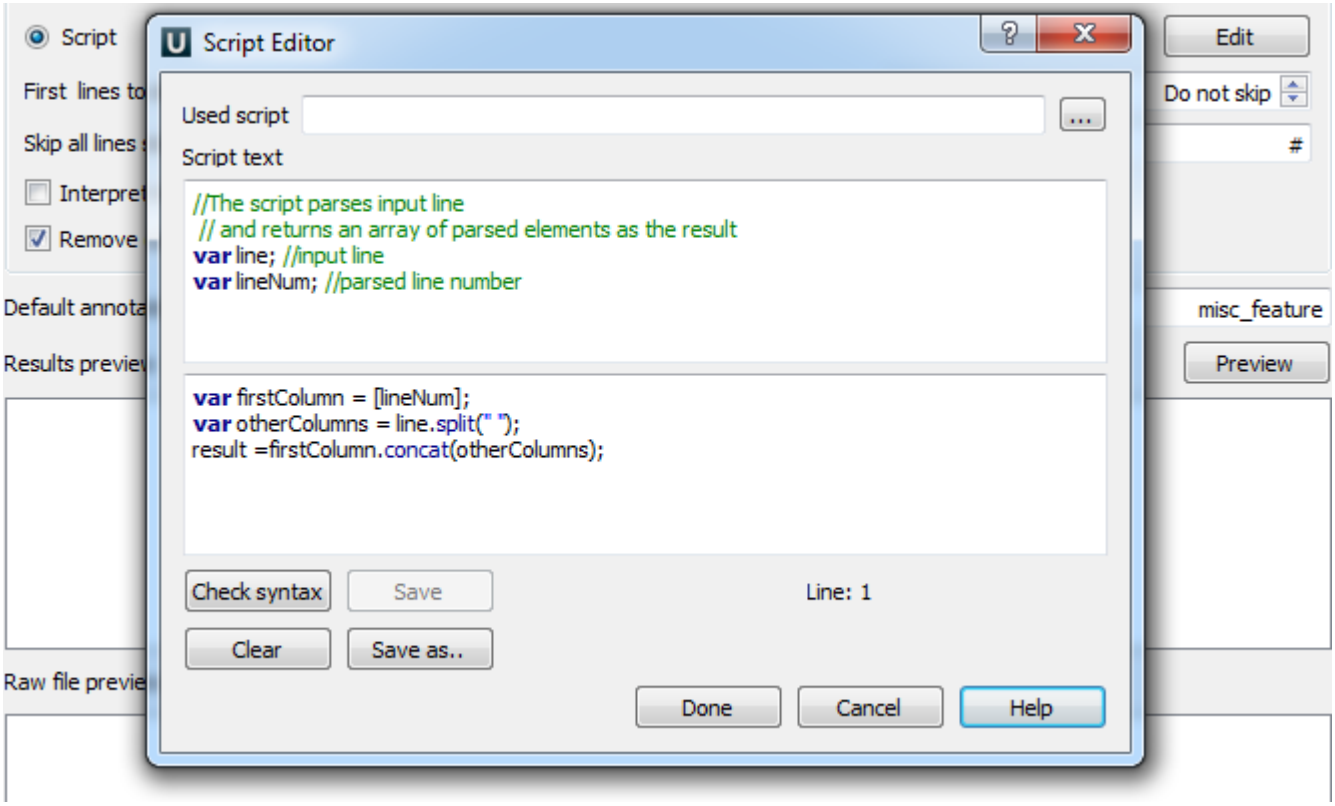
Check *Add result file to project* to link the annotations to the currently opened sequence.



To use a separator to split the table, check the *Column separator* item and specify the separator symbols. Also you can press *Guess* to try to detect the separator from the input file.



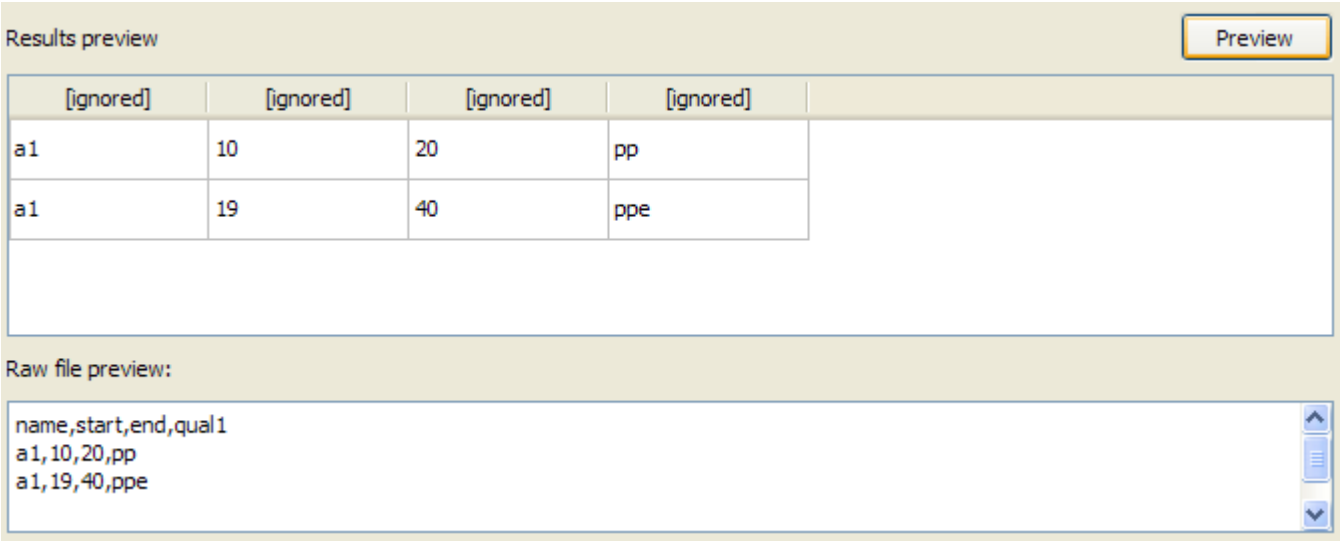
Alternatively, you can press *Edit* and edit the script which will specify the separator for each parsed line. It is possible to use line number in the script.



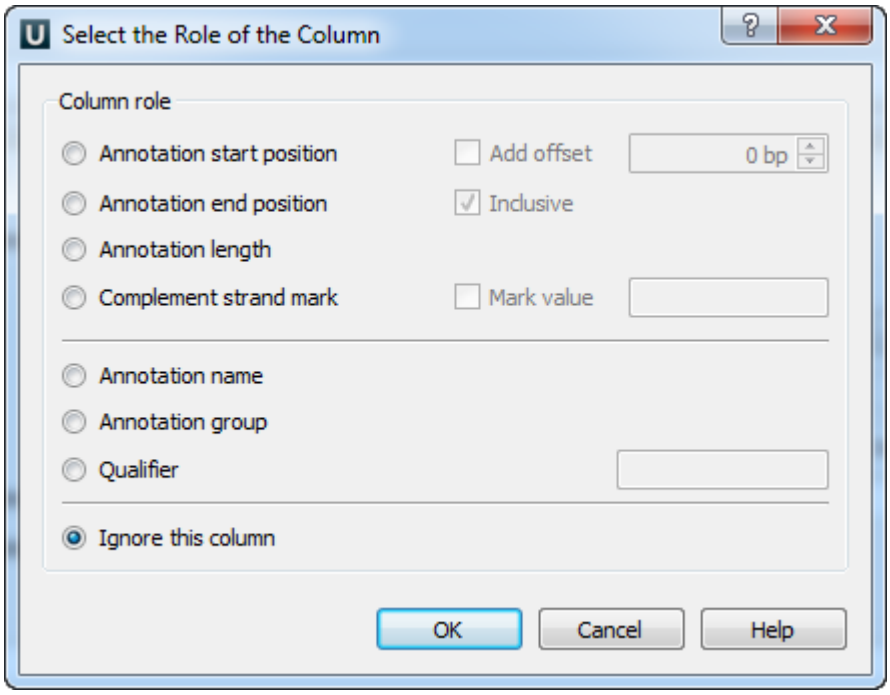
Using the arrows, you exclude the necessary number of lines at the beginning of the document from parsing. You can also skip all lines that start with the specified text.



By pressing *Preview* one can bring up the view of the current annotations table (which is produced from the input file with the specified parameters values). The input file contents will also be shown at the bottom part of the dialog.

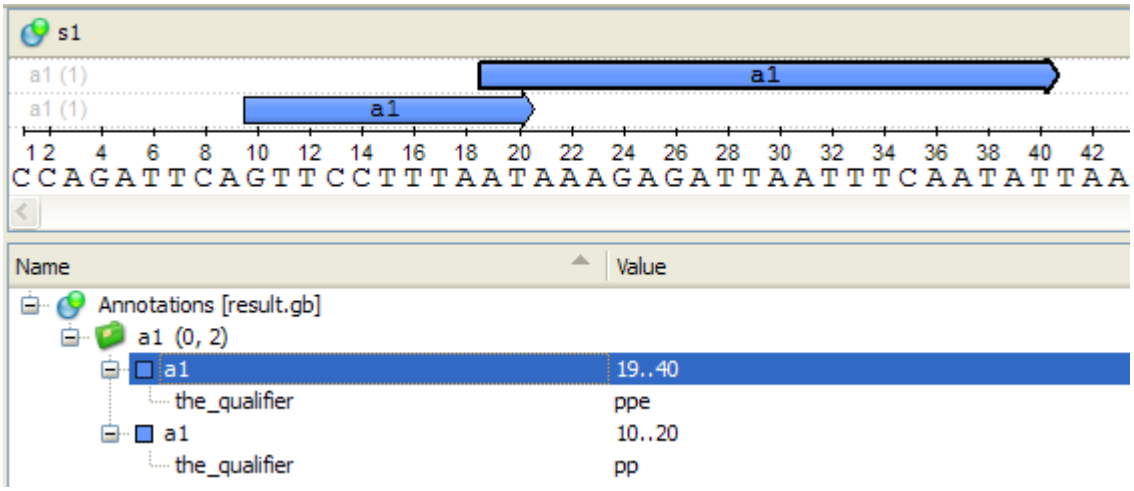


The preview table headline indicates the types of the information contained in the corresponding columns. By default the values are *[ignored]*. To specify a column role, click on the corresponding headline element:



The annotation start and end positions must be specified. It is possible to add an offset to every read start position by checking the *Add offset* checkbox, and to shorten annotations by one from the end by uncheking the *Inclusive* checkbox.

When all the roles are specified, press *Run*. With the *Add to project* checkbox specified and a *Sequence View* opened, on success you will see the *Sequence View* with annotations linked:

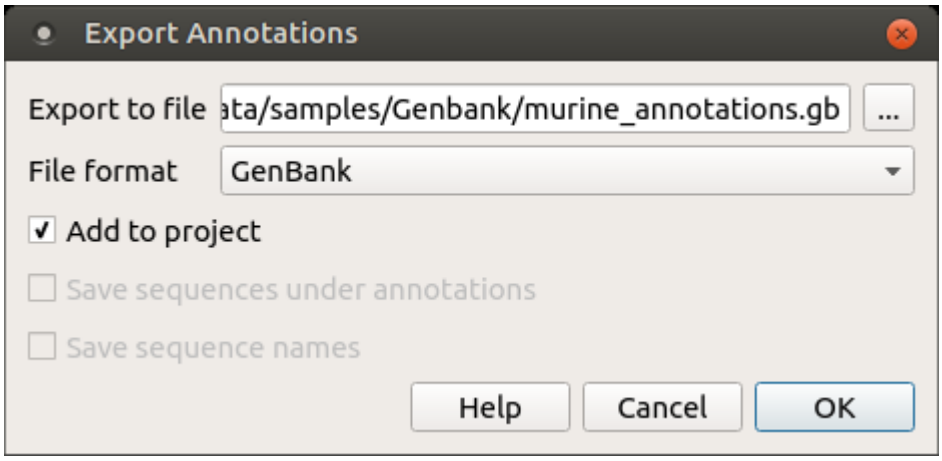


Exporting Annotations

To export annotation(s) open the *Sequence View* with a document that contains annotations. Select a single or several annotations or annotation groups in the *Annotation editor*, select the *Export Export annotations* context menu item.

Also, you can export annotation(s) from the *Project View* by the *Export Export annotations* context menu item.

The *Export Annotations* dialog will appear:

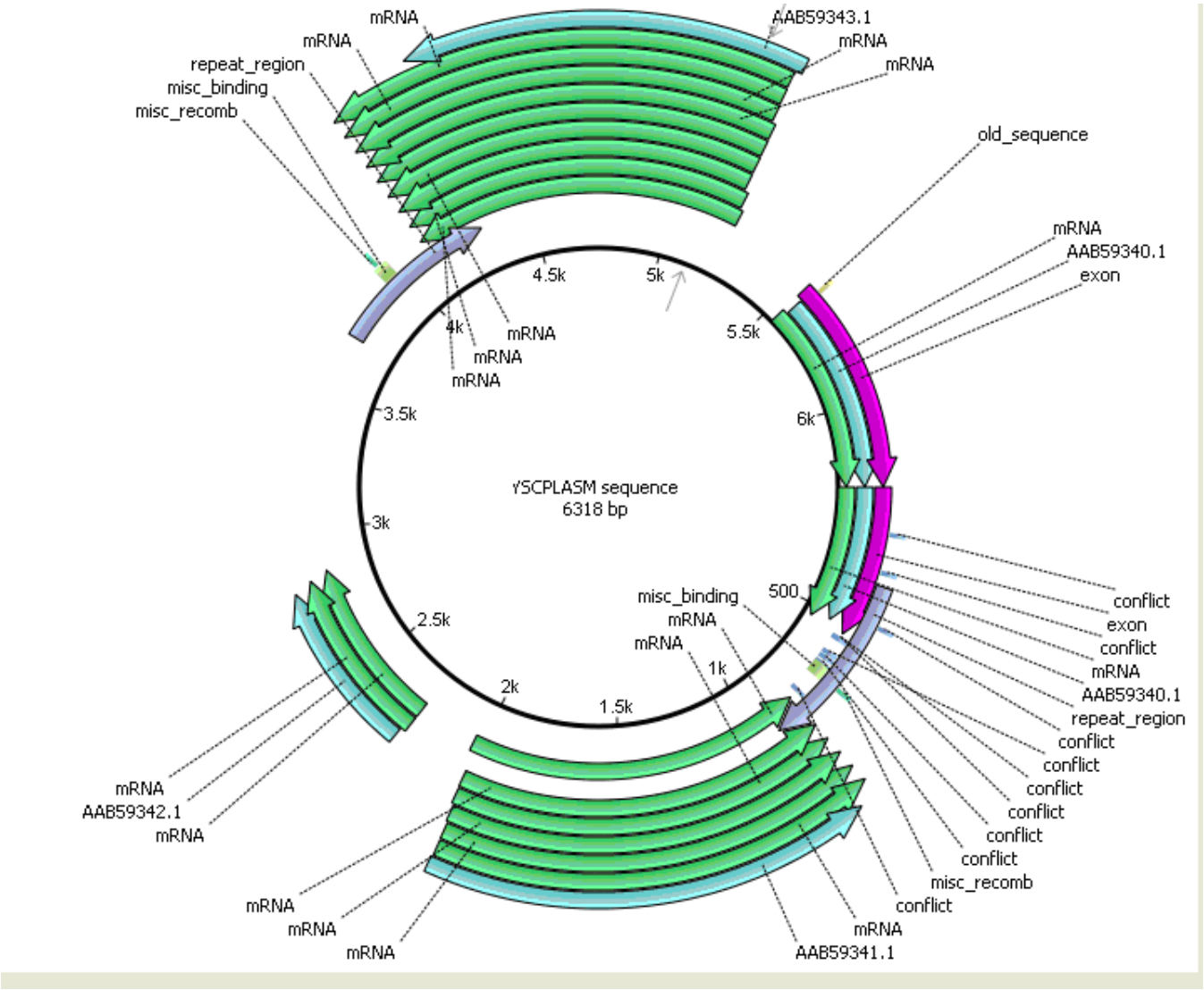


Here you can add the document to project, set the path to the file, choose the file format and optionally for CSV format you can save the sequence along with annotations and save sequence names.

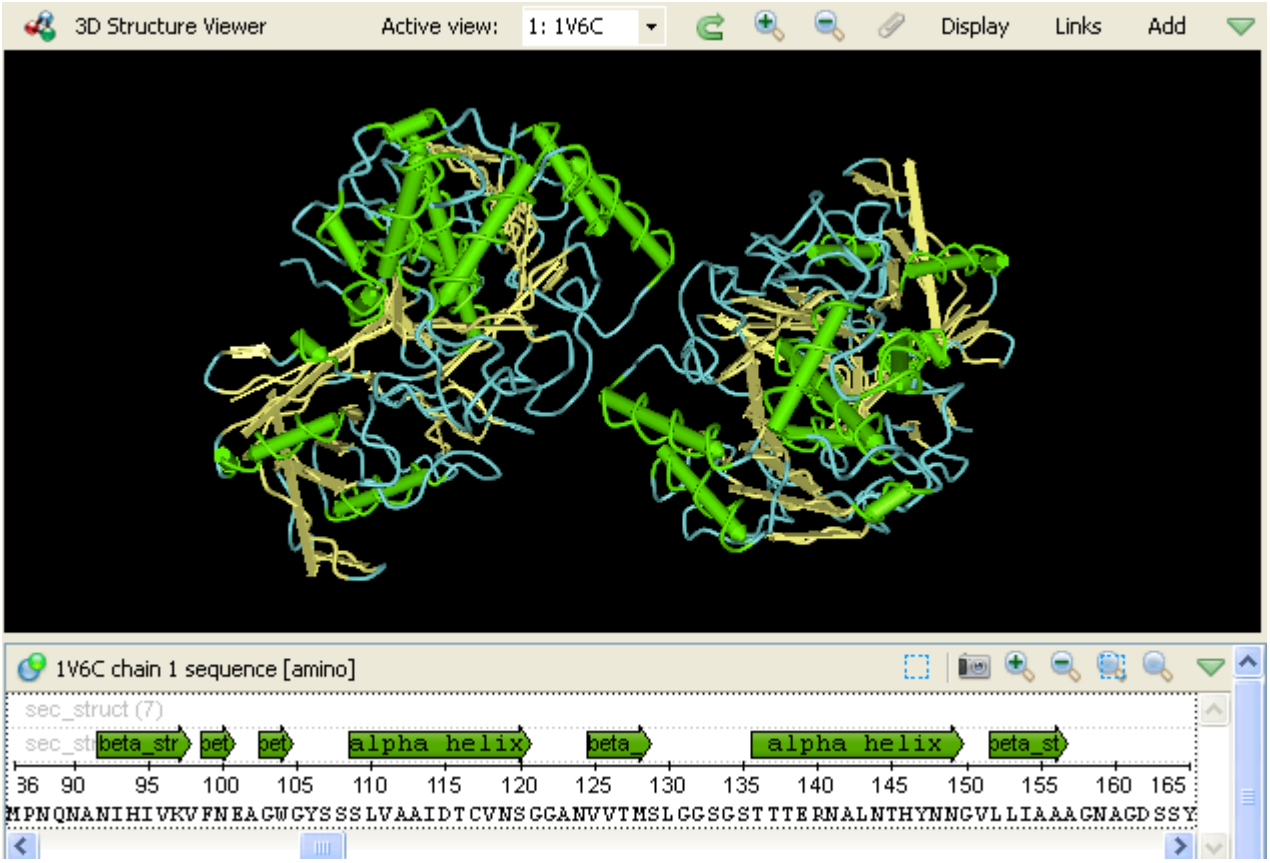
Sequence View Extensions

The functionality of the *Sequence View* can be significantly increased with *Sequence View Extensions*. Below is the demonstration its functionality.

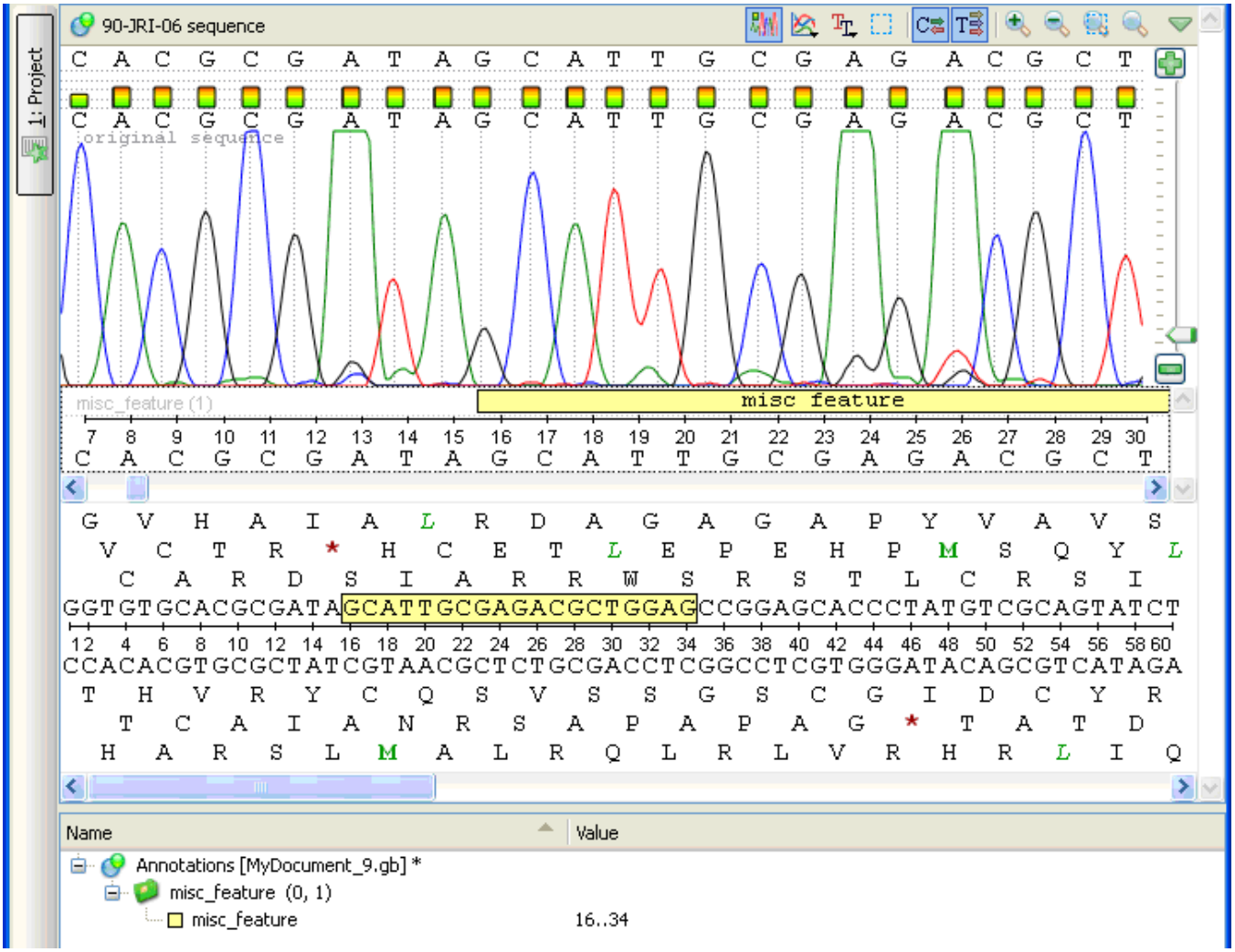
The *Circular Viewer* shows the circular view of a sequence:



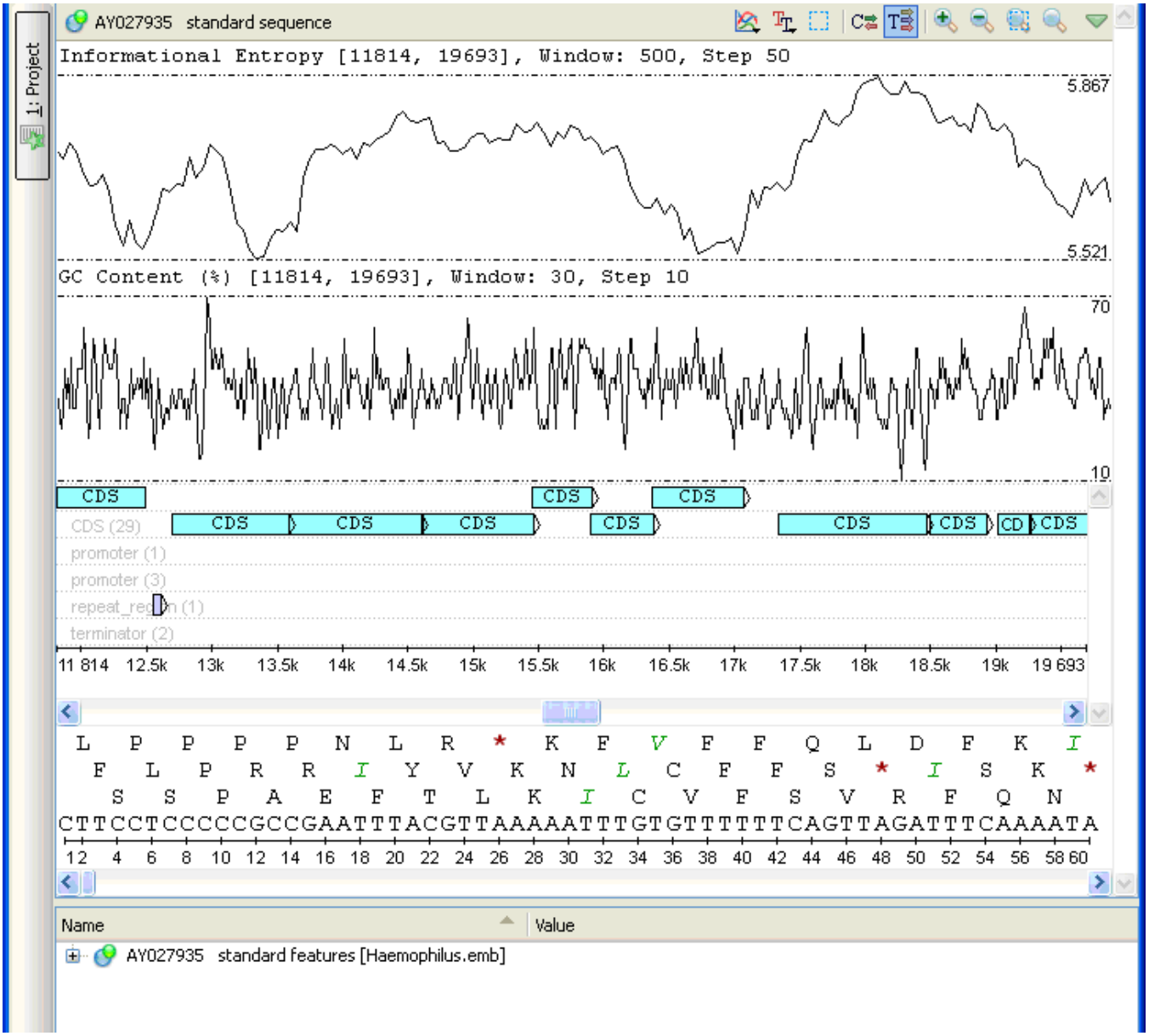
The *3D Structure Viewer* adds 3D visualization for PDB and MMDB files:



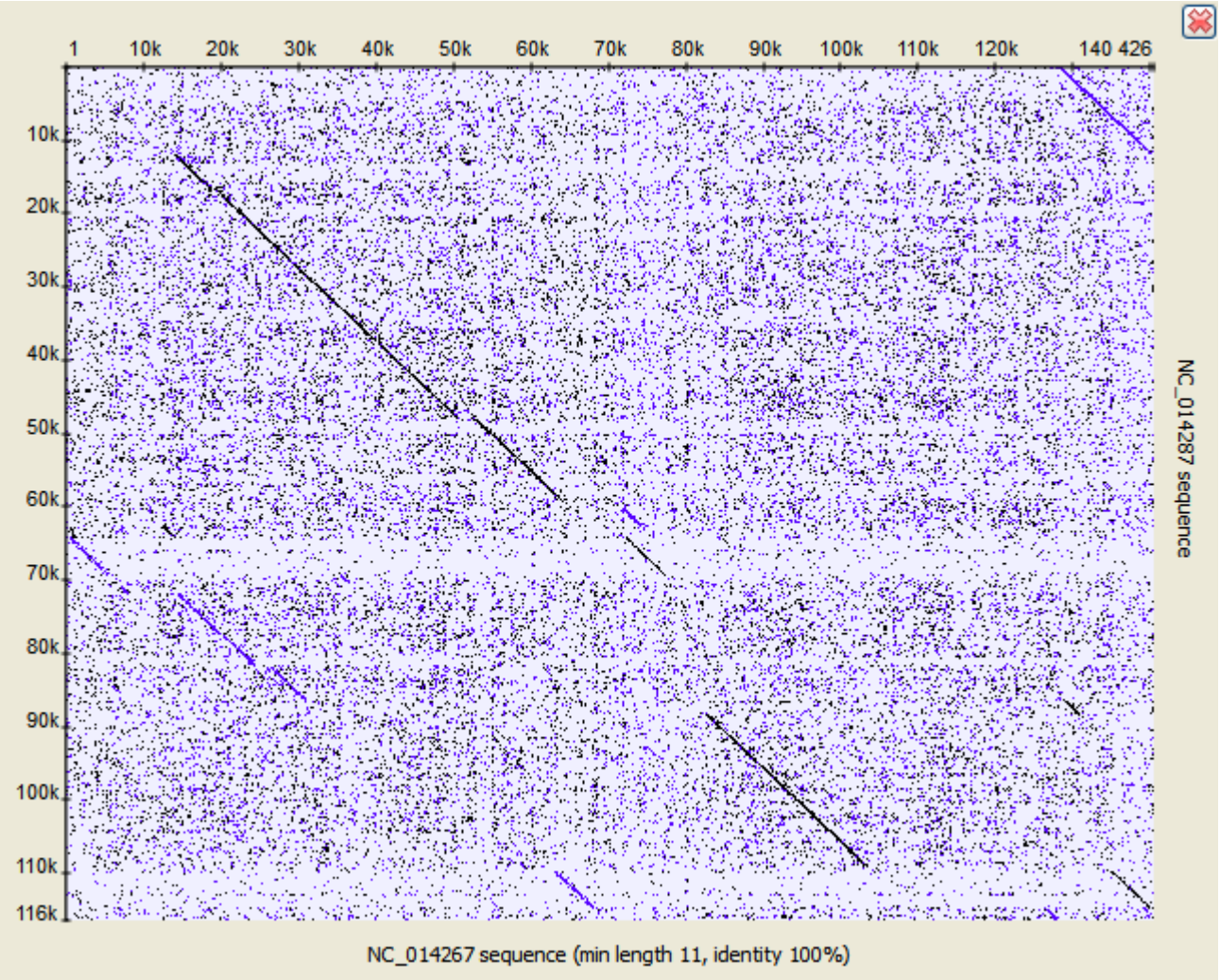
The *Chromatogram Viewer* adds support for chromatograms visualization and editing:



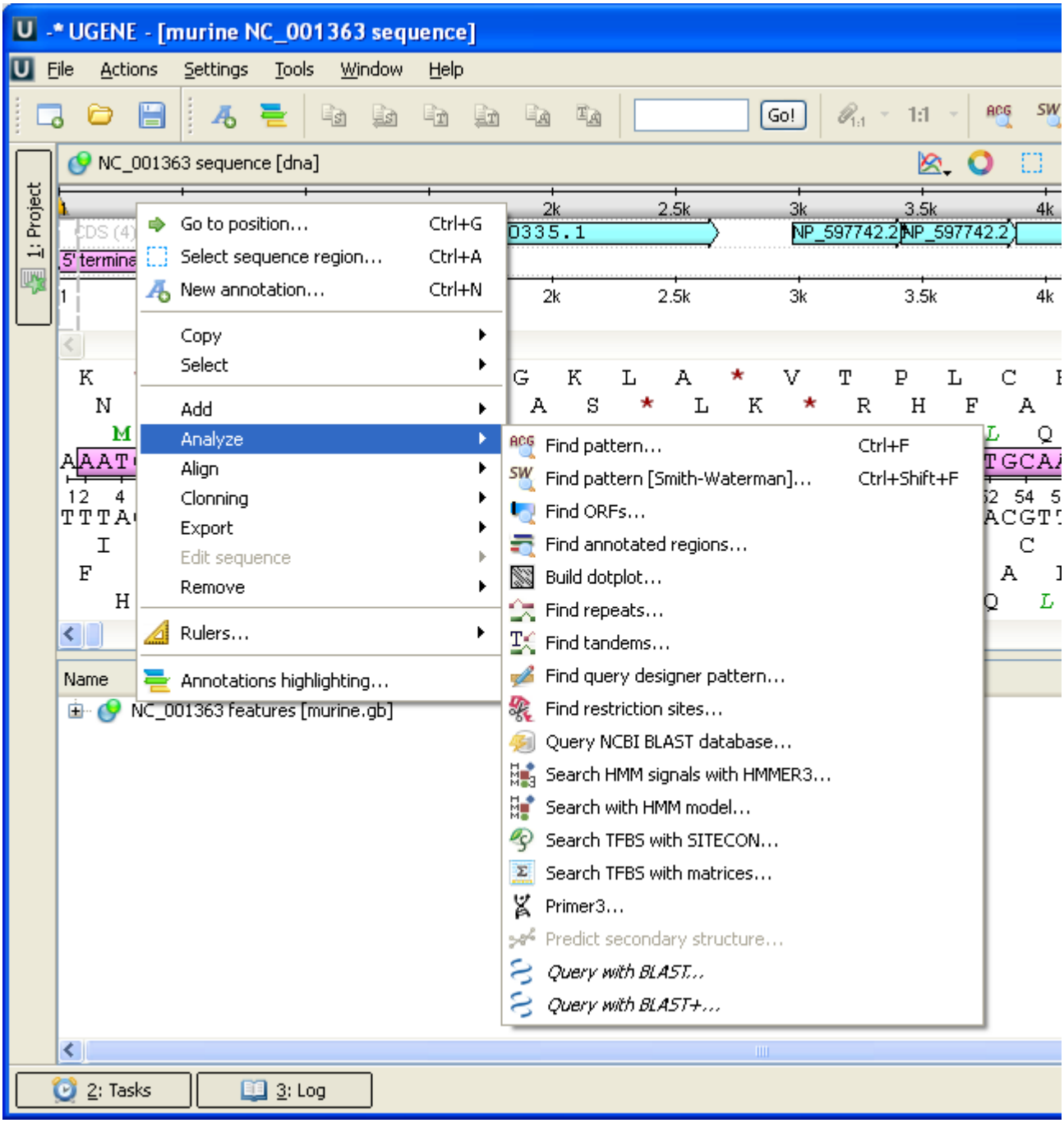
The *DNA Graphs Package* shows various graphs for sequences:



The *Dotplot* provides a tool to build dotplots for DNA or RNA sequences.



A number of other instruments add the graphical interface for popular sequence analysis methods:



For details see the next sections of the documentation:

- Circular Viewer
  - Circular View Settings
- 3D Structure Viewer
  - Opening 3D Structure Viewer
  - Changing 3D Structure Appearance
    - Selecting Render Style
    - Selecting Coloring Scheme
    - Calculating Molecular Surface
    - Selecting Background Color
    - Selecting Detail Level
    - Enabling Anaglyph View
  - Moving, Zooming and Spinning 3D Structure
  - Highlight Region on 3D Structure
  - Selecting Models to Display
  - Structural Alignment
  - Exporting 3D Structure Image

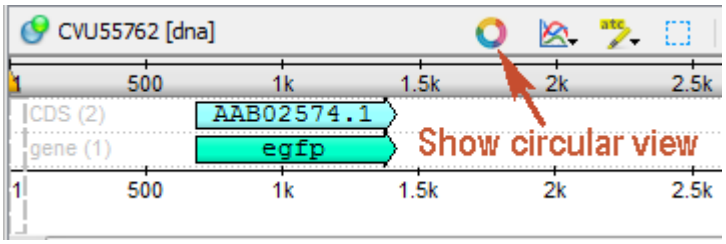
- Working with Several 3D Structures Views
- Chromatogram Viewer
  - Exporting Chromatogram Data
  - Viewing Two Chromatograms Simultaneously
- Graphs Package
  - Description of Graphs
  - Graph Settings
  - Saving Graph Cuttoffs as Annotations
- Dotplot
  - Creating Dotplot
  - Navigating in Dotplot
  - Zooming to Selected Region
  - Selecting Repeat
  - Interpreting Dotplot: Identifying Matches, Mutations, Invertions, etc.
  - Editing Parameters
  - Filtering Results
  - Saving Dotplot as Image
  - Saving and Loading Dotplot
  - Building Dotplot for Currently Opened Sequence
  - Comparing Several Dotplots

Circular Viewer

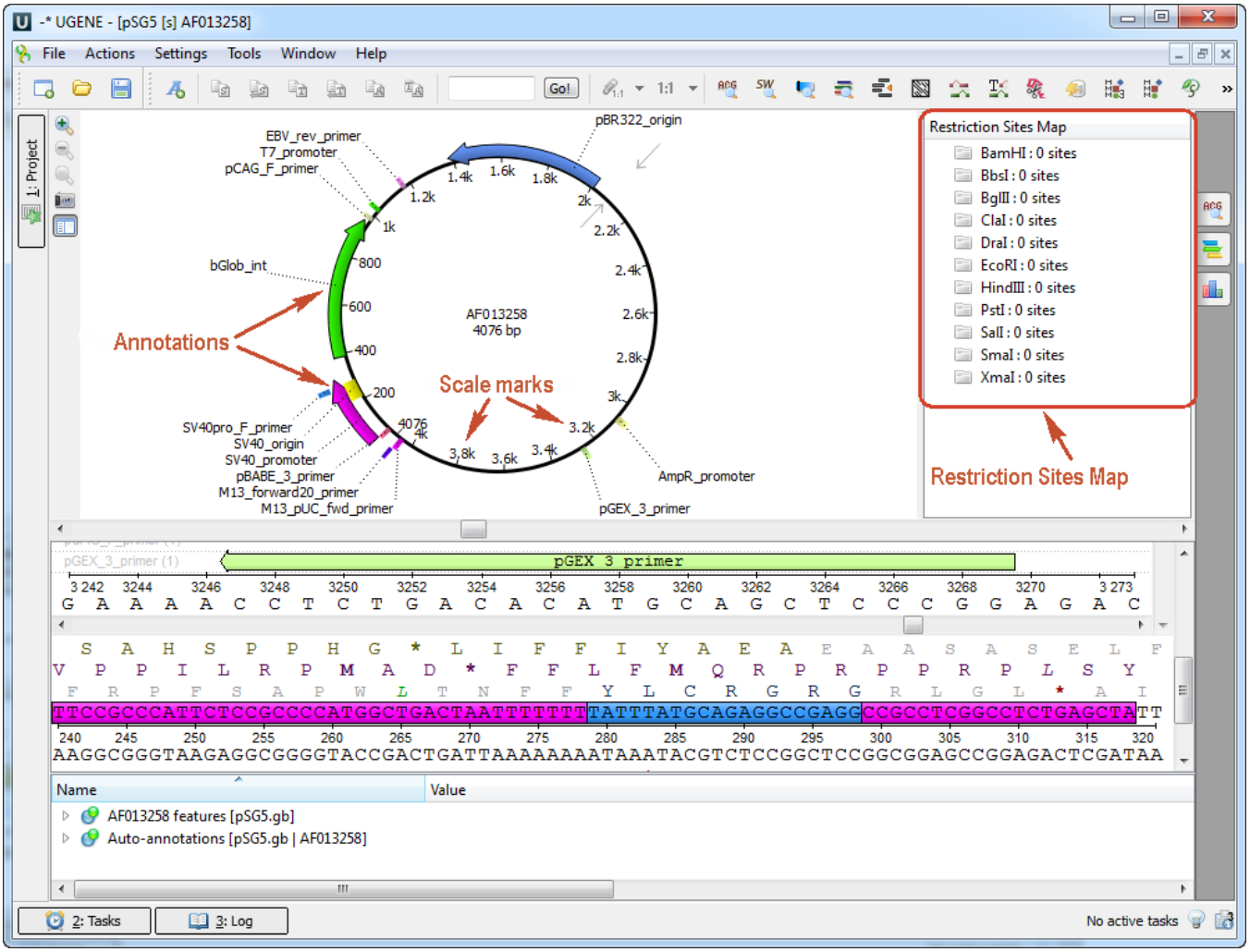
The *Circular Viewer* plugin provides capability to show the circular view of a nucleotide sequence.

Usage example:

Open a nucleotide sequence object in the *Sequence View*. The *Show circular view* button is available on the sequence toolbar:



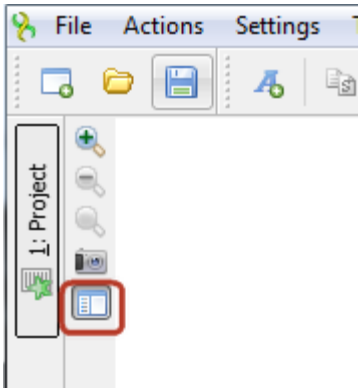
Pressing the button will show the circular view of the sequence:



If you work with file with many sequences the button closes circular views if some circular views are opened and if all circullar views are closed, it opens all of them.

Also you can mark sequences as circular in UGENE by the *Mark as circular* sequence context menu item. When the sequences are marked as *Circular*, the Circular View is automatically opened for them in all opened Sequence View windows.

The *Restriction Sites Map* will appear automatically. To show restriction sites the *Show Restriction Sites* menu should be checked. To hide the map click on the following button:

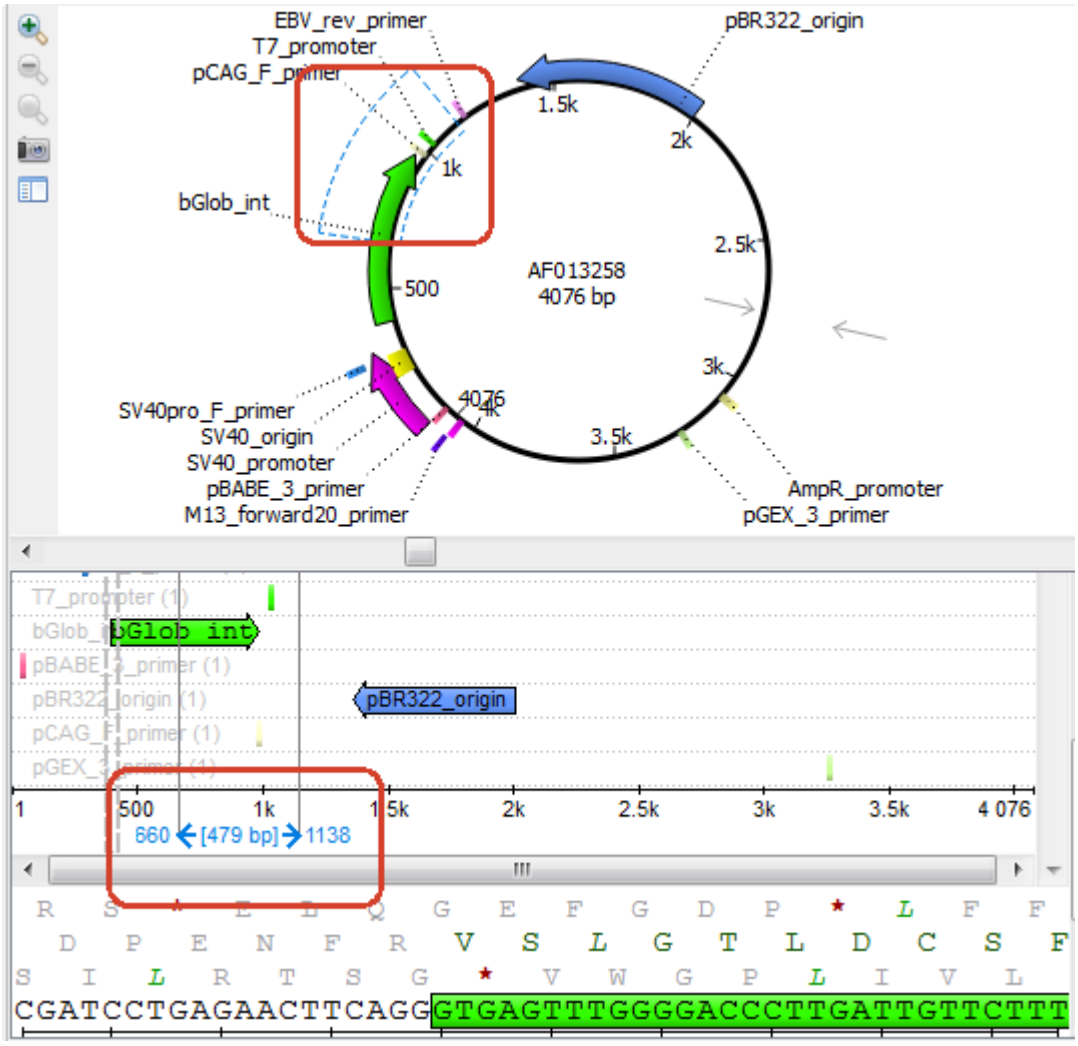


The *Circular Viewer* is opened automatically when the *Sequence View* is opened for a plasmid.

The inner circle represents the sequence clockwise and the scale marks show the corresponding sequence positions. The sequence annotations are represented as curved colored regions at the outer side of the circle.

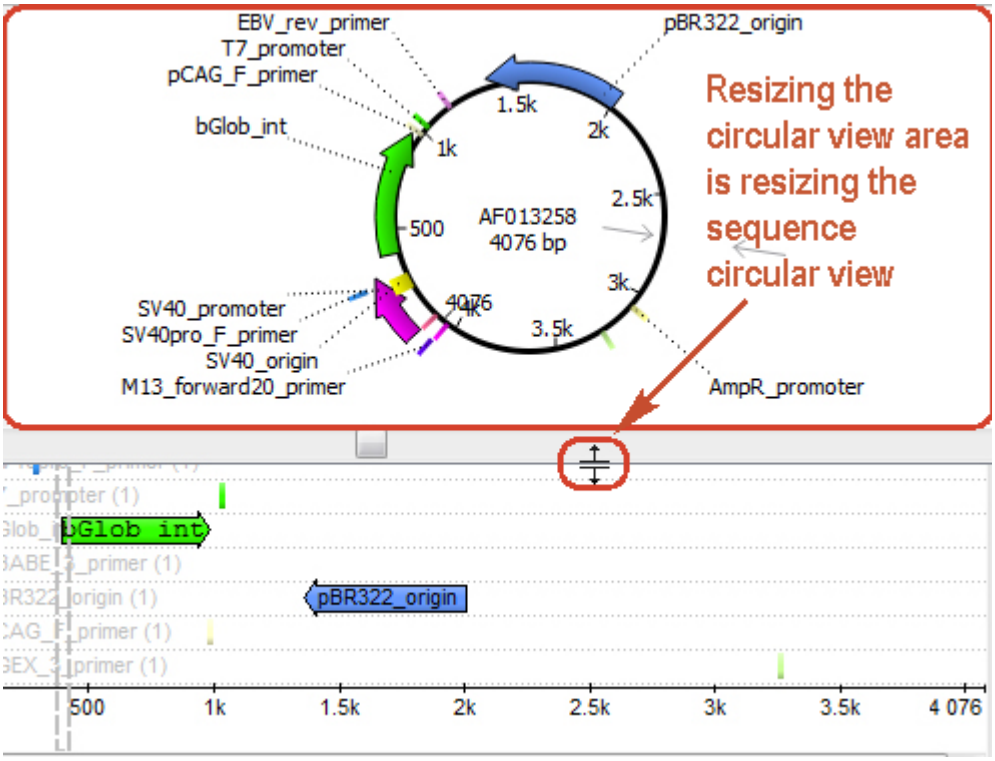
The *Circular Viewer* helps to navigate within the sequence. You can select an annotation on the circular view and the annotation will also be focused and highlighted in all *Sequence View* areas: *Sequence overview*, *Sequence zoom view*, *Sequence details view* and *Annotations editor*.

You can also select a sequence region:



This will also affect the *Sequence View*. You can select a sequence region with *Ctrl* and the selection will be inverted.

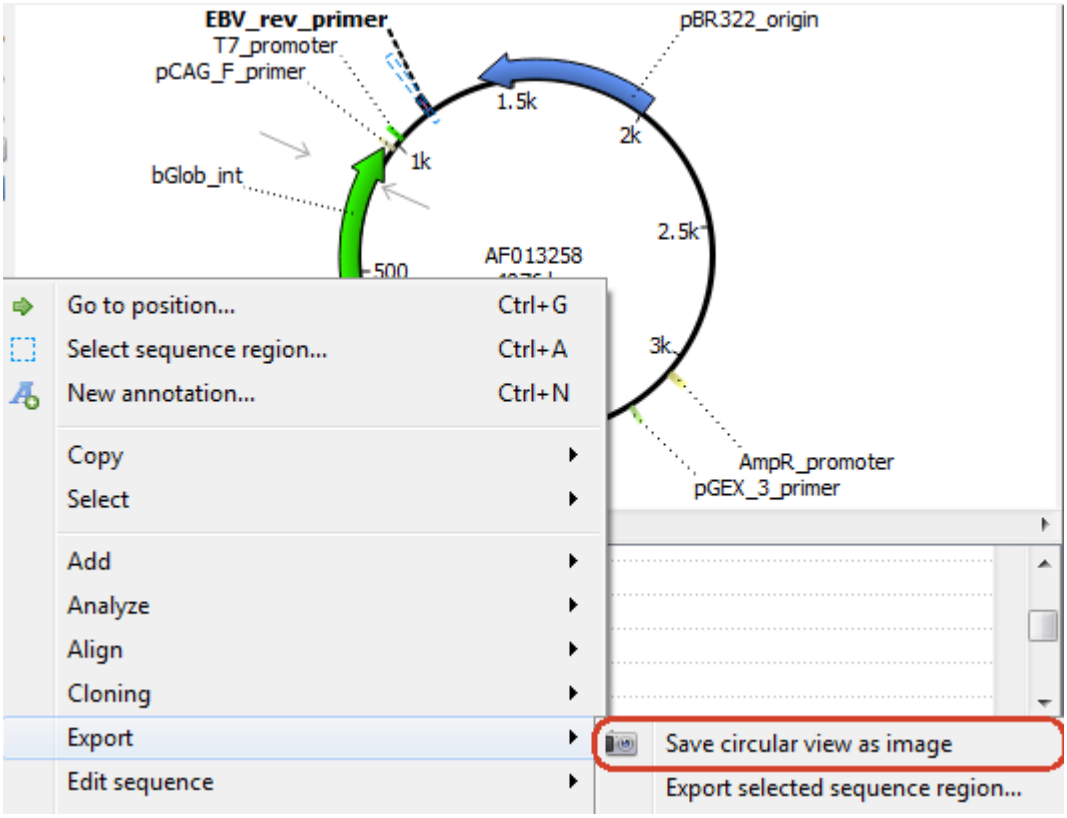
Note that the circular view is zoomed automatically when the *Circular Viewer* area is resized:



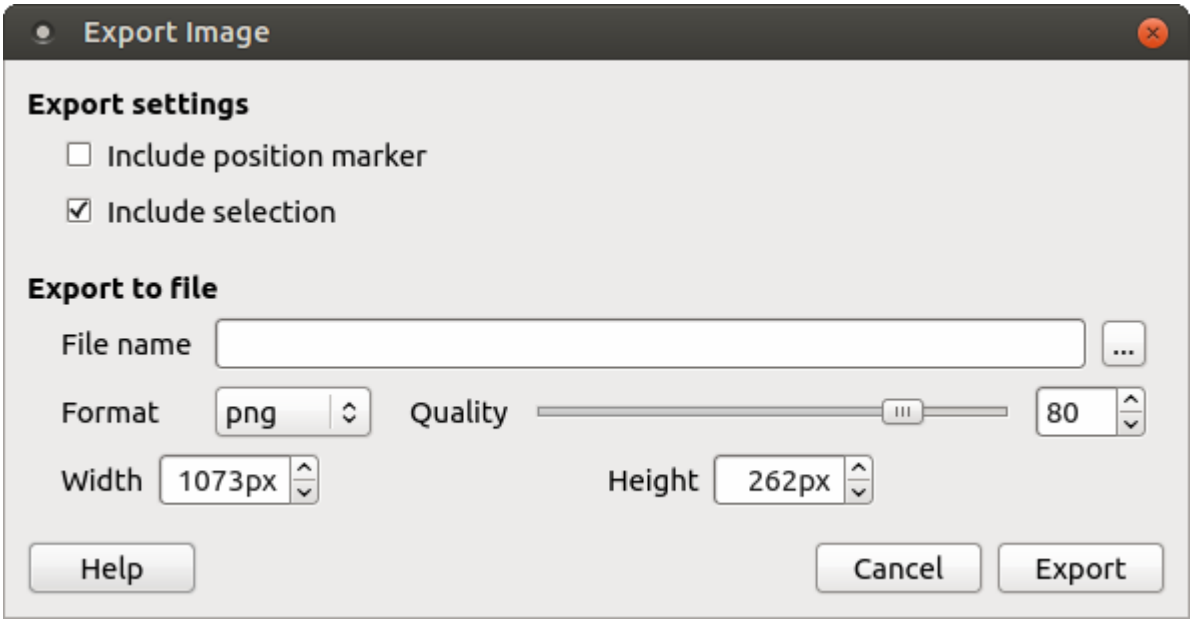
So you can adjust it to an appropriate size.

It is possible to rotate the circular view using the mouse wheel. Also it is possible to shift the start point of a circular molecule by *Edit sequence -> Set new sequence origin* context menu item.

Use the *Export Save circular view as image* context menu or the *Actions* main menu item to save the image of the circular view.

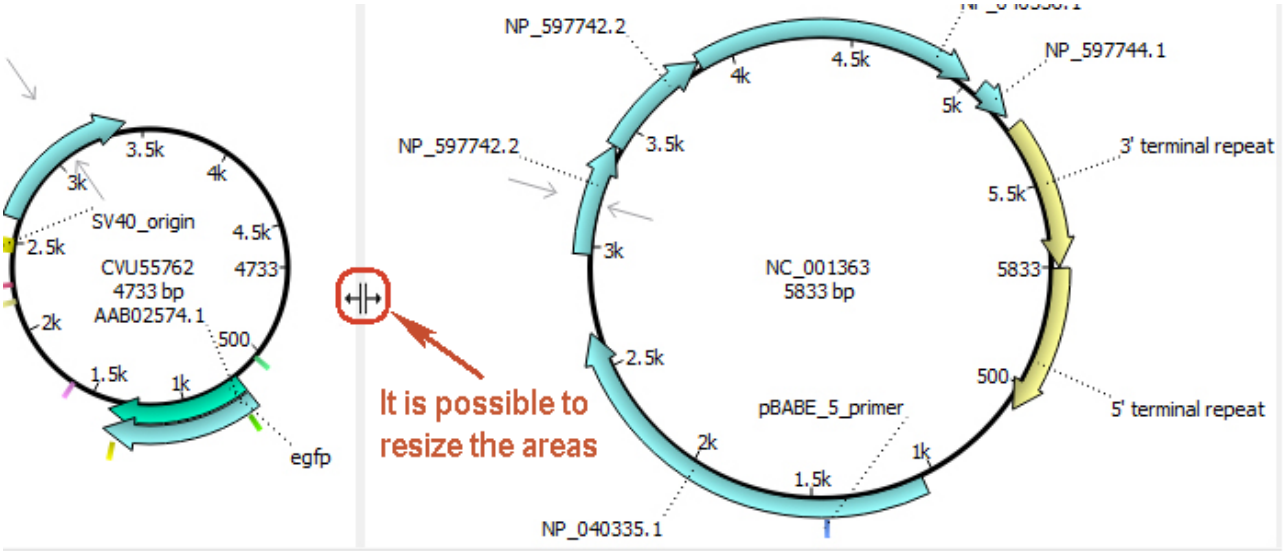


The *Export Image* dialog will appear:



Here you can browse for the file name, select the width, height and resolution of the image as well as its format: svg, ps, pdf, bmp, jpeg, jpg, png, ppm, tif or tiff. Also you can include position and selection markers to the image by the corresponding checkboxes.

Note, that if a sequence file contains several sequences it is possible to view the circular views of the sequences in the same *Circular Viewer* area.

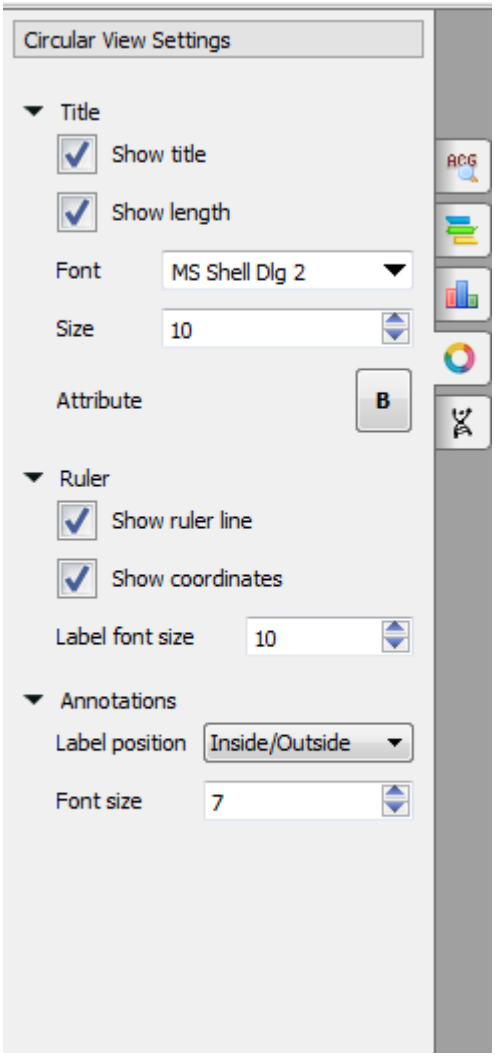


You can work with these circular views at the same time.

Circular View Settings

To configure circular view settings go to the *Circular View Settings* tab in the *Options Panel*.

Activate the circular view for a sequence and the following settings will appear:



In the title section you can show or hide title and length, change font, size and attribute.

In the ruler section you can show or hide ruler line and coordinates and change the label font size.

In the annotation section you can select the label position and change the label size.

The following label positions are available:

- inside - all labels are inside of the annotations
- outside - all labels are outside of the annotations
- inside/outside - if the label can fit the annotation and it is not auto-annotation, it's located inside. Otherwise outside.
- none - no labels at all

3D Structure Viewer

The 3D Structure Viewer is intended for visualization of 3D structures of biological molecules.

Using the 3D Structure Viewer you can work with data from the Protein Data Bank (PDB) - a repository for the 3D structural data of large biological molecules, such as proteins and nucleic acids, maintained by the Worlwide Protein Data Bank (wwPDB).

You can work as well with data from the NCBI Molecular Modeling DataBase (MMDB), also known as “Entrez Structure”, a database of experimentally determined structures obtained from the RCSB Protein Data Bank.

Find the description of the 3D Structure Viewer’ features below.

- Opening 3D Structure Viewer
- Changing 3D Structure Appearance
  - Selecting Render Style
  - Selecting Coloring Scheme

    - Calculating Molecular Surface
    - Selecting Background Color
    - Selecting Detail Level
    - Enabling Anaglyph View
- Moving, Zooming and Spinning 3D Structure
- Highlight Region on 3D Structure
- Selecting Models to Display
- Structural Alignment
- Exporting 3D Structure Image
- Working with Several 3D Structures Views

Opening 3D Structure Viewer

The *3D Structure Viewer* is opened automatically when you open a PDB or MMDB file.

For example, *open* $UGENE/data/samples/PDB/1CF7.PDB. The 3D Structure Viewer adds a view to the upper part of the *Sequence View*:



Notice the *Links* button on the toolbar. When you click the button the menu appears with quick links to online resources with detailed information about the molecule opened:

- PDB Wiki
- RSCB PDB
- PDBsum
- NCBI MMDB

Note that if you're online, you can access the Protein Data Bank directly from UGENE and load a required file by its PDB ID (see *Fetching Data from Remote Database* for details).

Hint

Don't forget to select the correct database (PDB) while fetching.

Changing 3D Structure Appearance

This chapter describes how you can change a 3D stucture appearance.

- Selecting Render Style
- Selecting Coloring Scheme

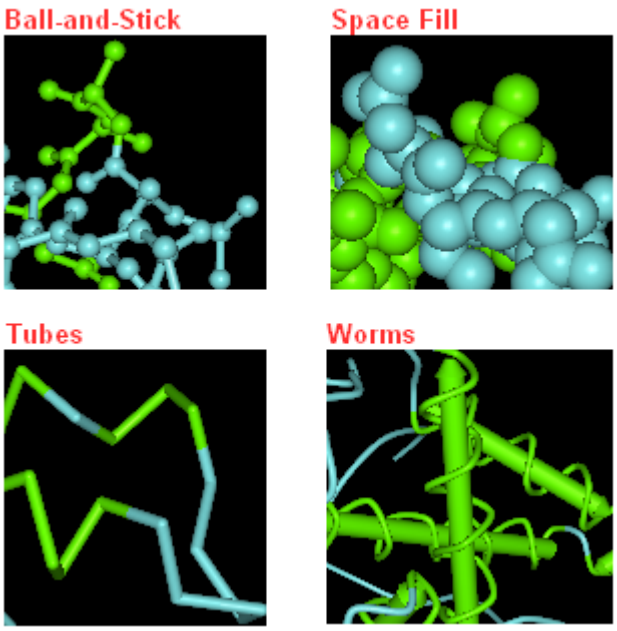
- Calculating Molecular Surface
- Selecting Background Color
- Selecting Detail Level
- Enabling Anaglyph View

Selecting Render Style

The following render styles are available:

- *Ball-and-Stick*
- *Space Fill*
- *Tubes*
- *Worms*

To change the render style select an appropriate item in the *Render Style* menu (it can be found either in the 3D Structure Viewer context menu or in the the *Display* menu on the toolbar).

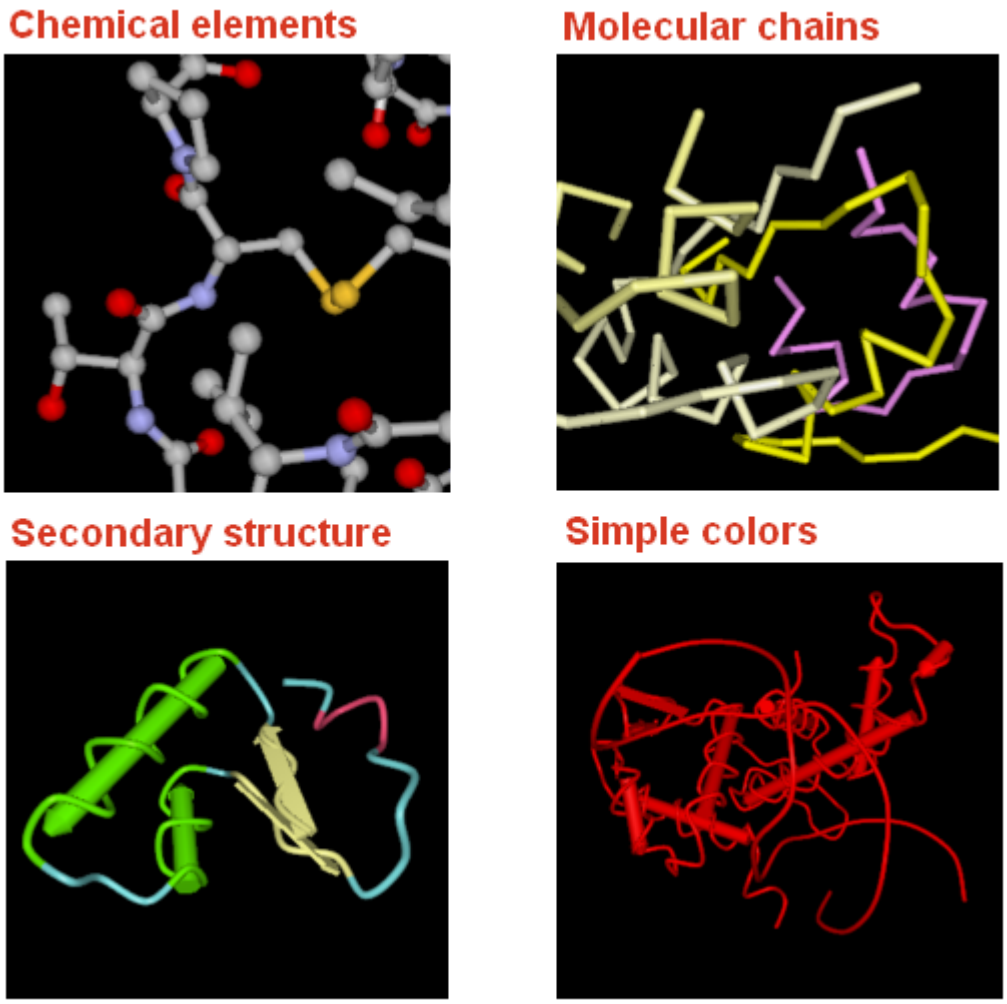


Selecting Coloring Scheme

You can select one of the following coloring schemes:

- *Chemical Elements*
- *Molecular Chains*
- *Secondary Structure*
- *Simple colors*

To change the coloring scheme open the *Coloring Scheme* menu (available in the context menu and in the *Display* menu on the toolbar).



Calculating Molecular Surface

To calculate the molecular surface of a molecule select the *Molecular Surface* item in the 3D Structure Viewer context menu or in the *Display* menu on the toolbar and check one of the following items:

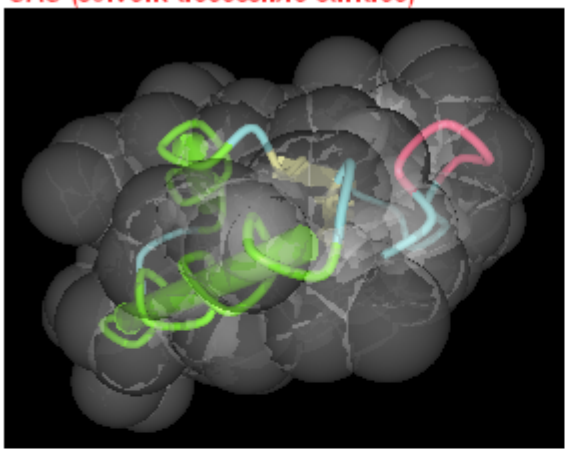
- *SAS* (solvent-accessible surface)
- *SES* (solvent-excluded surface)
- *vdWS* (van der Waals surface)

To remove the molecular surface that has already been calculated select the *Off* item.

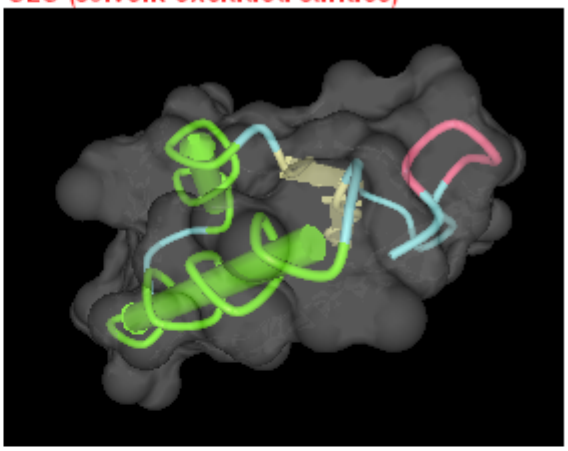
You can also select the *Molecular Surface Render Style* to modify the calculated molecular surface appearance:

- *Convex Map*
- *Dots*

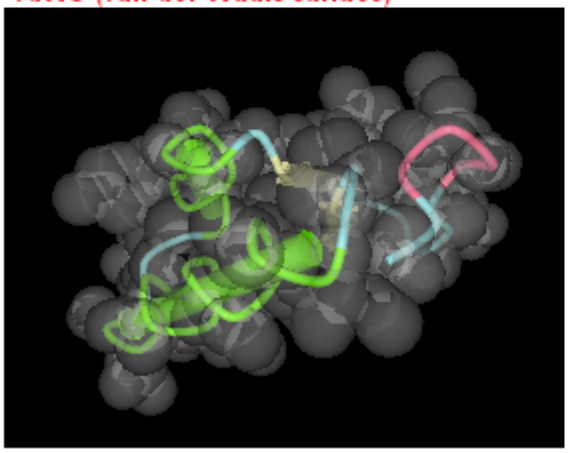
SAS (solvent-accessible surface)



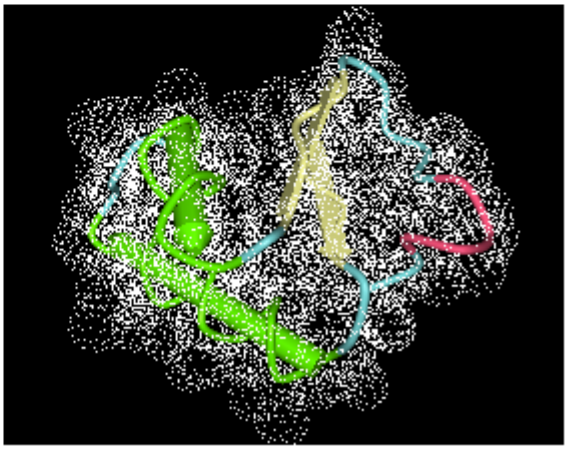
SES (solvent-excluded surface)



vdWS (van der Waals surface)



vdWS with dots



Selecting Background Color

To change the background color open the *Settings* dialog (choose the *Settings* item in the 3D Structure Viewer context menu or in the *Display* menu on the toolbar), press the *Set background color* button and select a color in the dialog appeared.

Selecting Detail Level

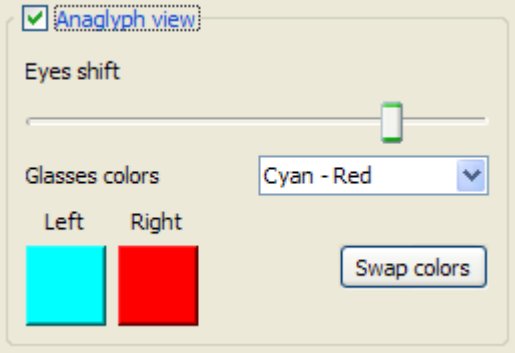
To select the detail level of a 3D Structure representation open the *Settings* dialog of the 3D Structure Viewer and drag the *Detail level* slider.

Enabling Anaglyph View

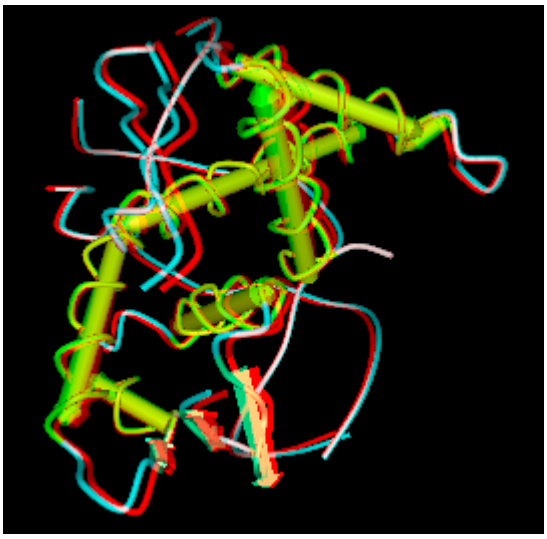
UGENE allows you to view a molecule in the anaglyph mode. To enable the anaglyph view open the *Settings* dialog of the 3D Structure Viewer and check the *Anaglyph view* check box.

You can modify the color settings: select one of the available *Glasses colors* or set custom colors, swap the colors.

The offset of the color layers can be adjusted by dragging the *Eyes shift* slider.



See the result the anaglyph view is applied to a molecule below:



Moving, Zooming and Spinning 3D Structure

A 3D structure can be easily spinned, moved and resized:

- To spin the 3D structure drag the mouse on the 3D structure while holding the left mouse button.
- To move the 3D structure hold the Ctrl keyboard button and drag the mouse with the left button pressed.
- To resize the 3D structure either use the mouse wheel or *Zoom In* and *Zoom Out* buttons on the toolbar.

At any time you can restore the default view by pressing the *Restore Default View* button on the toolbar.



You can also overview the whole structure by spinning it automatically. Select the *Spin* item either in the 3D Structure Viewer context menu or in the *Display* menu on the toolbar to do it.

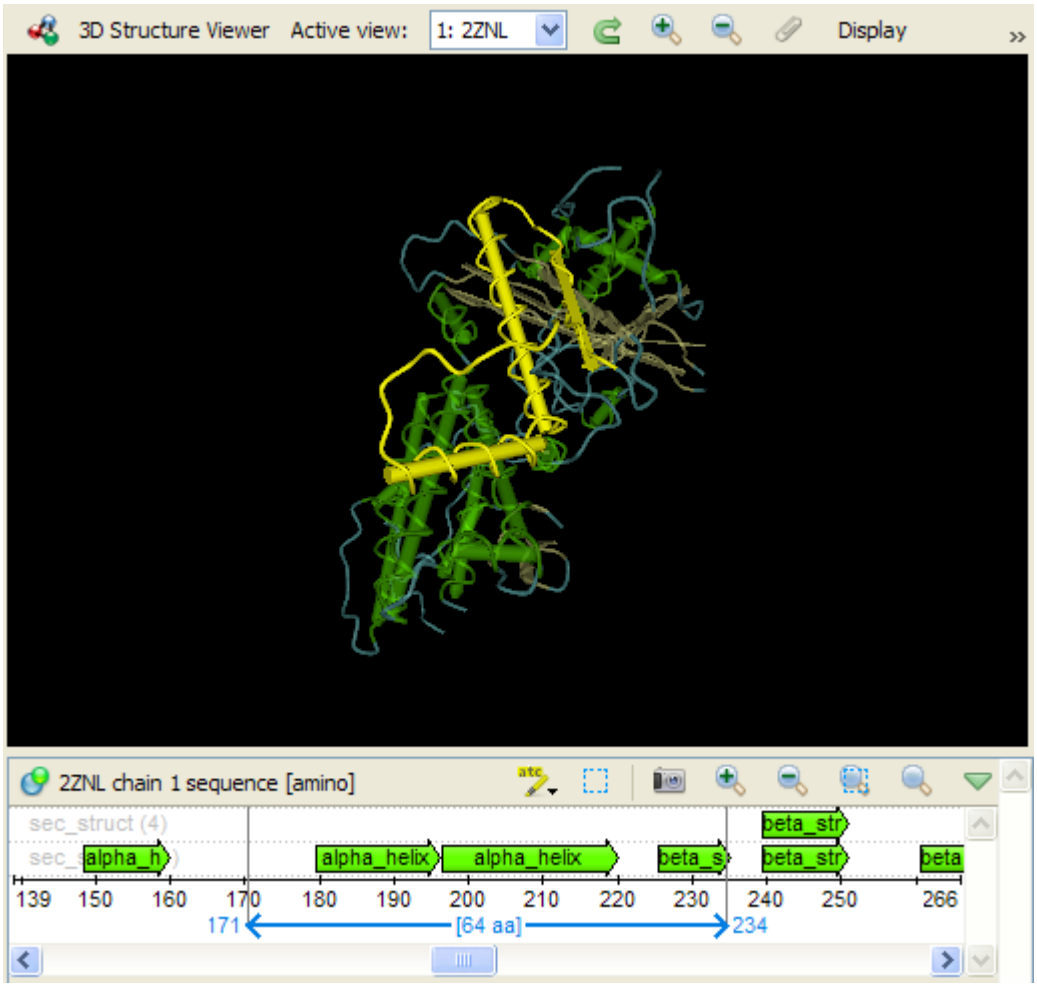
To stop the spinning uncheck the *Spin* item.

Highlight Region on 3D Structure

When you *are selecting a region of a sequence* e.g in the *Sequence zoom view* the corresponding region on the 3D structure is being highlighted while the rest regions of the 3D structure are being shaded.

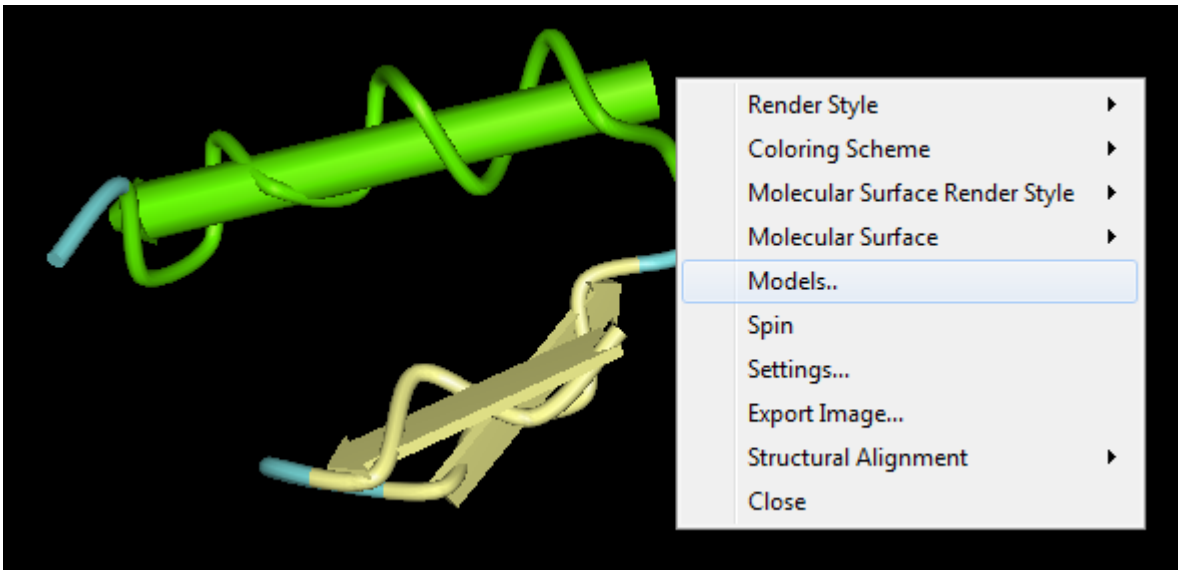
To configure the color of a region selected open the *Settings* dialog (press the *Settings* item in the 3D Structure Viewer context menu or in the *Display* menu on the toolbar to do it), press the *Set selection color* button and select a color in the dialog appeared.

To adjust the shading drag the *Unselected regions shading* slider in the *Settings* dialog.

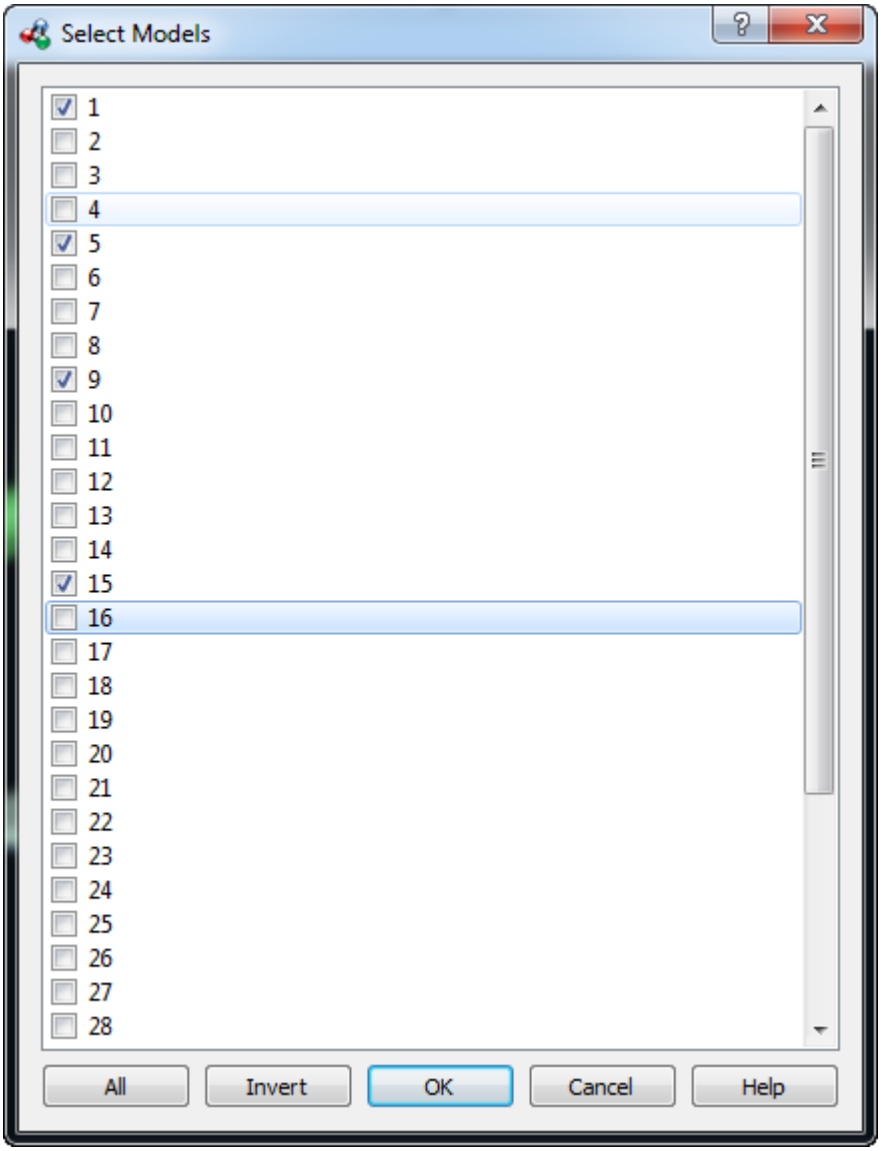


Selecting Models to Display

When a molecular structure contains multiple models (e.g. NMR ensembles of models), the *Models* item appears in the 3D Structure Viewer context menu and in the *Display* menu on the toolbar. For example, for protein with PDB ID 1ZNF the menu looks as follows:



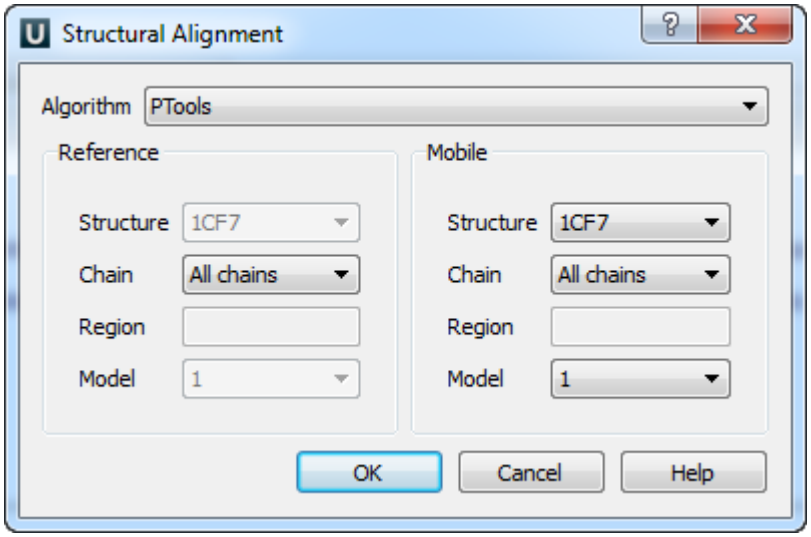
The dialog will appear:



To show all the models check the *All* item. To show only one model check the item and click the *OK* button. To show several models select it and click *OK* button. To show the inverted selection click the *Invert* button and click *OK* button.

Structural Alignment

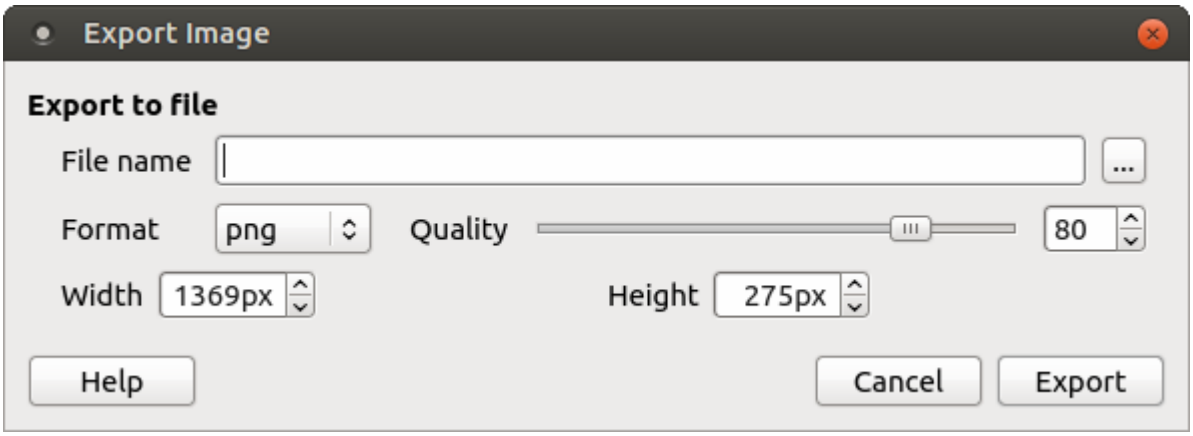
To use the structural alignment call the *Structural alignment->Align with* context menu item. The following dialog will appear:



Here you can change reference and mobile settings. After that click on the *OK* button. To reset structural alignment call the *Structural alignment->Reset* context menu item.

Exporting 3D Structure Image

To export a 3D structure image select the *Export Image* item in the 3D Structure Viewer context menu or in the *Display* menu on the toolbar. The *Export Image* dialog will appear:

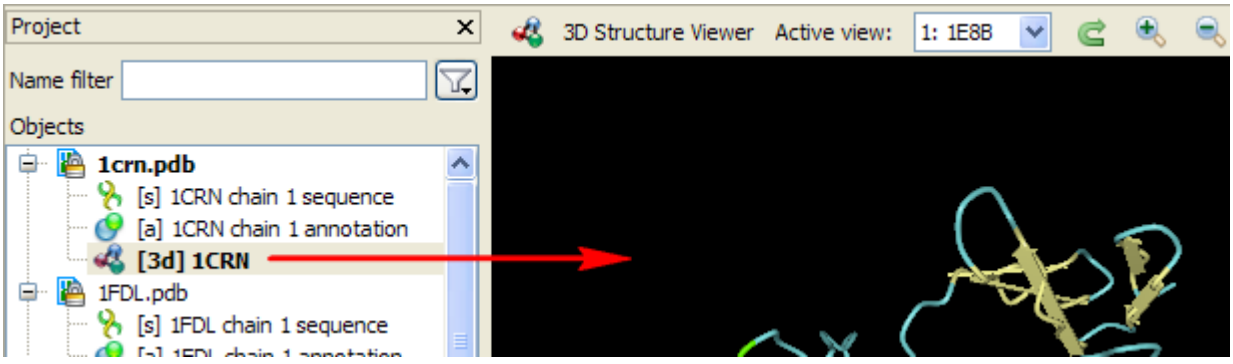


Here you can browse for the file name, select the width and height of the image as well as its format: svg, png, ps, jpg, jpeg, tiff, tif, pdf, bmp or ppm. For jpg, jpeg formats the quality score parameter is available.

Working with Several 3D Structures Views

To add another view to the 3D Structure Viewer you can:

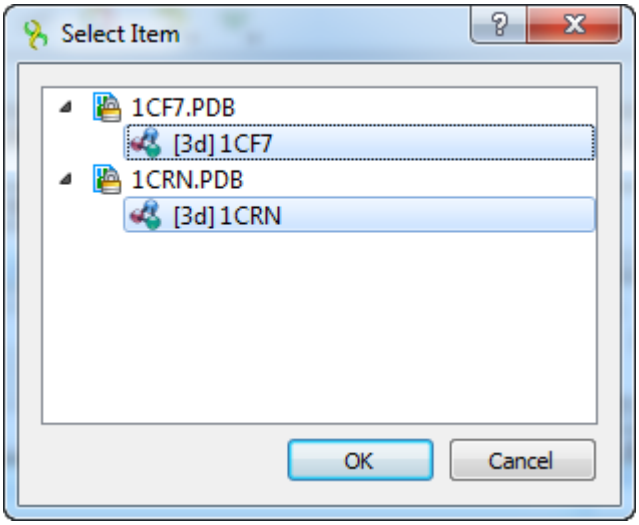
- Drag a required [3d] object from the *Project View* to the 3D Structure Viewer.



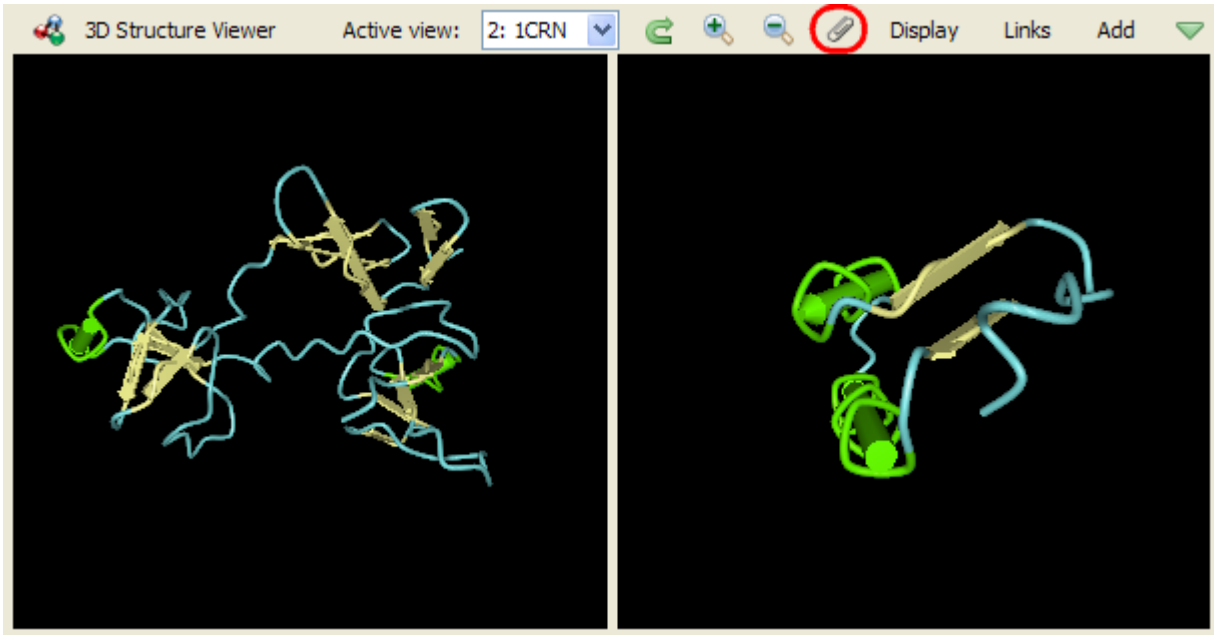
- Press the *Add* button on the toolbar. The *Select Item* dialog will appear. Select [3d] objects to add.

  Hint

  Use the Ctrl keyboard button to select several objects.



Below you can see the 3D Structure Viewer with two views:

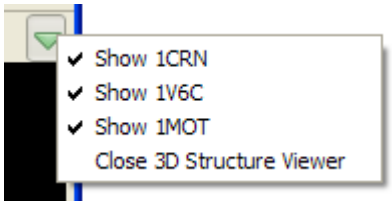


To select an active view click on the view area or select an appropriate value in the *Active view* combo box on the toolbar.

To synchronize the views press the *Synchronize 3D Structure Views* sticky button on the toolbar (see the image above). When the button has been pressed the 3D structures are *moved, zoomed and spinned* synchronously. Press the button again to stop the views synchronization.

The views that are no more required can be closed by selecting the *Close* button in the 3D Structure Viewer context menu.

Also you can hide/show views for a while. Use the menu of the green arrow button on the toolbar to do it:



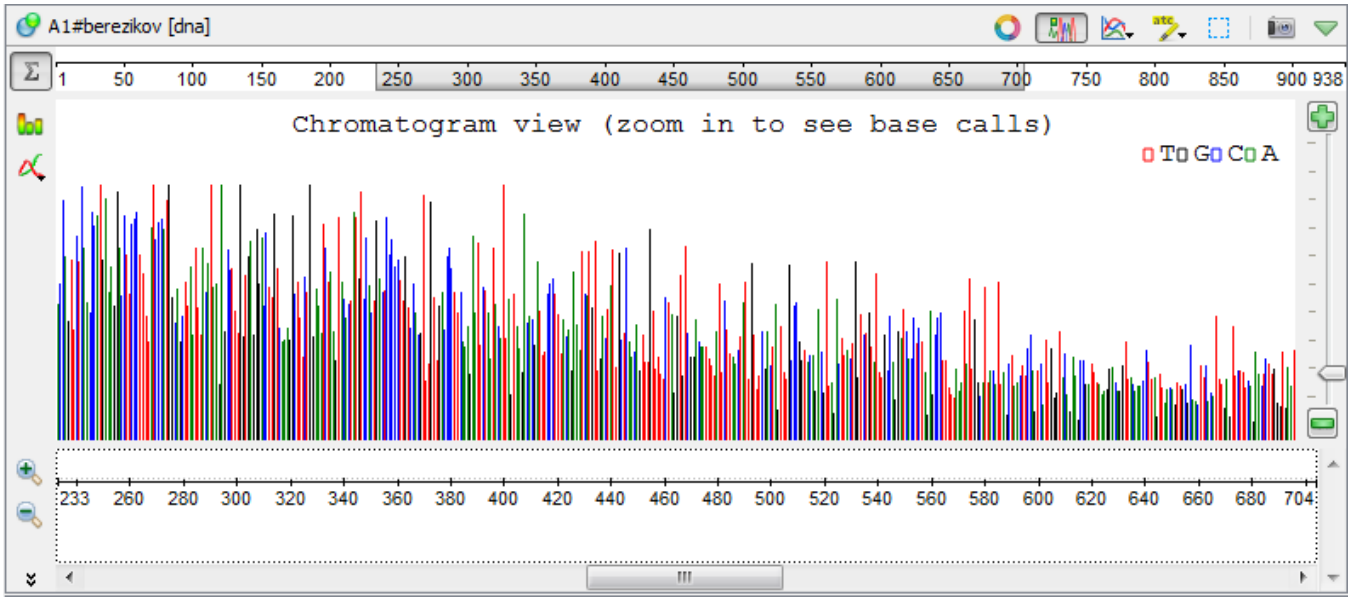
Notice that the 3D Structure Viewer can be closed from this menu.

Chromatogram Viewer

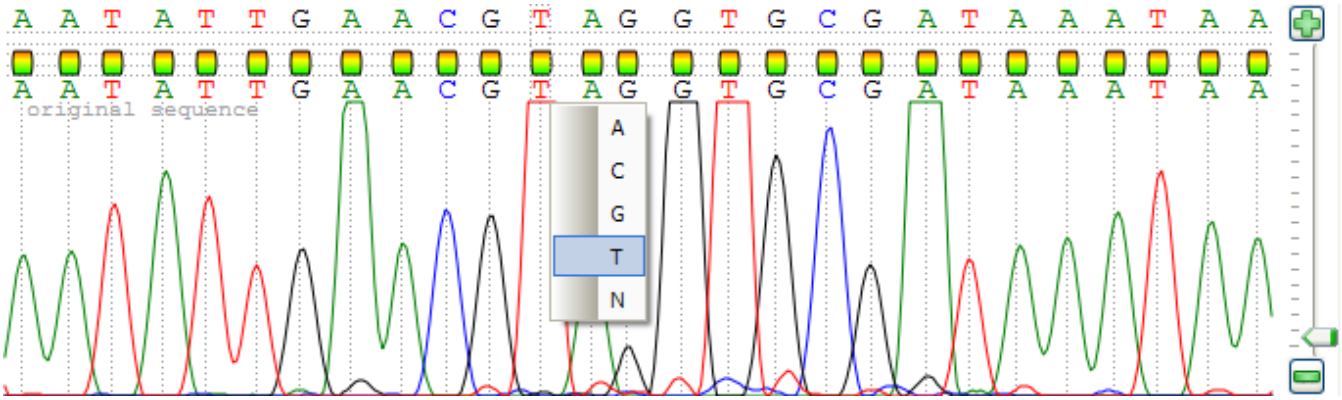
The *Chromatogram Viewer* plugin brings DNA chromatogram data viewing and editing capabilities into UGENE.

Currently supported chromatogram file formats are ABIF and SCF.

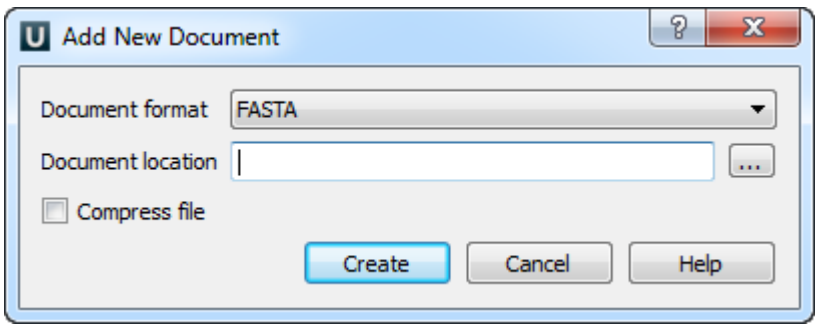
To view a chromatogram, just open an interesting file in UGENE by standard means (e.g. drag&drop the file or press the Ctrl-O shortcut). The *Chromatogram Viewer* is automatically embedded into the generic *Sequence View* if chromatogram data are found, as on the screenshot below:



After zooming in, more chromatogram details are available:



To edit a sequence data, right-click on the chromatogram view and select the *Edit new sequence* item in the appeared context menu. The following dialog will appear:

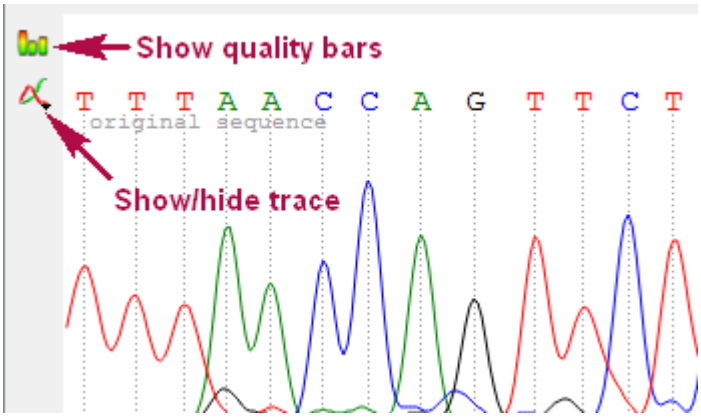


Select new document format and location and click on the *Create* button.

The original DNA sequence is not allowed to be changed; however you can add and modify a new sequence stored in a separate file.

The sequence being edited is displayed right above the original one. Symbols can be changed by clicking on interesting value, modifications are shown in bold.

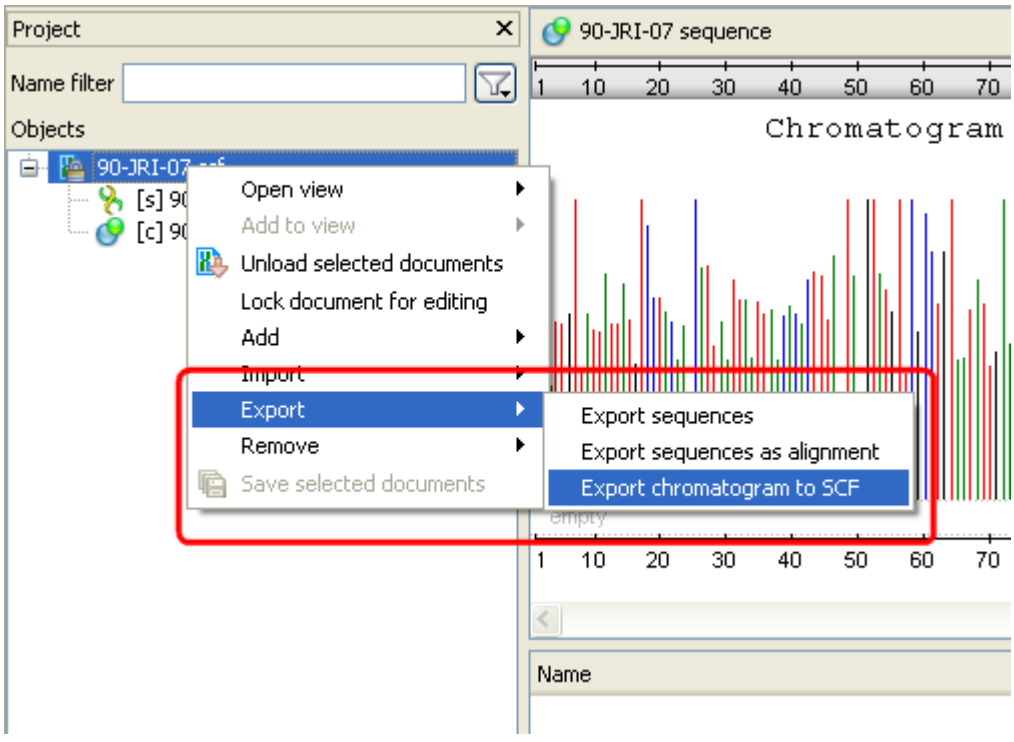
Also you can show/hide different signals of chromatogram and quality bars with a help of the *Show/hide trace* and *Show quality bars* toolbar buttons correspondingly:



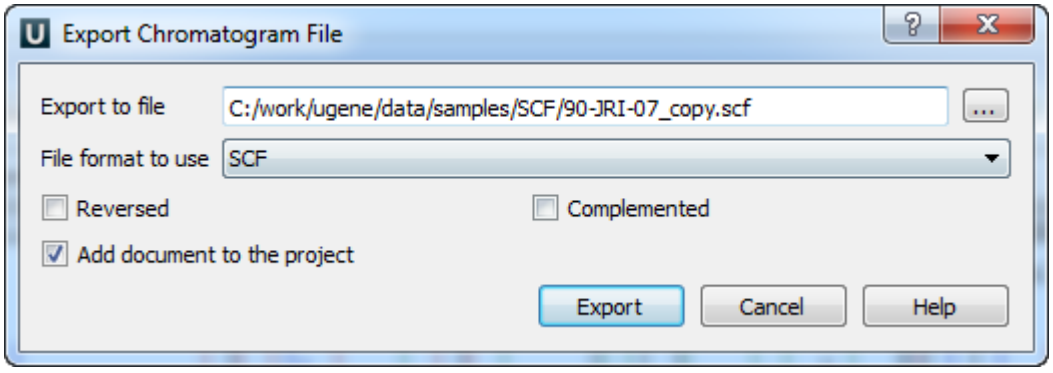
- Exporting Chromatogram Data
- Viewing Two Chromatograms Simultaneously

Exporting Chromatogram Data

Open, for example, the $UGENE/data/samples/SCF/90-JRI-07.srf file. In the *Project View* context menu there is *Export chromatogram to SCF* item:



After clicking on the item, the *Export chromatogram file* dialog will appear:

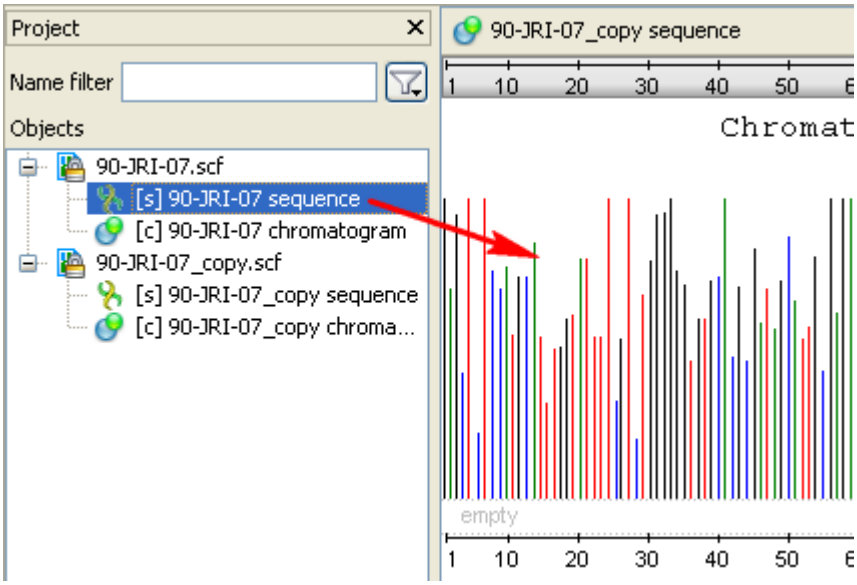


Check the *Reversed* and *Complemented* options if you want to create a reverse and complement chromatogram. Press the *Export* button.

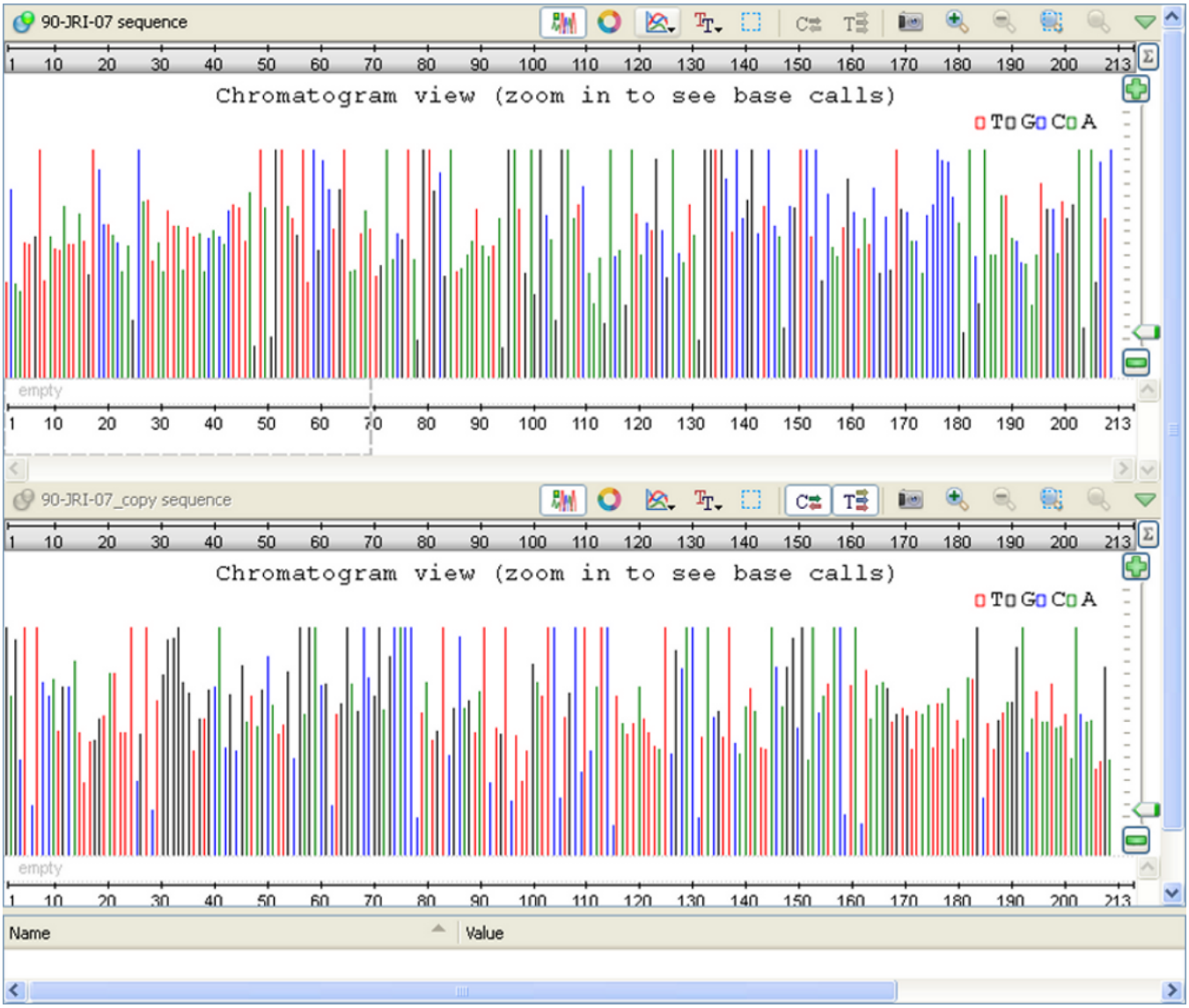
The exported file will be opened in the *Sequence View*.

Viewing Two Chromatograms Simultaneously

To add another sequence to the *Sequence View*, drag the required sequence object from the *Project View* and drop it in the *Sequence View* area. (Note that the dragged object is the sequence object, not the chromatogram object.)

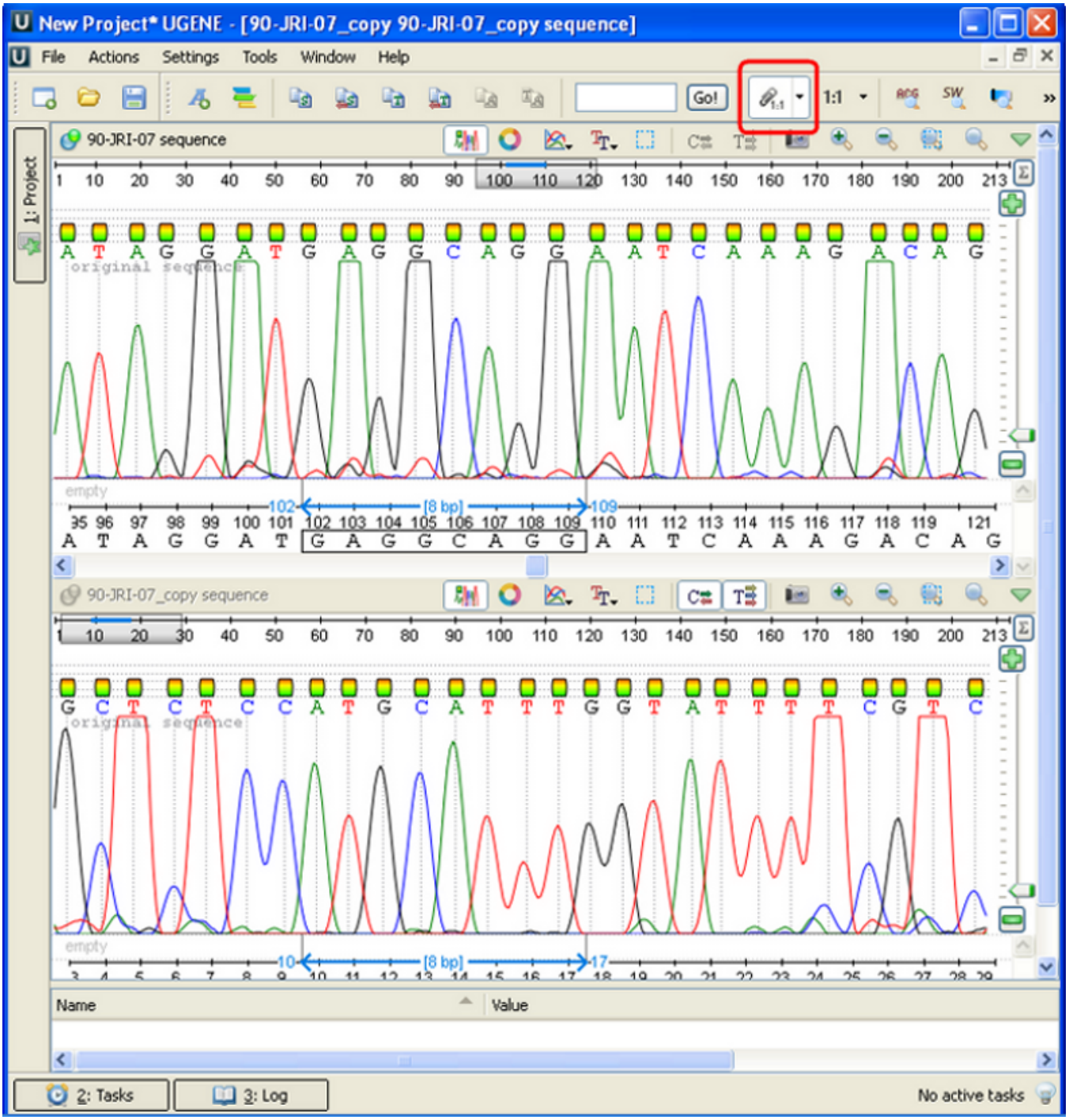


The result will look like this:



You can also use the *Lock scales* and *Adjust scales* global actions for the chromatograms.

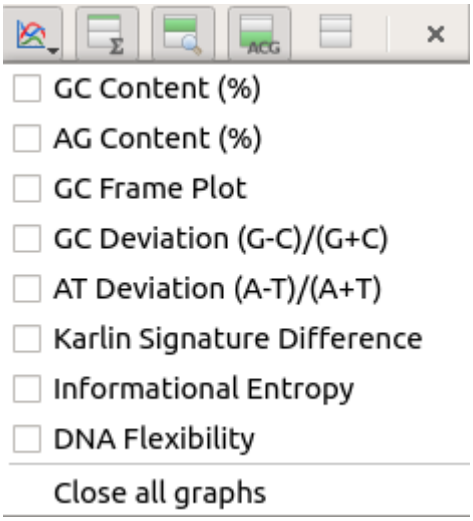
For example if you lock the scales you are able to scroll the sequences simultaneously. Also when you select a sequence region in one sequence, the same region is selected in the second sequence.



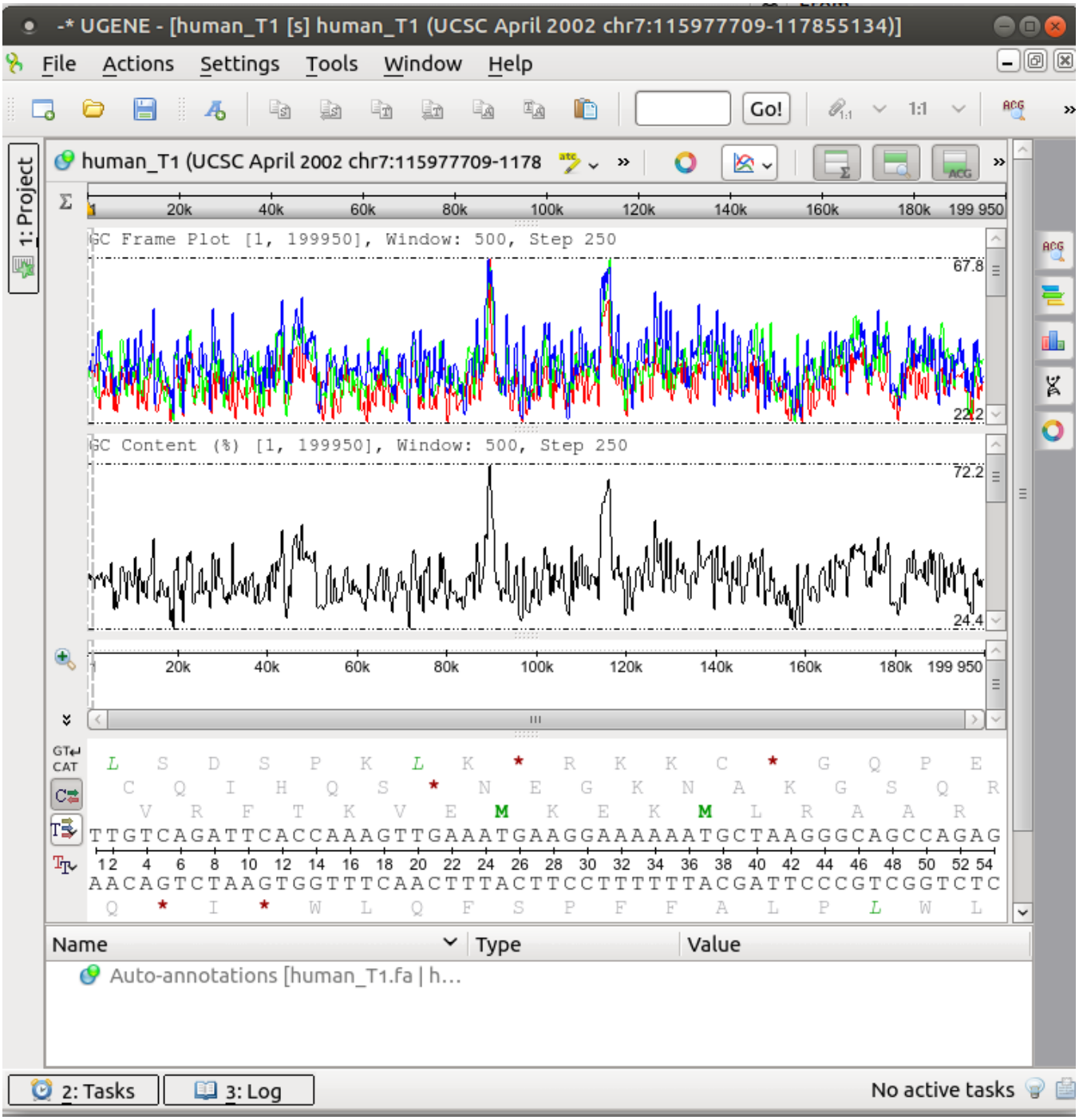
Graphs Package

The *DNA/RNA Graphs Package* draws contextual graphs for sequences. The *DNA/RNA Graphs Package* is available for the Standard DNA and Standard RNA alphabets.

Open a sequence in the *Sequence View* and click the *Graphs* icon on the toolbar. The popup menu appears:

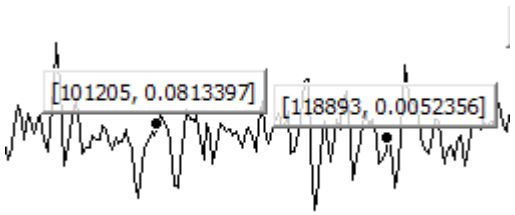


To see a graph select the corresponding graph item in the popup menu. A new area with the graph appears right above the *Sequence zoom view*:



Each point on a graph is calculated for a window of a specified size. The window is moved along the sequence by a step. See *Graph Settings* for instructions on how to modify these parameters.

It is possible to get information about each point of a graph. When a mouse is moved in the *Graphs* area, a small circle shows on the graph. A coordinates hint shows above it. When you hold Shift and click on a graph, the circle and the hint locks:



To remove it click on the hint. Also, you can delete all labels by *Graph->Delete all labels* context menu. To select all extremum points use the *Graph->Select all extremum points* context menu item.

All graphs are always aligned with the range shown in the *Sequence zoom view*. It means that if you change the visible range in the overview (either by zooming or scrolling) the graph will also be updated. The minimum and maximum values of the visible range are shown at the right lower and upper corners of the graph.

To close a graph, uncheck its item in the popup menu.

- Description of Graphs
- Graph Settings
- Saving Graph Cuttoffs as Annotations

Description of Graphs

Find below the detailed description of each graph. Note that characters A, C, G and T in the formulas denote the number of corresponding nucleotide in a window.

- *DNA Flexibility* — searches for regions of high DNA helix flexibility in a DNA sequence. The average *Threshold* in a window is calculated by the following formula:

```
(sum of flexibility angles in the window) / (the window size - 1)
```

For more detailed information see *DNA Flexibility* paragraph.

- *GC Content (%)* — shows the percentage of nitrogenous bases (either guanine or cytosine) on a DNA molecule. It is calculated by the following formula:

```
(G+C)/(A+G+C+T)*100
```

- *AG Content (%)* — shows the percentage of nitrogenous bases (either adenine or guanine) on a DNA molecule. It is calculated by the following formula:

```
(A+G)/(A+G+C+T)*100
```

- *GC Frame Plot* — this graph is similar to the GC content graph but shows the GC content of the first, second and third position independently. It is most effective in organisms with GC rich genomic sequence but it also works on all microbial sequences.
- *GC Deviation (G-C)/(G+C)* — shows the difference between the “G” content of the forward strand and the reverse strand. *GC Deviation* is calculated by the following formula:

```
(G-C)/(G+C)
```

- *AT Deviation (A-T)/(A+T)* — shows the difference between the “A” content of the forward strand and the reverse strand. *AT Deviation* is calculated by the following formula:

```
(A-T)/(A+T)
```

- *Karlin Signature Difference* — dinucleotide absolute relative abundance difference between the whole sequence and a sliding window. Let:

```
f(XY) = frequency of the dinucleotide XY
f(X)  = frequency of the nucleotide X

p(XY) = f(XY) / f(X) * f(Y)

p_seq(XY) = p(XY) for the whole sequence
p_win(XY) = p(XY) for a window
```

The *Karlin Signature Difference* for a window is calculated by the following formula:

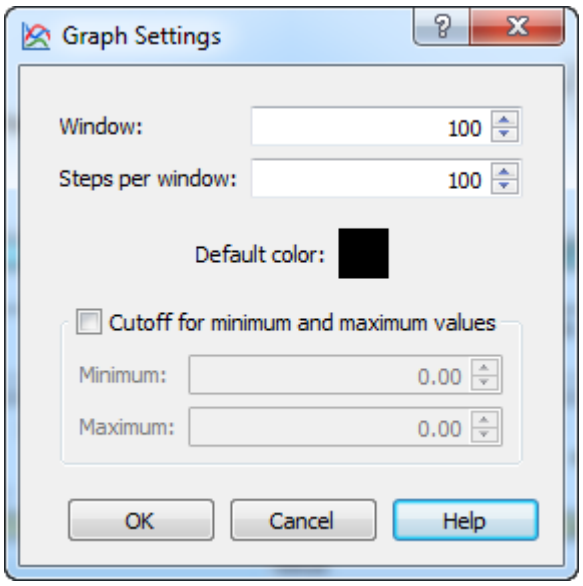
```
sum(p_seq(XY) - p_win(XY)) / 16
```

- *Informational Entropy* — is calculated from a table of overlapping DNA triplet frequencies. The use of overlapping triplets smooths the frame effect. *Informational Entropy* is calculated by the following formula:

```
-(triplet frequency)*log10(triplet frequency)/log10(2)
```

Graph Settings

To change settings of a graph, select the *Graph->Graph settings* item in the graph context menu. The *Graph Settings* dialog appears:



The following parameters are available:

Window — the number of bases in a window.

Steps per window — the number of steps in window. The *Step* is calculated as *Window* / *Steps per window*.

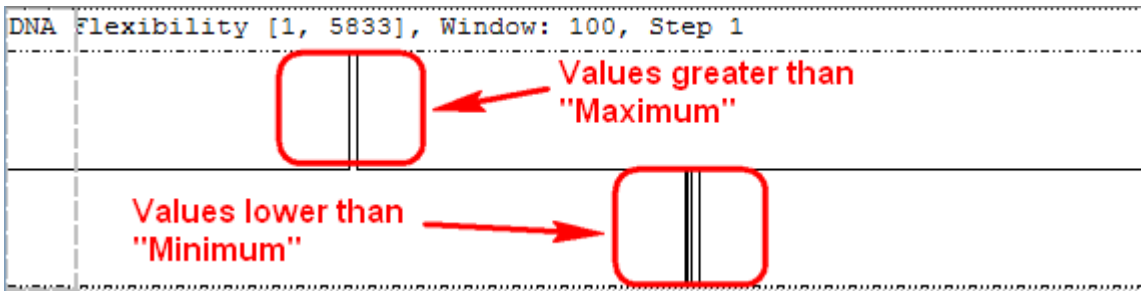
Default color — the default color of line of graph (or lines of graphs for *GC Frame Plot*).

Checking of the *Cutoff for minimum and maximum values* checkbox enables the following settings:

Minimum — the minimum value for cutoff.

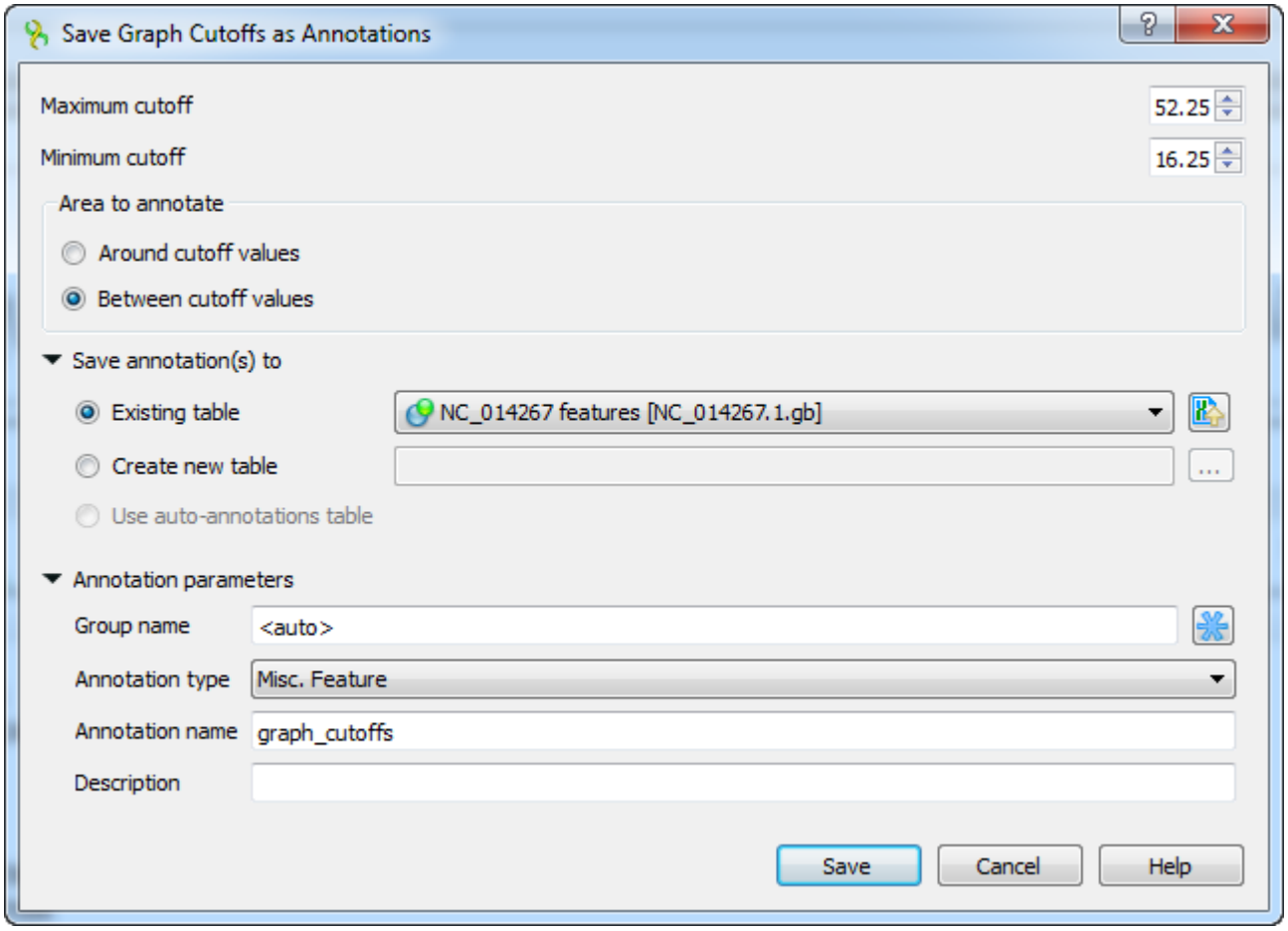
Maximum — the maximum value for cutoff.

Select an appropriate minimum and maximum value and click the *OK* button to show the graph of cutoffs. The graph is divided into 2 parts. The upper part shows values greater than the specified *Maximum* value. The lower part of the graph shows values lower than the specified *Minimum* value. For example:



Saving Graph Cuttoffs as Annotations

To save graph cuttoffs as annotations select the *Graph->Save cuttoffs as annotations* item in the graph contex menu. The following dialog will appear:



The following parameters are available:

Maximum cutoff - maximum cutoff value.

Minimum cutoff - minimum cutoff value.

Around cutoff values - saves the values around cutoffs values.

Between cutoff values - saves the values between cutoffs values.

In the *Save annotation(s) to* group you can set up a file to store annotations. It could be either an existing annotation table object, a new annotation table or auto-annotations table (if it is available).

In the *Annotation parameters* group you can specify the name of the group and the name of the annotation. If the group name is set to <auto> UGENE will use the group name as the name for the group. You can use the '/' characters in this field as a group name separator to create subgroups. If the annotation name is set to *by type* UGENE will use the annotation type from the *Annotation type:* table as the name for the annotation. Also you can add a description in the corresponding text field.

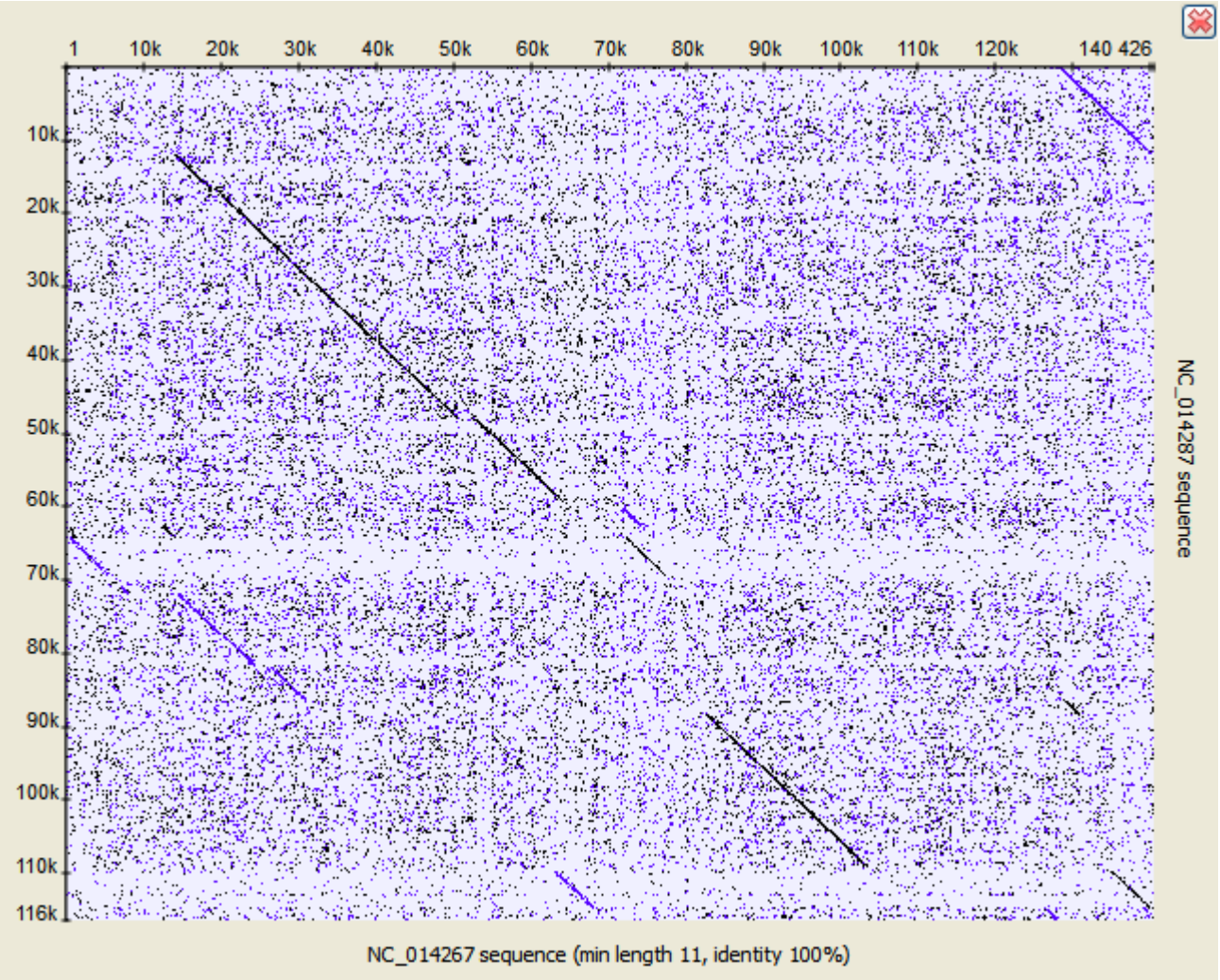
Select the parameters and click on the *Save* button. The corresponding annotations will be saved.

Dotplot

The *Dotplot* plugin provides a tool to build dotplots for DNA or RNA sequences. This allows comparing these sequences graphically. Using a dotplot, you can easily identify such differences between sequences as mutations, inversions, insertions, deletions and low-complexity regions.

Also the plugin provides advanced features: comparing multiple dotplots, navigation in a dotplot, dotplots synchronization, saving and loading a dotplot, etc.

An example of a dotplot view:



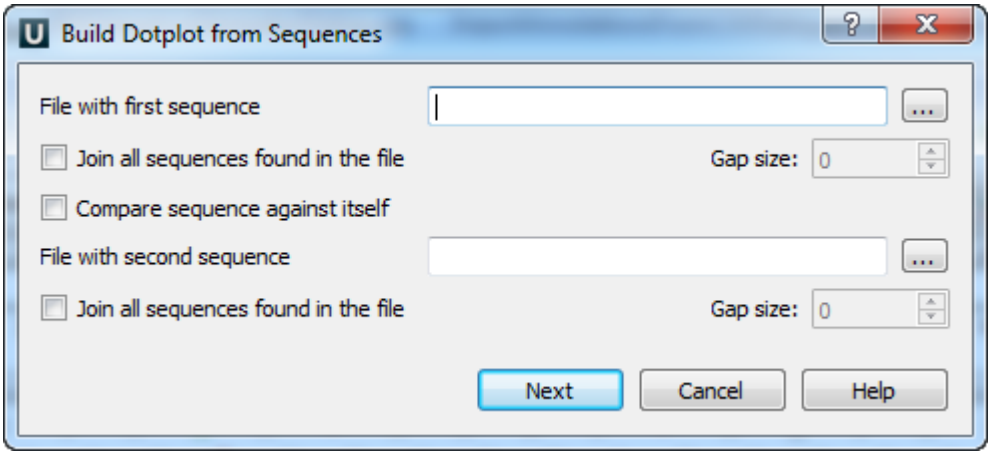
The *Dotplot* plugin uses the *Repeat Finder* plugin to build a dotplot, make sure you have the *Repeat Finder* plugin installed.

The *Dotplot* features are described in more details below.

- Creating Dotplot
- Navigating in Dotplot
- Zooming to Selected Region
- Selecting Repeat
- Interpreting Dotplot: Identifying Matches, Mutations, Invertions, etc.
- Editing Parameters
- Filtering Results
- Saving Dotplot as Image
- Saving and Loading Dotplot
- Building Dotplot for Currently Opened Sequence
- Comparing Several Dotplots

Creating Dotplot

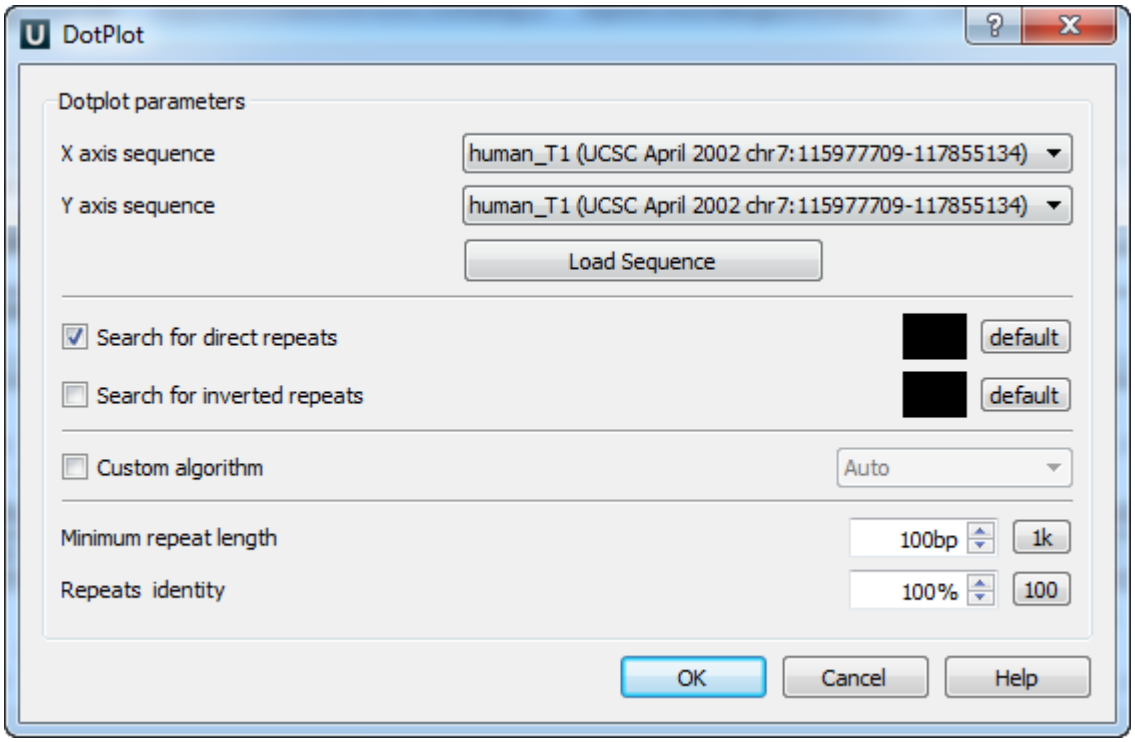
To create a dotplot select the *Tools Build dotplot* main menu item. The *Build dotplot from sequences* dialog will appear:



Here you should specify the *File with first sequence*. Also you should either check the *Compare sequence against itself* option or select the *File with second sequence*.

Optionally you can select to *Join all sequences found in the file* (for the first and/or for the second file). If you select to join the sequences you can also select the *Gap size*. The gap of the specified size will be inserted between the joined sequences.

After you press the *Next* button, the dialog to configure the dotplot parameters will appear:



The following parameters are available:

X axis sequence — the sequence for the X dotplot axis.

Y axis sequence — the sequence for the Y dotplot axis.

If there are several sequences in the specified (the first or the second) file and you haven't selected to join the sequences in the previous dialog, then you can select a sequence in these fields.

If you have selected to *Join all sequences found in the file*, then you can't select a separate sequence from the file, the joined *Sequence* can be selected instead.

Search direct repeats — check this option to search for direct repeats in the specified sequences. You can also select the color with which the repeats will be displayed in the picture. The *default* button sets the default color.

Search inverted repeats — check this option to search for inverted repeats in the specified sequences. You can also select the color with which the repeats will be displayed in the picture. The *default* button sets the default color.

Custom algorithm — optionally you can select an algorithm to calculate the repeats:

- Auto
- Suffix index
- Diagonals

The specified algorithm is provided to the *Repeat Finder* plugin as an input parameter. In most cases the *Auto* value is appropriate.

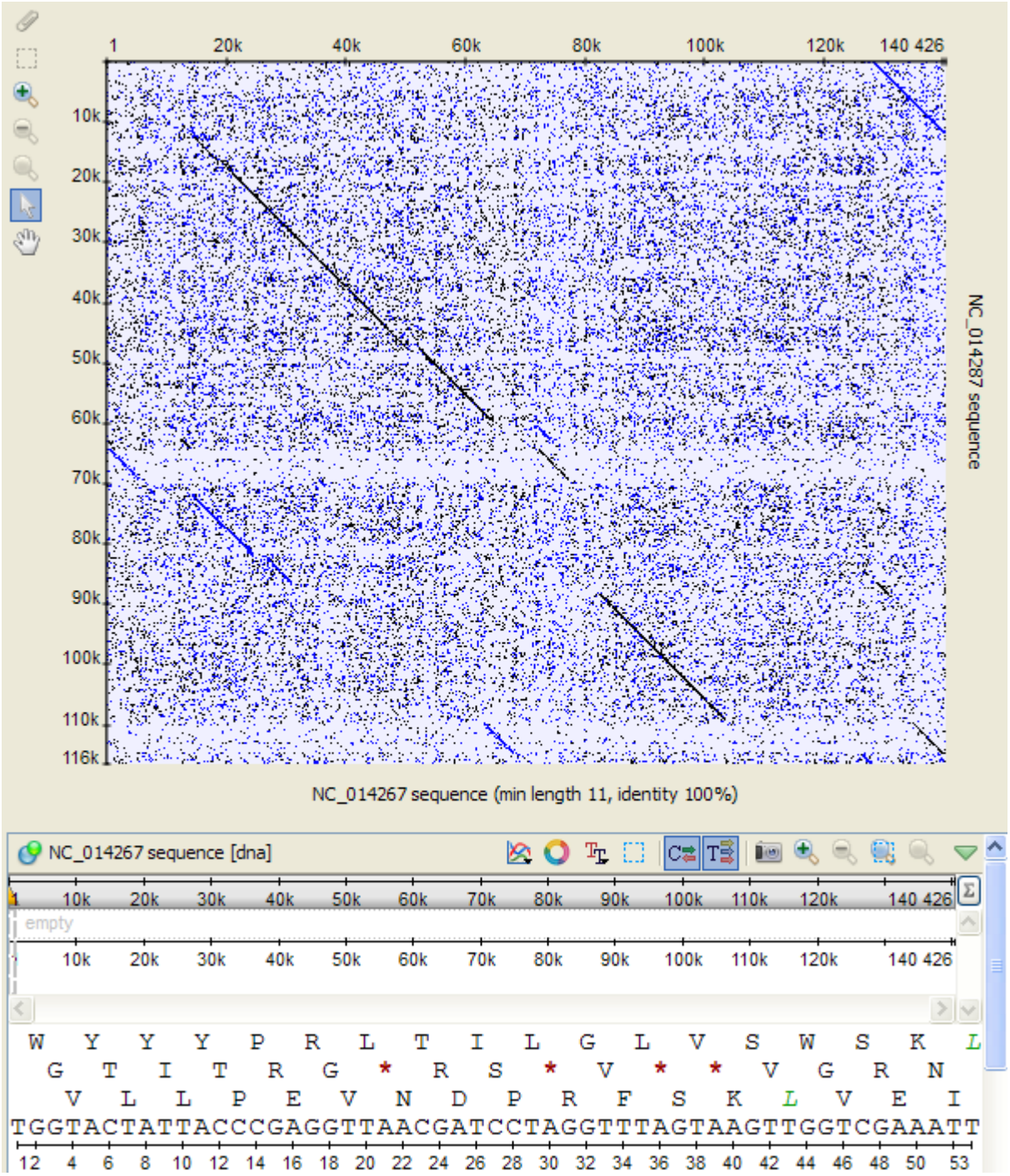
Minimum repeat length — allows to draw only such matches between the sequences that are continuous and long enough. For example if it equals to *3bp*, then only repeats will be found that contain 3 and more base symbols.

Press the *1k* button to automatically adjust the *Minimum repeat length* value. Such value will be set, that there will be about 1000 repeats found.

Repeats identity — specifies the percents of the repeats identity.

Press the *100* button to set the *100%* identity.

After the parameters are set, press the *OK* button. The dotplot will appear in the *Sequence View*:



It is a two-dimentional plot consisted of dots.

Each dot on the plot corresponds to a matched base symbol at the “x” position of the horizontal sequence and the “y” position of the vertical sequence.

Visible diagonal lines indicate matches between sequences in the given particular region.

See also:

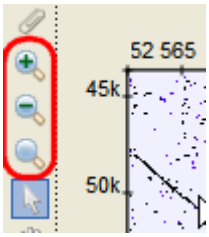
- *Interpreting Dotplot: Identifying Matches, Mutations, Invertions, etc.*
- *Building Dotplot for Currently Opened Sequence*

Navigating in Dotplot

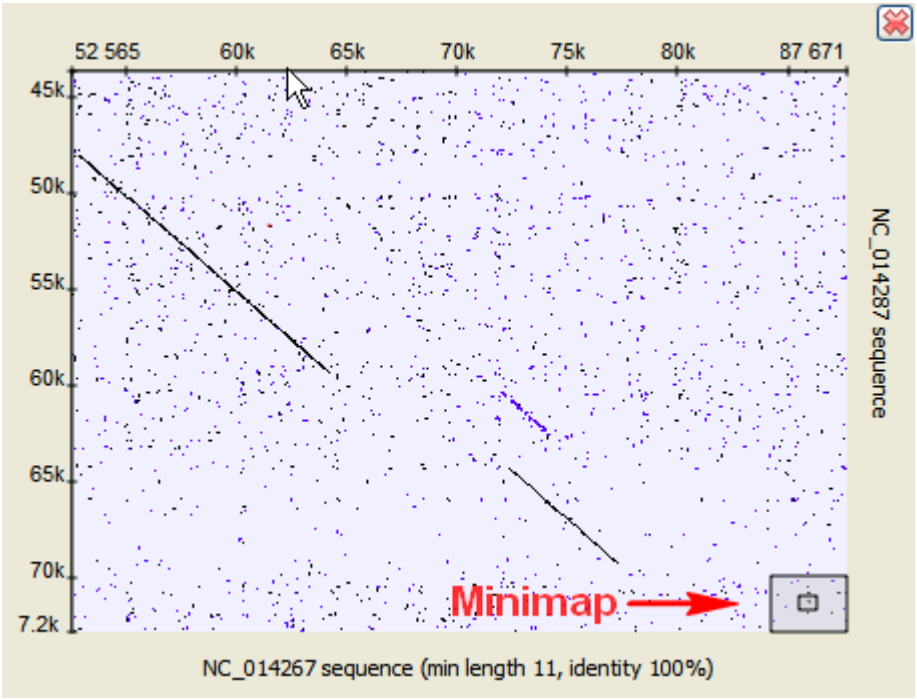
To zoom in / zoom out a dotplot you can:

- Rotate the mouse wheel.

- Press corresponding zoom buttons located on the left:



To move the zoomed region you can:



- Hold the middle mouse button and move the mouse cursor over the zoomed region of the doplot.
- Click on the desired region of the *minimap* in the right bottom corner.
- Activate the *Scroll tool*, hold the left mouse button and move the mouse cursor over the zoomed region:



Zooming to Selected Region

To select a dotplot region activate the *Select tool*:

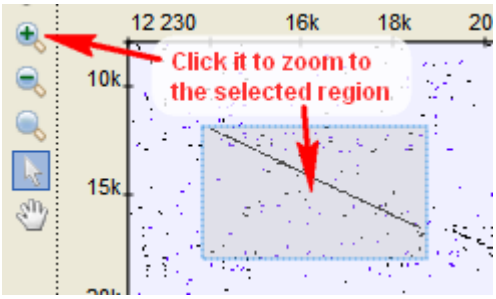


hold down the left mouse button and drag the mouse cursor over the dotplot.

When you select a region on a dotplot the corresponding region is also selected in other *Sequence View* areas (*Sequence details view*, *Sequence zoom view*, etc.).

The opposite is true as well: if you select a region in a *Sequence View* area, the corresponding region is also selected in the dotplot view.

To zoom to the region selected click the *Zoom in* on the left.

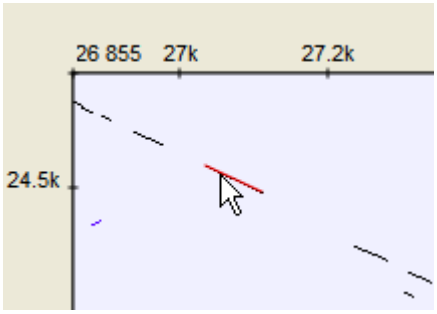


Selecting Repeat

To select a repeat activate the *Select tool*:



and click on the repeat:



To deselect the repeat either click on other repeat or hold Ctrl and click somewhere on the dotplot.

Interpreting Dotplot: Identifying Matches, Mutations, Invertions, etc.

Using a dotplot graphic, you can identify such the following differences between the sequences:

1. Matches

A *match* between sequences looks like a diagonal line on the dotplot graphic, representing the continuous match (or repeat).

2. Frame shifts

a. Mutations

Mutations are distinctions between sequences. On the graphic they are represented by gaps in diagonal lines. They interrupt matches.

b. Insertions

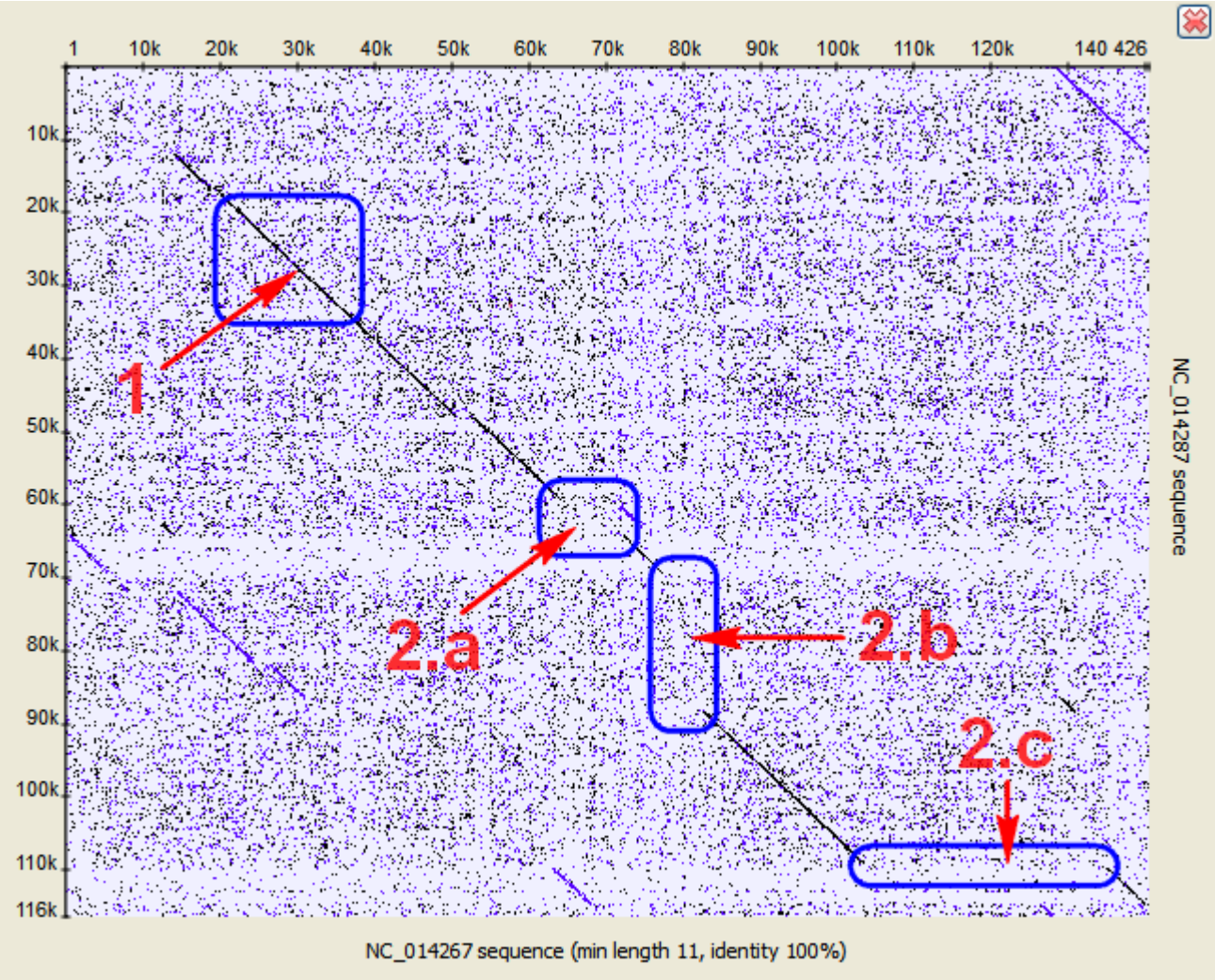
Insertions are parts of one sequence that are missed in the another, while the surrounding parts match. In other words, an insertion is a subsequence that was inserted into a sequence.

Graphically, insertions are represented by gaps which lie only on one axis. A little shift towards the other axis indicates a mutation involved.

c. Deletions

A deletion is a subsequence that was deleted from a sequence.

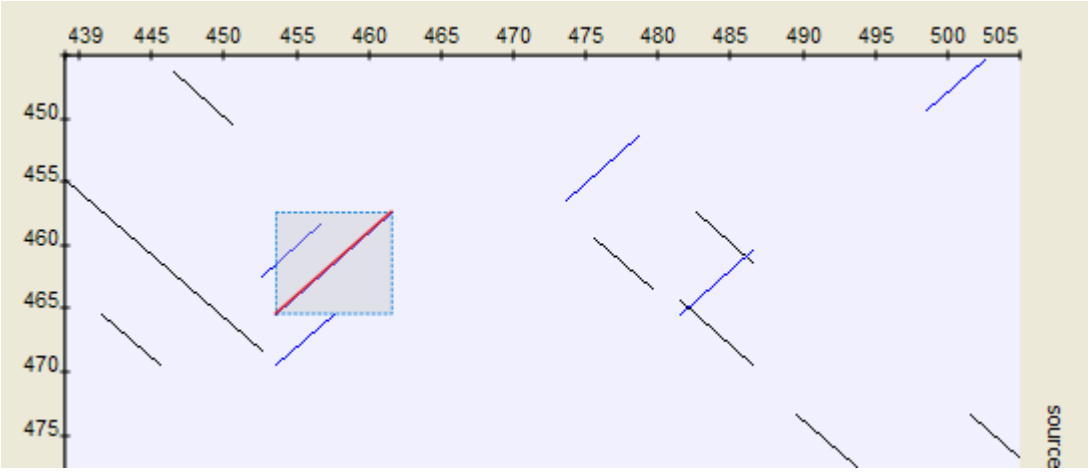
A deletion from sequence A found in sequence B can be considered as an insertion into sequence B and contained in sequence A.



3. Inverted repeats

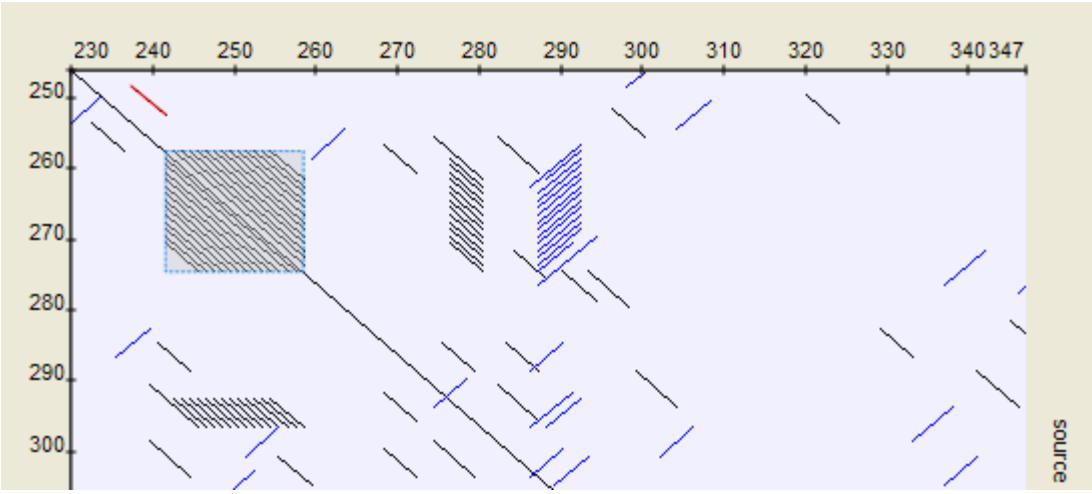
The *Dotplot* plugin allows to search for inverted repeats as well. Inverted repeats are shown contrary to the direct repeats.

Use the *Search direct repeats* and *Search inverted repeats* options of the *Dotplot* parameters dialog to select which repeats to draw (the dialog is described *here*).



4. Low-complexity regions

A low-complexity region is a region produced by redundancy in a particular part of the sequence. It is represented on a plot as a rectangular area filled with the matches.

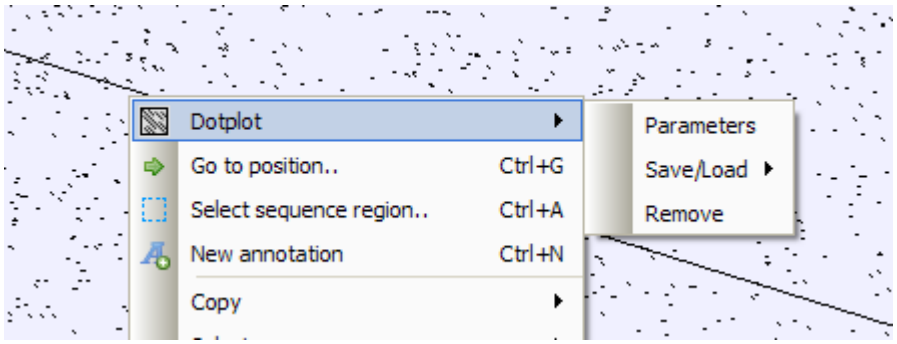


Hint

Compare sequence with itself to easily find low-complexity regions in it.

Editing Parameters

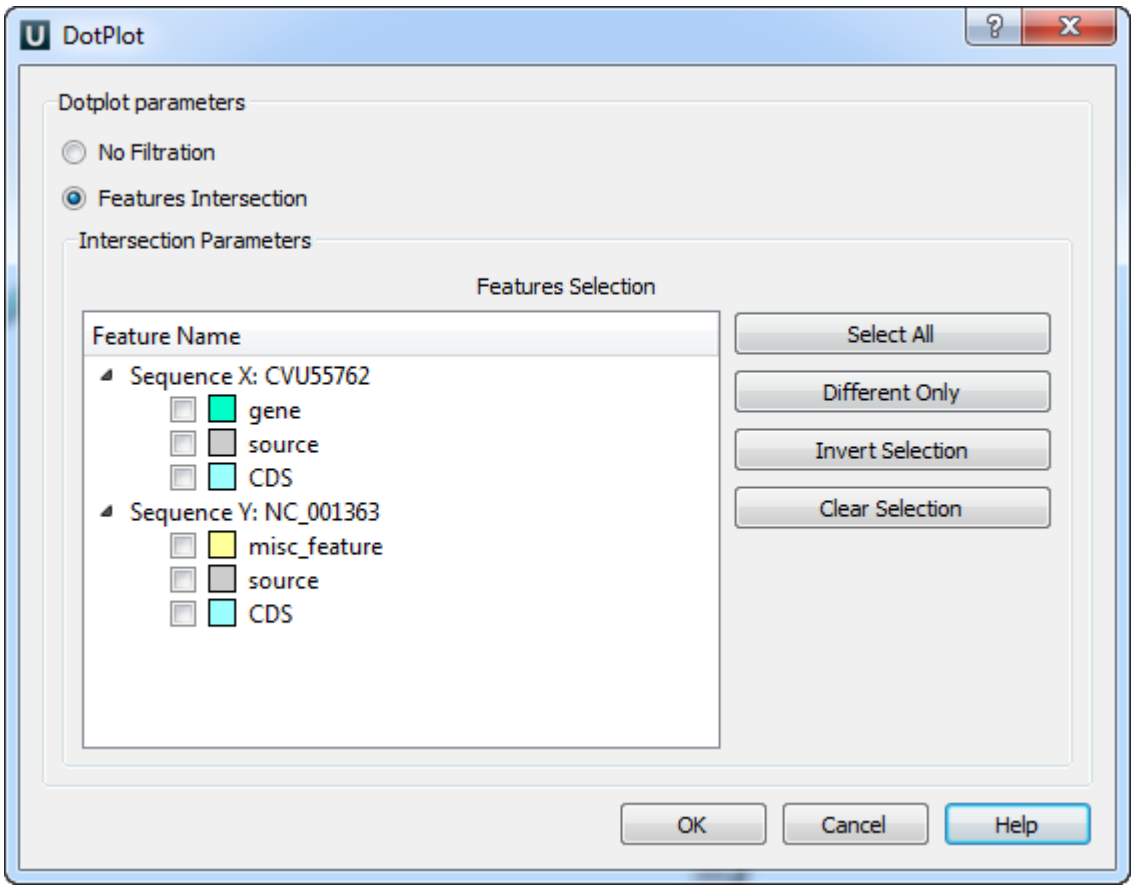
It is possible to edit parameters of a built dotplot. Right-click on the dotplot and select the *Dotplot Parameters* context menu item:



The parameters dialog will be re-opened. See description of the available parameters *here*.

Filtering Results

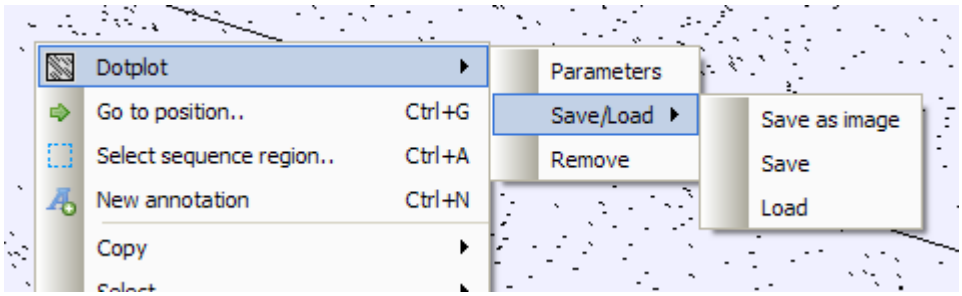
It is possible to find features intersections and filter dotplot results. Right-click on the dotplot and select the *Dotplot Filter results* context menu item. The following dialog will appear:



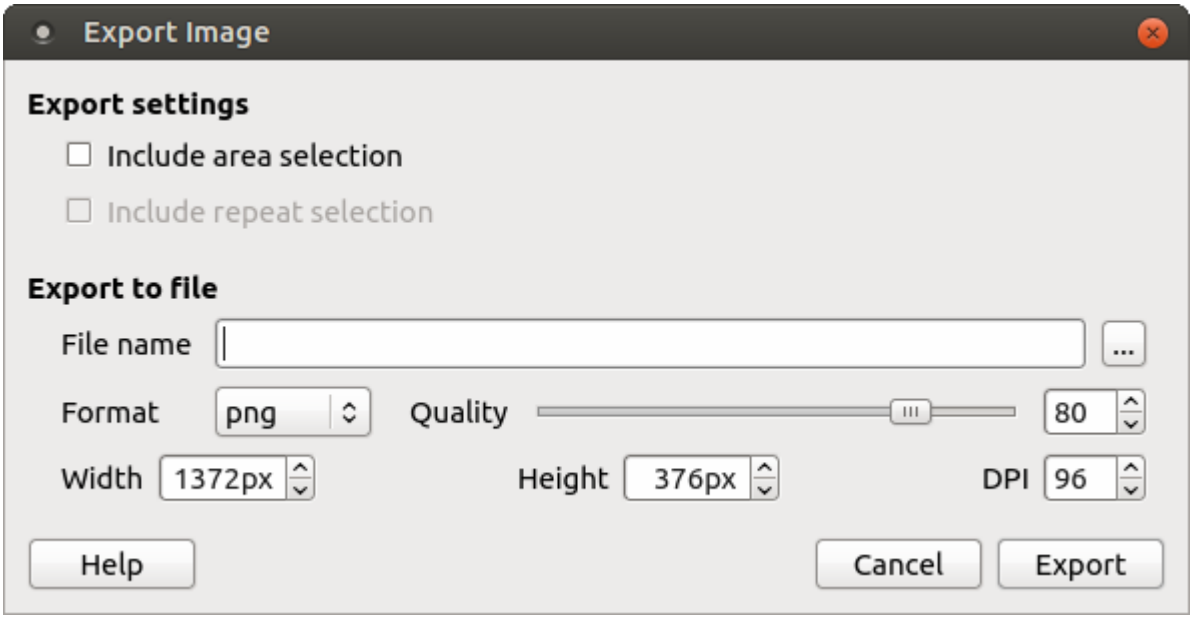
Select features and click OK button. The filtered dotplot will appear.

Saving Dotplot as Image

To save a dotplot as image right-click on the dotplot and select the *Dotplot Save/Load Save as image* context menu item:



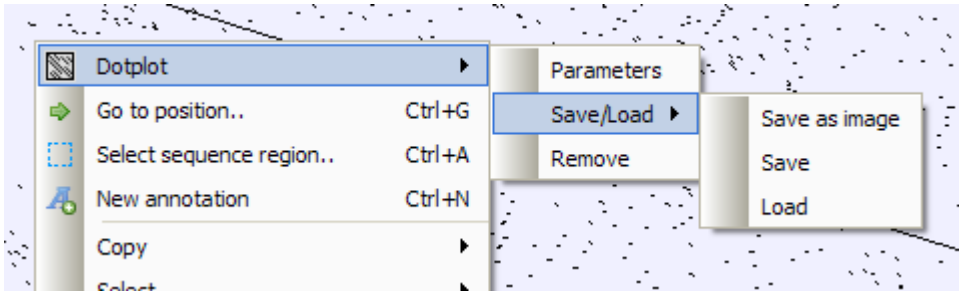
The following dialog will appear:



Available formats are *.png, *.jpg, *.bmp, *.jpeg, *.ppm, *.tif, *.tiff, *.xbm and *.xpm.

Saving and Loading Dotplot

To save a dotplot in a native format, right-click on the dotplot and select the *Dotplot Save/Load Save* context menu item:



The *Save Dotplot* dialog will appear. A dotplot is saved in a file with the *.dpt extension.

Later the dotplot can be loaded using the *Dotplot Save/Load Load* context menu item.

Building Dotplot for Currently Opened Sequence

To build a dotplot for currently opened sequences, create a multiple view containing these sequences. It can be arranged by dragging the corresponding sequence objects (the items strated with the “[s]”) into the same *Sequence View*.

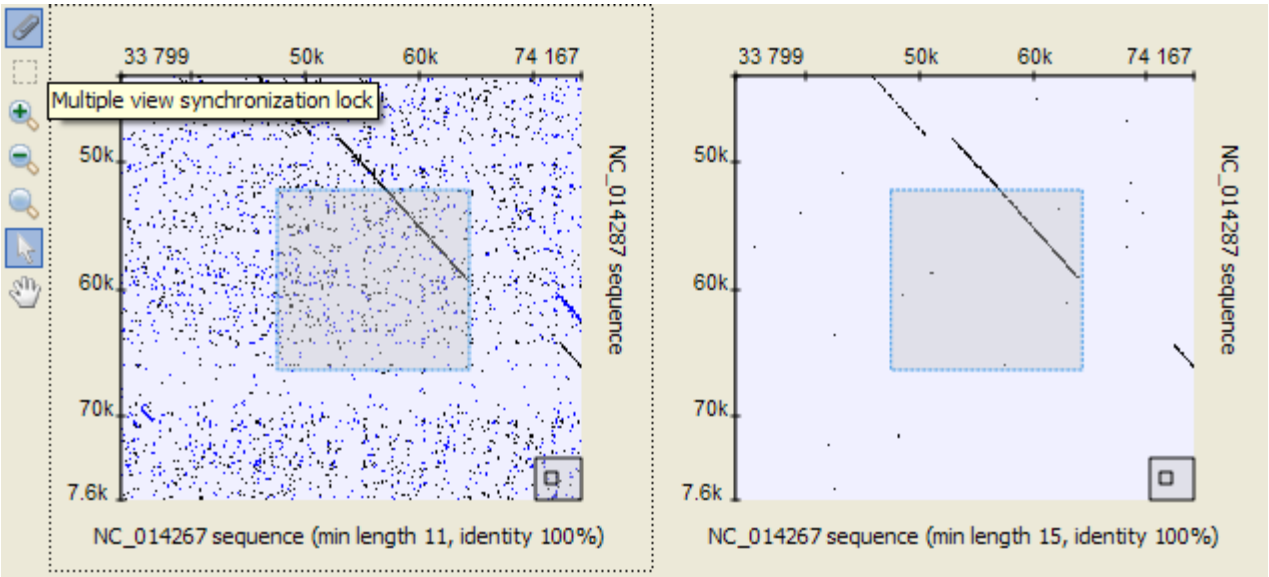
Then right-click on the created view and select the *Analyze Build dotplot* item in the context menu. Every sequence from the current multiple sequence view can be used to build a dotplot.

If you need to compare a sequence with itself, you can activate the menu from a single *Sequence View*.

Comparing Several Dotplots

Dotplots created for the same view are shown at the same view.

If the horizontal and vertical sequences of several dotplots are the same correspondingly, it is possible to lock all zooming and navigating operations for these dotplots. Press the *Multiple view synchronization lock* button on the left.



Alignment Editor

- Overview
  - Alignment Editor Features
  - Alignment Editor Components
  - Navigation
  - Coloring Schemes
    - Creating Custom Color Scheme
  - Highlighting Alignment
  - Zooming and Fonts
  - Consensus
    - Export Consensus
  - Alignment Overview
- Working with Alignment
  - Selecting Alignment Region
  - Moving Subalignment
  - Copying and Pasting Subalignment
  - Searching in Alignment
  - Editing Alignment
    - Removing Selection
    - Filling Selection with Gaps
    - Replacing Selected Character
    - Replacing with Reverse-Complement
    - Replacing with Reverse
    - Replacing with Complement
    - Removing Columns of Gaps
    - Removing Sequence
    - Removing All Gaps
    - Undo and Redo Framework
  - Saving Alignment
  - Aligning Sequences
  - Aligning Sequence to this Alignment
  - Pairwise Alignment
  - Working with Sequences List
    - Adding New Sequences
    - Renaming Sequences
    - Sorting Sequences
    - Shifting Sequences
    - Collapsing Rows
    - Copying Sequences
  - Exporting in Alignment
    - Extracting Selected as MSA
    - Exporting Sequence from Alignment
    - Exporting Alignment as Image
  - Importing APR and ACE Files
- Statistics
  - Distance Matrix
  - Grid Profile

- Advanced Functions
  - Building HMM Profile
- Building Phylogenetic Tree
  - PHYLIP Neighbor-Joining
  - MrBayes
  - PhyML Maximum Likelihood

Overview

This chapter gives an overview of the *Alignment Editor* components and explains basic concepts of browsing an alignment.

- Alignment Editor Features
- Alignment Editor Components
- Navigation
- Coloring Schemes
  - Creating Custom Color Scheme
- Highlighting Alignment
- Zooming and Fonts
- Consensus
  - Export Consensus
- Alignment Overview

Alignment Editor Features

The *Alignment Editor* is a powerful tool for visualization and editing DNA, RNA or protein multiple sequence alignments. The editor supports different multiple sequence alignment (MSA) formats, such as ClustalW, MSF, and Stockholm. The full list of file formats supported in UGENE is *here*.

The editor provides the interactive visual representation which includes:

- Navigation through an alignment;
- Optional coloring schemes (for example Clustal, Jalview like, etc.);
- Flexible zooming for large alignments;
- Export publication-ready images for alignment;
- Several consensus calculation algorithms.

Using the *Alignment Editor* you can:

- Perform multiple sequence alignment using integrated MUSCLE and KAlign algorithms;
- Edit an alignment: delete/copy/paste symbols, sequences, and subalignments;
- Build phylogenetic trees;
- Generate grid profiles;
- Build Hidden Markov Model profiles to use with HMM2/HMM3 tools.

Alignment Editor Components

Here is the default layout of the editor:



The *Alignment Editor* components:

For example, let's assume that the coordinate of the first visible base of the row is *N*, but the row contains *K* gaps before the position *N*. The starting offset value will be *N-K*. The same rule is true for the ending offset.

You can turn off the *Sequence offsets* by unchecking the *Actions* → *View* → *Show offsets* main menu item or *View* → *Show offsets* context menu item.

Navigation

The *Sequence area* provides several flexible ways to navigate through an alignment. The simplest way is to use the mouse and the scrollbars.

Alternatively, you can use arrow keys on the keyboard to navigate.

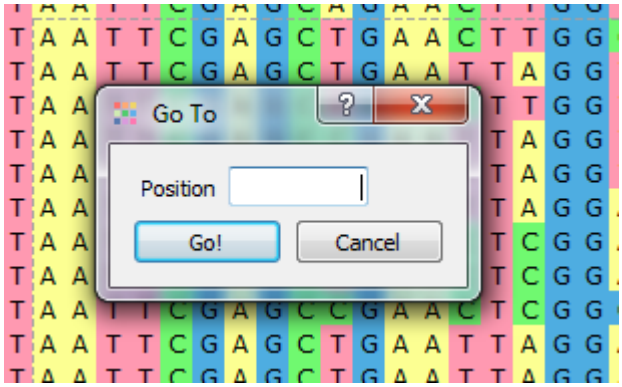
The list of hotkeys for quick navigation:

- PageUp — to move one screen left.
- PageDown — to move one screen right.
- Home — to center the starting columns of the alignment.
- End — to move to the trailing columns of the alignment

Hint

if you use Shift key with the hotkeys above you will navigate through the rows. For example, Shift-PageDown will move one screen down.

Finally, you can use the *Go to position* dialog from the *Actions* menu, the context menu or the editor toolbar.



Enter the column number (base coordinate) and the view will be centered to the corresponding base.

Coloring Schemes

It is possible to apply different coloring schemes for an alignment, so that nucleotide or amino acid residues are colored depending on some rules.

To set another coloring scheme, select the required item in the main (*Actions Colors*) or context menu (*Colors*) of the *Alignment Editor*, or in the *Color* com bo box on the *Highlighting* tab on the *Options Panel*.



Note that the coloring schemes setting may interfere with other alignment highlighting settings (see e.g. *Highlighting* combo box on the *Highlighting* tab of the *Options Panel*). Thus, the colors may vary in this case from the described below.

Coloring Schemes for Nucleotides

The following schemes options are available for a nucleotide alignment:

- *No colors* — do not color the residues background (i. e. all residues have white background and black font).
- *Jalview* — use the following background colors: A - green, C - orange, G - red, T/U - blue. Gaps and characters of the extended nucleotide alphabet are not colored.
- *Percentage identity* — a residue background color depends on its presence in the column of the alignment: <=40% - white, 41-60% - light slate blue, 61-80% - medium slate blue, 81-100% dark slate blue.
- *Percentage identity (colored)* — colors depend on the residues percentage presence in a column. The following rules are applied (in the order of priority):
  - If all residues (100%) are the same and there are no gaps, the residues in the column have yellow background and red font.
  - If there are 50% of one residue in the column and 50% of another (gaps are not taken into account), one half of the residues has green background and black font (the half to be colored is determined based on the following priorities: T>U>G>C>A>B>D>H>K>M>R>S>V>W>Y>N).
  - If there is one or several residues in a column with percentage number more or equal to the threshold parameter (see the *Threshold* para meter on the *Highlighting* tab of the *Options Panel*), then most frequent residues are colored as follows: light blue background color and blue font. Note that gaps are taken into account when calculating percentage number in the column in this case. Also, if required, the same residues priorities are applied as in the previous rule.
  - If the rules above are not applied for a residue, then it remains uncolored, i.e. white background color and black font are used.
- *Percentage identity (gray)* — this coloring scheme is similar to *Percentage identity* scheme, but shades of gray color are used instead of the slate blue color.
- *UGENE* — this is the default coloring scheme, the following background colors are used: A - yellow, C - light green, G - light blue, T/U - light red, gap - white. Other colors may are used for characters of the extended nucleotide alphabet.

- *UGENE Sanger* — this coloring scheme is used in the *UGENE Sanger Reads Editor*, the background colors correspond to the peaks of Sanger chromatogram peaks: A - green, C - blue, G - light gray, T - red. The orange background color is used for a gap, magenta background color is used for characters of the extended nucleotide alphabet.
- A custom scheme can be created and used with colors that depend on a residue.

Coloring Schemes for Amino Acids

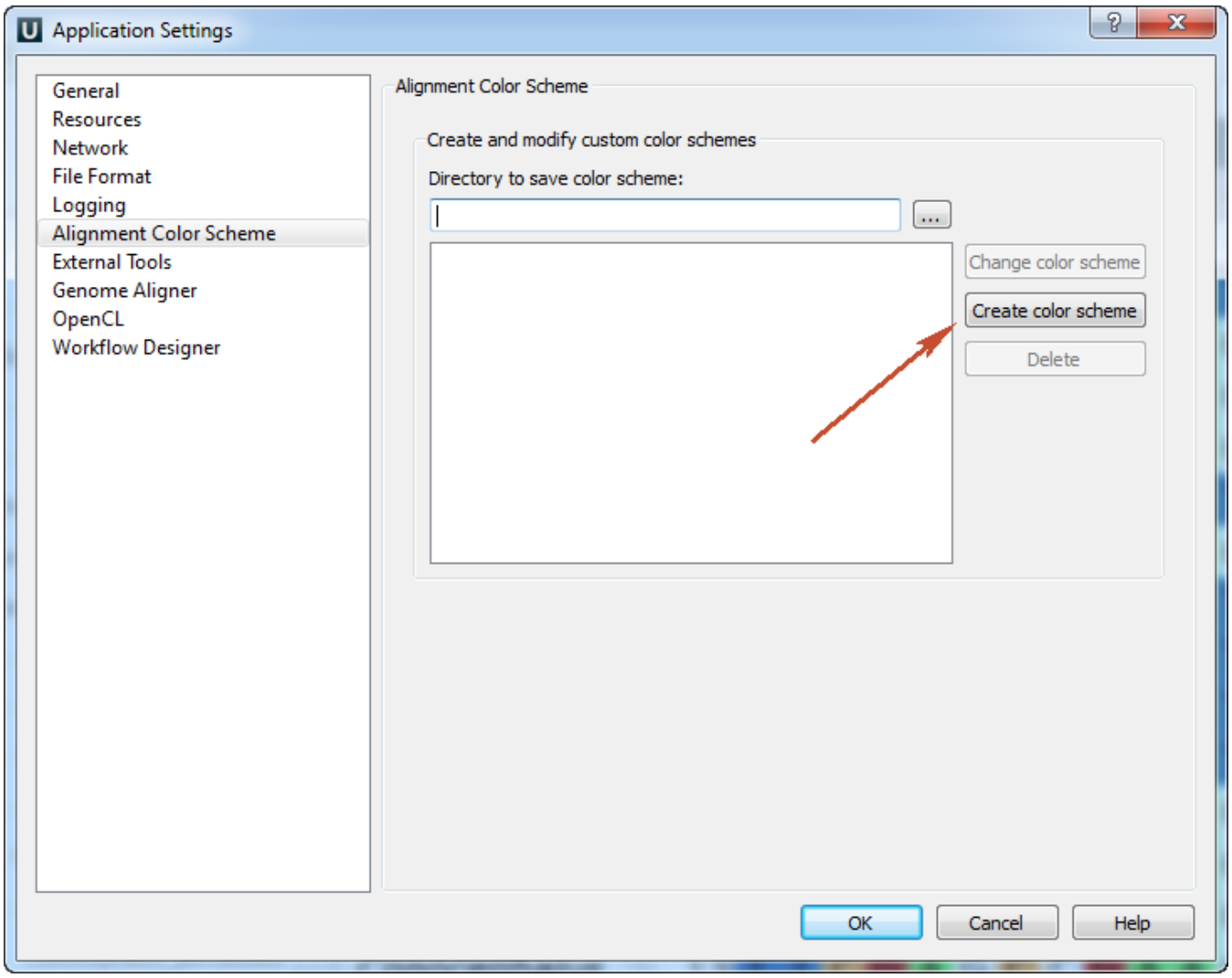
The following schemes options are available for an amino acids alignment:

- *No colors* — do not color the residues background (i. e. all residues have white background and black font).
- *Buried index* — depending on the frequency of occurrence inside a globule the amino acids residues are colored from light green (less frequent) to blue (most frequent).
- *Clustal X* — an emulation of the default color scheme in Clustal X, a graphical interface for the ClustalW multiple alignment program.
- *Helix propensity* — depending on helix propensity the residues are colored from magenta (the highest helix propensity) to green (the lowest one).
- *Hydrophobicity* — residues are colored based on their hydrophobicity. Most hydrophobic residues are red, most hydrophilic are blue.
- *Percentage identity* — a residue background color depends on its presence in the column of the alignment: <=40% - white, 41-60% - light slate blue, 61-80% - medium slate blue, 81-100% dark slate blue.
- *Percentage identity (gray)* — this coloring scheme is similar to *Percentage identity* scheme, but shades of gray color are used instead of the slate blue color.
- *Strand propensity* — residues with highest strand propensity are yellow, with lowest are blue.
- *Tailor* — see this publication by William R.Taylor.
- *Turn propensity* — residues with highest turn propensity are red, with lowest are cyan.
- *UGENE* — this is the default coloring scheme, the following background colors are used for the residues: positive (KRH) - yellow, aromatic (FWY) - green, large alphatic hydrophobic (ILM) - blue, small hydrophobic (ST) - rose, GP - red, EX - gray.
- *Zappo* — amino acids residues are colored according to their physicochemical properties: alphatic/hydrophobic (residues ILVAM) - rose, aromatic (FWY) - orange, positive (KRH) - dark blue, negative (DE) - red, hydrophilic (STNQ) - light green, conformationally special (PG) - magenta, cysteine (C) - yellow.
- A custom scheme can be created and used with colors that depend on a character.

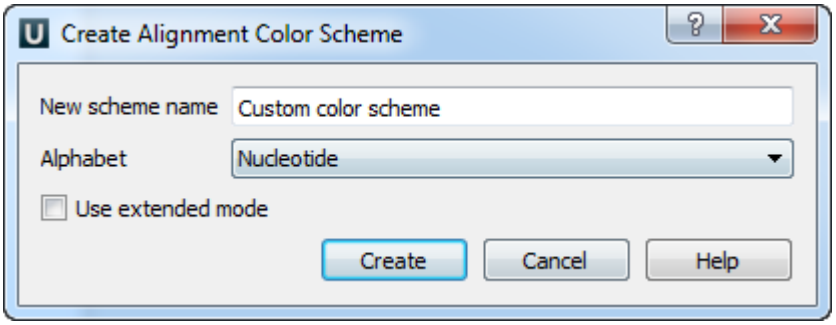
- Creating Custom Color Scheme

Creating Custom Color Scheme

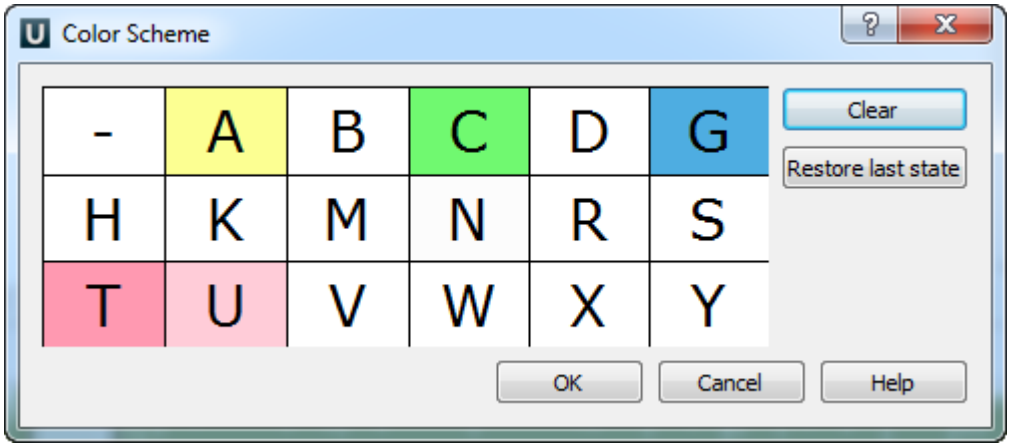
To create custom color scheme use the *Colors->Custom schemes->Create new color scheme* context menu item. The *Application Settings* dialog will appear. Click on the *Create color scheme* button:



The following dilaog will appear:



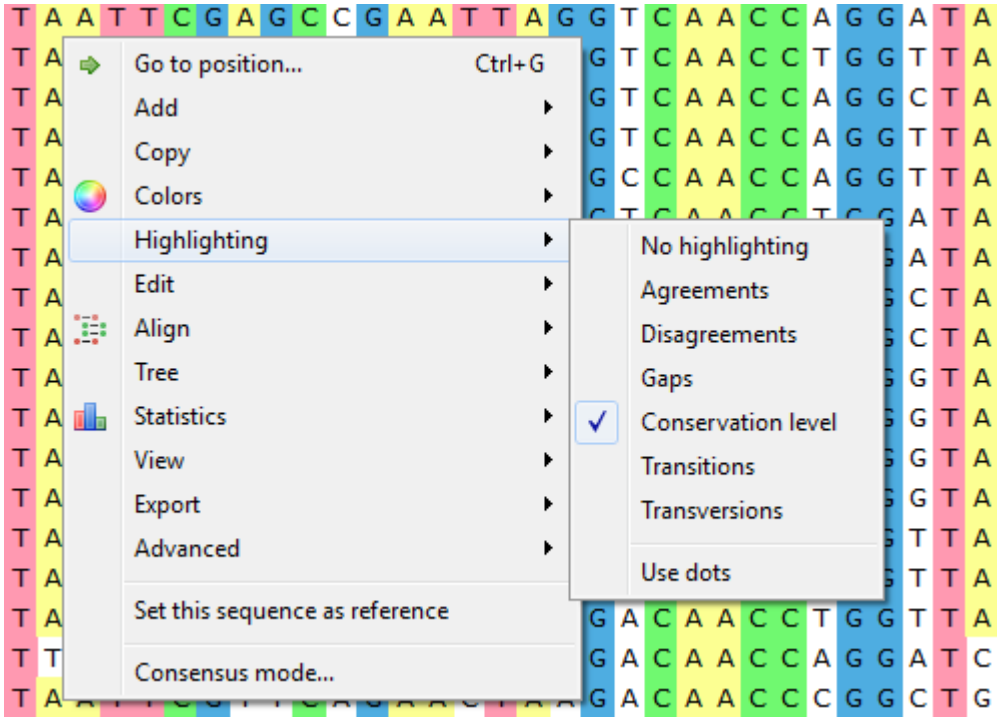
Select the new scheme name, alphabet and click on the *Create* button. The next dialog will appear for nucleotide extended mode:



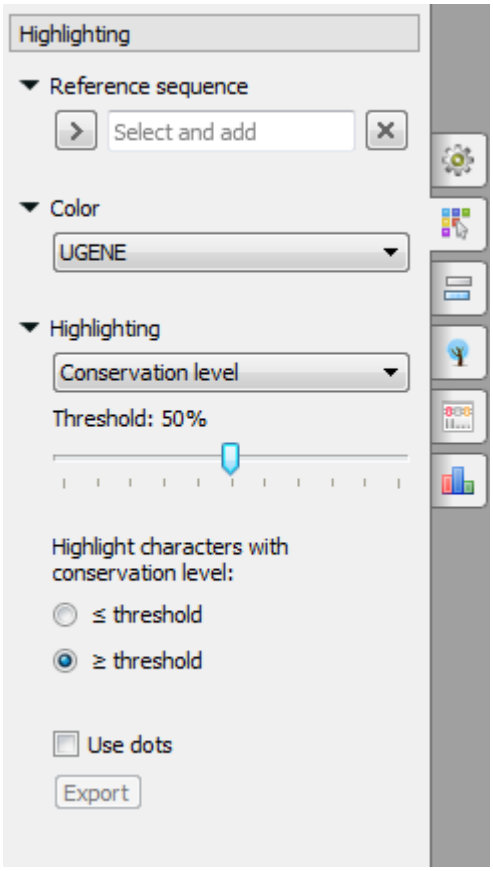
Here you can select a color for each element. Click on the element for it. The new scheme will be created after clicking the *OK* button. The new custom scheme will be available in the *Colors->Custom schemes* context menu.

Highlighting Alignment

To apply an alignment highlighting mode, select it in the *Highlighting* context menu:



or on the *Highlighting* tab of the *Options Panel:*



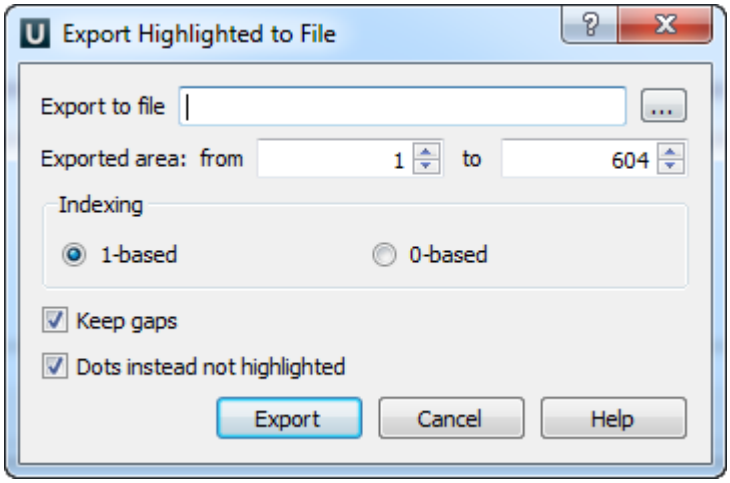
The following modes are available:

- *Agreements* — highlights symbols that coincide with the reference sequence.
- *Disagreements* — highlights nucleotides that differ from the reference sequence.
- *Gaps* - highlights gaps.
- *Conservation level* - highlights conservation level of symbols in a multiple alignment >= or <= treshhold. To select the conservation parameters use the *Highlighting* Options Panel tab.
- *Transitions* - highlights transitions.
- *Transversions* - highlights transversions.

To use dots instead of symbols which are not highlighted check the *Use dots* checkbox in the *Options Panel* or use the *Highlighting->Use dots* context menu item.

To select a reference sequence use the *Set this sequence as reference* context menu or *Reference sequence* field in the *Highlighting* tab of the *Options Panel*.

Also you can export highlighting with a help of the *Export* button in the *Options Panel* or by the *Export->Export highlighted* context menu item. The following dialog will appear:



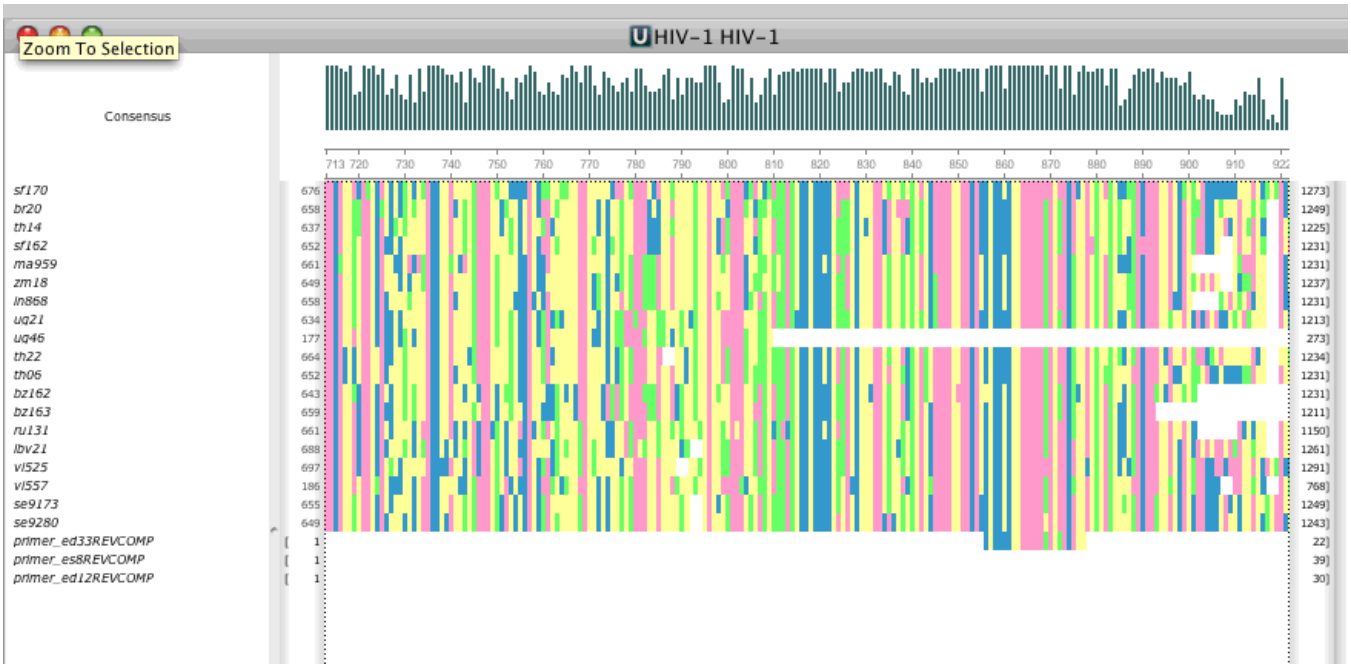
Select file to export, exported area and click on the *Export* button. The task report will appear in the *Notifications*.

Zooming and Fonts

To perform zoom operations use the corresponding buttons on the editor toolbar.



By default, the base characters are visible when zooming. But for rather long sequences there is another zoom mode available. In this mode the bases are not shown. This allows viewing very large sequence regions (up to 500 bp).



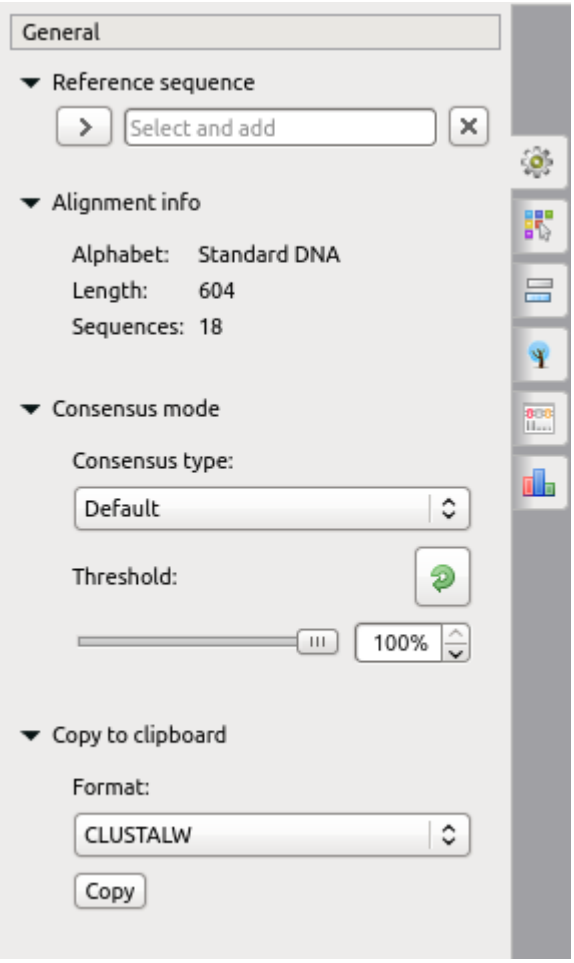
You can zoom to the selected region by clicking the *Zoom to selection* button. It is very convenient operation, when the alignment size is rather large. For example, you can zoom out to some percentage, select an interesting region and then zoom to the selection.

You can change font by clicking the *Change font* button.

To reset zoom and font click the *Reset zoom* button.

Consensus

Each base of a consensus sequence is calculated as a function of the corresponding column bases. There are different methods to calculate the consensus. Each method reveals unique biological properties of the aligned sequences. The *Alignment Editor* allows switching between different consensus modes. To switch the consensus mode go to the *General tab* of the *Options Panel* or activate the context menu (using the right mouse button) or the *Actions* menu and select the *Consensus mode* item and *General tab* will be opened automatically:



There are several consensus modes:

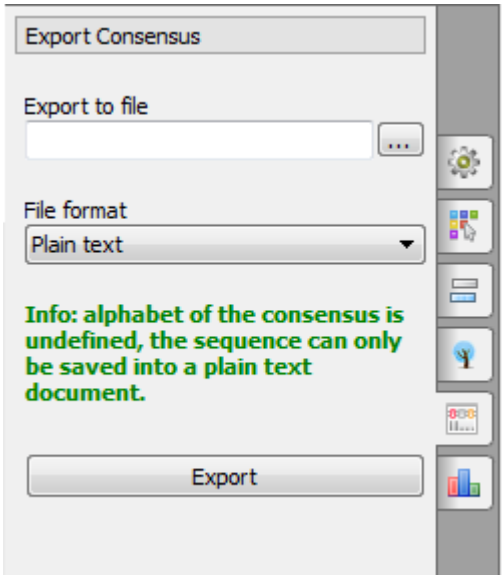
- **JalView (Default)** — the mode is based on the JalView algorithm. A character in the consensus is calculated based on the characters in the corresponding column and the value of the threshold:
  - If the percentage value of a character is greater than the threshold, the character is shown in upper case in the consensus.
  - If there are two characters with high frequency in a column, the '+" sign is shown.
  - If the percentage value of a character is high, but lower than the specified threshold, the character is shown in lower case in the consensus.
- **ClustalW** — emulates the ClustalW program behavior.
- **Levitsky** — proposed by Victor Levitsky, this mode calculates the consensus of a DNA alignment, taking into account frequencies of characters in the whole alignment, i.e. not only one column. The algorithm includes these steps:
  - Collect global alignment frequencies for every character using the extended DNA alphabet.
  - For every column select the rarest character in the whole alignment with the percentage in the column greater or equal to the threshold value.
- **Simple extended** — a character of the consensus is calculated based on the characters in the corresponding column. The extended DNA alphabet is used for the result consensus, the threshold value is also taken into account.
- **Strict** — if a character percentage in a column is higher than the specified threshold, the character is shown in the consensus at this position. Otherwise, a gap character ('—') is shown.

Also the *General tab* shows the general information about an alignment and allows one to set up a reference sequence. The following chapter describes how to export a consensus sequence:

- Export Consensus

Export Consensus

To export a consensus sequence use the *Export consensus* tab of the *Options Panel*:



The following parameters are available:

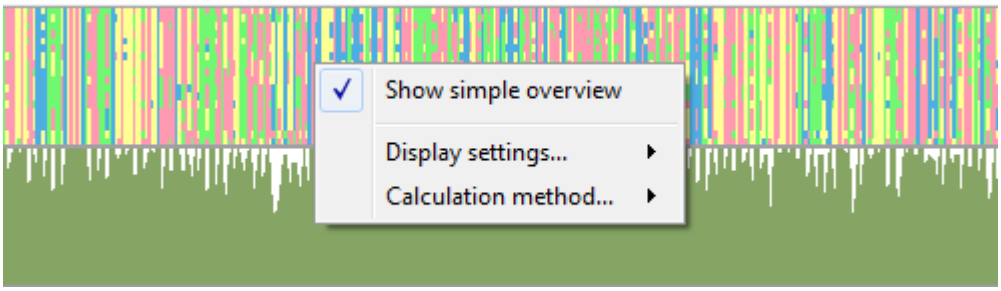
Export to file - the output file path.

File format - format of the output file. The possible values depend on the mode, selected for the consensus calculation. For *Default* and *ClustalW* modes the *Plain text* value is only available. For *Levitsky*, *Simple extended* and *Strict* consensus modes sequence formats FASTA and GenBank are also available.

Click on the *Export* button to export the consensus sequence to the specified output file.

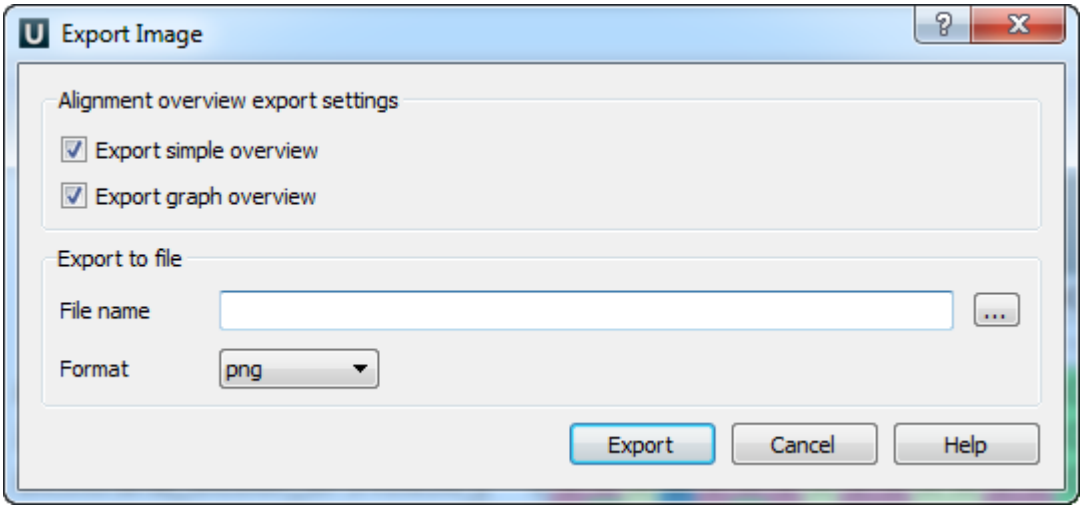
Alignment Overview

The alignment overview is shown automatically in the Alignment Editor. To close the overview click on the *Overview* toolbar button. To show the simple alignment overview use the *Show simple overview* context menu item of the overview.



The following settings of the alignment overview are available:

Export as image - you can export multiple alignment overview and simple alignment overview as image. Use this context menu item to do it. In the following dialog select the required parameters and click on the *Export* button:



Dispaly settings:

Graph type - sets the graph type: histogram, line graph or area graph.

Orientation - sets the orientation: top to buttom or buttom to top.

Set color - sets the gpaph color.

Calculation method - sets the calculation method: strict, gaps, clustal or highlighting.

To use these settings go to the corresponding context menu items of the alignment overview.

Working with Alignment

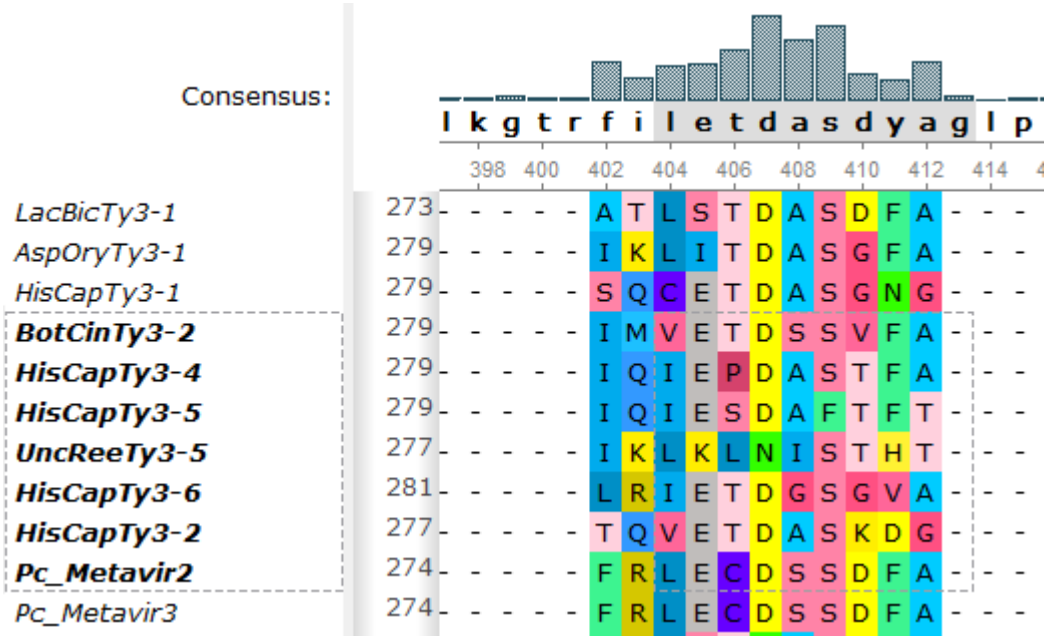
This chapter explains how to work efficiently with the *Alignment Editor*. You will learn how to modify an alignment, remove gaps, align sequences, copy and paste regions, add new sequences and extract subalignments as new alignments.

- Selecting Alignment Region
- Moving Subalignment
- Copying and Pasting Subalignment
- Searching in Alignment
- Editing Alignment
  - Removing Selection
  - Filling Selection with Gaps
  - Replacing Selected Character
  - Replacing with Reverse-Complement
  - Replacing with Reverse
  - Replacing with Complement
  - Removing Columns of Gaps
  - Removing Sequence
  - Removing All Gaps
  - Undo and Redo Framework
- Saving Alignment
- Aligning Sequences
- Aligning Sequence to this Alignment
- Pairwise Alignment
- Working with Sequences List
  - Adding New Sequences
  - Renaming Sequences
  - Sorting Sequences
  - Shifting Sequences
  - Collapsing Rows
  - Copying Sequences
- Exporting in Alignment
  - Extracting Selected as MSA
  - Exporting Sequence from Alignment
  - Exporting Alignment as Image
- Importing APR and ACE Files

Selecting Alignment Region

Selection in the Alignment Editor components

In the *Name List* and the *Sequence Area* selection is visualized as a rectangle with a dashed border. In the *Consensus Area* the consensus sequence character of the corresponding columns are shown with light gray background:



Information about selection is shown in the statistic bar under alignment.



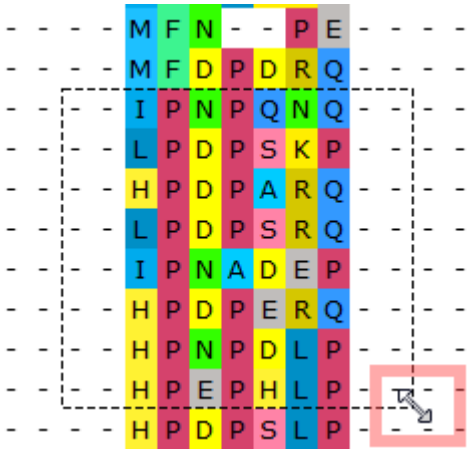
Selecting a region using the mouse and keyboard shortcuts

Using the mouse one can select:

- Several consequent sequences in the *Name List*.
- Several consequent columns in the *Consensus Area*.
- An alignment region in the *Sequence Area*.

By holding the *Shift* key it is possible to select a region between two coordinates of a mouse click. For example, if there is a big alignment, one can click on a character (the left upper coordinate of the region), press *Shift*, scroll to the required location of the alignment, and click on another character (the right bottom coordinate of the region).

It is possible to modify a selection in the *Sequence Area* by dragging the selection rectangle border:



Press the *Esc* key to clear the selection.

Moving Subalignment

To move subalignment there are different ways:

1. Select a subalignment and drag and drop it. The subalignment will be moved.
2. With a help of the *Space,* the subalignment will be moved to the right by the size of the selection. With a help of the *Backspace*, the subalignment will be returned to the first state.
3. With a help of the *Ctrl+Space,* the subalignment will be moved to the right by one column. With a help of the *Ctrl* + *Backspace*, the subalignment will be returned to the first state.
4. With a help of the *Ctrl*, you can join the parts of alignment. When there is a fragment of characters, surrounded by gaps in a row, select continuous block without gaps on *Ctrl* pressed and move the selection to the right/left.

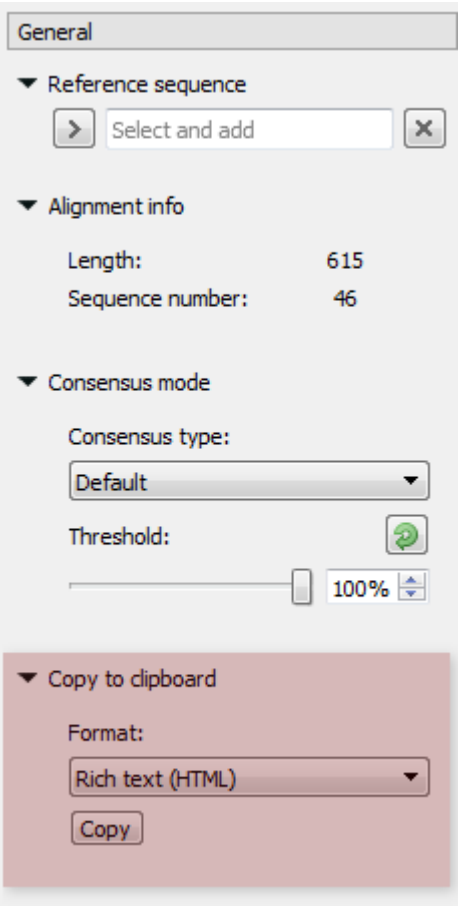
Copying and Pasting Subalignment

To copy current selection click the *Copy/Paste Copy selection* item in the *Actions* main menu or the context menu. The hotkey for this action is Ctrl-C.

Also you can copy selection in different formats, for example FASTA, CLUSTALW, RICH TEXT(HTML), NEXUS, MSF.

To copy subalignment in selected format do the following:

- Select the subalignment;
- Select the *Copy/Paste->Copy formatted* context menu item in the context menu or use hotkey *Ctrl+Shift+C.*
- The global Options Panel tab will open:

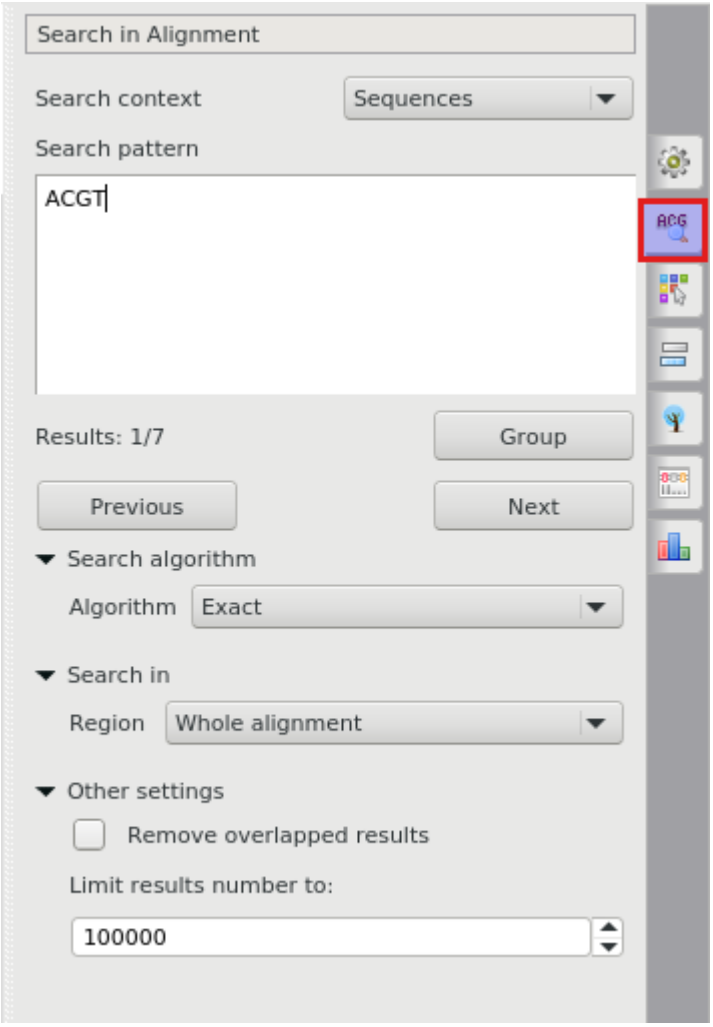


- Select format and press the *Copy* button.

To paste any of copied selection use the *Copy/Paste->Paste* context menu item or *Ctrl+V* shortcut.

Searching in Alignment

To search for a pattern(s) in alignment go to the *Search in Alignment* tab of the *Options Panel*.



The following parameters are available:

Search context

This group specifies the search context that should be used to search for a pattern. It is possible to search patterns in sequences or in sequence names.

Search pattern

To search for a pattern(s) input the value you want to search in the text field.

Press the *Next* button to search in the direction “From left to right, from top to bottom”. Press the *Previous* button to search in the direction “From right to left, from bottom to top”. If the pattern is found, the result will be focused and highlighted in the *Sequence area*. You can continue the search in any direction from this position.

To group results press *Group* button. The results will be selected and grouped to the top after the first click or to the bottom after the second click to the button.

Search Algorithm

This group specifies the algorithm that should be used to search for a pattern. The algorithm can be one of the following:

- *InsDel* — there could be insertions and/or deletions, i.e. a pattern and the searched region can vary in their length. You can specify the percentage of the pattern and a searched region match in the field nearby. Note that this value also depends on the pattern length and is disabled when the pattern hasn’t been specified.
- *Substitute* — a pattern may contain characters different from the characters in the searched region. When this algorithm has been selected you can also specify the match percentage and additionally it is possible to take into account ambiguous bases.
- *Regular expression* — a regular expression may be specified instead of a pattern. For example character ‘.’ matches any character, ‘.*’ matches zero or more of any characters. There is also the *Limit result length* option that specifies the maximum length of a result.
- *Exact* - find a place where one or several patterns are found within a larger pattern.

Search in

In this group, you can specify the sequence range where to search for a pattern. You can search in the whole alignment, specify a custom region or search in the selected region.

Other Settings

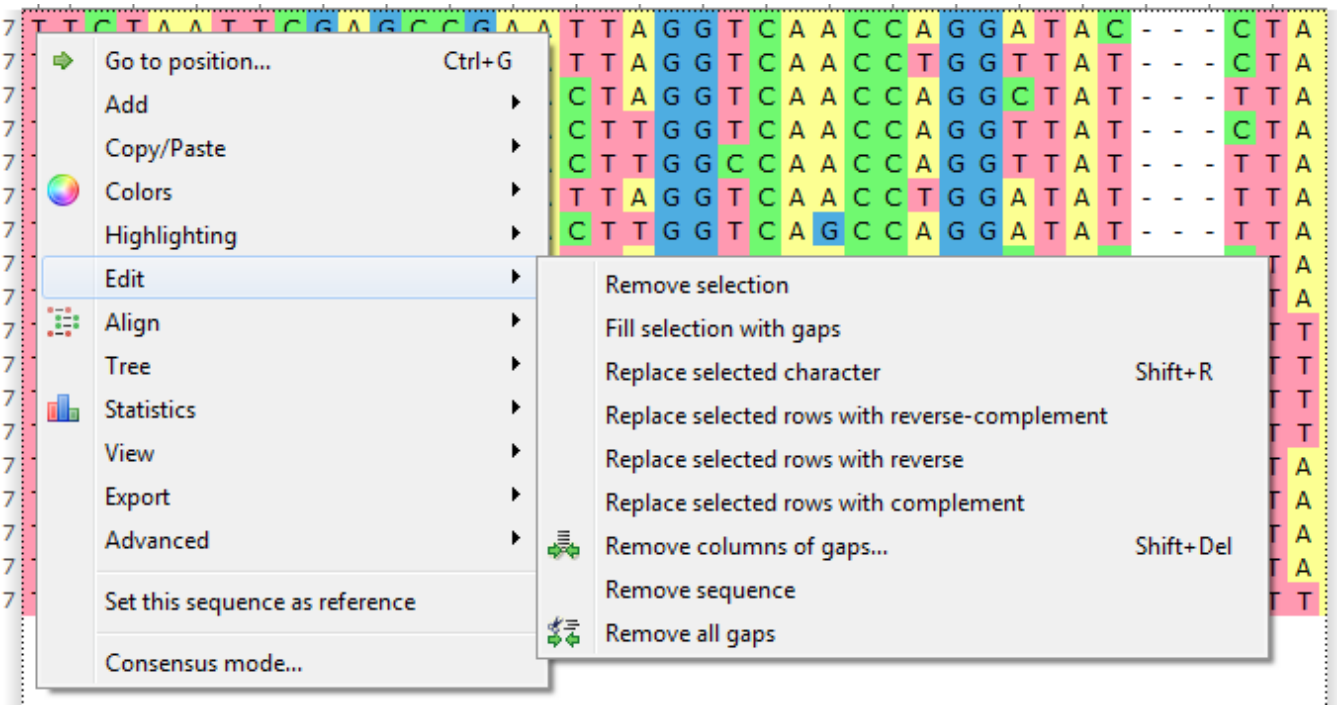
This group contains additional common settings:

Remove overlapped results — annotates only one of the overlapped results.

Limit results number to — limits the number of the searched results to the specified value.

Editing Alignment

Select the *Edit* submenu in the *Alignment Editor* context menu:



The actions available from this menu are described below.

- Removing Selection
- Filling Selection with Gaps
- Replacing Selected Character
- Replacing with Reverse-Complement
- Replacing with Reverse
- Replacing with Complement
- Removing Columns of Gaps
- Removing Sequence
- Removing All Gaps
- Undo and Redo Framework

Removing Selection

To remove a subalignment select it and choose the *Edit Remove selection* item in the context menu or press the Delete key. For macOS use the Fn+Delete key instead of the Delete key.

Filling Selection with Gaps

Select a region in the alignment and choose the *Edit Fill selection with gaps* item in the context menu or press the Spacebar. The region is filled with gaps shifting the subalignment from the region to the right.

Replacing Selected Character

To replace a symbol in the alignment select the nucleotide and press *Shift+R* for switching the edit mode on.

While the selection is blinking, press a keyboard character that you want to replace the nucleotide with.

After that the edit mode is switched off. For replacing another nucleotide press *Shift+R* again.

You always can cancel the changes pressing *Ctrl+Z*.

Replacing with Reverse-Complement

To replace sequence(s) in the alignment with reverse-complement select it and use the *Edit->Replace with reverse-complement* item in the context menu.

Replacing with Reverse

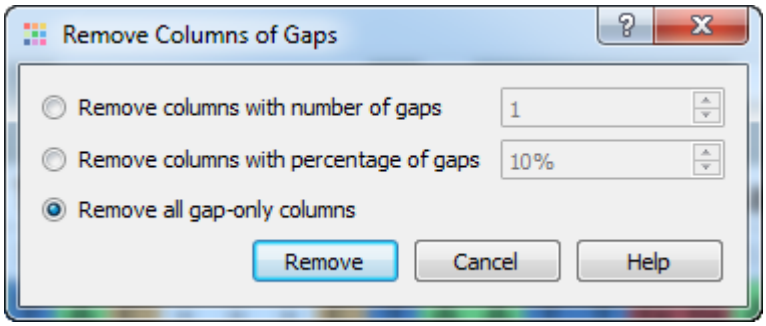
To replace sequence(s) in the alignment with reverse select it and use the *Edit->Replace with reverse* item in the context menu.

Replacing with Complement

To replace sequence(s) in the alignment with complement select it and use the *Edit->Replace with complement* item in the context menu.

Removing Columns of Gaps

To remove colums containg certain number of gaps select the *Edit Remove columns of gaps* item in the context menu. The dialog appears:



There are the following options:

Remove columns with number of gaps — removes columns with number of gaps greater than or equal to the specified value.

Remove columns with percentage of gaps — removes columns with percentage of gaps greater than or equal to the specified value.

Remove all columns of gaps — this option is selected by default. It specifies to remove columns from the alignment if they entirely consist of gaps.

Select the option required and press the *Remove* button.

Removing Sequence

Select a sequence and call the *Edit Remove sequence(s)* item in the *Actions* main menu or in the context menu to remove sequence from the alignment.

Removing All Gaps

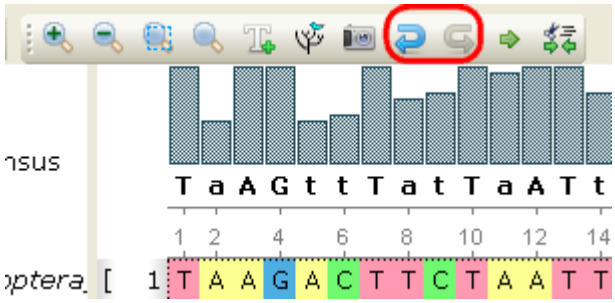
Use the *Edit Remove all gaps* item in the *Actions* main menu or in the context menu to remove all gaps from the alignment.

Undo and Redo Framework

The editor tracks all modifications of the aligned sequences.

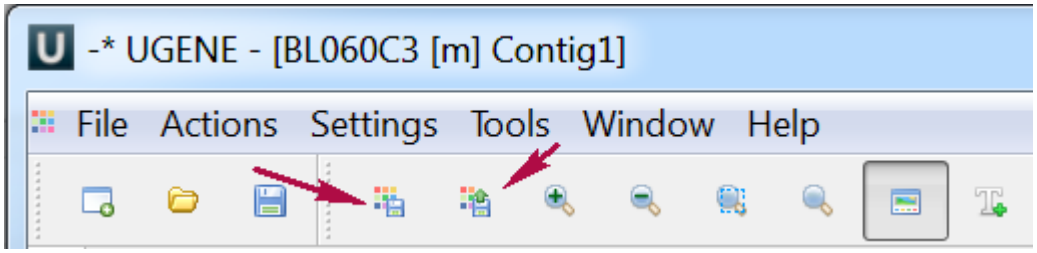
When a modification happens the current state of the multiple sequence alignments object is being recorded.

You can apply any previous state and redo the modifications using the corresponding buttons on the toolbar:



Saving Alignment

To save current alignment click the *Save alignment* button, to the the alignment into another file click the *Save alignment as* button.



Aligning Sequences

The *Alignment Editor* integrates several popular multiple sequence alignment algorithms. Below is the list of available algorithms and links to the documentation:

- Port of the popular *MUSCLE3* algorithm.
- KAlign plugin: effective work with huge alignments.
- ClustalW and MAFFT: these algorithms appeared in the version 1.7.2 of UGENE with the *External Tools* plugin.
- T-Coffee: this alignment algorithm is available since version 1.8.1 of UGENE with the *External Tools* plugin.
- ClustalO: Clustal Omega is the latest addition to the Clustal family. It offers a significant increase in scalability over previous versions, allowing hundreds of thousands of sequences to be aligned in only a few hours.

To align sequences choose a preferred alignment method in the *Actions* main menu, in the context menu or by *Align* main toolbar button .

Also you may find useful the following video tutorials devoted to the multiple sequence alignment:

- Making a multiple sequence alignment from FASTA file
- Working with large alignments in UGENE
- Performing profile-to-profile and profile-to-sequence MUSCLE alignments
- Running remote MUSCLE task

Aligning Sequence to this Alignment

To align a sequence to an opened alignment, click the *Align sequence to this alignment* toolbar button:

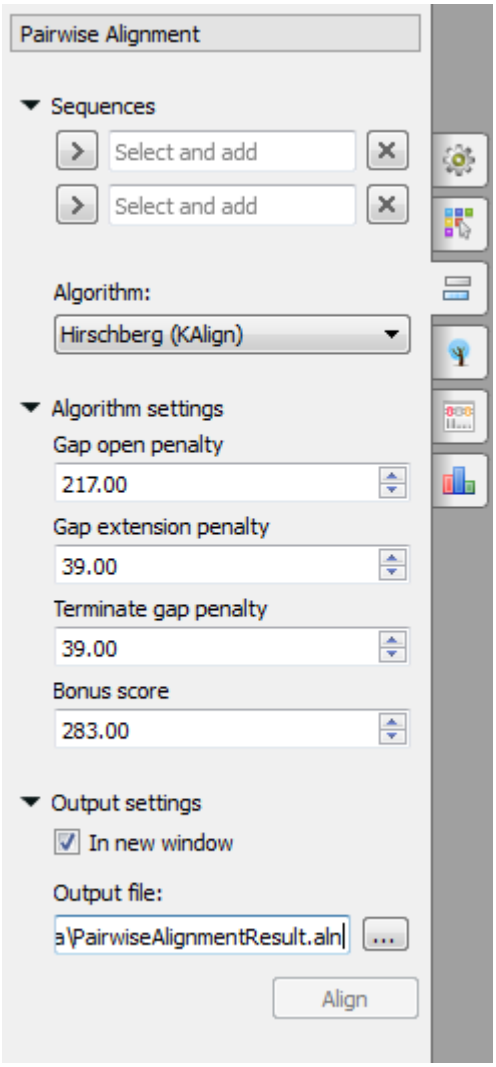


Choose a file with the sequence from the files system, and click *Open*.

Also, you can add an already opened sequence or sequences to the alignment. To do it, select the sequence object(s) in the Project View and click the *Alig n sequence to this alignment* toolbar button. The sequence(s) will be aligned to the alignment automatically.

Pairwise Alignment

To align two sequences go to the *Pairwise Alignment* tab of the *Options Panel*:



Select two sequence from the original alignment, select the parameters and click on the *Align* button. The following parameters are available:

Algorithm - algorithm of the pairwise alignment. There are two algorithms:

Hirschberg (KAlign) - algorithm has the following parameters:

Gap open penalty - indicates the penalty applied for opening a gap. The penalty must be negative.

Gap extension penalty - indicates the penalty applied for extending a gap.

Terminate gap penalty - the penalty to extend gaps from the N/C terminal of protein or 5'/3' terminal of nucleotide sequences.

Bonus score - a bonus score that is added to each pair of aligned residues.

Smith-Waterman - the following parameters are available:

Algorithm version - version of the algorithm implementation. Non-classic versions produce the same results as classic but much faster. To use these optimizations our system must support these capabilities: OPENCL, SSE2 or SW_classic.

Scoring matrix - scoring matrix.

Gap open penalty - penalty for opening a gap.

Gap extension penalty — penalty for extending a gap.

Output settings - settings of the otput file.

Working with Sequences List

- Adding New Sequences
- Renaming Sequences
- Sorting Sequences
- Shifting Sequences
- Collapsing Rows
- Copying Sequences

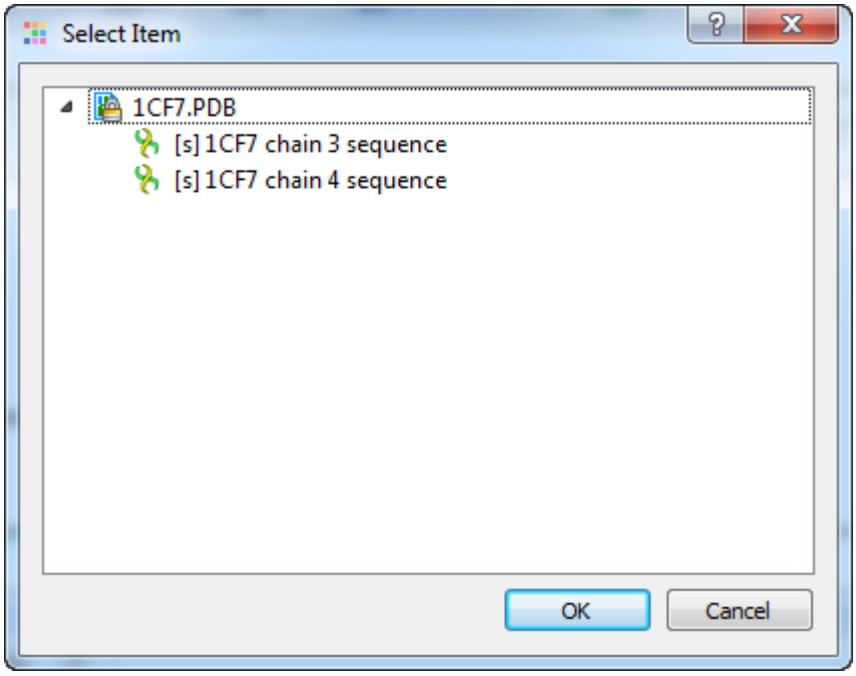
Adding New Sequences

You can add new sequences to an alignment using the *Add* submenu in the *Actions* main menu or the context menu.

There are two ways to add a new sequence to the current alignment:

- From a file in the compatible format (FASTA, GenBank etc.). The list of the supported data formats can be found *here*.
- From the current project.

If you activate this item, the following dialog will appear:



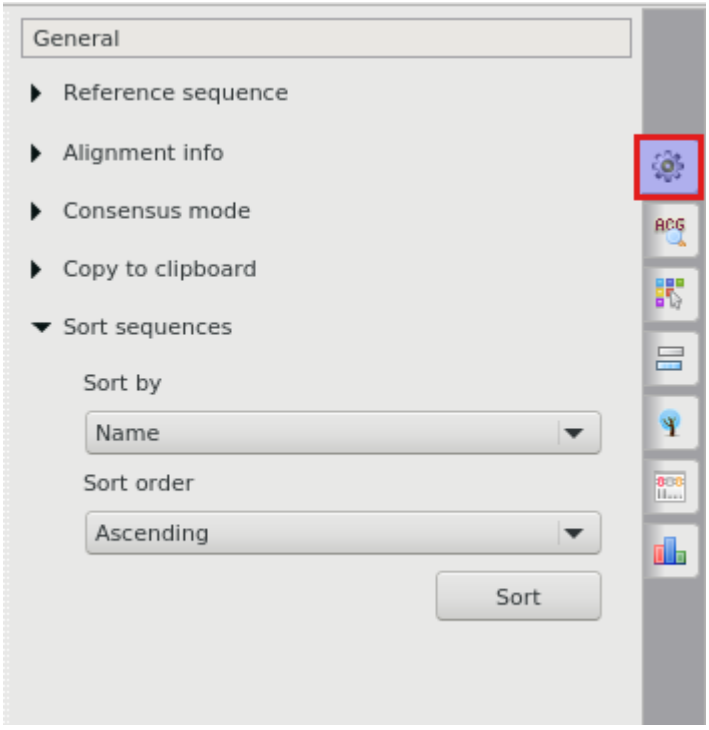
You will see the *Project View* tree filtered to show only appropriate sequences. Select the items to add and press the *Ok* button.

Renaming Sequences

To rename a sequence double click on the name of this sequence and enter a new sequence name in the dialog.

Sorting Sequences

You can sort sequences in alignment using the *Sort* submenu in the *Actions* main menu or from the context menu. Also, you can use the *General* Options Panel tab:



There are four ways to sort sequences:

- By name, ascending
- By name, descending
- By length, ascending
- By length, descending

Shifting Sequences

To change an order of sequences in a multiple sequence alignment do the following:

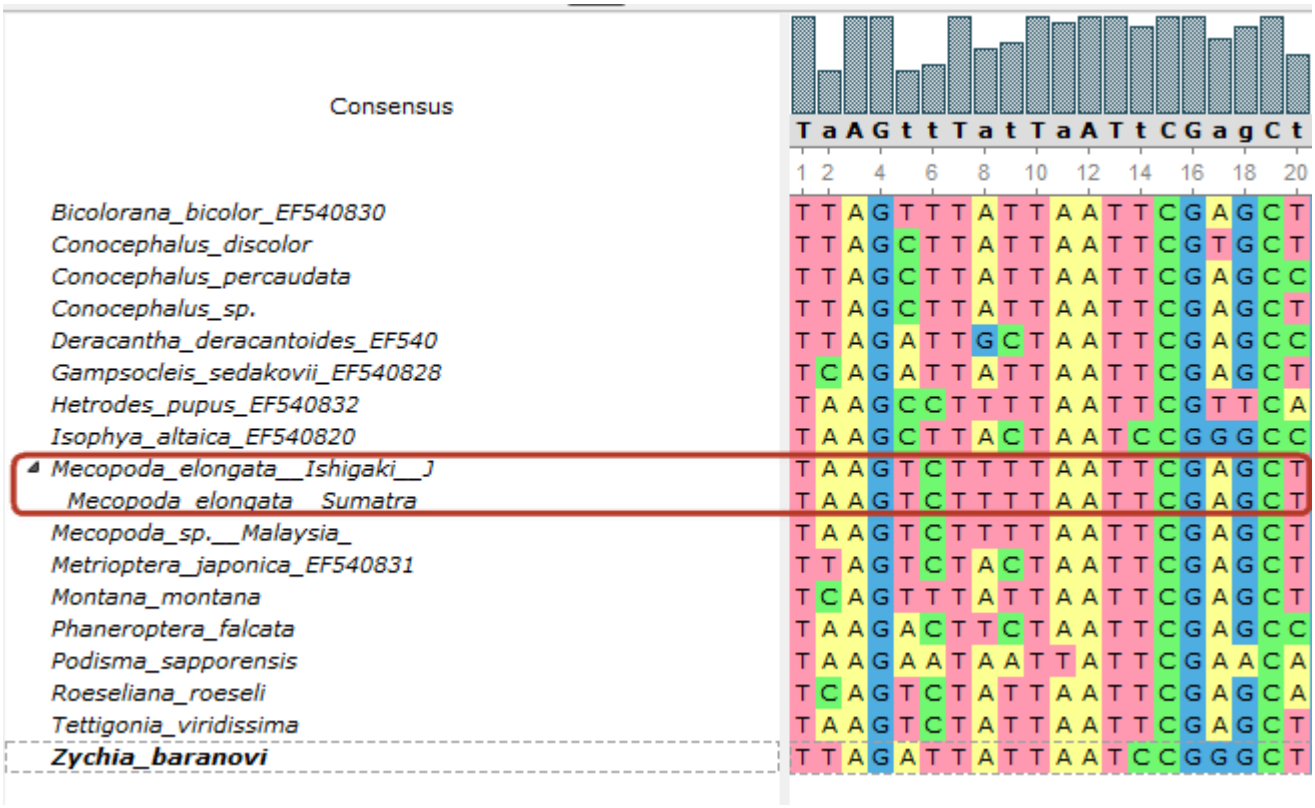
- select sequence or sequences in the sequences names list by click or by click and drag correspondingly.
- click and drag on selected region to shift it.

Collapsing Rows

It is able to collapse the sequential rows. To collapse rows click on the *Switch on/off collapsing* main toolbar button:



The triangle will appear near collapsed sequences. Click on the triangle to show the whole tree of the collapsed rows. Also, it is possible to control the collapsed group by a keyboard with a help of *Right* and *Left* keyboard arrows.

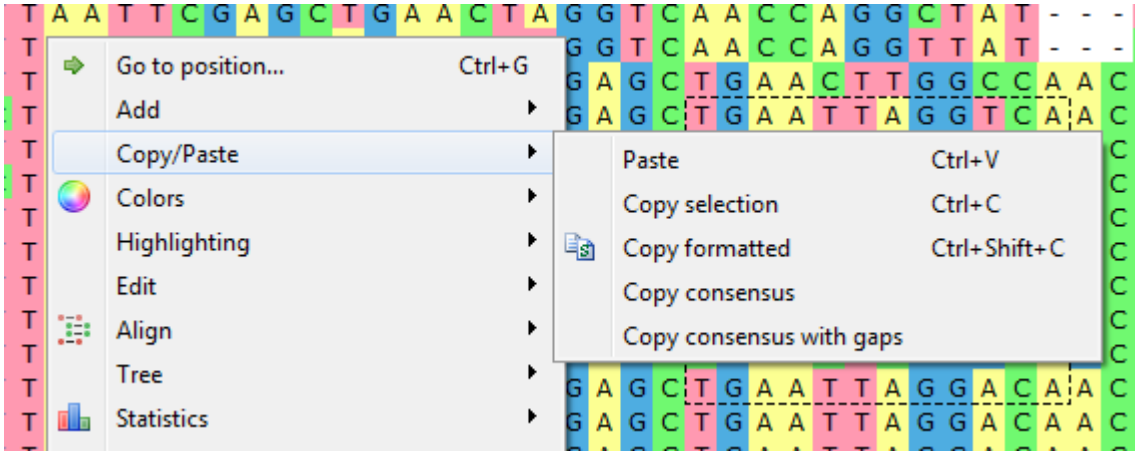


To update the collapsed groups click on the corresponding main toolbar button.

Copying Sequences

To copy one or several sequences do the following:

- Select the sequences in the *Sequence list* area;
- Select the *Copy/Paste Copy selection* context menu item in the *Sequence area* or use hot key combination. Note, that if you activate context menu in the *Sequence list* area you will lose your current selection.



To copy consensus sequence use the *Copy/Paste Copy consensus* item. To copy consensus with gaps use the corresponding menu item.

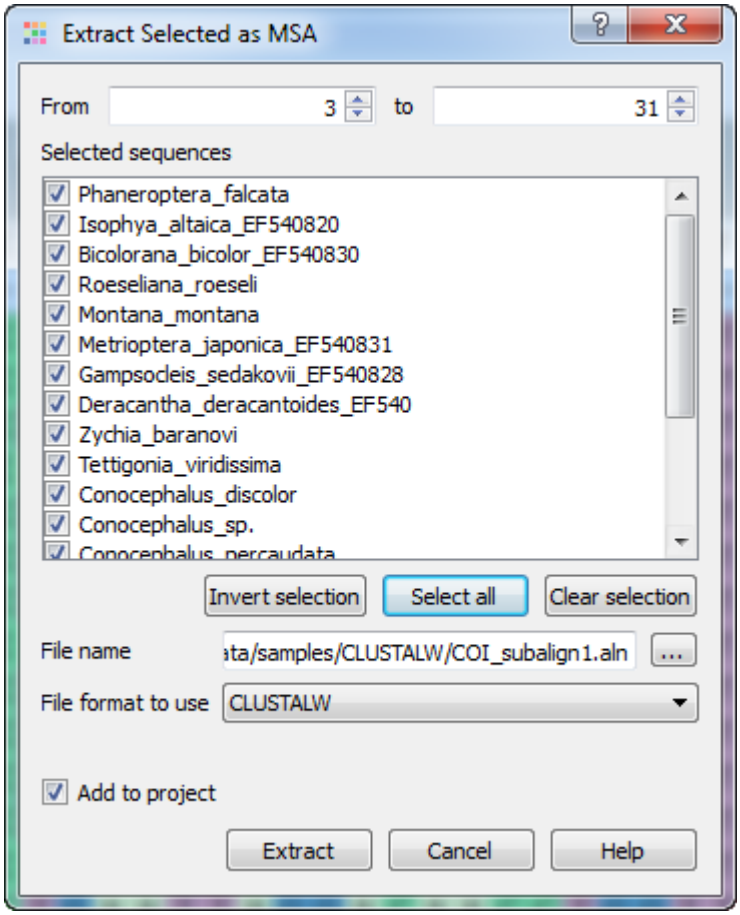
Exporting in Alignment

- Extracting Selected as MSA
- Exporting Sequence from Alignment
- Exporting Alignment as Image

Extracting Selected as MSA

It is possible to extract a subalignment and save it as new multiple sequence alignment (MSA).

Select a subalignment and choose the *Export Save subalignment* item in the *Actions* main menu or in the context menu. The following dialog appears:



Specify the name and format of the new MSA file in the *File name* and *File format to use* fields. The currently selected region is extracted by default when you press the *Extract* button.

You can change the columns to be extracted using the *From* and *to* fields. And change the rows to be extracted by checking/unchecking required sequences in the *Selected sequences* list.

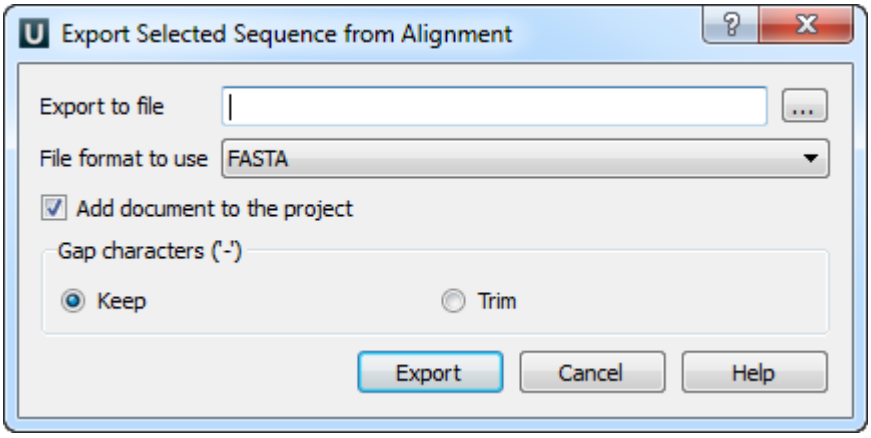
Use buttons:

- *Invert selection* — to invert the selection of the sequences.
- *Select all* — to select all sequences.
- *Clear selection* — to clear the selection of all sequences.

The *Add to project* check box specifies to add the MSA file created from the subalignment to the active project.

Exporting Sequence from Alignment

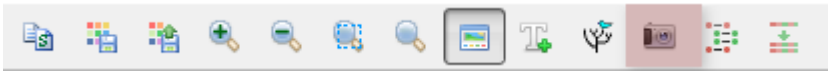
To export one sequence from an alignment select the sequence in the sequence list or in the sequence area and use the *Export->Save sequence* context menu item. The following dialog will appear:



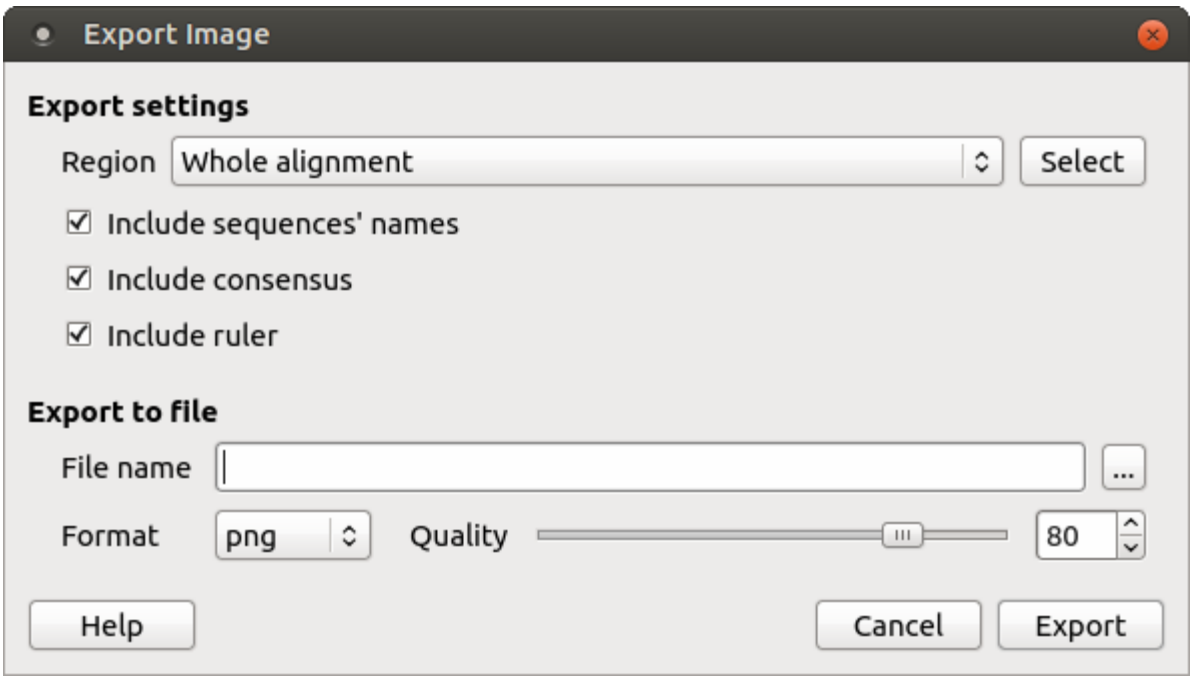
Here it is possible to specify the result file location, to select a sequence file format, to define whether to keep or remove gaps ('—' chars) in the sequence and optionally add the created document to the current project.

Exporting Alignment as Image

To export an alignment as image click the *Export as image* button on the editor toolbar or call the *Export->Export as image* context menu item.



The Export Image dialog will appear where you should set name, location, export settings and format of the picture:

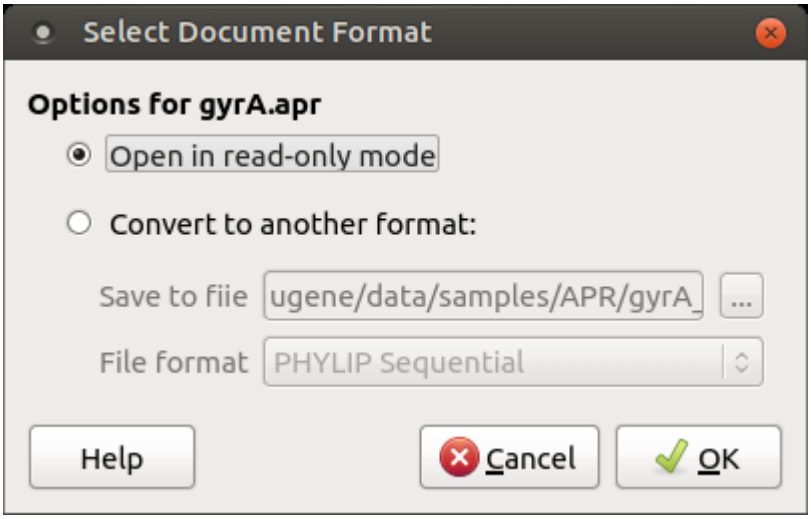


UGENE supports export to the BMP, CUR, DDS, ICNS, JP2, JPEG, JPG, PBM, PGM, PNG, PPM, TIF, TIFF, WBMP, WEBP, XBM, XPM and SVG image formats. You can export whole alignment or custom region. To select the custom region click on the *Select* button.

Importing APR and ACE Files

To start working with APR file you can open it in the Alignment Editor in read-only mode or convert it to another format.

To do this, *open* the *.apr file. The following dialog will appear:

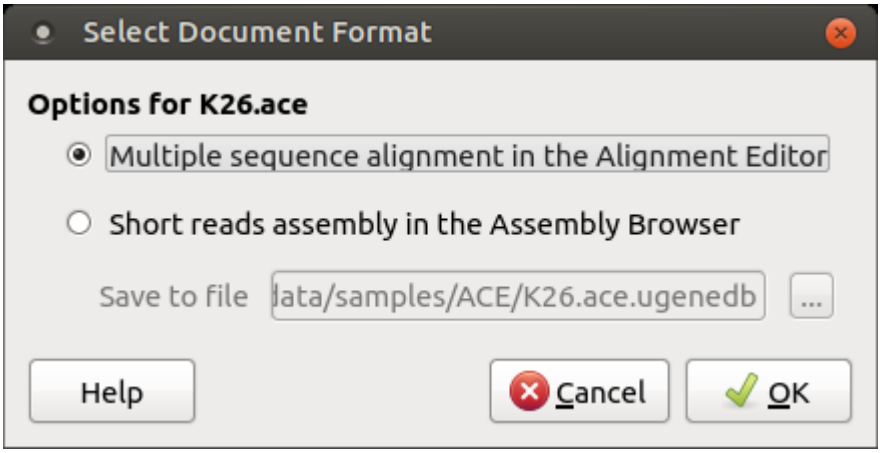


If you choose the first option the file will be opened in the Alignment Editor as multiple sequence alignment in read-only mode. If you choose the second option the file will be converted to a corresponding format and open in UGENE.

The following formats are available: CLUSTALW, FASTA, Mega, MSF, NEXUS, Phylip Interleaved, Phylip Sequential, Stockholm.

To start working with ACE file you can open it in the Alignment Editor or import it to the UGENE database file.

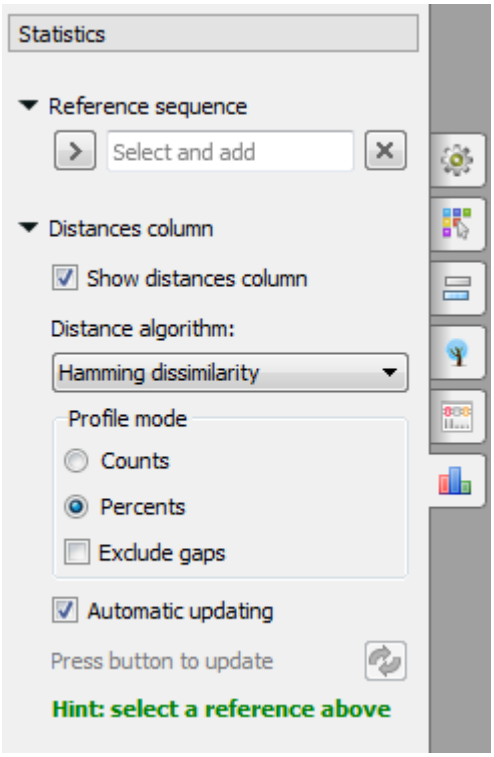
To do this, *open* the *.ace file. The following dialog will appear:



If you choose the first option the file will be opened in the Alignment Editor as multiple sequence alignment. If you choose the second option the file will be opened in Assembly Browser.

Statistics

To show statistics use the *Statistic* tab of the *Options Panel*:



Here you need to select a reference sequence.

You can show/hide distance column by *Show distances column* checkbox.

Also you can change the distance algorithm, select the profile mode and exclude gaps.

To update the column use *Automatic update* checkbox or press corresponding button to update.

To generate distance matrix and grid profile see the documentation below:

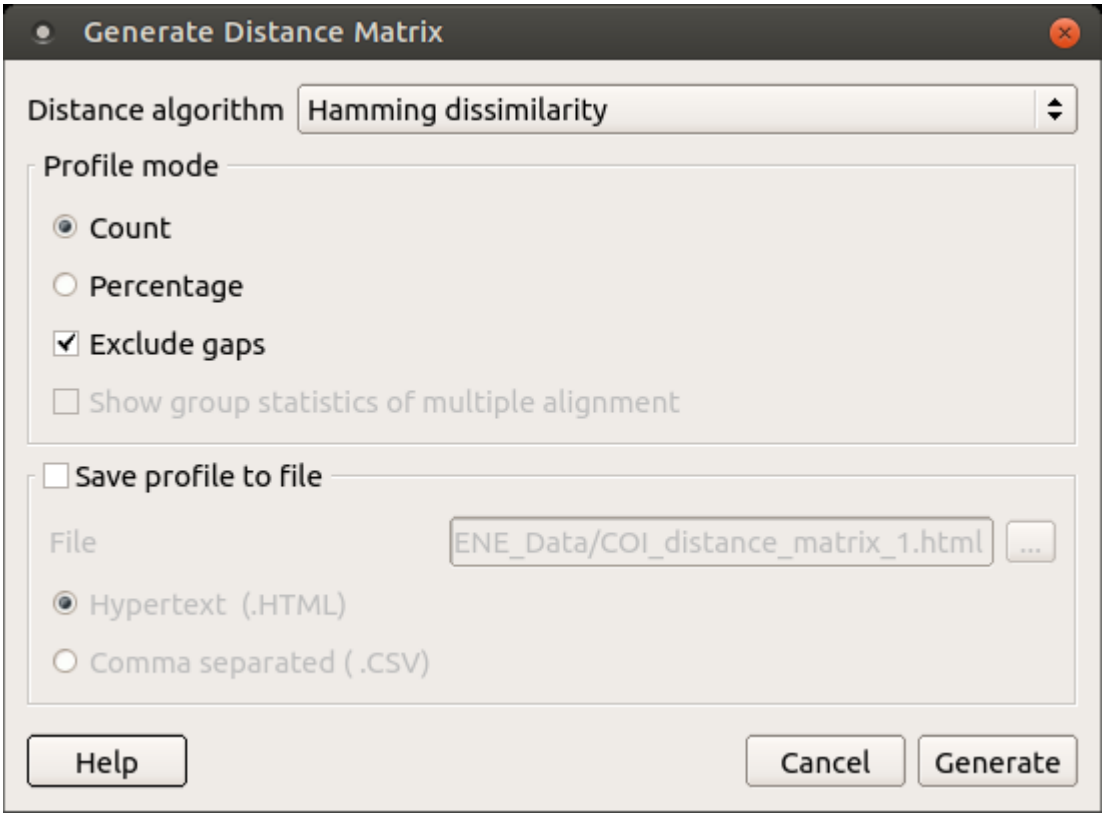
- Distance Matrix
- Grid Profile

Distance Matrix

Using the *Alignment Editor* you can also create a distance matrix of a multiple sequence alignment.

To create a distance matrix, use the *Statistics Generate distance matrix* item in the *Actions* main menu or in the context menu.

The dialog will appear:



The following parameters are available:

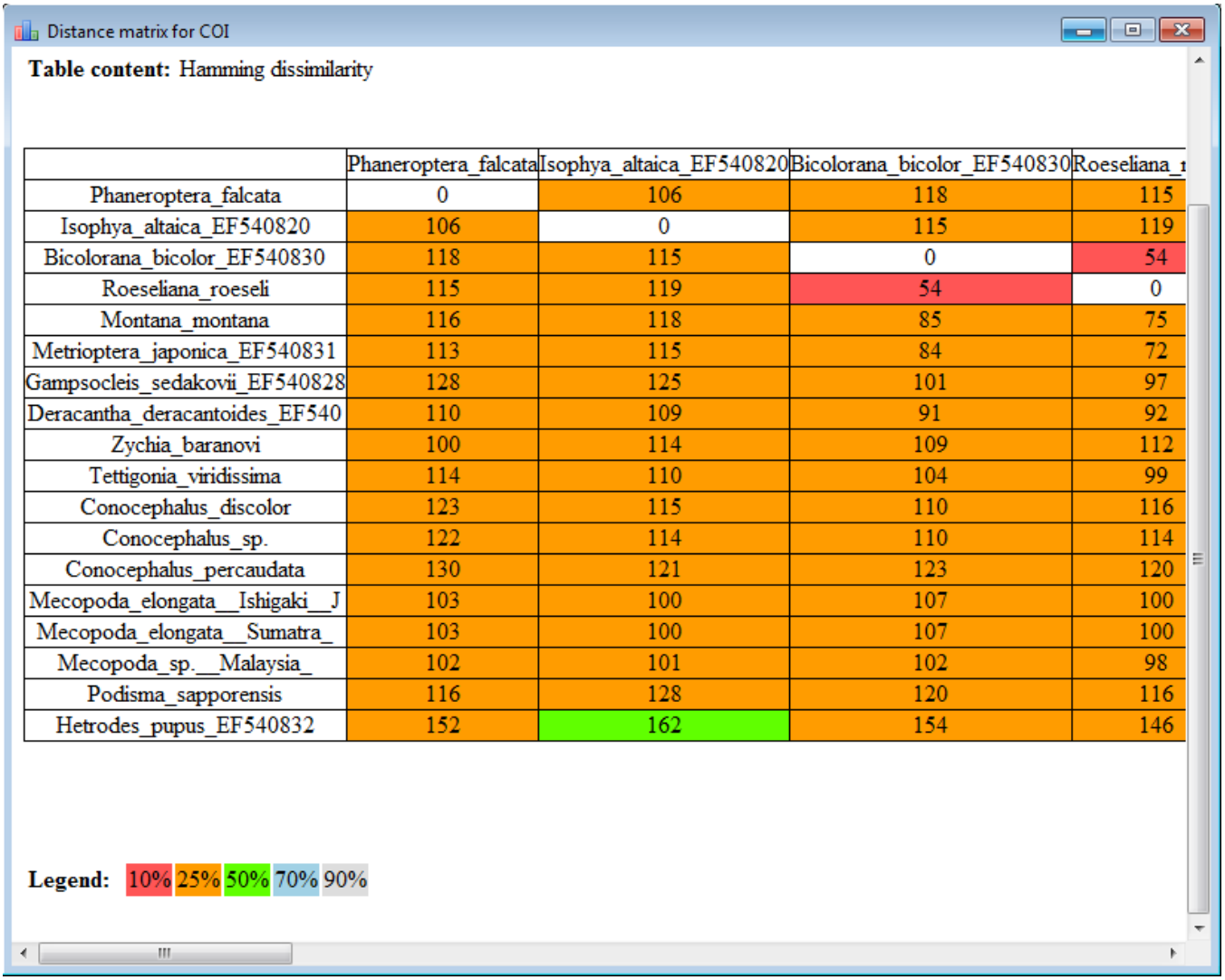
Distance algorithm - there are two distance algorithms: "Hamming distance" for dissimilarity and "Simple similarity" for similarity.

Profile mode: Counts/Percents — select the *Percents* to have scores shown as percents in the report. Also, you can *Exclude gaps*.

Show group statistics of multiple alignment - shows group statistics when the collapsing is switched on.

Save profile to file — allows saving the profile to a file in the HTML or CSV format. The CSV format is convenient for further processing in worksheets editors like Excel.

The resulting profile in the HTML mode:



	Phaneroptera_falcata	Isophya_altaica_EF540820	Bicolorana_bicolor_EF540830	Roeseliana_r
Phaneroptera_falcata	0	106	118	115
Isophya_altaica_EF540820	106	0	115	119
Bicolorana_bicolor_EF540830	118	115	0	54
Roeseliana_roeseli	115	119	54	0
Montana_montana	116	118	85	75
Metrioptera_japonica_EF540831	113	115	84	72
Gampsocleis_sedakovii_EF540828	128	125	101	97
Deracantha_deracantoides_EF540	110	109	91	92
Zychia_baranovi	100	114	109	112
Tettigonia_viridissima	114	110	104	99
Conocephalus_discolor	123	115	110	116
Conocephalus_sp.	122	114	110	114
Conocephalus_percaudata	130	121	123	120
Mecopoda_elongata__Ishigaki__J	103	100	107	100
Mecopoda_elongata__Sumatra_	103	100	107	100
Mecopoda_sp.__Malaysia_	102	101	102	98
Podisma_sapporensis	116	128	120	116
Hetrodes_pupus_EF540832	152	162	154	146

Grid Profile

Using the *Alignment Editor* you can create a statistic profile of a multiple sequence alignment.

The alignment grid profile shows positional amino acid or nucleotide counts highlighted according to the frequency of symbols in a row.

To create a grid profile, use the *Statistics Generate grid profile* item in the *Actions* main menu or in the context menu.

To learn more about this feature, refer to the *DNA Statistics* plugin documentation.

Advanced Functions

This chapter is devoted to the advanced functions of the *Alignment Editor*. You will learn how to build a grid profile, export a picture of an alignment and build HMM profiles.

- Building HMM Profile

Building HMM Profile

The editor has capabilities to build a Hidden Markov Model profile based on the multiple sequence alignment.

This functionality is based on the Sean Eddy's HMMER package.

To build a HMM profile select the *Advanced Build HMMER2 profile* or the *Advanced Build HMMER3 profile* item in the *Actions* main menu or in the context menu.

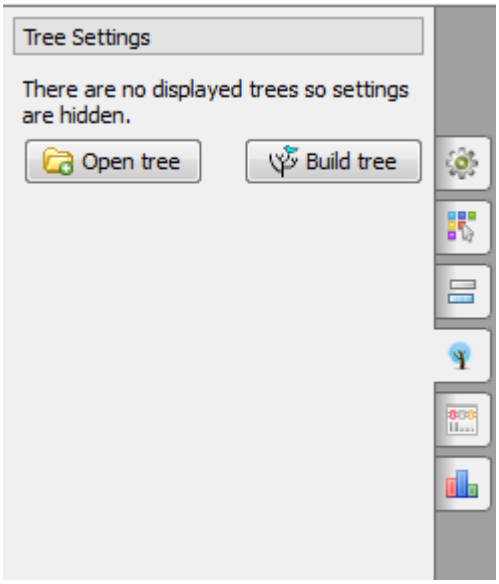
Learn more about the HMM tool in the documentation pages of the *HMM2* and the *HMM3* plugins.

Building Phylogenetic Tree

To build a tree from an alignment either press the *Build Tree* button on the toolbar, select the *Tree Build Tree* item in the alignment context menu or the *Actions Tree Build Tree* item in the main menu.



Also you can use *Tree Settings* tab of the *Options Panel*:



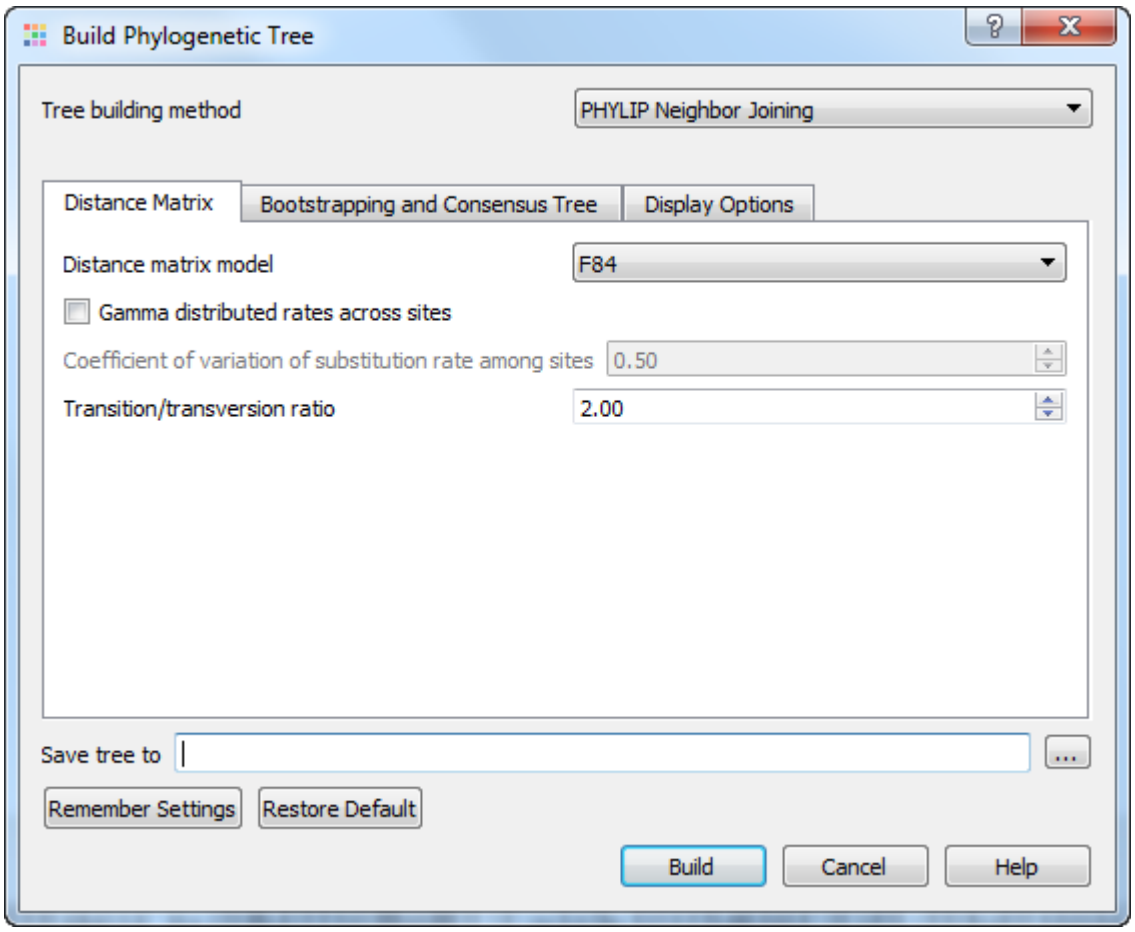
Three methods for building phylogenetic trees are supported:

1. The *PHYLIP Neighbour-Joining* method. The PHYLIP package implementation of the method is used under the hood.
2. The *MrBayes* external tool. Check MrBayes Web Site for more details.
3. PhyML Maximum Likelihood method. Check PhyML Maximum Likelihood Web Site for more details.

- PHYLIP Neighbor-Joining
- MrBayes
- PhyML Maximum Likelihood

PHYLIP Neighbor-Joining

The *Building Phylogenetic Tree* dialog for the *PHYLIP Neighbour-Joining* method has the following view:



The following parameters are available:

Distance matrix model — model to compute a distance matrix. The following values are available for a nucleotide multiple sequence alignment:

- F84
- Kimura
- Jukes-Cantor
- LogDet

The following models are available for a protein alignment:

- Jones-Taylor-Thornton
- Henikoff/Tillier PMB
- Dayhoff PAM
- Kimura

Gamma distributed rates across sites — specifies to take into account unequal rates of change at different sites. It is assumed that the distribution of the rates follows the Gamma distribution.

Coefficient of variation of substitution rate among sites — becomes available if the *Gamma distributed rates across sites* parameter is checked. Specifies the coefficient of the distribution of the rates.

Transition/transversion ratio — expected ratio of transitions to transversions.

To enable bootstrapping check the *Bootstrapping and Consensus Trees* group check box. The following parameters are available:

Number of replicates — number of replicate date sets.

Seed — random number seed. By default, it is generated automatically. You can manually change this value in order to make results of different runs (of a tree building) reproducible. The should must be an integer greater than zero and less than 32767 and which is of the form 4n+1, that is, it leaves a remainder of 1 when divided by 4. Any odd number can also be used, but may result in a random number sequence that repeats itself after less than the full one billion numbers. Usually this is not a problem.

Consensus type — specifies the method to build the consensus tree. Select one of the following:

- *Strict* — specifies that a set of species must appear in all input trees to be included in the strict consensus tree.

- *Majority Rule (extended)* — specifies that any set of species that appears in more than 50% of the trees is included. The program then considers the other sets of species in order of the frequency with which they have appeared, adding to the consensus tree any which are compatible with it until the tree is fully resolved. This is the default setting.
- *M1* — includes in the consensus tree any sets of species that occur among the input trees more than a specified fraction of the time (see the *Fraction* parameter below). The *Strict* consensus and the *Majority Rule* consensus are extreme cases of the MI consensus, being for fractions of 1 and 0.5 respectively.
- *Majority Rule* — specifies that a set of species is included in the consensus tree if it is present in more than half of the input trees.

Fraction — becomes available when the *Consensus type* parameter is set to *M1*. Specifies the fraction.

Display tree in new window - displays tree in new window.

Display tree with alignment editor - displays tree with alignment editor.

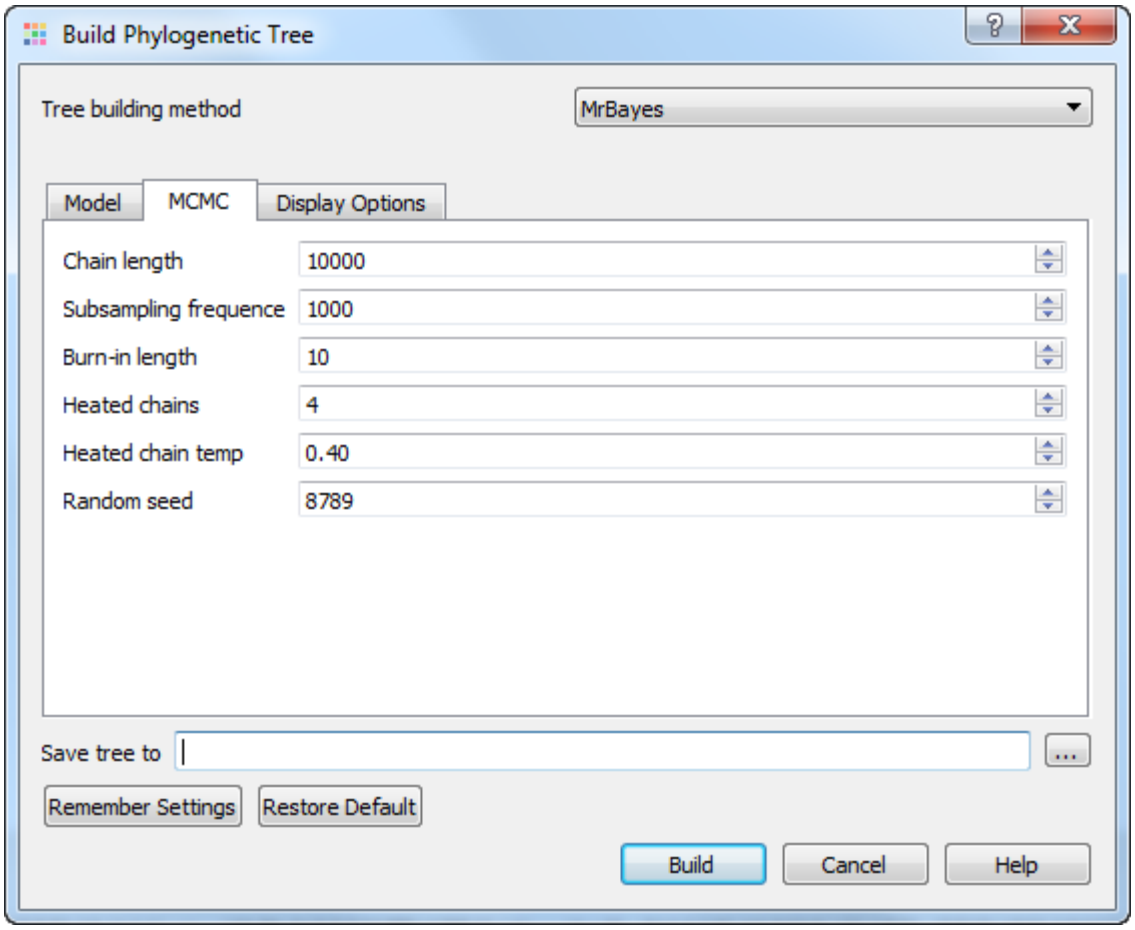
Synchronize alignment with tree - synchronize alignment and tree.

Save tree to — file to save the tree built.

Press the *Build* button to build a tree with the parameters selected.

MrBayes

The *Building Phylogenetic Tree* dialog for the *MrBayes* method has the following view:



There are two steps to a phylogenetic analysis using *MrBayes*:

1. Set the evolutionary model.
2. Run the Markov chain Monte Carlo (MCMC) analisys.

The evolutionary model is defined by the following parameters:

Substitution model — specifies the general structure of a DNA substitution model. This parameter is available for the nucleotide sequences. It corresponds to the Nst setting of MrBayes. You may select one of the following:

- JC69 (Nst=1)
- HKY85 (Nst=2)
- GTR (Nst=6)

Rate matrix (fixed) — specifies the fixed-rate amino-acid model. This parameter is available for amino-acid sequences. The following models are available:

- poisson
- jones
- dayhoff
- mtrev
- mtmam
- wag
- rtrev
- cprev
- vt
- blosum
- equaline

The following parameters are common for nucleotide and amino-acid sequences:

Rate — sets the model for among-site rate variation. Select one of the following:

- equal — no rate variation across sites.
- gamma — gamma-distributed rates across sites. The rate at a site is drawn from a gamma distribution. The gamma distribution has a single parameter that describes how much rates vary.
- propinv — a proportion of the sites are invariable.
- invgamma — a proportion of the sites are invariable while the rate for the remaining sites are drawn from a gamma distribution.

Gamma — sets the number of rate categories for the gamma distribution.

You can select the following parameters for the MCMC analisys:

Chain length — sets the number of cycles for the MCMC algorithm. This should be a big number as you want the chain to first reach stationarity, and then remain there for enough time to take lots of samples.

Subsampling frequency — specifies how often the Markov chain is sampled. You can sample the chain every cycle, but this results in very large output files.

Burn-in length — determines the number of samples that will be discarded when convergence diagnostics are calculated.

Heated chains — number of chains will be used in Metropolis coupling. Set 1 to use usual MCMC analysis.

Heated chain temp — the temperature parameter for heating the chains. The higher the temperature, the more likely the heated chains are to move between isolated peaks in the posterior distribution.

Random seed — a seed for the random number generator.

Display tree in new window - displays tree in new window.

Display tree with alignment editor - displays tree with alignment editor.

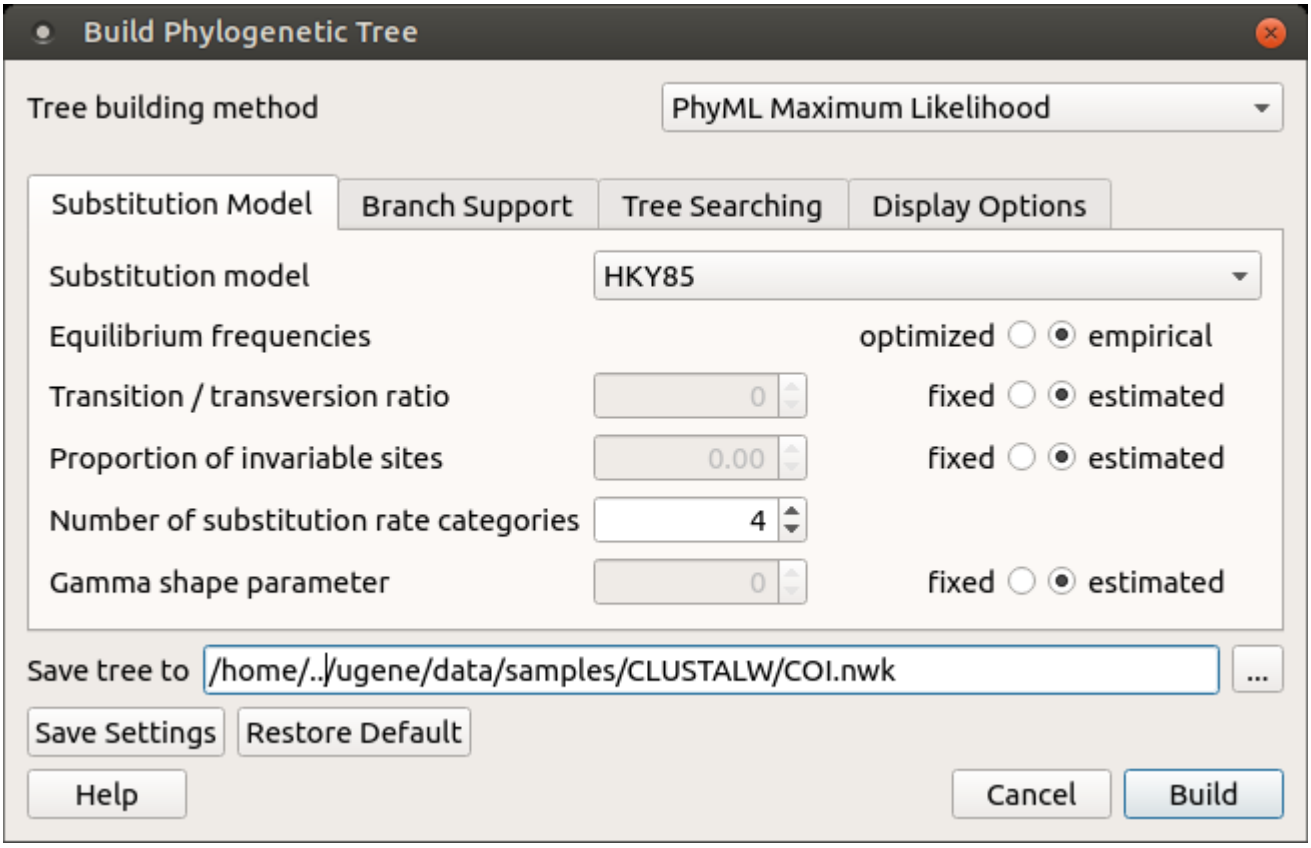
Synchronize alignment with tree - synchronize alignment and tree.

Save tree to — file to save the built tree.

Press the *Build* button to run the analysis with the parameters selected and build a consensus tree.

PhyML Maximum Likelihood

The *Building Phylogenetic Tree* dialog for the *PhyML Maximum Likelihood* method has the following view:



The following parameters are available:

Substitution model parameters - a selection of the Markov model of substitution:

Substitution model - model of substitution.

Equilibrium frequencies - equilibrium frequencies.

Transition/transversion ratio - fix or estimate the transition/transversion ratio in the maximum likelihood framework.

Proportion of invariable sites - the proportion of invariable sites, i.e., the expected frequency of sites that do not evolve, can be fixed or estimated.

Number of substitution rate categories - the number of substitution rate categories.

Gamma shape parameter - the shape of the gamma distribution determines the range of rate variation across sites.

Branch support parameters - the selection of the method that is used to measure branch support:

Use fast likelihood method - use fast likelihood method.

Perform bootstrap - the support of the data for each internal branch of the phylogeny can be estimated using non-parametric bootstrap.

Tree searching parameters - the selection of the tree topology searching algorithm:

Make initial tree automatically - initial tree automatically.

Type of tree improvement - type of tree improvement.

Set number of random starting tree - the number of random starting tree.

Optimize tree topology - the tree topology is optimised in order to maximise the likelihood.

Optimize branch lengths - optimize branch lengths.

Optimise substitution rate - substitution rate parameters are optimized.

Display tree in new window - displays tree in the new window.

Display tree with alignment editor - displays tree with alignment editor.

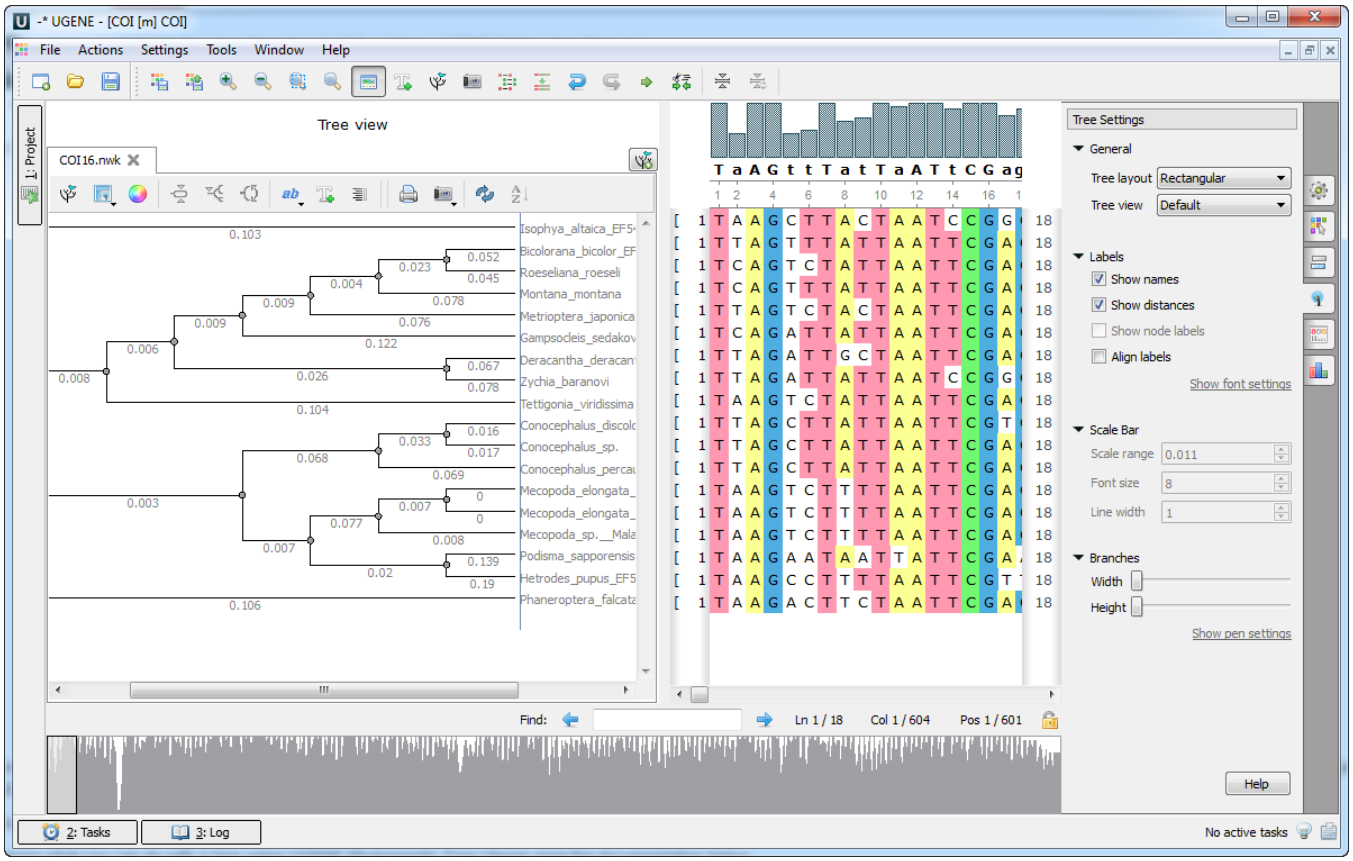
Synchronize alignment with tree - synchronize alignment and tree.

Save tree to - file to save the built tree.

Press the *Build* button to run the analysis with the parameters selected and build a consensus tree.

Phylogenetic Tree Viewer

The *Phylogenetic Tree Viewer* is intended to display a phylogenetic tree built from an alignment or loaded from a file (e.g. a Newick file).



To load a tree from a file follow the instruction described in the *Opening Document* paragraph or use the *Tree settings* tab of the *Options Panel*. For example, you may open the $UGENE\data\samples\Newick\COI.nwk sample file provided within UGENE package.

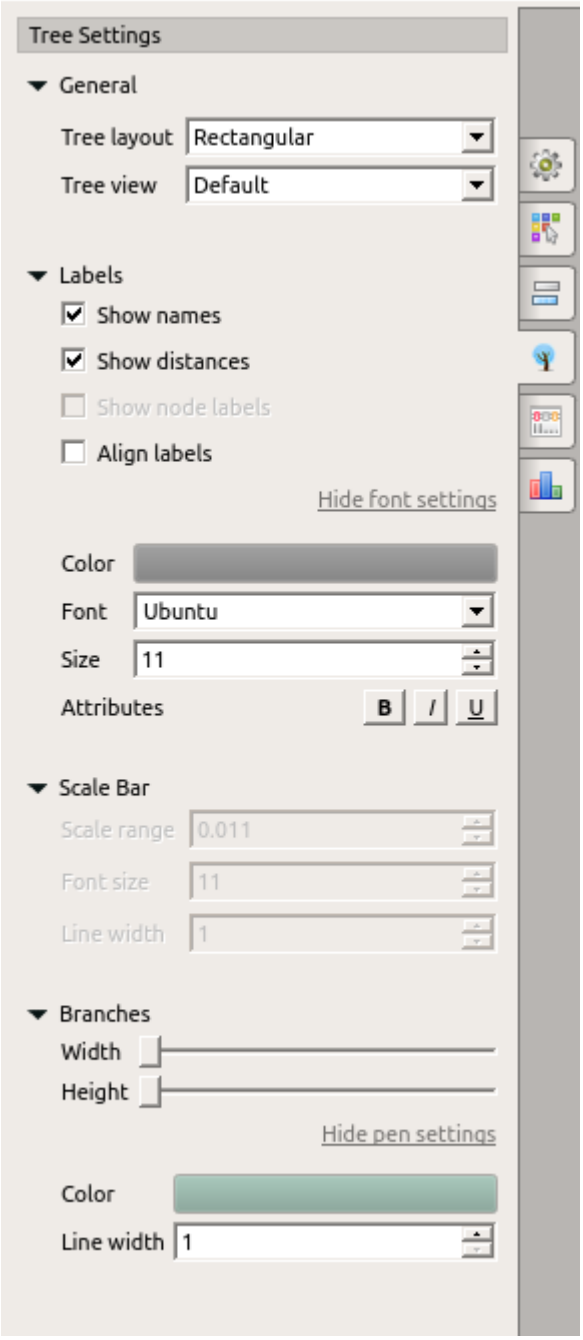
To build a tree from a multiple sequence alignment see the *Building Phylogenetic Tree* paragraph.

To learn what you can do with a tree using UGENE Phylogenetic Tree Viewer read the documentation below.

- Tree Settings
  - Selecting Tree Layout and View
  - Modifying Labels Appearance
    - Showing and Hiding Labels
    - Aligning Labels
    - Changing Labels Formatting
  - Adjusting Branch Settings
- Zooming Tree
- Working with Clade
  - Selecting Clade
  - Collapsing and Expanding Branches
  - Swapping Siblings
  - Zooming Clade
  - Adjusting Clade Settings
  - Changing Root
- Exporting Tree Image
- Printing Tree

Tree Settings

To adjust a tree settings select either the *Tree Settings* toolbar button or the *Tree settings* tab of the *Options Panel*. The *Tree settings* tab:



Detailed information about tree setting see below:

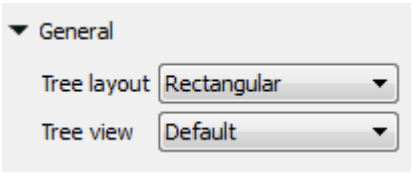
- Selecting Tree Layout and View
- Modifying Labels Appearance
  - Showing and Hiding Labels
  - Aligning Labels
  - Changing Labels Formatting
- Adjusting Branch Settings

Selecting Tree Layout and View

You can select one of the following tree layouts:

- *Rectangular*
- *Circular*
- *Unrooted*

To do it press the *Layout* toolbar button and check the required item in the appeared menu or select it in the *Tree settings Options Panel* tab:



See the example of the *Circular* layout:



Also you can select one of the following tree view:

- *Default*
- *Phylogram*
- *Cladogram*

Modifying Labels Appearance

From this paragraph you can learn how to show/hide taxon and distance labels, align them and change their formatting (font, color, etc.).

- Showing and Hiding Labels
- Aligning Labels
- Changing Labels Formatting

Showing and Hiding Labels

When you open a tree all available labels are shown by default.

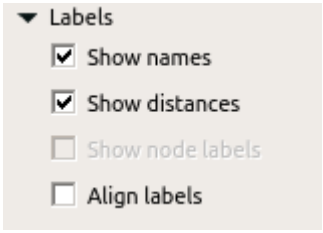
To hide the taxon (sequence name) labels select the *Show labels* toolbar button or in the *Tree settings Options Panel* tab uncheck the *Show names* item.

To hide the distance labels uncheck the *Show distances* item.

To show/hide node labels (if it is available) use the *Show node labels* item.

To show the labels again check an appropriate item.

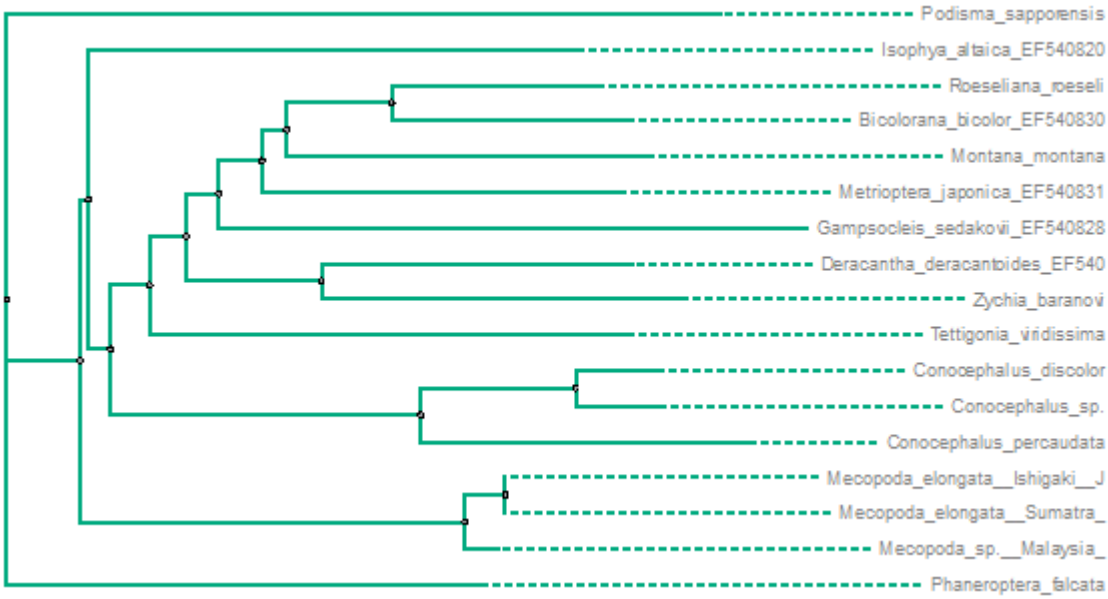
Labels settings in the *Options Panel*:



Aligning Labels

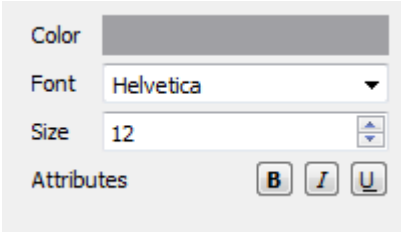
To align a tree labels press the *Align labels* toolbar button or in the *Tree settings Options Panel* tab check the *Align label* item.

See the example of aligning labels below:



Changing Labels Formatting

To change formatting of a tree labels select the *Labels Formatting* toolbar button or the *Tree settings Options Panel* tab:



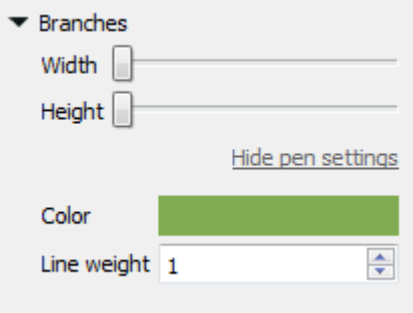
Here you can select color, font, size and attributes (bold, italic, etc.) of the labels.

Note that when *a clade has been selected* the labels formatting settings are applied to the clade only.

Adjusting Branch Settings

To adjust branch settings select the *Branch Settings* toolbar button, the *Branch Settings* context menu item or the *Tree settings Options Panel* tab.

The following settings are available:



Here you can select the color and the line width of the tree branches.

Note that when *a clade has been selected* the branch settings are applied to the clade only.

Zooming Tree

To change the size of a tree use the *Zoom In* and *Zoom Out* toolbar button. You can use the *Restore Zooming* toolbar button to set the default size.

Or use the corresponding items in the *Actions* main menu.

See also: *Zooming Clade*.

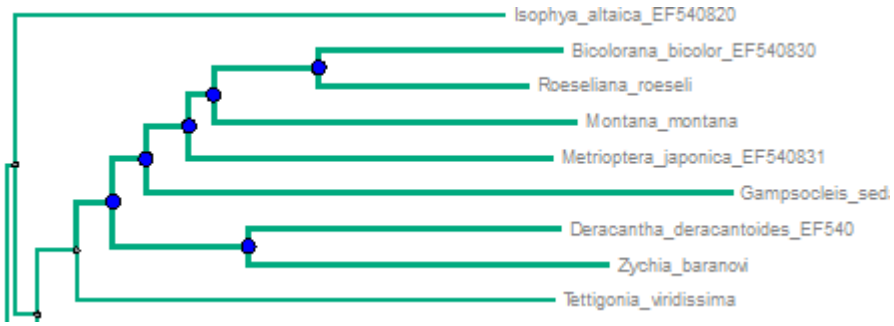
Working with Clade

This paragraph describes how to select a clade and modify it's appearance.

- Selecting Clade
- Collapsing and Expanding Branches
- Swapping Siblings
- Zooming Clade
- Adjusting Clade Settings
- Changing Root

Selecting Clade

To select a clade click on it's root node:

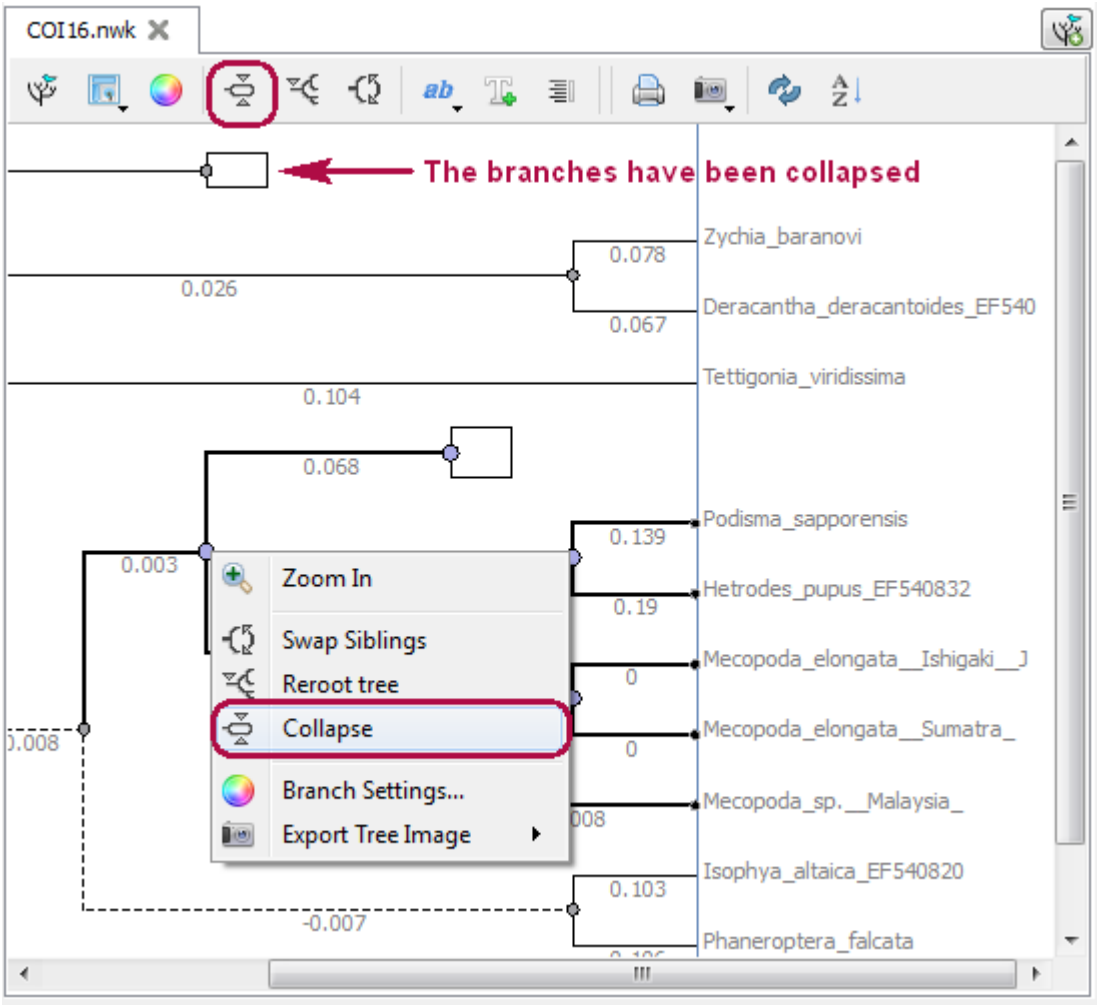


You can see that the corresponding branches are highlighted.

To select several clades at the same time hold the Shift key and click on the root nodes of the clades.

Collapsing and Expanding Branches

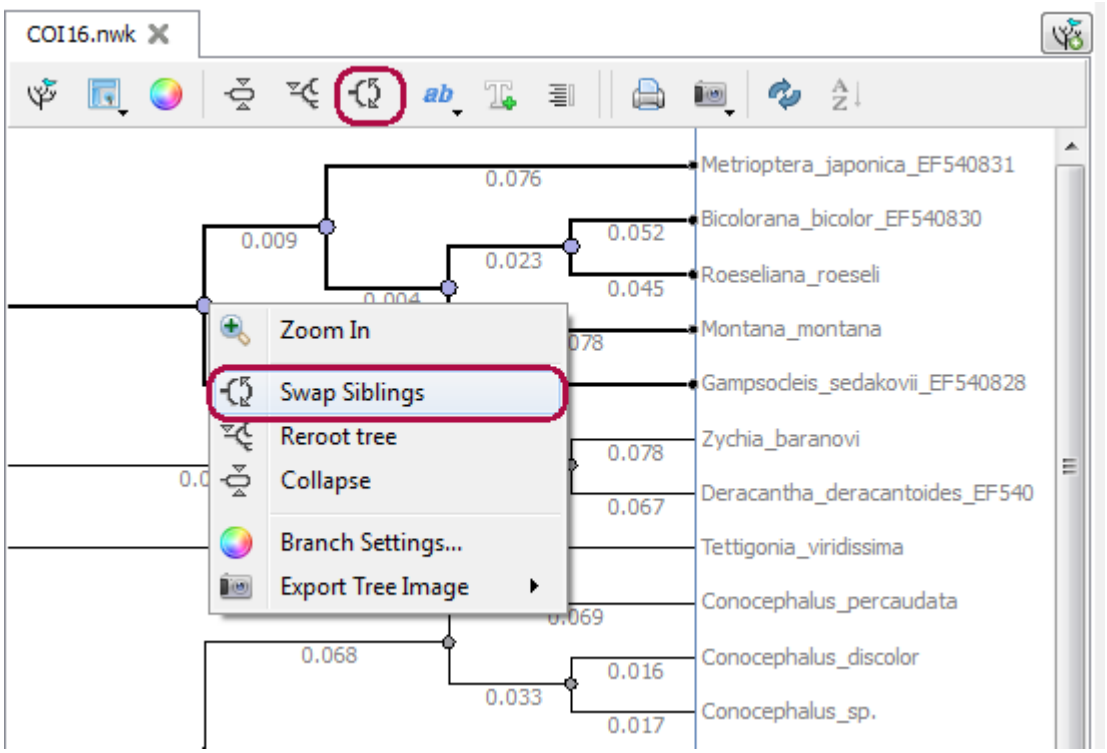
You can hide branches of a clade by selecting the *Collapse* item in the context menu of the clade's root node or use the *Collapse* button on the tree toolbar:



To show the collapsed clade select the *Expand* item in the node's context menu.

Swapping Siblings

To rearrange two branches of an internal node, select the *Swap Siblings* item in the node context menu or click the *Swap Siblings* button on the tree toolbar, while the node is selected:



Zooming Clade

Additionally to other *zooming options* you can use the *Zoom In* item in the context menu of the root node of a clade.

Adjusting Clade Settings

When a clade is selected the *branch* and the *labels formatting* settigns are applied to the clade only.

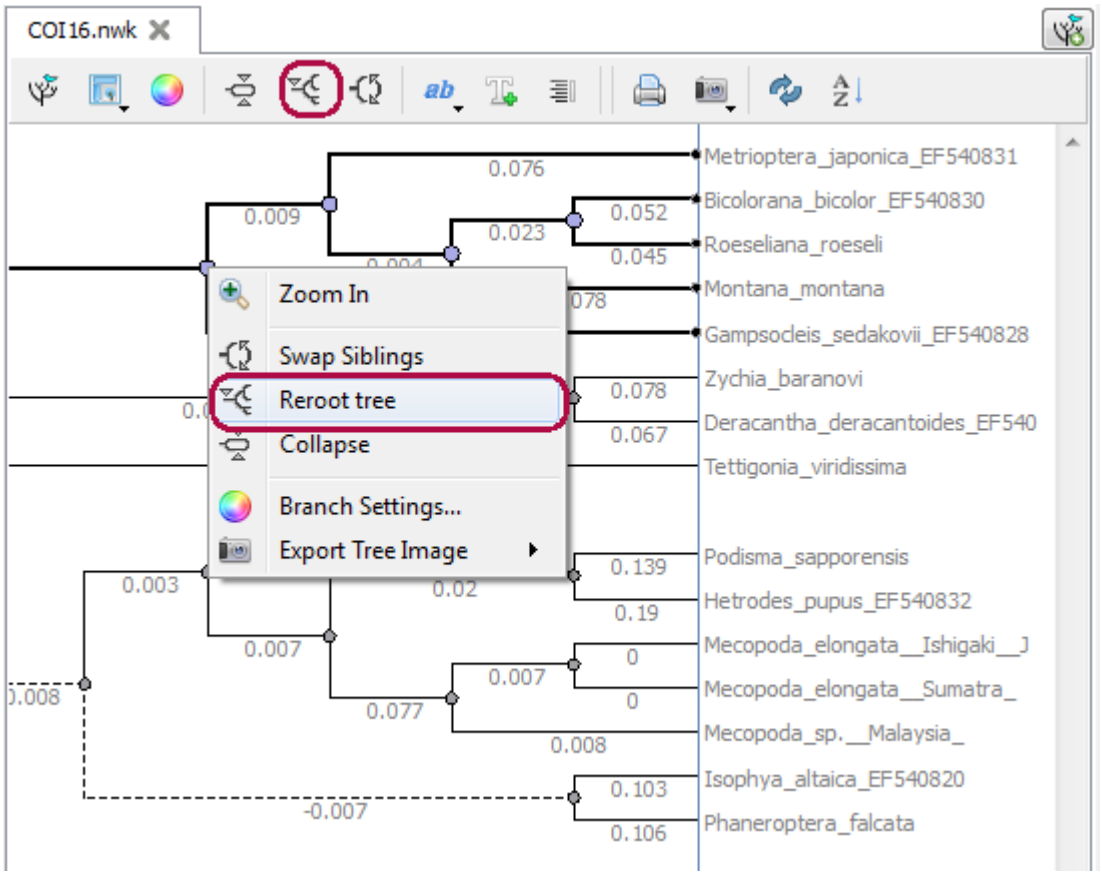
Note that the settings are not applied to the *collapsed* branches (if any).

See an example of changing branch settings for a clade:



Changing Root

To change root of a tree select the root and call the *Reroot tree* context menu item or use the *Reroot tree* button on the tree toolbar:



Exporting Tree Image

A tree image can be exported to a raster format (.png, .jpg, .bmp, etc.) or to a vector format (.svg).

Select either the *Export Tree Image* toolbar button or the *Actions Export Tree Image* item in the main menu.

In the submenu appeared select the *Screen Capture* item to save the tree image to a raster format. The standard *Save As* dialog will appear where you can select the file name and format.

To export a tree image to a vector format select the *As SVG* item in the *Export Tree Image* submenu.

Printing Tree

To print a tree select either the *Print Tree* toolbar button or the *Actions Print Tree* item in the main menu.

The standard print dialog will appear where you can select a printer to use and specify other settings.

Sanger Reads Editor

- Sanger Reads Editor Overview
  - Sanger Reads Editor Features
  - Sanger Reads Editor Components
- Working with Chromatogram
  - Mapping Reads to Reference
  - Alignment Statistics
  - Alignment Appearance
  - Overview and Show and Hide Chromatogram
  - Sanger Reads Consensus
    - Export Chromatogram Consensus
  - Navigation in Sanger Reads Alignment
  - Editing Sanger Reads
    - Inserting Character
    - Replacing Character and Gap
    - Removing Character and Gap
    - Inserting Gap
    - Removing Gap at the Left
    - Removing All Columns of Gaps
    - Trimming Left End

        - Trimming Right End
        - Renaming Read
        - Removing Read
        - Undo and Redo
    - Exporting Alignment without Chromatograms

Sanger Reads Editor Overview

This chapter gives an overview of the *Sanger Reads Editor* components and explains basic concepts of browsing a chromatogram.

- Sanger Reads Editor Features
- Sanger Reads Editor Components

Sanger Reads Editor Features

Sanger Reads Editor is dedicated to DNA sequence analysis and manipulation. You can trim, map to reference and view Sanger sequencing trace files and create consensus sequences.

The editor provides interactive visual representation which includes:

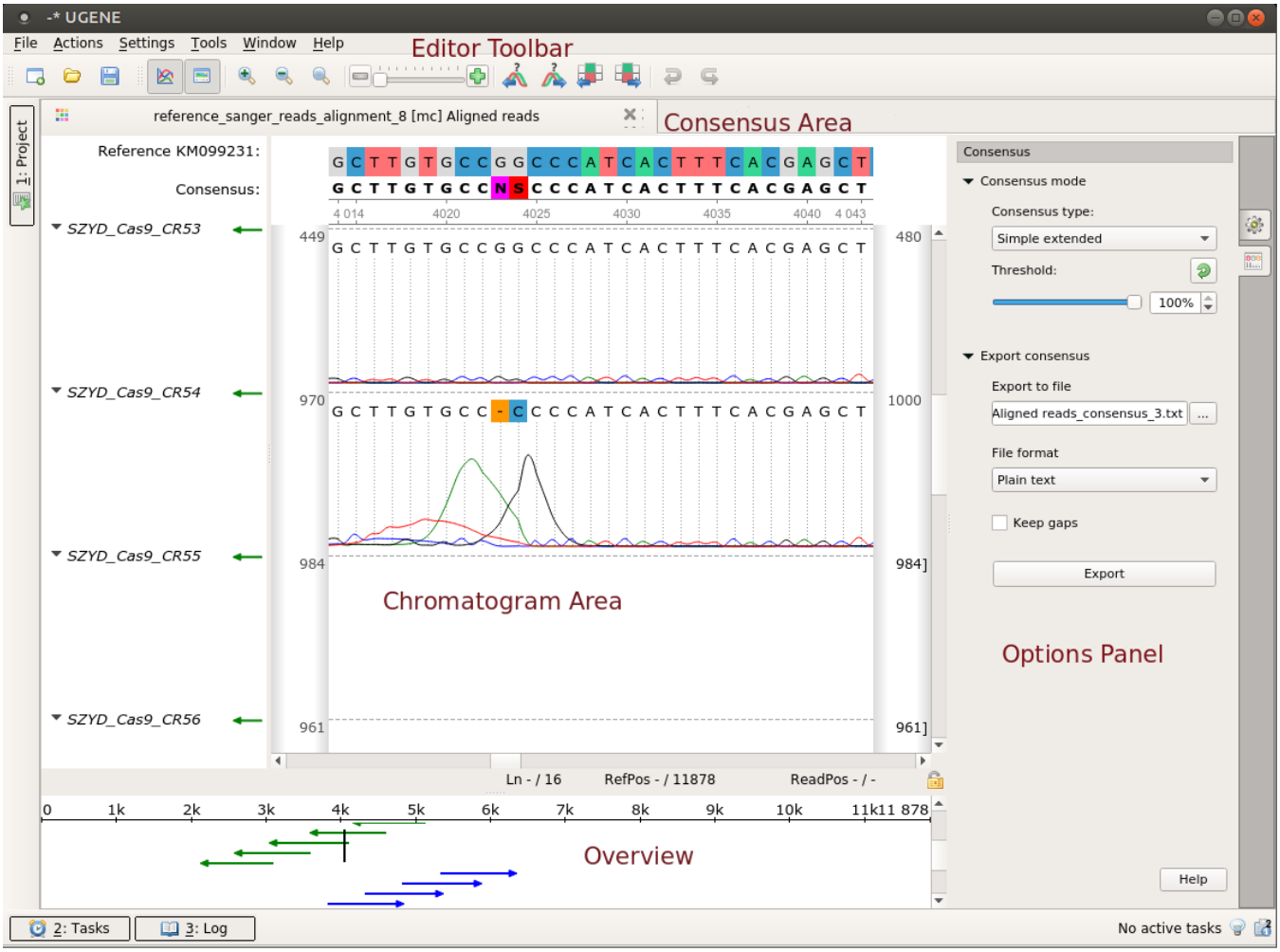
- Navigation through an chromatogram;
- Flexible zooming;
- Several consensus calculation algorithms.

Using the *Sanger Reads Editor* you can:

- Export alignment without chromatogram;
- Edit chromatogram: remove/replace/trim.

Sanger Reads Editor Components

Here is the default layout of the editor:



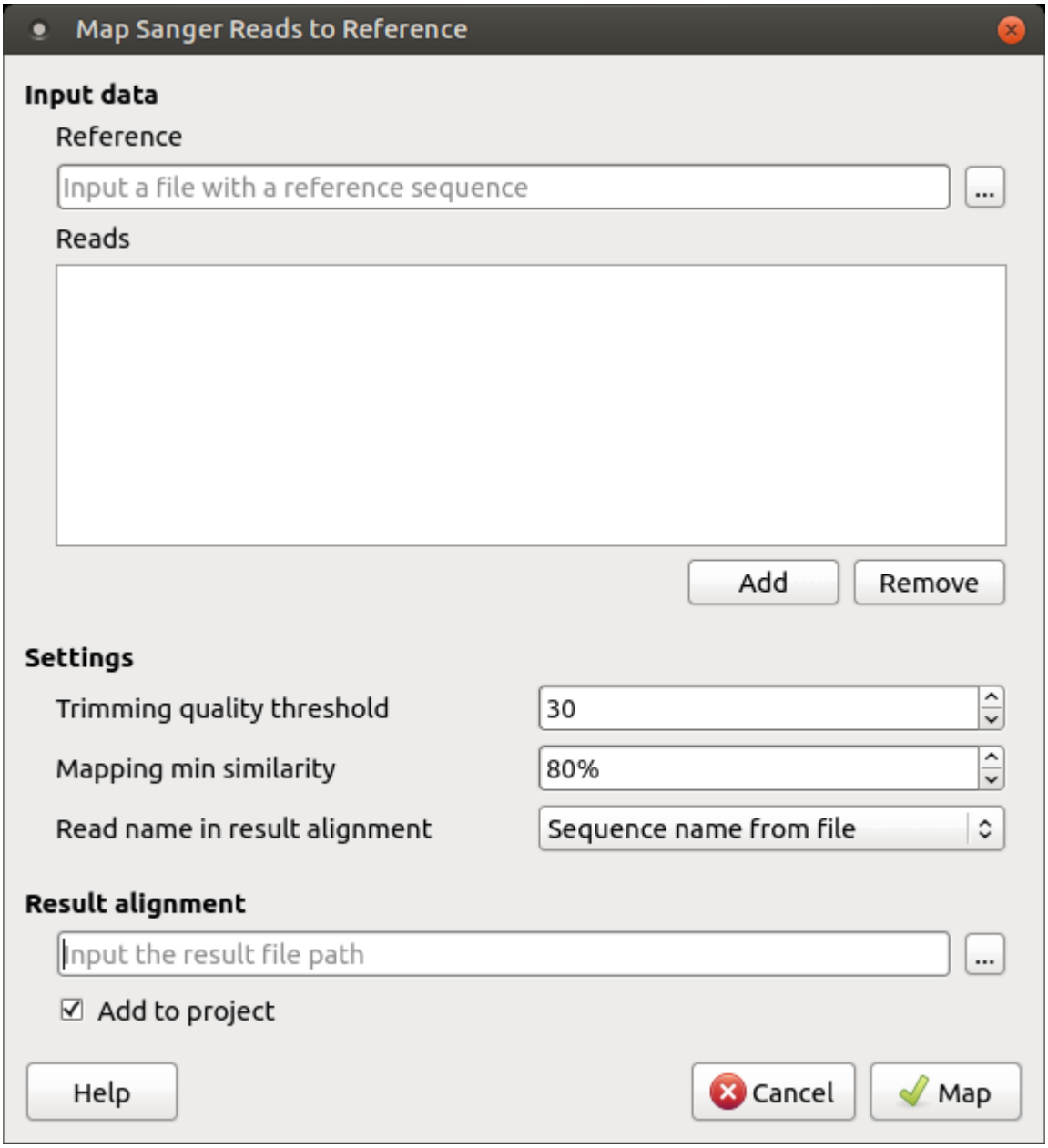
Working with Chromatogram

This chapter explains how to work efficiently with the *Sanger Reads Editor*.

- Mapping Reads to Reference
- Alignment Statistics
- Alignment Appearance
- Overview and Show and Hide Chromatogram
- Sanger Reads Consensus
  - Export Chromatogram Consensus
- Navigation in Sanger Reads Alignment
- Editing Sanger Reads
  - Inserting Character
  - Replacing Character and Gap
  - Removing Character and Gap
  - Inserting Gap
  - Removing Gap at the Left
  - Removing All Columns of Gaps
  - Trimming Left End
  - Trimming Right End
  - Renaming Read
  - Removing Read
  - Undo and Redo
- Exporting Alignment without Chromatograms

Mapping Reads to Reference

To map Sanger sequencing reads to a reference use the *Tools–> Sanger data analysis–> Map reads to reference* main menu item. The following dialog appears:



It is required to input:

Reference sequence — a file with a single DNA sequence of any supported file format (e.g. FASTA or GenBank). This parameter is mandatory.

Reads — a set of files in *.ab1 or *.scf format with the Sanger sequencing data to be mapped to the reference sequence. It is required to add at least one read. A read orientation (forward or reverse) will be automatically detected during the mapping.

During the mapping task execution, these data are processed as follows. First of all, to enhance the further mapping sensitivity, the low-quality ends of the reads are trimmed off. Then the reads are mapped using a UGENE original method that works in two steps: rough mapping of a read using BLAST+ and enhancement of the read alignment with the Smith-Waterman algorithm. After that the percentage similarity of the two sequences is calculated, that is the number of all edit operations (i.e., insertions, deletions, and substitutions) required to transform a read sequence into the corresponding region of the reference sequence. Reads that have low similarity with the reference sequence are filtered out.

One can configure the following parameters of the mapping task

Trimming quality threshold — all bases at the ends of the reads with quality lower than the specified value are trimmed. To skip the trimming, set this value to zero.

Mapping min similarity — all reads mapped to the reference with lower percentage similarity than the specified value are filtered out.

Read name in result alignment — reads in the resulting alignment can be named either by names of the sequences in the input files or by the input files names. Set this value to "File name", for example, if the sequences in the input *.ab1 files have the same name, this will help in distinguishing of the reads in the resulting alignment.

The resulting alignment is stored in a native UGENEDB format. One can set up the file location and name in the *Result alignment* field. Note that thereafter it is also possible to export data to standard alignment formats without chromatograms such as FASTA, ClustalW, etc.

To initiate the mapping task execution click the *Map* button in the dialog. Note that when the task is finished, the task statistics can be found in a report, available by clicking the corresponding notification:

Task report [Map Sanger reads to reference]

Status	Finished
Time	0h 00m 03.652s

Details

Reference sequence: KM099231

Mapped reads (16):

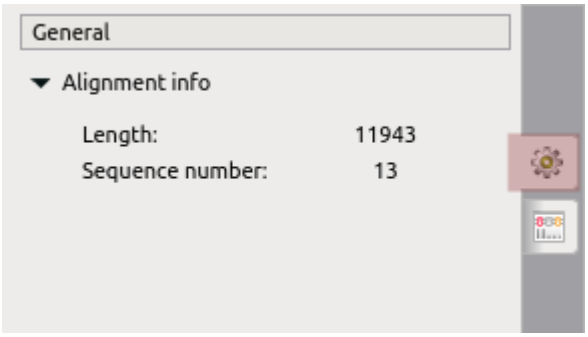
←	SZYD_Cas9_5B70	similarity 99%
→	SZYD_Cas9_5B71	similarity 93%
←	SZYD_Cas9_CR50	similarity 98%
←	SZYD_Cas9_CR51	similarity 98%
←	SZYD_Cas9_CR52	similarity 99%
←	SZYD_Cas9_CR53	similarity 99%
←	SZYD_Cas9_CR54	similarity 99%
←	SZYD_Cas9_CR55	similarity 98%
←	SZYD_Cas9_CR56	similarity 99%
→	SZYD_Cas9_CR60	similarity 99%
→	SZYD_Cas9_CR61	similarity 98%
→	SZYD_Cas9_CR62	similarity 99%
→	SZYD_Cas9_CR63	similarity 99%
→	SZYD_Cas9_CR64	similarity 99%
→	SZYD_Cas9_CR65	similarity 100%
→	SZYD_Cas9_CR66	similarity 98%

Filtered by low similarity (4):

	SZYD_Cas9_CR49	similarity 0%
	SZYD_Cas9_CR57	similarity 58%
	SZYD_Cas9_CR58	similarity 0%
	SZYD_Cas9_CR59	similarity 48%

Alignment Statistics

To show statistics use the *General* tab of the *Options Panel*:



Alignment Appearance

To perform zoom operations use the corresponding buttons on the editor toolbar.

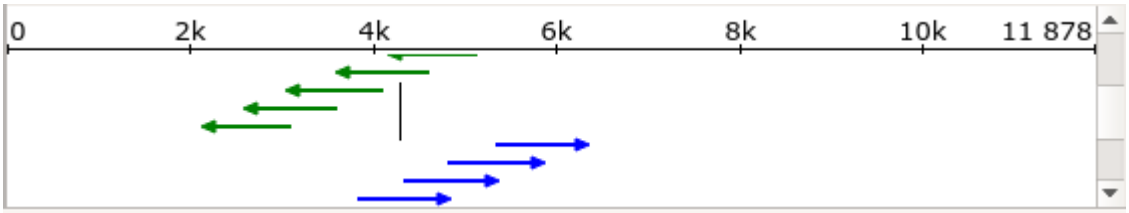


You can zoom in and zoom out by the corresponding toolbar buttons. To reset zoom use the *Reset zoom* button. To increase/decrease peaks height use the corresponding buttons.

You can change font by *Appearance–>Change characters font* context menu item or by the main *Actions–>Appearance* menu item.

Overview and Show and Hide Chromatogram

The chromatogram overview is shown automatically in the *Sanger Reads Editor*. To close the overview click on the *Show overview* toolbar button. Drag and drop vertical cursor to change the current position.



To show/hide chromatogram click on the corresponding toolbar button:



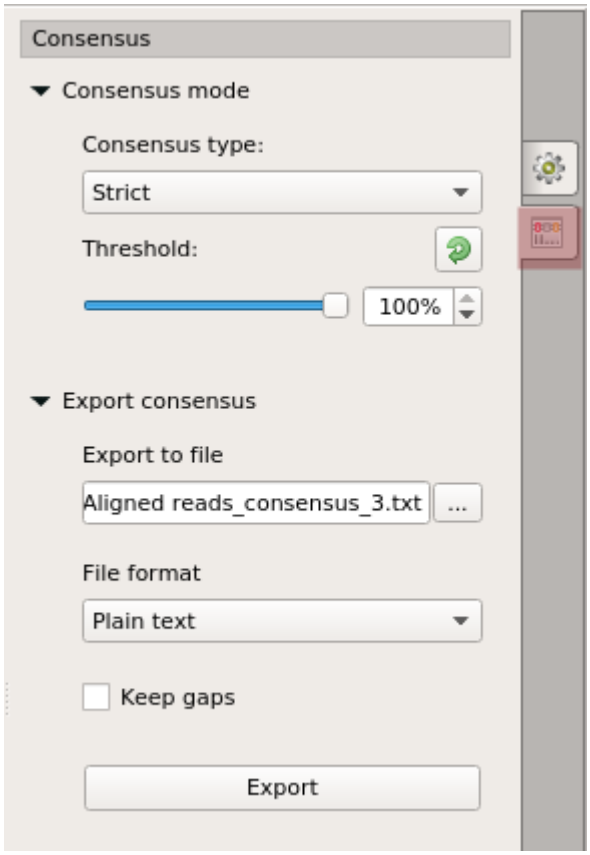
Sanger Reads Consensus

Each base of a consensus sequence is calculated as a function of the corresponding column bases. The *Sanger Reads Editor* allows switching between different consensus modes: Simple extended and Strict.

The Simple extended algorithm selects the best character from the extended DNA alphabet. Only bases with frequences which are greater than a threshold value are taken into account.

The Strict algorithm returns gap character ('-') if symbol frequency in a column is lower than threshold specified.

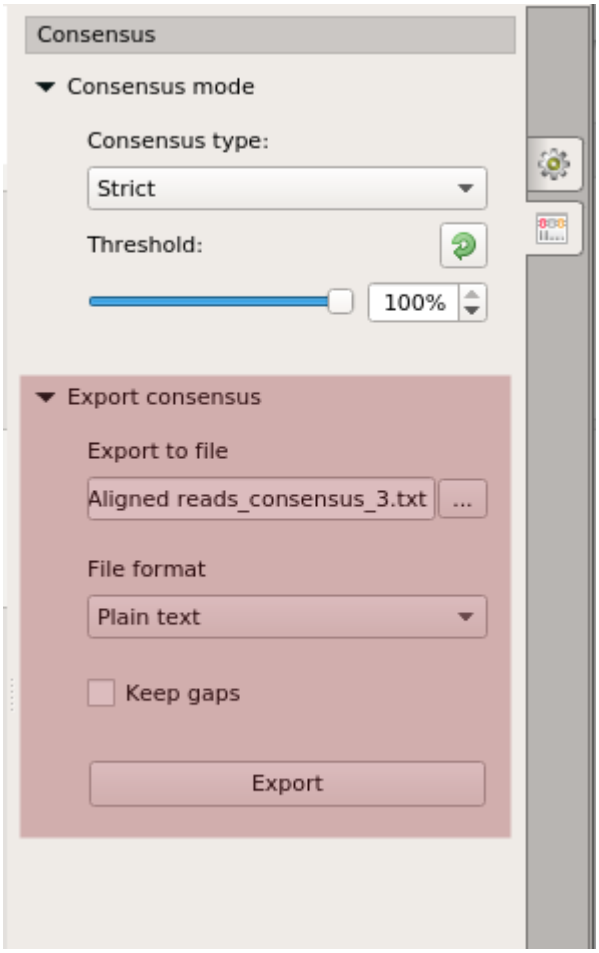
To switch the consensus mode go to the *Consensus tab* of the *Options Panel*:



- Export Chromatogram Consensus

Export Chromatogram Consensus

To export consensus sequence use the *Consensus* tab of the *Options Panel*:



The following parameters are available:

Export to file - here you need to select path for the output file.

File format - format for the output file.

Keep gaps - set this parameter if the result consensus must keep gaps.

When you click on the *Export* button the consensus sequence will be exported into selected output file.

Navigation in Sanger Reads Alignment

You may jump to previous/next ambiguous character in the reads alignment and edit the characters.

After you finish the editing, use the navigation by previous/next variations to find differences between the reference sequence and the consensus.

Use the main *Actions–> Navigation* menu, context menu or the following toolbar buttons:



Editing Sanger Reads

Select the *Edit* submenu in the *Sanger Reads Editor* context menu.

The actions available from this menu are described below:

- Inserting Character
- Replacing Character and Gap
- Removing Character and Gap
- Inserting Gap
- Removing Gap at the Left
- Removing All Columns of Gaps

- Trimming Left End
- Trimming Right End
- Renaming Read
- Removing Read
- Undo and Redo

Inserting Character

To insert a nucleotide into a read in the *Sanger Reads Editor*, select a nucleotide or a gap in the corresponding location in the *Chromatogram Area* and switch to the editing mode by pressing *Shift+I*, selecting *Edit–>Insert character/gap* in the context menu, or selecting *Actions->Edit->Insert character/gap* in the main menu. While the cursor is blinking, press a character on the keyboard that you want to insert. After that the editing mode is switched off.

Note that if you press the '-' character on the keyboard while being in the editing mode, a gap is inserted. The other way to insert a gap is by pressing *Space* , see the Inserting Gap chapter.

You can always cancel the changes by pressing *Ctrl+Z*.

Replacing Character and Gap

To replace read symbol or gap in the *Sanger Reads Editor* select the nucleotide and press *Shift+R* or *Edit–>Replace character/gap* context menu for switching the edit mode on.

While the selection is blinking, press a keyboard character that you want to replace.

After that the edit mode is switched off. For replace another symbol press *Shift+R* again.

You always can cancel the changes pressing *Ctrl+Z*.

Removing Character and Gap

To delete read symbol or gap in the *Sanger Reads Editor* select the nucleotide and press *Del* or *Edit–>Remove character/gap* context menu for switching the edit mode on.

While the selection is blinking, press a delete button.

After that the edit mode is switched off. For remove another symbol press *Del* again.

You always can cancel the changes pressing *Ctrl+Z*.

Inserting Gap

To insert a gap into a read in the *Sanger Reads Editor* select a character or a gap in the corresponding location in the *Chromatogram Area* and press *Spac e,* select *Edit–>Insert gap* in the context item, or *Actions->Edit->Insert gap* in the main menu.

Thus, by holding the *Space* key, one can shift a Sanger read to the right. To shift the read to the left hold the *Backspace* key, see the Removing Gap at the Left chapter.

You can always cancel the changes by pressing *Ctrl+Z*.

Removing Gap at the Left

To remove a gap in the *Sanger Reads Editor* select the next character or gap after the gap you want to remove and press *Backspace*. You can also select *Edit->Remove gap at the left* in the context menu or *Actions->Edit->Remove gap at the left*.

If there is a region of gaps or it is the beginning of a read, by holding the *Backspace* key, one can shift a Sanger read to the left. To shift the read to the right hold the *Space* key, see the Inserting Gap chapter.

You can always cancel the changes by pressing *Ctrl+Z*.

Removing All Columns of Gaps

Use the *Shift+Del* or *Edit Remove all columns of gaps* item in the *Actions* main menu or in the context menu to remove all columns of gaps from the alignment. This can be useful, for example, when an insertion in a read turned out not to be real, it was replaced by a gap, and a column of gaps was left in the alignment.

Trimming Left End

To trim left end in the *Sanger Reads Editor* select the next symbol after trimming part and use *Ctrl+Shift+Backspace* or *Edit–>Trim left end* context menu for switching the edit mode on.

After that the edit mode is switched off.

You always can cancel the changes pressing *Ctrl+Z*.

Trimming Right End

To trim right end in the *Sanger Reads Editor* select the previous symbol before trimming part and use *Ctrl+Shift+Del* or *Edit–>Trim right end* context menu for switching the edit mode on.

After that the edit mode is switched off.

You always can cancel the changes pressing *Ctrl+Z*.

Renaming Read

To rename read use *F2* or *Rename read* context menu.

Removing Read

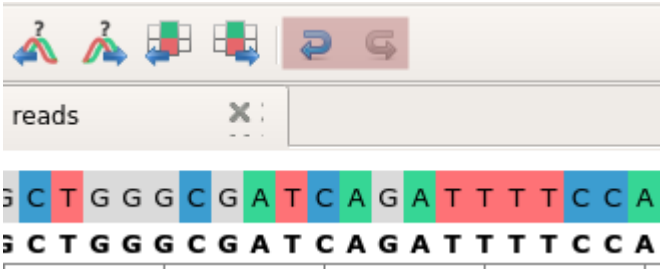
To remove read use *Delete* keyboard button or *Remove read* context menu.

Undo and Redo

The editor tracks all modifications.

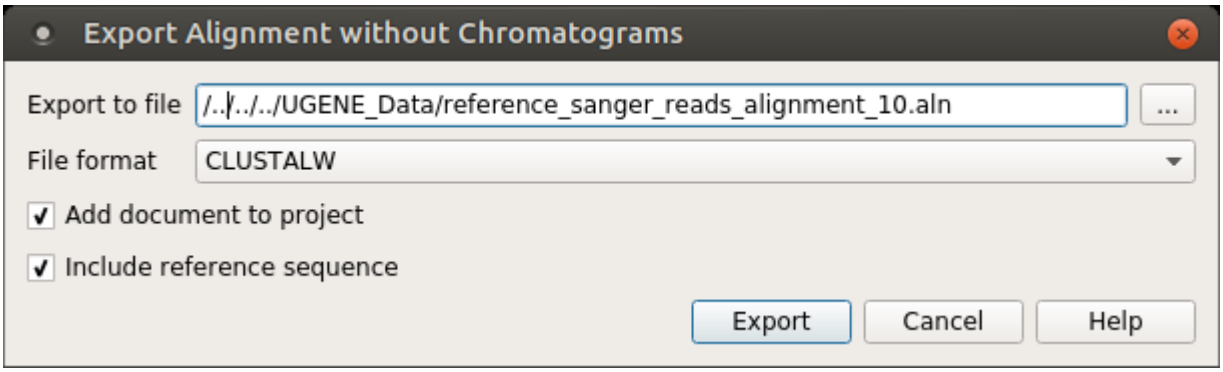
When a modification happens the current state is being recorded.

You can apply any previous state and redo the modifications using the corresponding buttons on the toolbar:



Exporting Alignment without Chromatograms

To export alignment without chromatogram use the Actions–>*Alignment–>Export alignment without chromatogram* main menu or the same context menu item. The following dialog will appear:



Here it is possible to specify the result file location, to select a sequence file format, optionally include consensus and add the created document to the current project.

Assembly Browser

The UGENE Assembly Browser project started in 2010 was inspired by Illumina iDEA Challenge 2011 and multiple requests from UGENE users.The main goal of the Assembly Browser is to let a user visualize and efficiently browse large next generation sequence assemblies.

Currently supported formats are SAM (Sequence Alignment/Map) and BAM, which is a binary version of the SAM format. Both formats are produced by SAMtools and described in the following specification: SAMtools. Support of other formats is also planned, so please send us a request if you're interested in a certain format.

To browse an assembly data in UGENE, a BAM or SAM file should be imported to a UGENE database file. After that you can convert the UGENE database file into a SAM file. The import to a UGENE database file has both advantages and disadvantages. The disadvantages are that the import may take time for a large file and there should be enough disk space to store the database file.

On the other hand, this allows one to overview the whole assembly and navigate in it rather rapidly. In addition, during the import you can select contigs to be imported from the BAM/SAM file. So, there is no need to import the whole file if you're going to work only with some contigs. Note that in the future there are plans to support the other approach as well, namely, when a BAM/SAM file is opened directly.

The Assembly Browser has been tested on different BAM/SAM files from the 1000 Genomes Project and other sources.

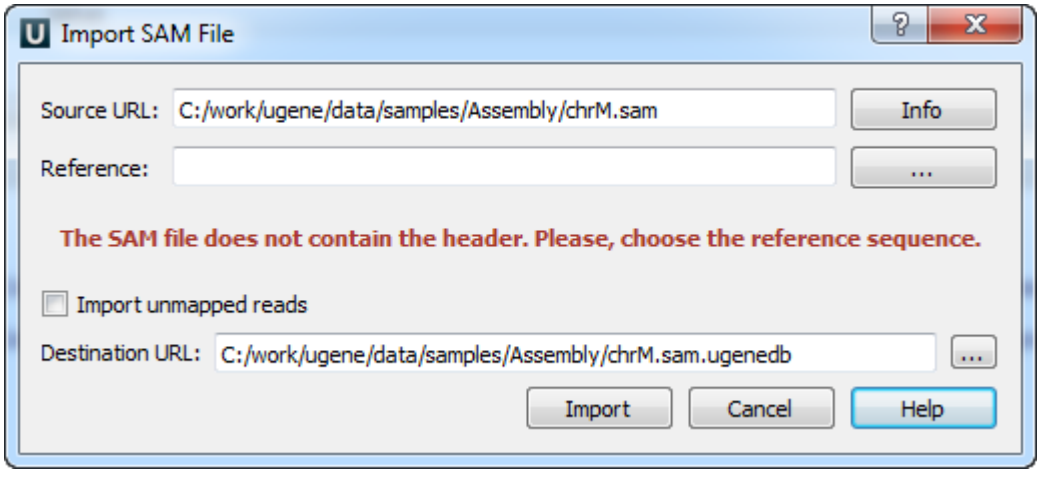
Read the documentation below to learn more about the Assembly Browser features.

- Import BAM and SAM Files
- Browsing and Zooming Assembly
  - Opening Assembler Browser Window
  - Assembly Browser Window
  - Assembly Browser Window Components
  - Reads Area Description
  - Assembly Overview Description
  - Ruler and Coverage Graph Description
  - Go to Position in Assembly
  - Using Bookmarks for Navigation in Assembly Data
- Getting Information About Read
- Short Reads Vizualization
  - Reads Highlighting
  - Reads Shadowing
- Associating Reference Sequence
- Associating Variations
- Consensus Sequence
- Exporting
  - Exporting Reads
  - Exporting Visible Reads
  - Exporting Coverage
  - Exporting Consensus
  - Exporting Consensus Variations
  - Exporting Assembly as Image
  - Exporting Assembly Region
- Options Panel in Assembly Browser
  - Navigation in Assembly Browser
  - Assembly Statistics
  - Assembly Browser Settings
- Assembly Browser Hotkeys
  - Assembly Overview Hotkeys
  - Reads Area Hotkeys

Import BAM and SAM Files

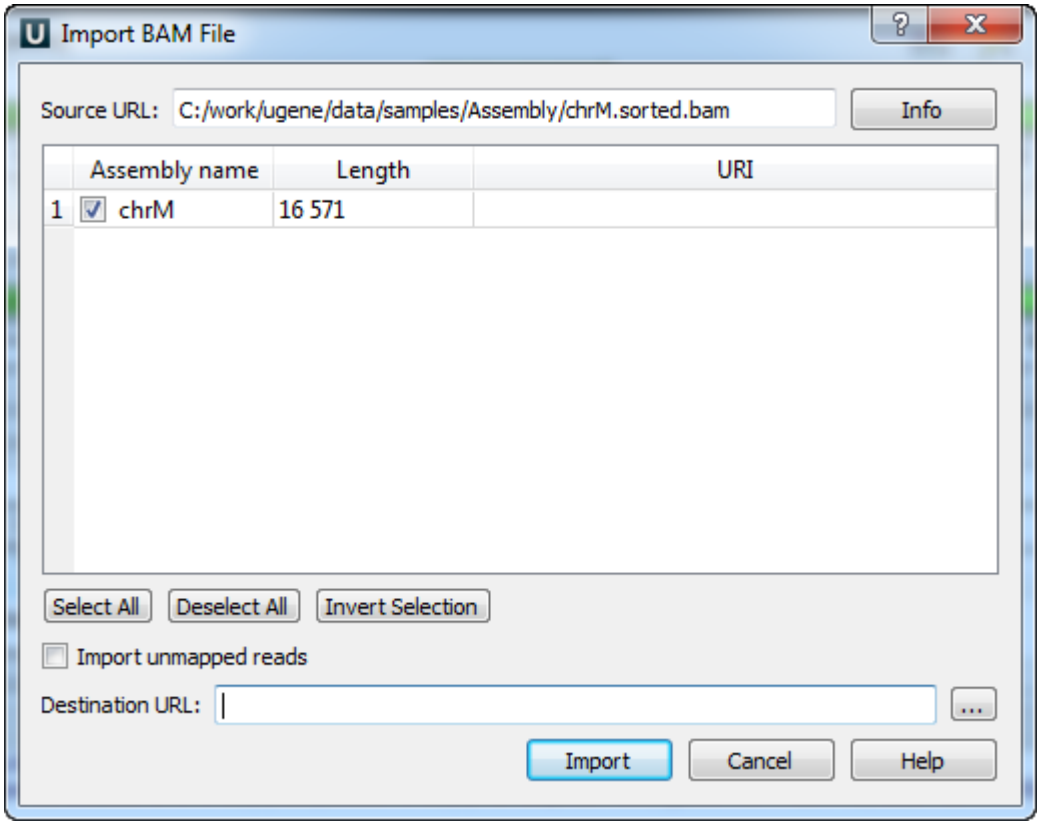
To start working with an assembly import it to the UGENE database file. To do this, *open* the assembly file.

For assembly file without header you need to choose a referenece sequence:



Select the referense sequence and click *Import* button.

For other assembly files the following dialog appears:



The *Source URL* field in the dialog specifies the file to import. The *Info* button nearby can be used to obtain additional information about the file.

There is a list of contigs below the *Source URL*. Check the contigs that you want to import to the database. You can use the *Select All*, *Deselect All* and *Invert Selection* buttons to manage the selection.

The *Destination URL* field specifies the output database file.

If you check the *Import unmapped reads*, then all unmapped reads in the assembly (i.e. read with the unmapped flag or without CIGAR) are imported. Note, however, that they are not vizualized in the current UGENE version.

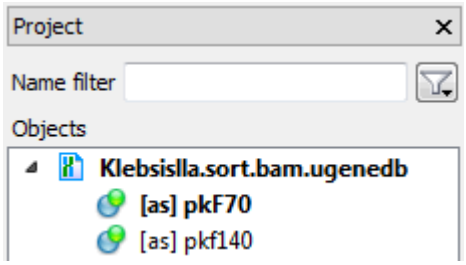
To start the import, click the *Import* button in the dialog. You can see the progress of the import in the *Task View*. To export a UGENE database file into the SAM format, select the *Actions Export assembly to SAM format* item in the main menu.

Browsing and Zooming Assembly

- Opening Assembler Browser Window
- Assembly Browser Window
- Assembly Browser Window Components
- Reads Area Description
- Assembly Overview Description
- Ruler and Coverage Graph Description
- Go to Position in Assembly
- Using Bookmarks for Navigation in Assembly Data

Opening Assembler Browser Window

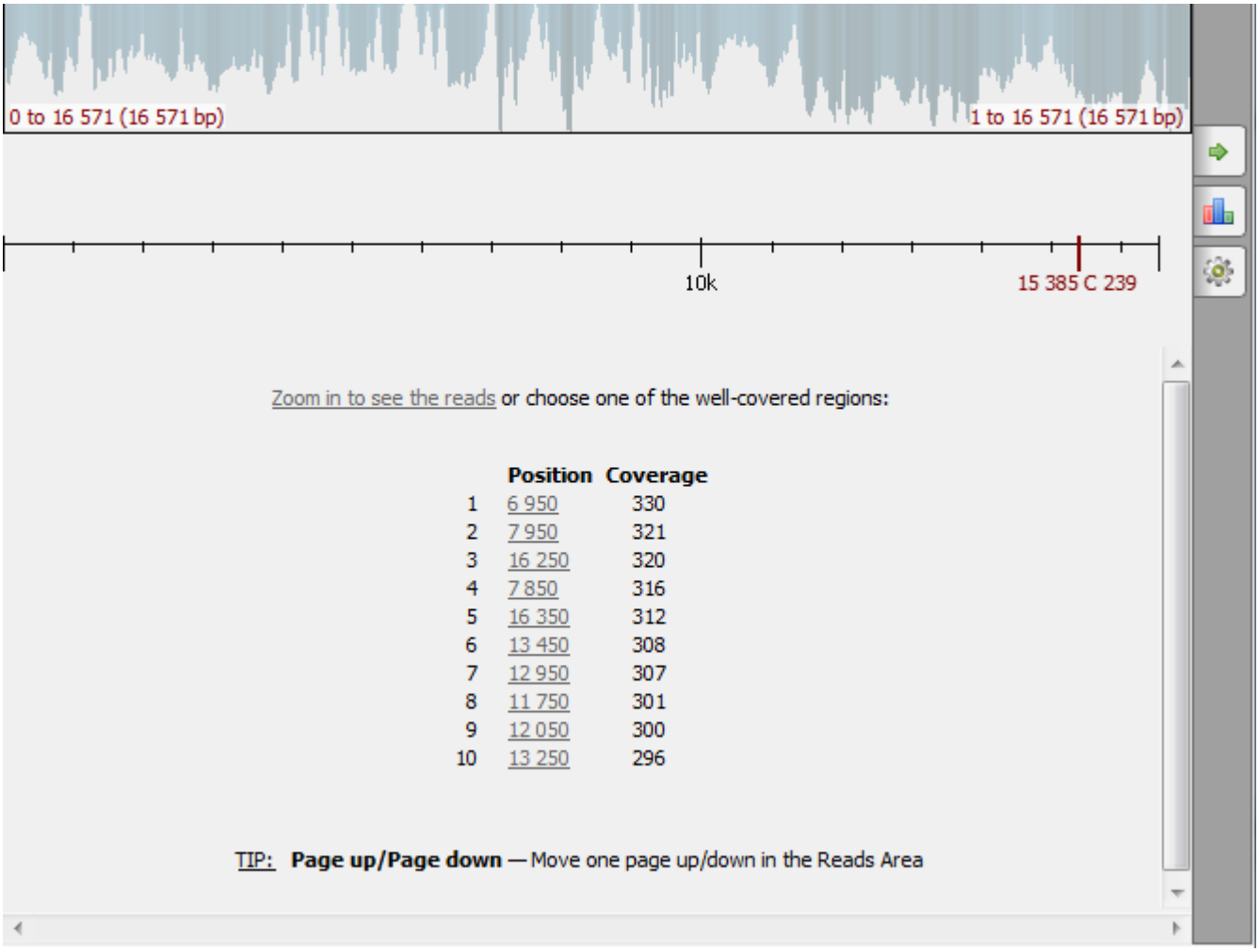
An imported assembly added to the project is shown in the *Project View* as follows:



Each [as] object corresponds to an imported contig. When you double-click on an [as] object a new Assembly Browser window with the assembly data is opened. A window for the first assembly object in the list is opened automatically after the import.

Assembly Browser Window

The opened window contains the list of well-covered regions of the assembly:



Note that for large assemblies it may take some time to calculate the overview and the well-covered regions.

To see the reads, either select a region from the list or zoom in, for example, by clicking the link above the well-covered regions or by rotating the mouse wheel.

You can also use the hotkeys. Tips about hotkeys are shown under the list of well-covered regions. To learn about available hotkeys refer to *Assembly Browser Hotkeys*.

Assembly Browser Window Components

An Assembly Browser window consists of:

Assembly Overview

By default, shows the whole assembly overview. Can be resized to provide an overview of an assembly part.

Reference Area

Shows the reference sequence.

Consensus Area

Shows the consensus sequence.

Ruler

Shows the coordinates in the *Reads Area*.

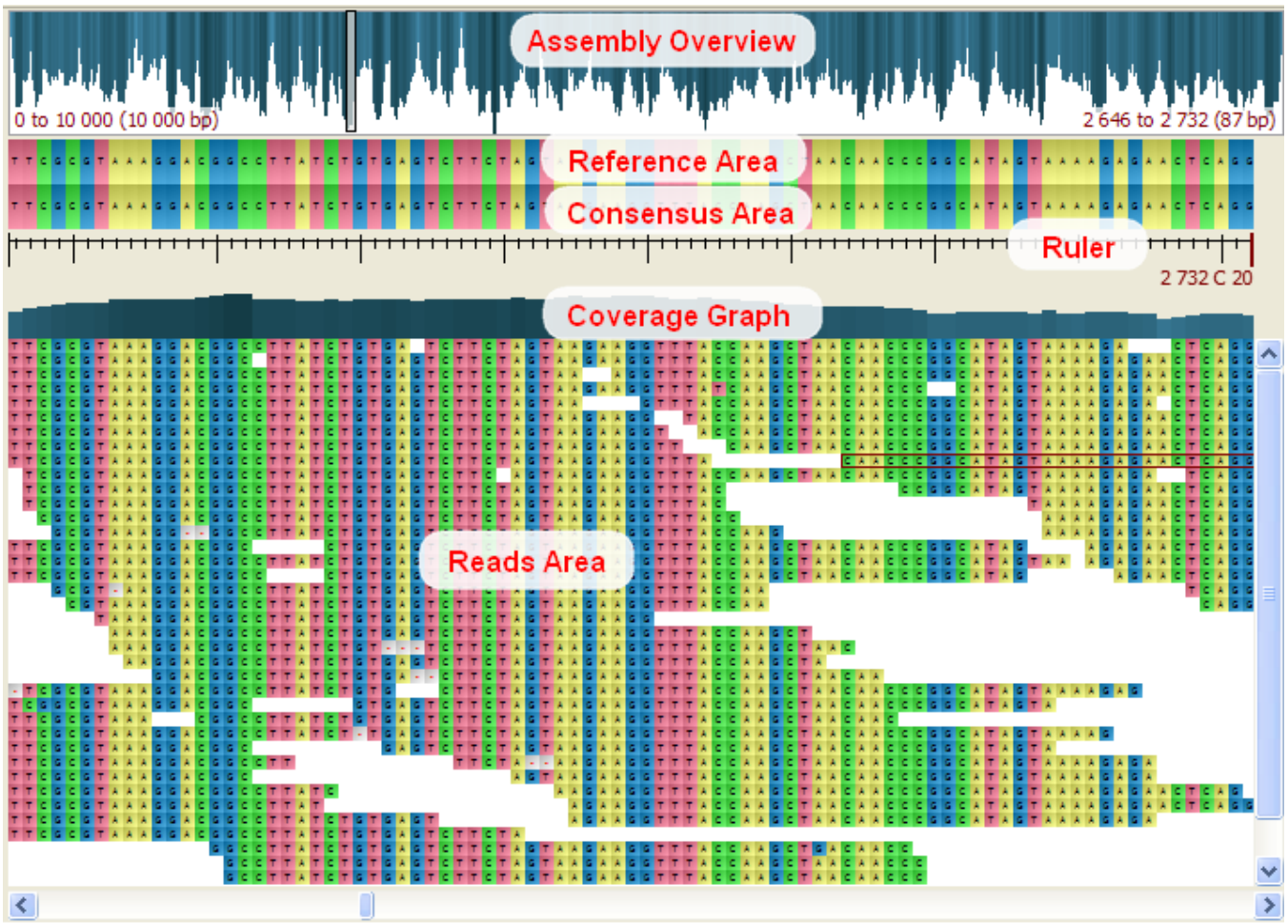
Reads Area

Displays the reads.

Coverage Graph

Shows the coverage of the Reads Area.

See the example below:



Reads Area Description

The *Reads Area* provides a visualization of reads of an assembly part. To zoom in or zoom out, rotate the mouse wheel.

To perform zooming you can also use the *Zoom In* and *Zoom Out* buttons on the toolbar or the *Actions Zoom In* and *Actions Zoom Out* items in the main menu.

Also, when you double-click on a read it is zoomed in and moved to the center of the window.

By dragging the mouse while holding the left mouse button you can navigate in the Reads Area.

To navigate long distances in the Reads Area use the *Assembly Overview* described *below*.

Other ways to navigate in the assembly are:

- Use the horizontal and vertical scroll bars of the Reads Area
- *Go to a specified position in an assembly*

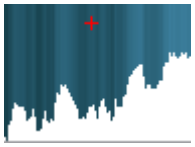
To learn about available hotkeys refer to *Assembly Browser Hotkeys*.

By default, assembly rendering is optimized while scrolling. While you are moving across an assembly, it shows the assembly in gray color, but when you stop it shows the assembly in different colors. To disable this option uncheck the *Optimize the rendering while scrolling* item in the context menu of the *Reads Area* or *Optimize scrolling* item on the *Assembly Browser Settings* tab of the *Options Panel*.

Assembly Overview Description

The *Assembly Overview* shows a coverage overview of the assembly. The longer the depth of a line in the overview and the deeper the color, the more reads are located in this region.

To open a region of the assembly in the *Reads Area* click on it in the Assembly Overview. On the overview, the selected region is displayed either as a gray rectangle, a red cross or a red rectangle. For example:

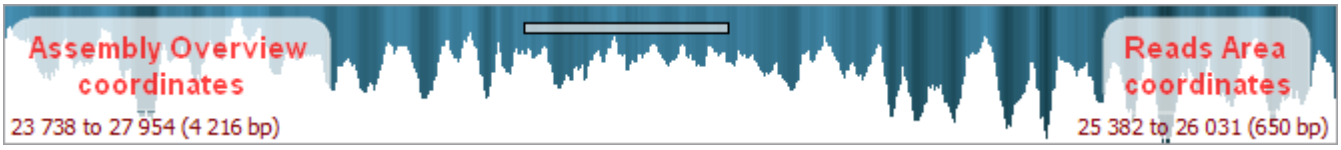


If you hold Shift and select a region on the overview, the overview is zoomed to the selection.

Note that when the Assembly Overview is in focus and you use either the zoom buttons on the toolbar, the zoom items in the *Actions* main menu, or a mouse wheel, the Reads Area is resized appropriately.

The Assembly Overview can also be resized. To zoom in the overview, select either the *Zoom in* or the *Zoom in 100x* item in the Assembly Overview context menu. You can scroll the resized overview by dragging the mouse while pressing down the mouse wheel. To zoom out the overview, select the *Zoom out* item in the context menu. The *Restore global overview* item in the context menu restores the default overview size when the whole contig overview is shown.

Notice that the Assembly Overview shows the coordinates of the assembly areas visible in the Reads Area and in the Assembly Overview:

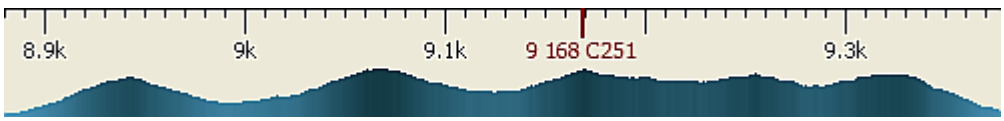


To scroll the resized overview, drag the mouse while pressing down the mouse wheel.

To learn about available hotkeys refer to *Assembly Browser Hotkeys*.

Ruler and Coverage Graph Description

The *Ruler* shows the coordinates in the *Reads Area*. When you move the mouse cursor in the Reads Area the coordinate of the selected location with the coverage of reads is shown on the ruler in dark red. The Coverage Graph shows the exact coverage of the sequence at each position. For example on the image below the coordinate is 9168 and the coverage of reads is 251.



To show/hide the coordinates on the ruler you can click the following button on the toolbar:



To show/hide the coverage on the ruler you can click the following button on the toolbar:



Alternatively, you can use the *Show coordinates* and *Show coverage under cursor* check boxes located on the *Assembly Browser Settings* tab of the *Options Panel*.

Go to Position in Assembly

To go to the required position in an assembly use the following field located on the Assembly Browser toolbar.



Input the location and click the *Go!* button. A similar *Go!* field is also available on the *Navigation* tab of the *Options Panel*.

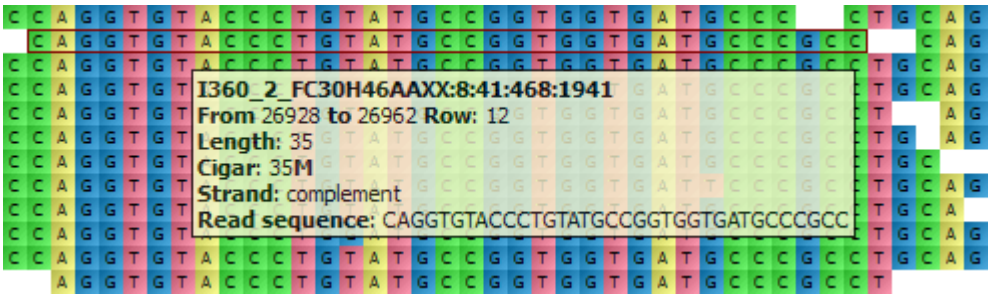
Using Bookmarks for Navigation in Assembly Data

Use *bookmarks* to save and restore visual state of an assembly, for example, position in the assembly, zoom scale, etc.

Getting Information About Read

A read displayed in the *Reads Area* consists of the bases (A, C, G, T). It may also contain the N character that stays for an ambigous base. Depending on the value of the *Cigar* parameter, the read can be shown partially or gaps can be inserted inside the read (see below).

By default when a read is hovered over in the *Reads Area* a hint appears:



To disable this behaviour click the following button on the toolbar:



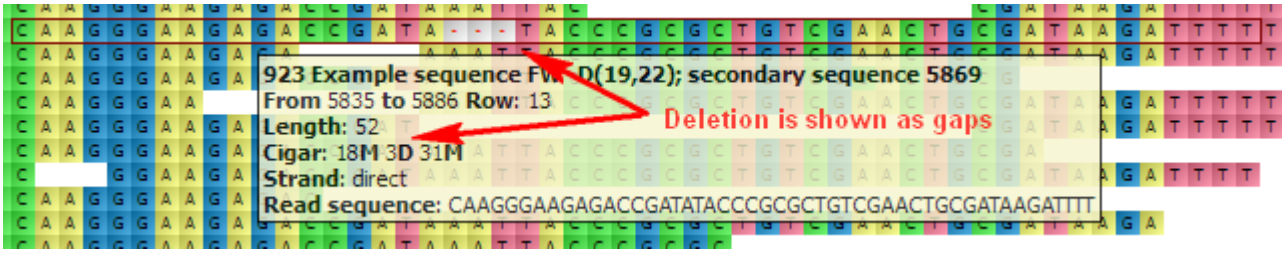
Or uncheck the *Show pop-up hint* check box on the *Assembly Browser Settings* tab of the *Options Panel*.

The hint shows the following information about the read:

- Read name
- Location
- Length
- Cigar
- Strand
- Read sequence

The operations in the *Cigar* parameter are described as follows:

- **M** — Alignment match (can be a sequence match or mismatch).
- **I** — Insertion to the reference. Skipped when the read is aligned to the reference, i.e. it is not shown in the Reads Area, but is present in the read sequence.
- **D** — Deletion from the reference. Gaps are inserted to the read when the read is aligned to the reference. For example:



- **N** — Skipped region from the reference. Behaves as **D**, but has a different biological meaning: for mRNA-to-genome alignment it represents an intron.
- **S** — Soft clipping (clipped sequences are present in the read sequence, i.e. behaves as **I**).
- **H** — Hard clipping (clipped sequences are not present in the read sequence).
- **P** — Padding (silent deletion from padded reference).
- **=** — Exact match to the reference.
- **x** — Reference sequence mismatch.

To copy the information about the read to the clipboard, select the *Copy read information to clipboard* item in the Reads Area context menu. Now you can paste it in any text editor.

To copy the current position of the read select the *Copy current position to clipboard* item in the *Reads Area* context menu.

Short Reads Vizualization

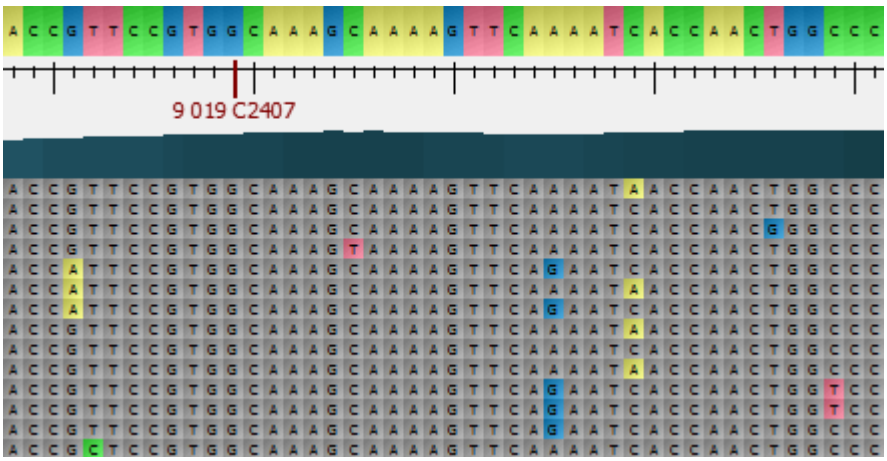
There are various modes of reads highlighting and shadowing.

- Reads Highlighting
- Reads Shadowing

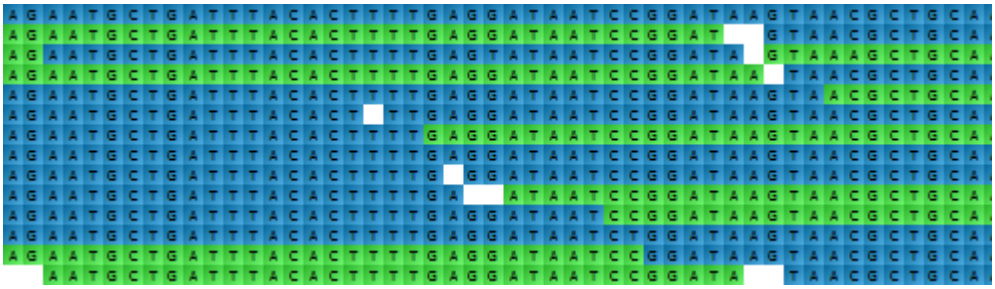
Reads Highlighting

To apply a reads highlighting mode, select it in the *Reads highlighting* menu of the *Reads Area* context menu or on the *Assembly Browser Settings* tab of the *Options Panel*. The following modes are available:

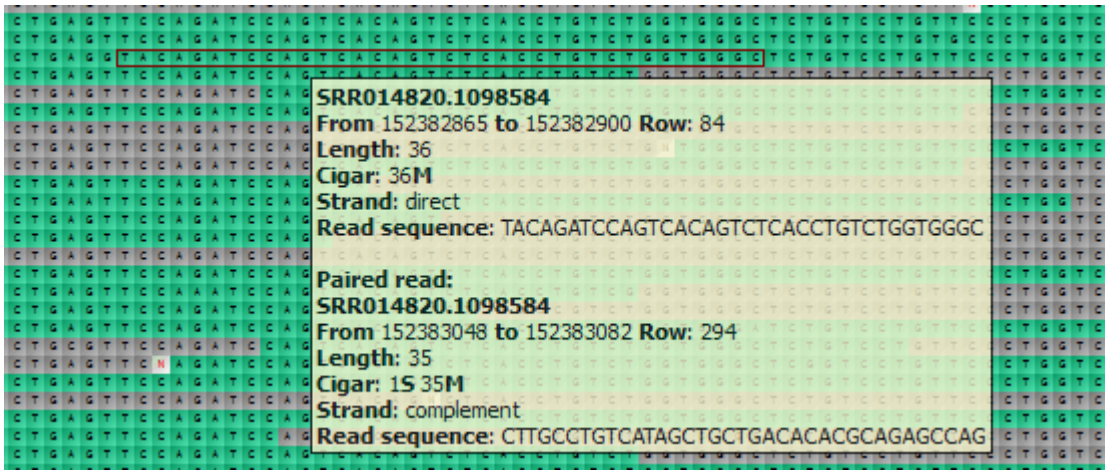
- *Nucleotide* — shows all nucleotides in different colors. It is used by default.
- *Difference* — highlights gaps and nucleotides that differ from the reference sequence. You should add a reference first for correct displaying of this highlighting.



- *Strand direction* — highlights reads located on the direct strand in blue and reads on the complement strand in green.



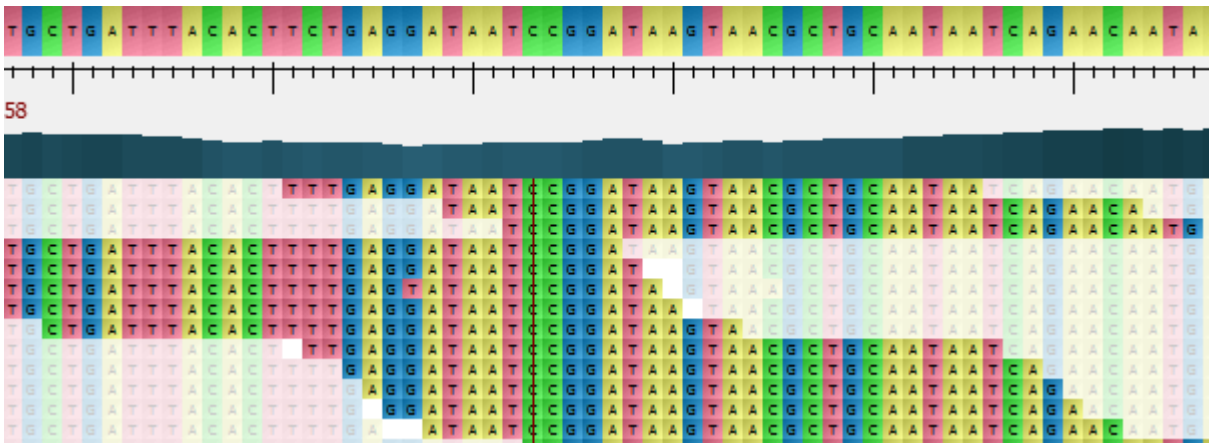
- *Paired reads* — highlights all paired reads in green. Note that the information about the pair is shown in the hint.



Reads Shadowing

Various modes of column highlighting are available from the *Reads shadowing* item in the context menu of the *Reads Area*:

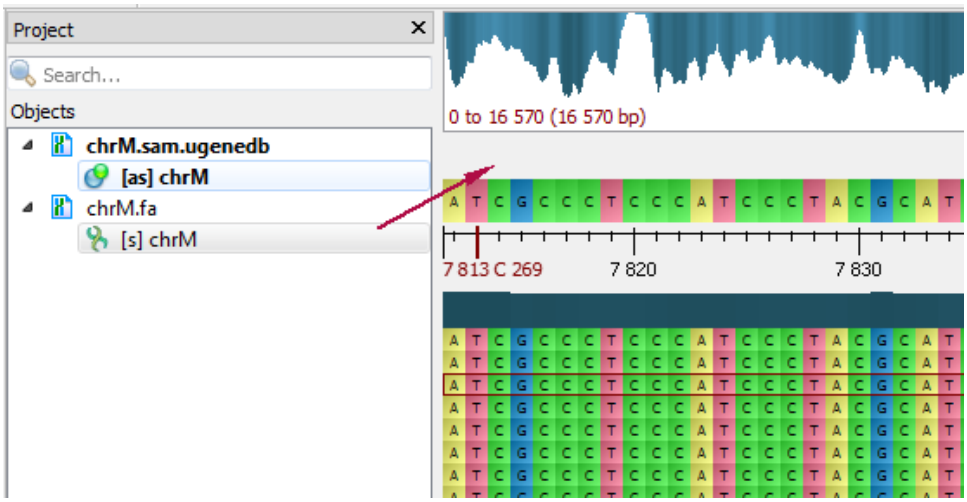
- *Disabled* — highlights all columns of nucleotides.
- *Free* — highlights all reads that intersect a given column. In this mode you can lock a position. Click the *Lock here* item in the context menu to do it. To return to a locked position, select the *Jump to locked base* item in the context menu.
- *Centered* — highlights all reads that intersect the column in the center of the screen.



Associating Reference Sequence

To associate a reference sequence with the assembly use one of the following ways:

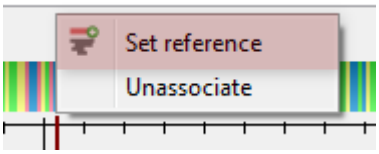
1. *Open the sequence* (the sequence must be loaded) and drag it to the *Assembly Reference Area*:



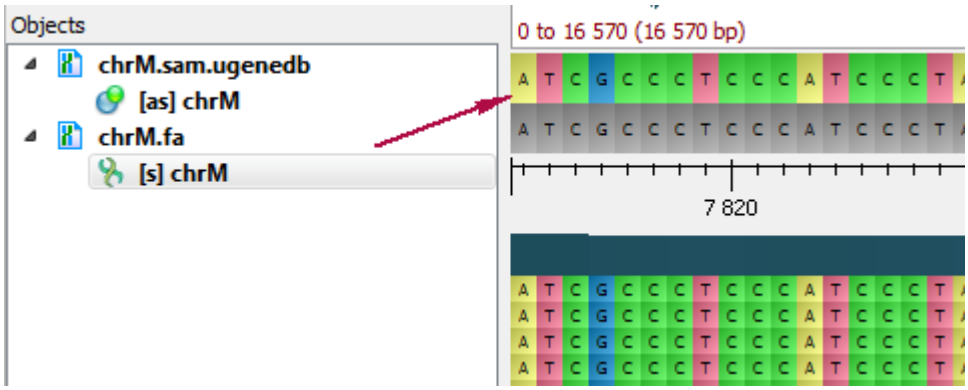
2. Use the corresponding toolbar button:



3. Use the corresponding context menu item:



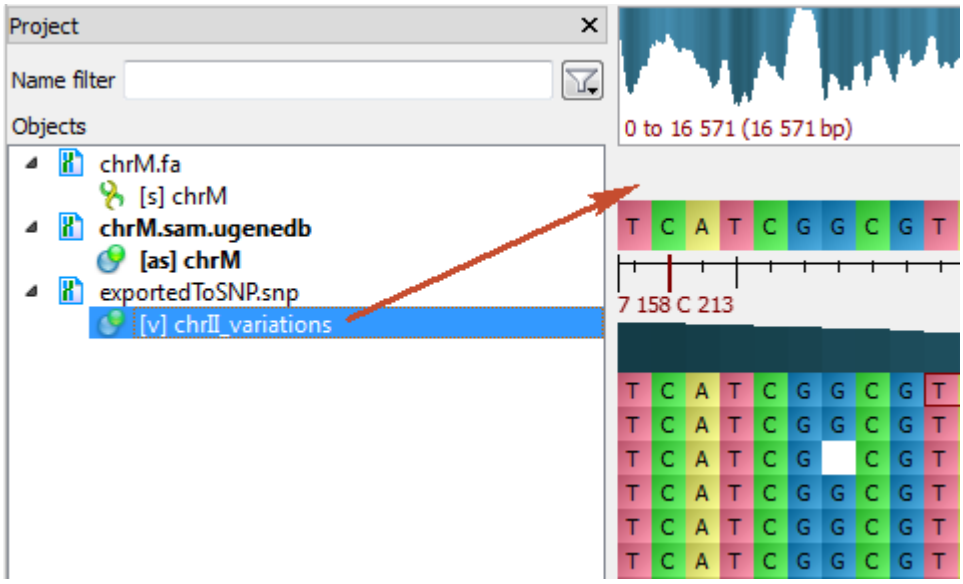
The reference sequence will appear in the *Reference Area*:



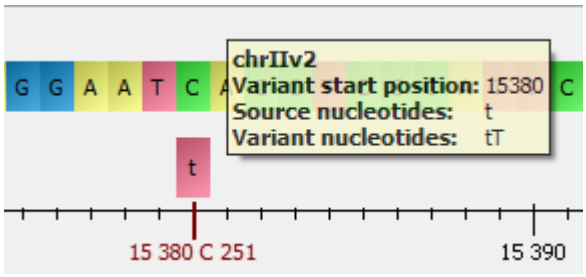
To remove the association, select the *Unassociate* item in the *Reference Area* context menu.

Associating Variations

To associate variations with the assembly, *open the sequence* (the sequence must be loaded) and drag it to the *Assembly Reference Area*:



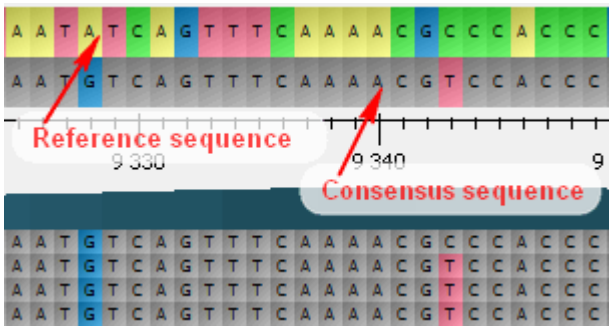
The variations will appear under the *Consensus Sequence*:



To remove the association, select the *Remove track from the view* item in the *Variations Area* context menu.

Consensus Sequence

A consensus sequence can be found in the *Consensus Area* under a reference sequence. It refers to the most common nucleotide at a particular position.

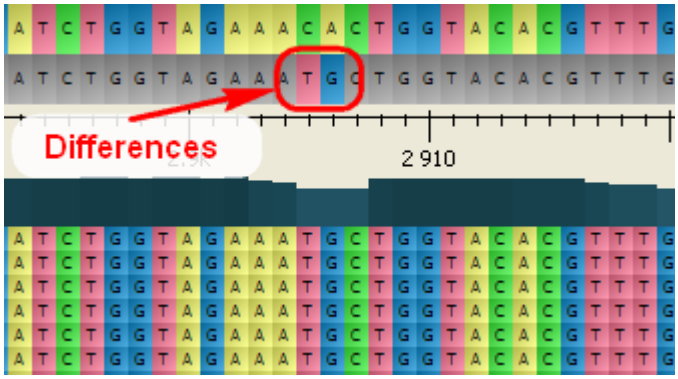


To choose a consensus algorithm select the *Consensus algorihtm* item either in the context menu of the *Consensus Area*, in the context menu of the *Read s Area* or on the *Assembly Browser Settings* tab of the *Options Panel*. .

The following algorithms are currently available:

- *Default* — shows the most common nucleotide at each position. When there is equal numbers of different nucleotides in a position, the consensus sequence resulting nucleotide is selected randomly from these nucleotides.
- *SAMtools* — uses an algorithm from the SAMtools Text Alignment Viewer to build the consensus sequence. The algorithm takes into account quality values of reads and nucleotides and works with the extended nucleotide alphabet.

To leave only differences between the reference and the consensus sequences highlighted on the consensus sequence, select the *Show difference from reference* item in the context menu of the *Consensus Area* or the *Difference from reference* item on the *Assembly Browser Settings*tab of the *Options Panel*



To export a *Consensus Sequence*, right-click on it in the *Consensus Area* and select the *Export Export consensus* item in the context menu. For more information about consensus exporting see *Exporting Consensus*.

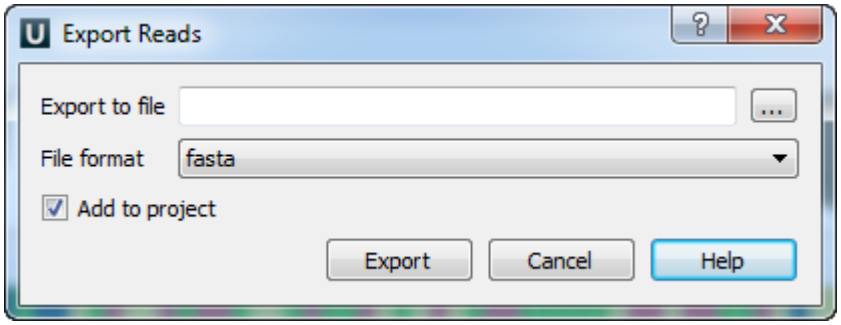
Exporting

- Exporting Reads
- Exporting Visible Reads
- Exporting Coverage
- Exporting Consensus
- Exporting Consensus Variations
- Exporting Assembly as Image
- Exporting Assembly Region

Exporting Reads

To export a read, right-click on it in the *Reads Area* and select the *Export Current read* item in the context menu.

The *Export Reads* dialog appears:



Select a file to export the read to and the file format. The read can be exported either to a FASTA or FASTQ file.

When the parameters are set click the *Export* button.

The read is exported to the file and if the *Add to project* check box has been checked it is added to the current *project* from where you can *open* it.

Exporting Visible Reads

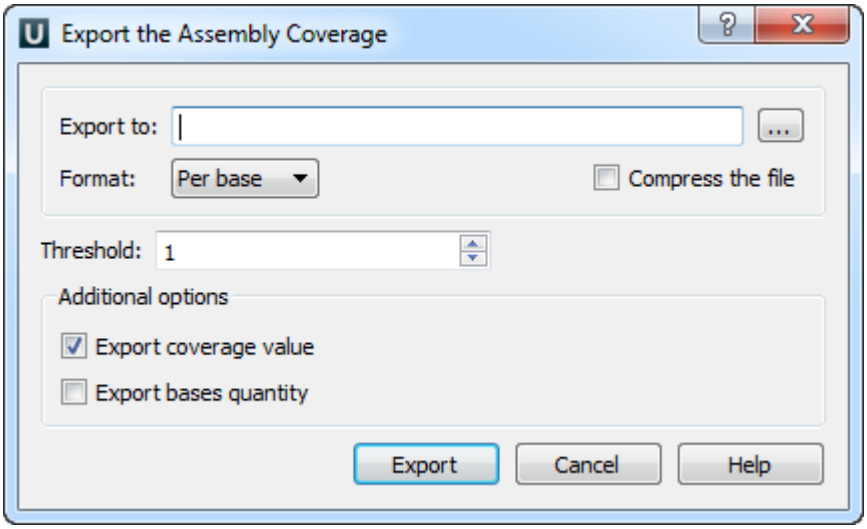
To export all reads visible in the *Reads Area* select the *Export Visible reads* item in the Reads Area context menu.

The *Export Reads* dialog appears. The dialog is described in the *Exporting Read* section.

Exporting Coverage

To export a coverage of the assembly, select either the *Export coverage* item in the *Consensus Area* context menu.

The *Export Coverage* dialog appears:



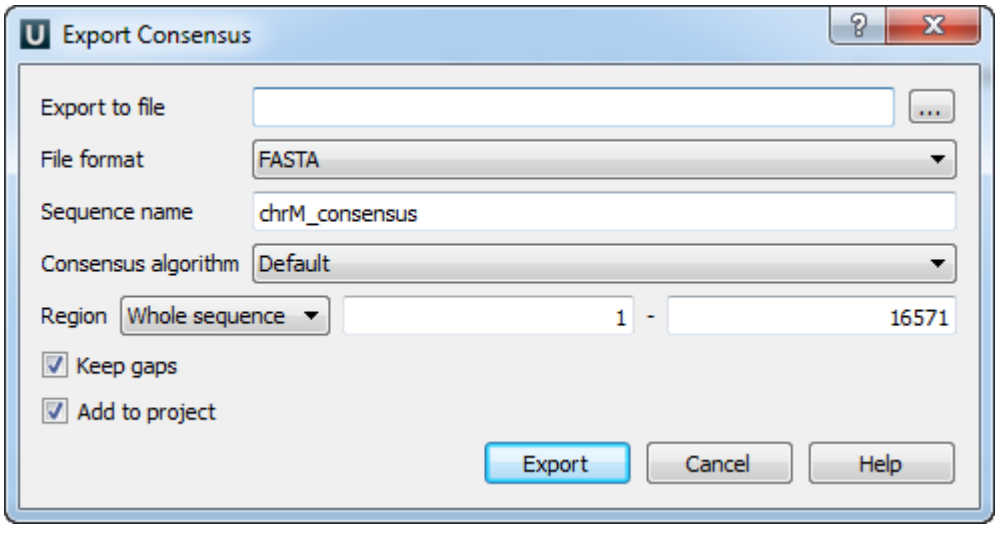
Select a file, threshold and format: *Histogram, Per base* or *Bedgraph.* Where *threshold* is the minimum coverage value to export. For *Per base* format the additional options are available: *Export coverage value* or *Export bases quantity* or both of them.

When all the parameters are set click the *Export* button.

Exporting Consensus

To export a consensus sequence of the assembly, select either the *Export consensus* item in the *Consensus Area* context menu or the *Export Consensus* i tem in the *Reads Area* context menu.

The *Export Consensus* dialog appears:



Select a file and the file format. The consensus can be exported to a FASTA, FASTQ, GFF or GenBank file.

Modify, if required, the exported sequence name and choose the *consensus algorithm*.

The consensus is exported with gaps if the *Keep gaps* check box has been checked.

Also you can select the exporting region. It can be either a *Whole sequence*, a *Visible* region, or a *Custom* region.

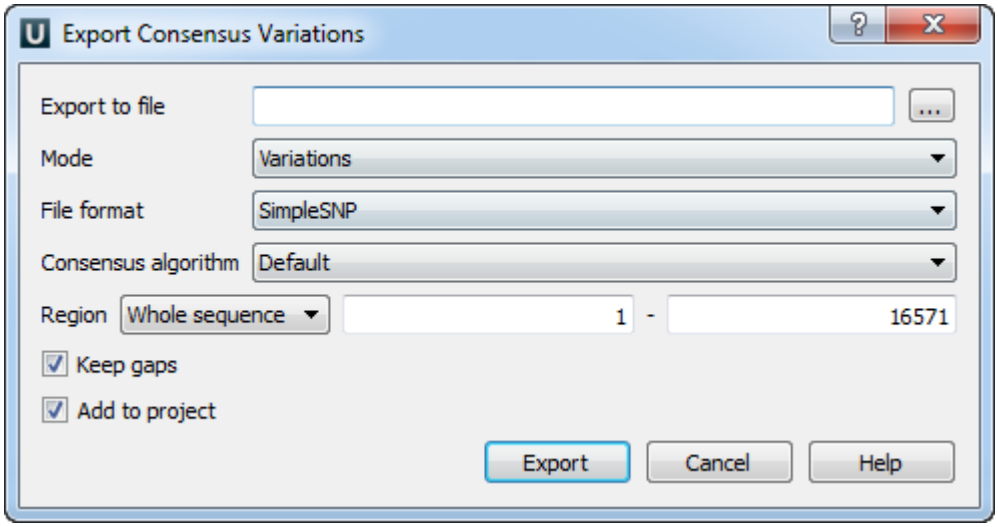
When all the parameters are set click the *Export* button.

The consensus sequence is exported to the file and if the *Add to project* check box has been checked it is added to the current *project* and opened.

Exporting Consensus Variations

To export a consensus sequence variations of the assembly, select the *Export consensus variations* item in the *Consensus Area* context menu.

The *following* dialog will appear:



Select a file, mode and the file format. The following modes are available: *Variations*, *Similar* and *All*. Variations can be exported as to a SimpleSNP or VCFv4 file.

Modify, if required, the *consensus algorithm*.

The consensus is exported with gaps if the *Keep gaps* check box has been checked.

Also you can select the exporting region. It can be either a *Whole sequence*, a *Visible* region, or a *Custom* region.

When all the parameters are set click the *Export* button.

The consensus sequence is exported to the file and if the *Add to project* check box has been checked it is added to the current *project* and opened.

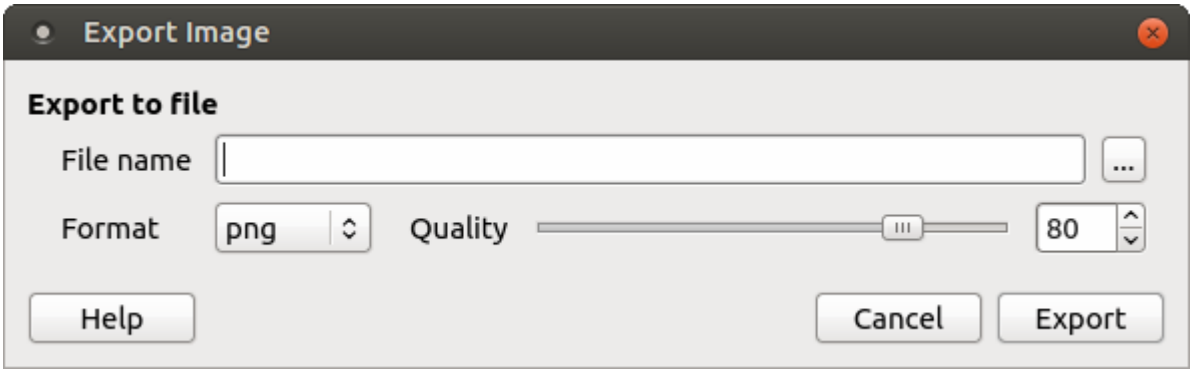
> The *Export consensus variations* feature is available when the reference sequence is associated with assembly.

Exporting Assembly as Image

To export the visible part of the assembly as an image, select either the *Actions* *Export as image* item in the main menu or the following button on the toolbar:



The *Export Image* dialog appears:



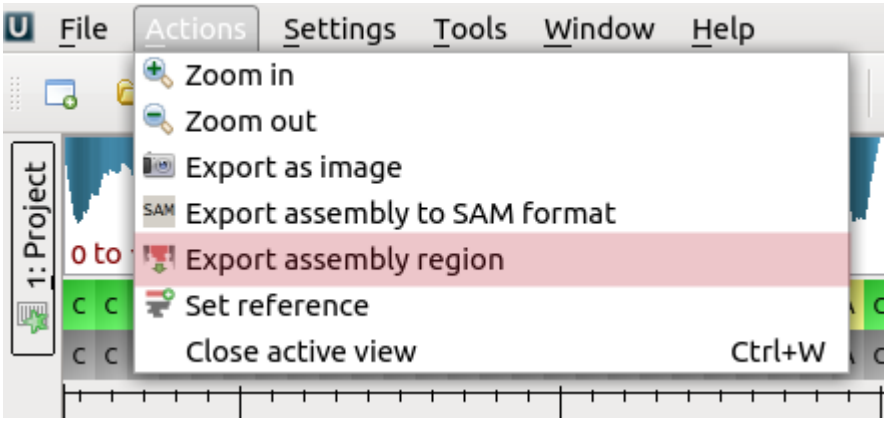
In the dialog you can select the image file name and its format (bmp, jpeg, png, etc.). For some file formats the *Quality* parameter also becomes available.

When the parameters are set click the *OK* button.

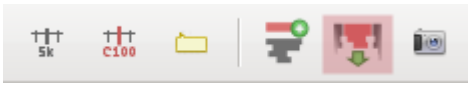
Exporting Assembly Region

To export an assembly region use one of the following ways:

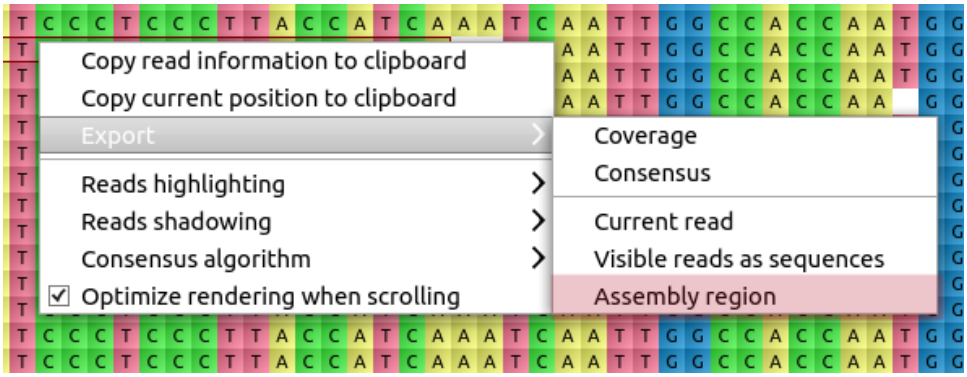
1. Use the *Actions->Export assembly region* main menu item:



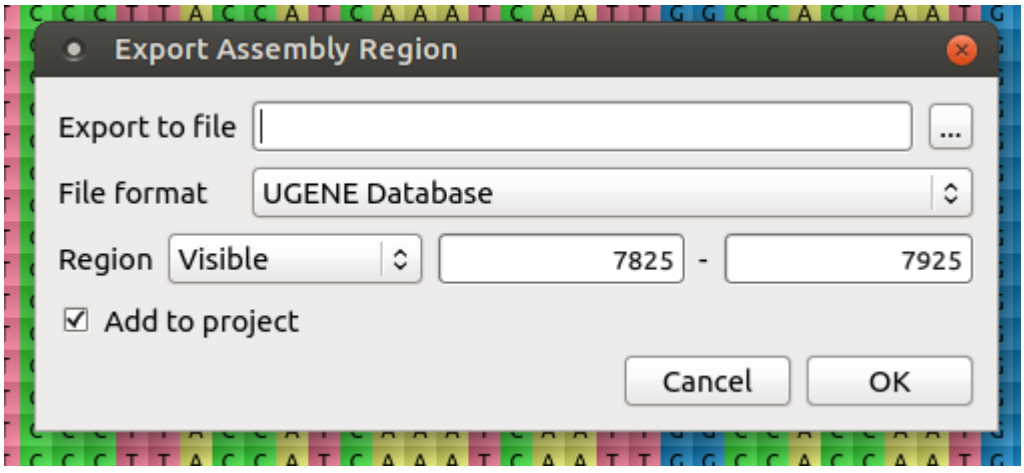
2. Use the corresponding toolbar button:



3. Use the corresponding context menu item:



The *following* dialog will appear:



Select a file, region and the file format.

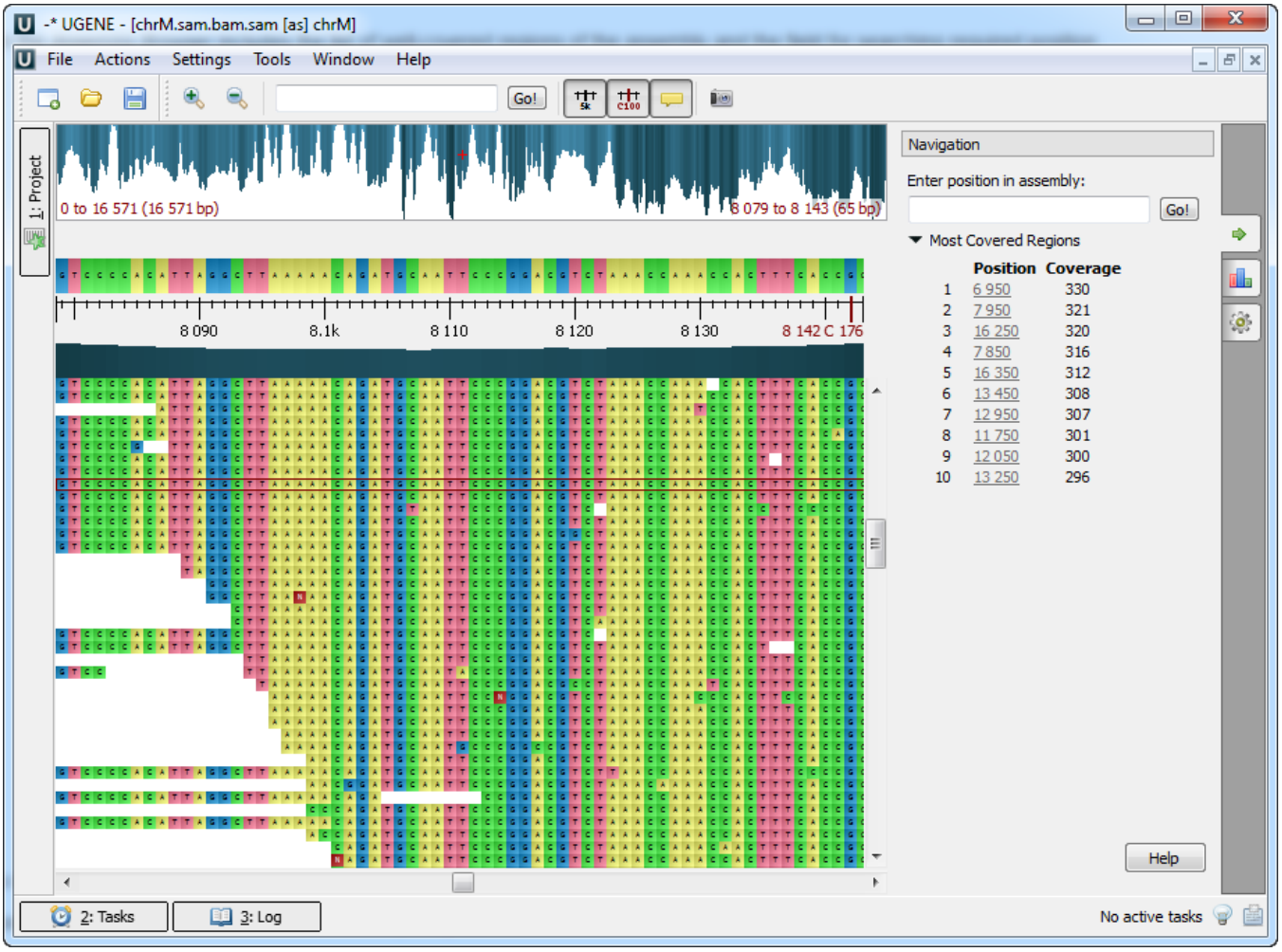
Options Panel in Assembly Browser

- Navigation in Assembly Browser

- Assembly Statistics
- Assembly Browser Settings

Navigation in Assembly Browser

The *Navigation* tab of the *Options Panel* in the *Assembly Browser* includes the list of well-covered regions of the assembly and the field for searching required position.



To learn more about well-covered regions refer to the *Assembly Browser Window* chapter.

To learn more about searching required position refer to the *Go to Position in Assembly* chapter.

Assembly Statistics

The *Assembly Statistics* tab includes the following *Assembly Information*:

- Name — the name of the opened assembly.
- Length — the length of the assembly.
- Reads — the number of reads in the assembly.

Also the tab can include the *Reference Information* if it is available in the assembly file. For example:

- MD5
- Species
- URI



Assembly Browser Settings

The *Assembly Browser Settings* tab includes *Reads Area*, *Consensus Area* and *Ruler* settings.



To learn more about *Reads Area* settings refer to the *Reads Area Settings* chapter.

To learn more about *Consensus* see the *Consensus Sequence* chapter.

To learn more about *Ruler* see the *Browsing and Zooming Assembly* chapter.

Assembly Browser Hotkeys

- Assembly Overview Hotkeys
- Reads Area Hotkeys

Assembly Overview Hotkeys

The following hotkeys are available for the *Assembly Overview*:

Hotkey	Action
Shift + move mouse	Zoom the Assembly Overview to selection
Ctrl + wheel	Zoom the Assembly Overview
Alt + click	Zoom the Assembly Overview in 100x
wheel + move mouse	Move the Assembly Overview

Reads Area Hotkeys

The following hotkeys are available for the *Reads Area*:

Hotkey	Action
wheel	Zoom the Reads Area
double-click	Zoom in the Reads Area
+ / -	Zoom in / zoom out the Reads Area
click + move mouse	Move the Reads Area
arrow	Move one base in the corresponding direction in the Reads Area
Ctrl + arrow	Move one page in the corresponding direction in the Reads Area
Page Up / Page Down	Move one page up / down in the Reads Area
Home / End	Move to the beginning / end of the assembly in the Reads Area
Ctrl+G	Focus to the *Go to position* field on the toolbar

Workflow Designer

- About the Workflow Designer
- Introduction
  - Launching Workflow Designer
  - Workflow Designer Window Components
  - Workflow Elements and Connections
  - Managing Parameters
  - UGENE Components and Workflow Designer
    - Task View, Notifications and Log View
    - Actions Menu
    - Toolbar
    - Context Menus
    - Application Settings
  - How to Create and Run Workflow
  - How to Use Sample Workflows
- Manipulating Element
  - Adding Element
  - Copying Element
  - Pasting Element
  - Cutting Element
  - Deleting Element
  - Selecting All Elements on Scene
- Manipulating Workflow
  - Creating New Workflow
  - Loading Workflow
  - Saving Workflow
  - Exporting Workflow as Image
  - Validating Workflow
  - Running Workflow
  - Dashboard
    - Dashboard Window Components
    - Using Dashboard
  - Stopping and Pausing Workflow
- Changing Appearance
- Custom Elements with Scripts
  - Functions Supported for Multiple Alignment Data
  - Functions Supported for Sequence Data
  - Functions Supported for Set of Annotations Data
  - Functions Supported for Files
  - Common Function
- Custom Elements with External Tools
  - Creating Element
  - Editing Element
  - Adding Existent Element
  - Removing Element
- Using Script to Set Parameter Value
- Running Workflow from the Command Line
- Running Workflow in Debugging Mode
  - Creating Breakpoints
  - Manipulating Breakpoints
- Workflow File Format
  - Header
  - Body
    - Elements
    - Dataflow
    - Metainformation
- Workflow Elements

- Data Readers
  - Read Alignment Element
  - Read Annotations Element
  - Read FASTQ File with SE Reads Element
  - Read FASTQ Files with PE Reads Element
  - Read File URL(s) Element
  - Read NGS Reads Assembly Element
  - Read Plain Text Element
  - Read Sequence Element
  - Read Sequence from Remote Database Element
  - Read Variants Element
- Data Writers
  - Write Alignment Element
  - Write Annotations Element
  - Write FASTA Element
  - Write NGS Reads Assembly Element
  - Write Plain Text Element
  - Write Sequence Element
  - Write Variants Element
- Data Flow
  - Filter Element
  - Grouper Element
  - Multiplexer Element
  - Sequence Marker Element
- Basic Analysis
  - Amino Translations Element
  - Annotate with UQL Element
  - CD-Search Element
  - Collocation Search Element
  - Export PHRED Qualities Element
  - Fetch Sequences by ID From Annotation Element
  - Filter Annotation by Name Element
  - Filter Annotations by Qualifier
  - Find Correct Primer Pairs Element
  - Find Pattern Element
  - Find Repeats Element
  - Gene-by-gene approach report
  - Get Sequences by Annotations Element
  - Group Primer Pairs Element
  - Import PHRED Qualities Element
  - Intersect Annotations Element
  - Local BLAST+ Search Element
  - Merge Annotations Element
  - ORF Marker Element
  - Remote BLAST Element
  - Sequence Quality Trimmer Element
  - Smith-Waterman Search Element
- Data Converters
  - Convert bedGraph Files to bigWig Element
  - Convert Text to Sequence Element
  - File Format Conversion Element
  - Reverse Complement Element
  - Split Assembly into Sequences Element
- DNA Assembly
  - Assembly Sequences with CAP3
- HMMER2 Tools
  - HMM2 Build Element
  - HMM2 Search Element
  - Read HMM2 Profile Element
  - Write HMM2 Profile Element
- HMMER3 Tools
  - HMM3 Build Element
  - HMM3 Search Element
  - Read HMM3 Profile
  - Write HMM3 Profile
- Multiple Sequence Alignment
  - Align Profile to Profile with MUSCLE Element
  - Align with ClustalO Element
  - Align with ClustalW Element
  - Align with Kalign Element
  - Align with MAFFT Element
  - Align with MUSCLE Element
  - Align with T-Coffee Element
  - Extract Consensus from Alignment as Sequence
  - Extract Consensus from Alignment as Text
  - In Silico PCR Element
  - Join Sequences into Alignment Element
  - Map to Reference Element
  - Split Alignment into Sequences Element

- NGS: Basic Functions
  - CASAVA FASTQ Filter Element
  - Cut Adapter Element
  - Extract Consensus from Assembly Element
  - Extract Coverage from Assembly Element
  - FASTQ Merger Element
  - FASTQ Quality Trimmer Element
  - FastQC Quality Control Element
  - Filter BAM/SAM Files Element
  - Genome Coverage Element
  - Improve Reads with Trimmomatic Element
  - Merge BAM Files Element
  - Remove Duplicates in BAM Files Element
  - Slopbed Element
  - Sort BAM Files Element
- NGS: ChIP-Seq Analysis
  - Annotate Peaks with peak2gene Element
  - Build Conservation Plot Element
  - Collect Motifs with SeqPos Element
  - Conduct GO Element
  - Create CEAS Report Element
  - Find Peaks with MACS Element
- NGS: Map/Assemble Reads
  - Assemble Reads with SPAdes Element
  - Map Reads with Bowtie Element
  - Map Reads with Bowtie2 Element
  - Map Reads with BWA Element
  - Map Reads with BWA-MEM Element
  - Map Reads with UGENE Genome Aligner Element
  - Map RNA-Seq Reads with TopHat Element
- NGS: Metagenomics Classification
  - Build CLARK Database
  - Build DIAMOND Database Element
  - Build Kraken Database Element
  - Classification Report Element
  - Classify Sequences with CLARK Element
  - Classify Sequences with DIAMOND Element
  - Classify Sequences with Kraken Element
  - Classify Sequences with MetaPhlAn2 Element
  - Ensemble Classification Data Element
  - Filter by Classification Element
  - Improve Classification with WEVOTE Element
- NGS: RNA-Seq Analysis
  - Assemble Transcripts with StringTie Element
  - Assembly Transcripts with Cufflinks Element
  - Extract Transcript Sequences with gffread Element
  - Merge Assemblies with Cuffmerge Element
  - StringTie Gene Abudance Report Element
  - Test for Diff. Expression with Cuffdiff Element
- NGS: Variant Analysis
  - Call Variants with SAMtools Element
  - Change Chromosome Notation for VCF Element
  - Convert SnpEff Variations to Annotations Element
  - Create VCF Consensus Element
  - SnpEff Annotation and Filtration Element
- Transcription Factor
  - Build Frequency Matrix Element
  - Build SITECON Model Element
  - Build Weight Matrix Element
  - Convert Frequency Matrix Element
  - Read Frequency Matrix Element
  - Read SITECON Model Element
  - Read Weight Matrix Element
  - Search for TFBS with SITECON Element
  - Search for TFBS with Weight Matrix Element
  - Write Frequency Matrix Element
  - Write SITECON Model Element
  - Write Weight Matrix Element
- Utils
  - DNA Statistics Element
  - Generate DNA Element

- Workflow Samples
  - Alignment
    - Align Sequences with MUSCLE
    - Extract Consensus as Sequence
    - Extract Consensus as Text
  - Conversions
    - Convert "seq/qual" Pair to FASTQ
    - Convert Alignments to ClustalW

        - Convert UQL Schema Results to Alignment
        - Convert Sequence to Genbank
- Custom Elements
    - CASAVA FASTQ Filter
    - FASTQ Trimmer
    - Dump Sequence Info
    - LinkData Fetch
    - Quality Filter
- Data Marking
    - Marking by Annotation Number
    - Marking by Length
- Data Merging
    - Find Substrings in Sequences
    - Merge Sequences and Shift Corresponding Annotations
    - Search for TFBS
- HMMER
    - Build HMM from Alignment and test it
    - Search Sequences with Profile HMM
- NGS
    - ChIP-Seq Coverage
    - ChIP-seq Analysis with Cistrome Tools
    - Extract Consensus from Assembly
    - Extract Coverage from Assembly
    - Extract Transcript Sequences
    - Quality Control by FastQC
    - De novo Assemble Illumina PE Reads
    - De novo Assemble Illumina PE and Nanopore Reads
    - De novo Assemble Illumina SE Reads
    - De Novo Assembly and Contigs Classification
    - Parallel NGS Reads Classification
    - Serial NGS Reads Classification
    - RNA-Seq Analysis with TopHat and StringTie
    - RNA-seq Analysis with Tuxedo Tools
    - Variation Annotation with SnpEff
    - Call Variants with SAMtools
    - Variant Calling and Effect Prediction
    - Raw ChIP-Seq Data Processing
    - Raw DNA-Seq Data Processing
    - Raw RNA-Seq Data Processing
    - Get Unmappet Reads
- Sanger Sequencing
    - Trim and Align Sanger Reads
- Scenarios
    - Filter Sequence That Match a Pattern
    - Search for Inverted Repeats
    - Find Patterns
    - Gene-by-gene Approach for Characterization of Genomes
    - Group Primer Pairs
    - Intersect Annotations
    - Filter out Short Sequences
    - Merge Sequences and Annotations
    - In Silico PCR Sample
    - Remote BLASTing
    - Get Amino Translations of a Sequence
- Transcriptomics
    - Search for Transcription Factor Binding Sites (TFBS) in Genomic Sequences

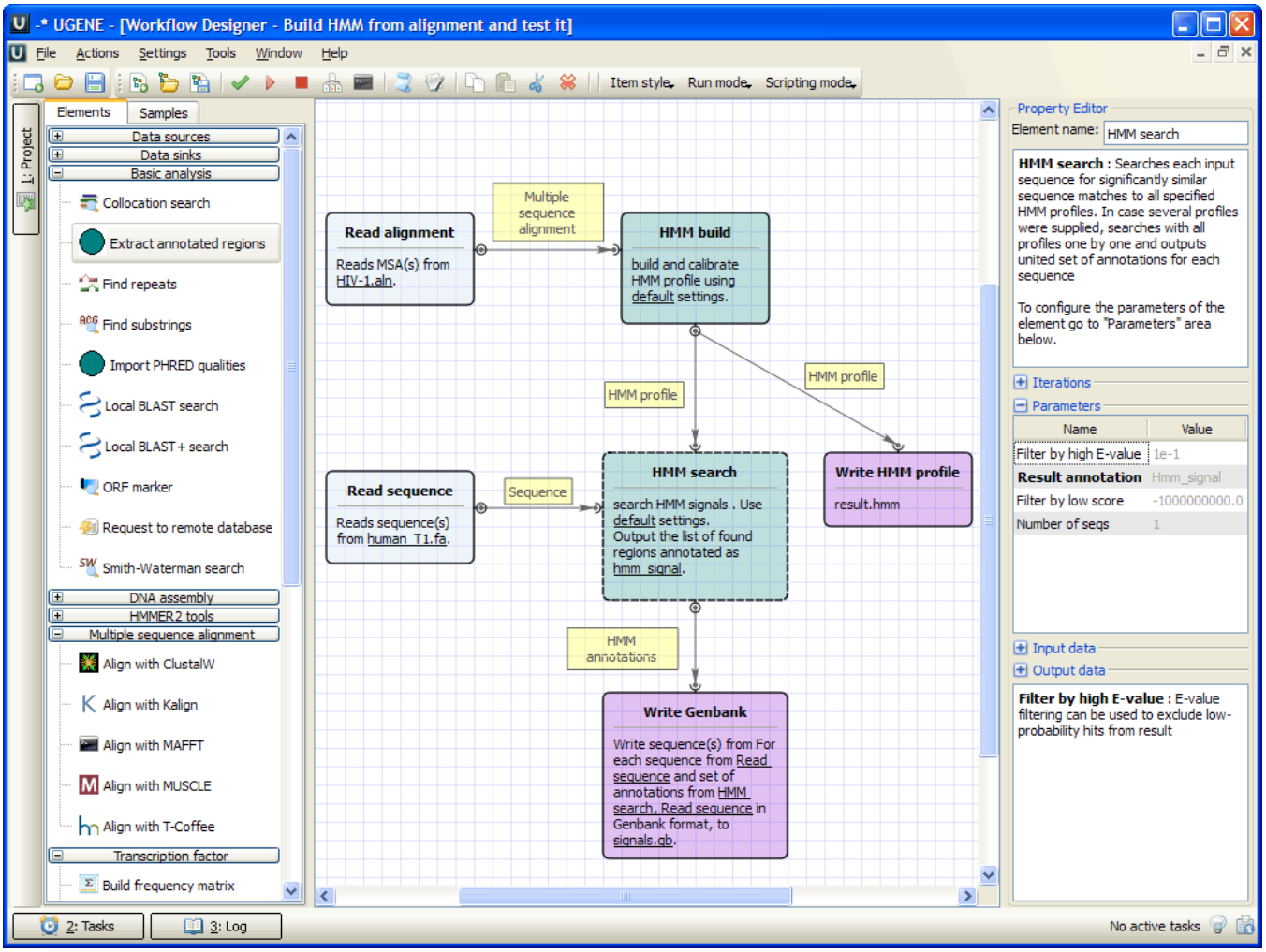
About the Workflow Designer

The *Workflow Designer* allows a molecular biologist to create and run complex computational workflow schemas even if he or she is not familiar with any programming language.

The workflow schemas comprise reproducible, reusable and self-documented research routines, with a simple and unambiguous visual representation suitable for publications.

The workflow schemas can be run both locally and remotely, either using a graphical interface or launched from the command line.

The elements that a schema consists of corresponds to the bulk of algorithms integrated into UGENE. Additionally, you can create custom workflow elements.



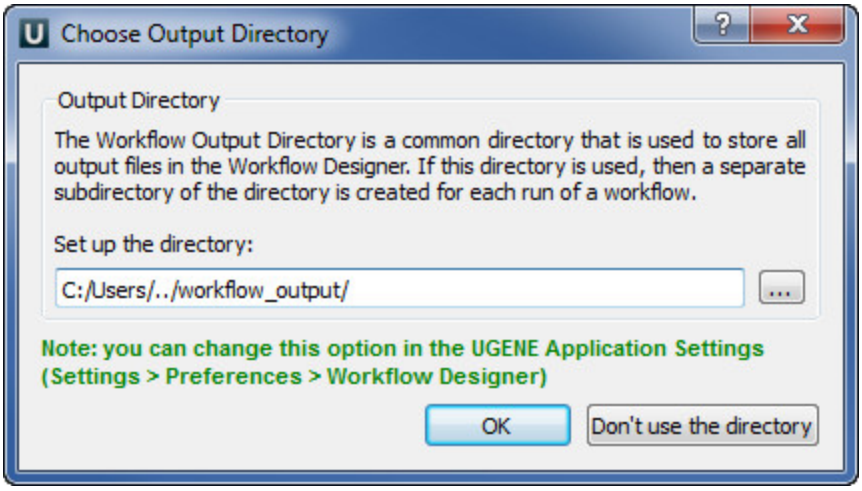
Introduction

This chapter describes the Workflow Designer key elements and provides an example on how to create and run a simple *workflow*.

- Launching Workflow Designer
- Workflow Designer Window Components
- Workflow Elements and Connections
- Managing Parameters
- UGENE Components and Workflow Designer
- How to Create and Run Workflow
- How to Use Sample Workflows

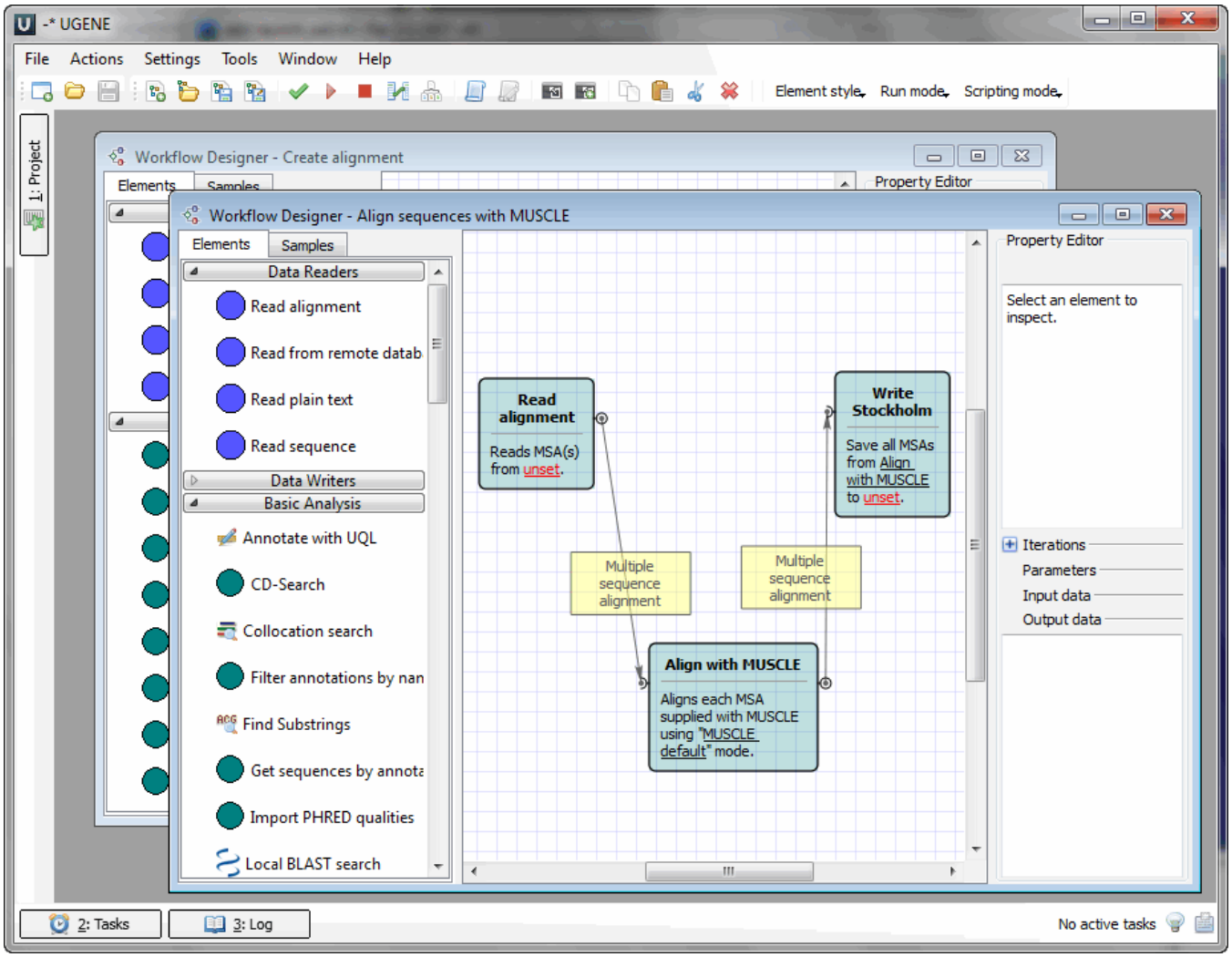
Launching Workflow Designer

To launch the Workflow Designer select the *Tools Workflow Designer* item in the UGENE main menu. The following Choose Output Directory dialog appears:



The output directory is a common directory that is used to store all output files in the Workflow Designer. If this directory is used, then a separate subdirectory of the directory is created for each run of a workflow. You can change this option in the *Application Settings* dialog.

The tool provides multi-window user interface, so you can open and use at the same time as many Workflow Designer windows as you need.



Workflow Designer Window Components

Each Workflow Designer window consists of:

Palette

The *Elements* tab of the palette contains *workflow elements* for most algorithms intergrated in UGENE and sets of common input / output routines. The elements are grouped into categories that reflect their uses and features. The *Samples* tab of the palette contains examples of *workflow*.

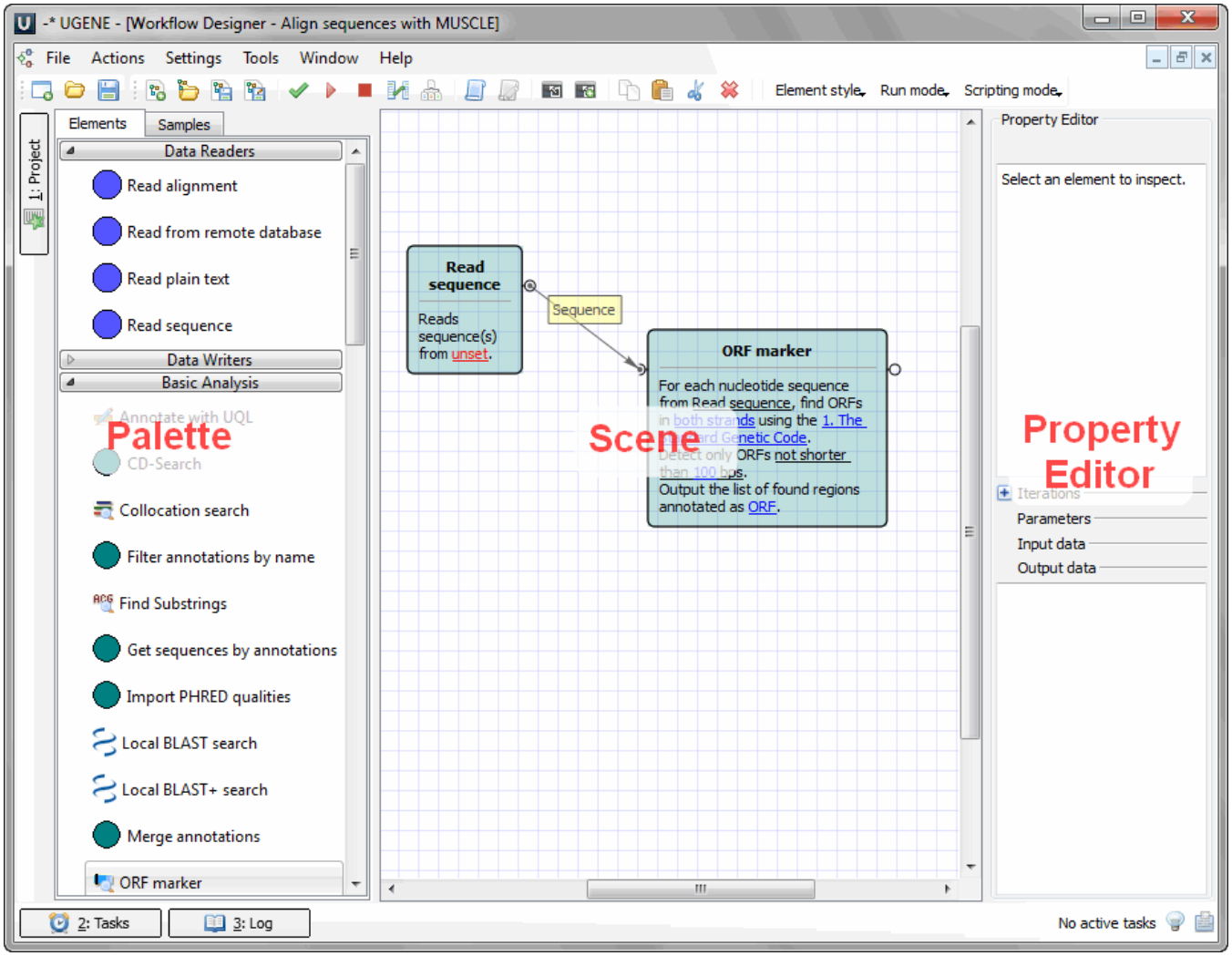
Scene

The main drawing scene is the place where the workflow elements are constructed into a workflow.

Property Editor

Provides information about a currently selected workflow element and allows configuring it.

On the image below you can see these components in a maximized Workflow Designer window:



All these components are resizable and can be adjusted to individual needs.

Workflow Elements and Connections

The *Scene* is initially empty and you start with creating a workflow on it:

workflow

A workflow is a visual representation of the dataflow. It consists of workflow elements and their connections.

workflow element

An element of a workflow. Different elements are used to read data from files on disk, perform some algorithms and to write data to files on disk. Each element contains one or several input and output ports.

element connection

Connection between two elements specifies that data in output port of one element should be passed to a matching input port of another element.

input port

An input port of an element is used to collect data from another element. A workflow element may have several input ports. On the Scene such port is displayed as a right semicircle.

output port

An output port of an element is used to provide data to another element. A workflow element may have one output port or none. On the Scene the port is displayed as a left semicircle.

slot

Each port has one or several slots. A slot is the smallest passageway to transfer the workflow data through. It has a certain type (e.g. "Sequence", "Set of annotations", etc.). So, for example, only sequence data can be passed through a sequence slot.

Thus, an input port has one or several **input slots**. These slots specify data that are expected as input by the element. An output port has one or several **output slots**. These slots specify data that the element produces.

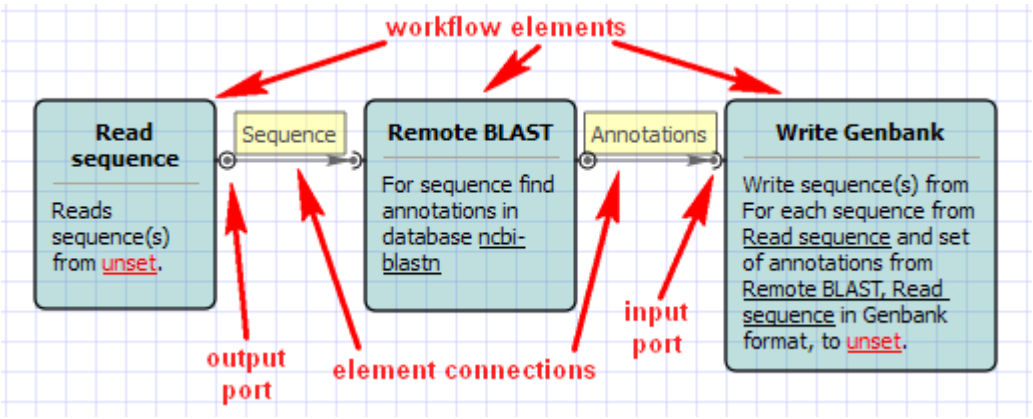
In a workflow, an element usually have access to slots of the connected elements, located in the workflow before it.

message

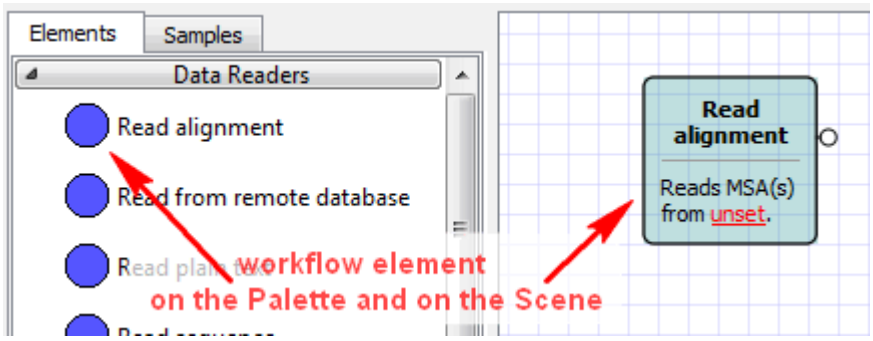
A message is a single data chunk, transferred from an output slot of one element to an input slot of another element. The slots must have the same type to make the transfer possible.

The Scene is initially empty and you start with creating a workflow on it:

See an example of a workflow on the image below:

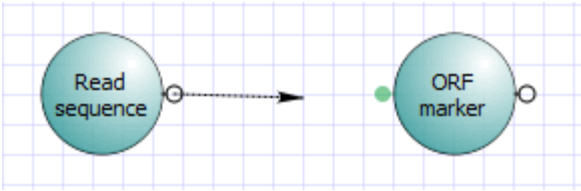


Your first step is to *add* necessary workflow elements, for example, by dragging them from the *Palette* to the Scene:



The added element can be moved around on the Scene by dragging it and can be resized by dragging its borders. Read chapter *Manipulating Element* to learn what else you can do with workflow elements.

If you have two elements with matching output and input ports, you can make the connection by dragging the arrow between the ports:



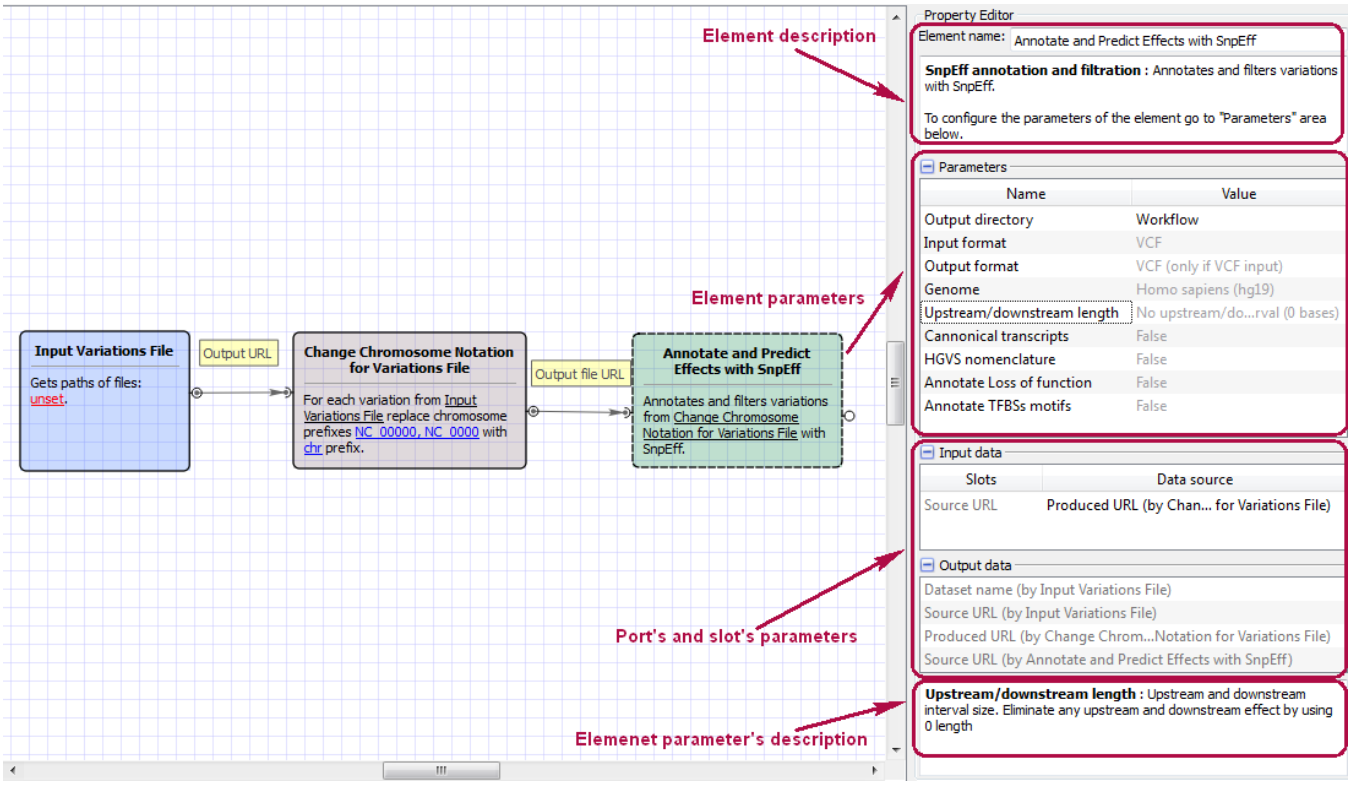
All matching ports of available processes are highlighted while you drag the arrow, besides the arrow sticks to a near match when you drag closer. If an element has a sole matching port, you can just drop the arrow on the element itself to create a correct connection.

Once created, a connection will follow movements of the linked elements; you cannot redirect or reshape the connection arrow but only remove it. You can move the port around an element that it belongs to by dragging it and holding the Alt key at the same time. This is helpful to fine-tune visual layout of a workflow.

Managing Parameters

When you select an *element* on the *Scene* the *Property Editor* displays detailed information about it: it's name, description, parameters, *input* and *output* po rts, etc. To change the name of the element displayed on the Scene edit the *Element name* value.

All the parameters available for the element are displayed in the *Parameters* area. Some parameters must have a value, they are displayed in bold. Notice, that when you select a parameter, it's description is shown below. To modify a value click on it. Depending on the parameter's type you may be required to either input a value or browse for a file(s). Also you can configure slots of a connected input port by selecting different (matching) data available through the dataflow. More advanced users can use their own scripts to set a parameter's value, read chapter *Using Script to Set Parameter Value* to learn more. The image below shows the *Property Editor*:



For *Data Readers* you can manipulate with file(s) or directory(ies) with a help of dataset(s):



Also, to remove files from dataset you can select it and press the *Delete* button.

For Data Writers, if the *Output file* parameter is empty, UGENE will generate output files names automatically. You can use the *Output file suffix* parameter to manipulate it.

UGENE Components and Workflow Designer

This paragraph provides an overview of uf UGENE components that affect your work with the Workflow Designer.

- Task View, Notifications and Log View
- Actions Menu
- Toolbar
- Context Menus
- Application Settings

Task View, Notifications and Log View

When a workflow is executed in the Workflow Designer a **task** is created.

Task View

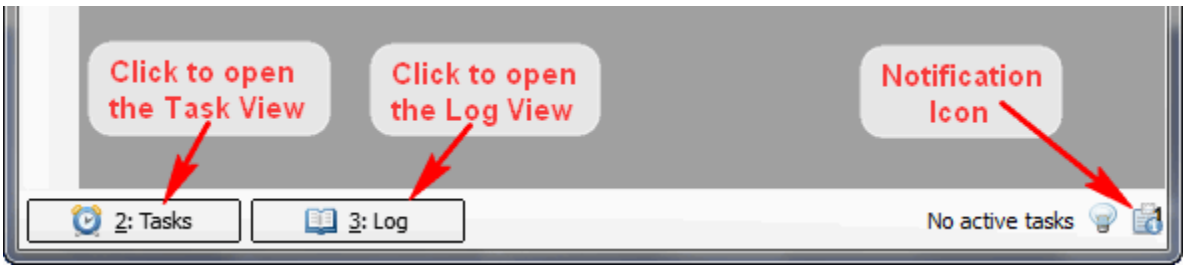
Here you can see the tasks currently executed in UGENE.

Notification Icon

When a task has finished it's execution, a notification is pop up. At any time you can watch the last notifications by clicking the *Notification Icon*.

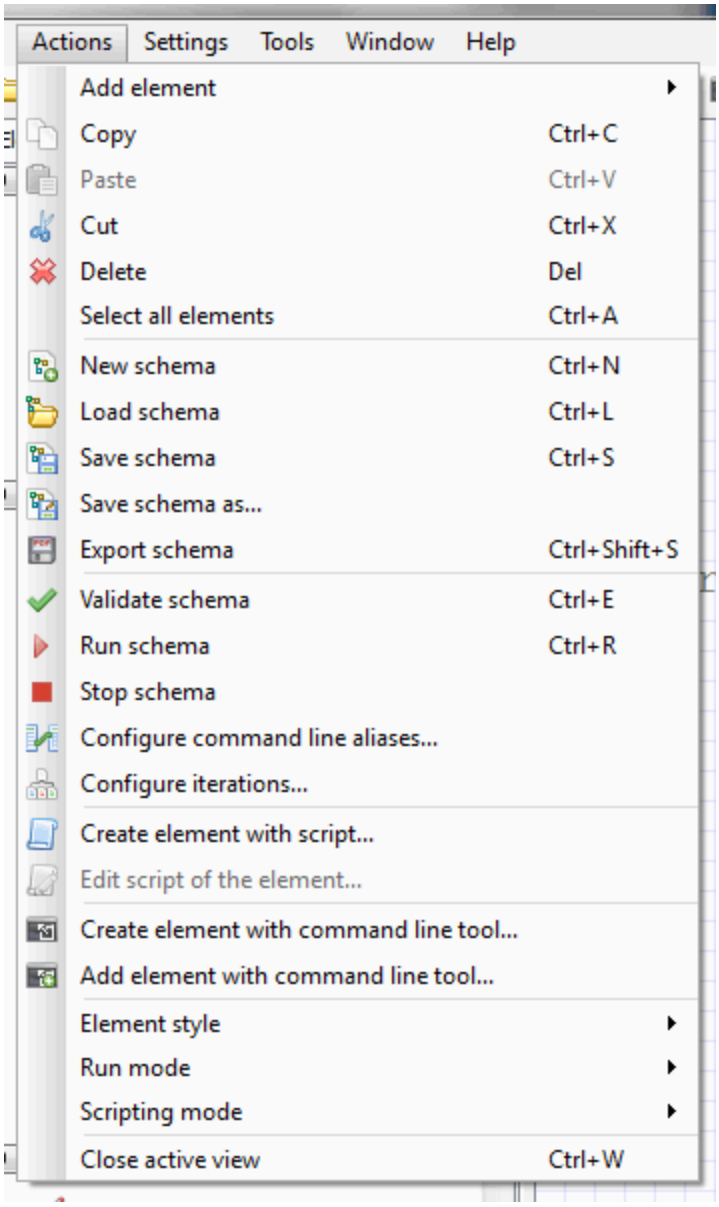
Log View

Here you can see UGENE logs.



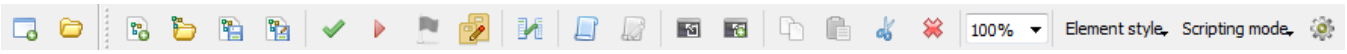
Actions Menu

When a Workflow Designer window is active, all standard actions to work with workflow are available from the *Action* main menu:



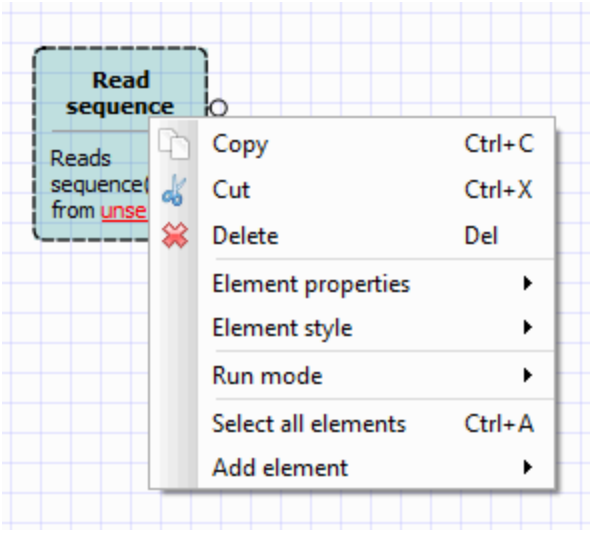
Toolbar

Most common actions are available on the main toolbar:



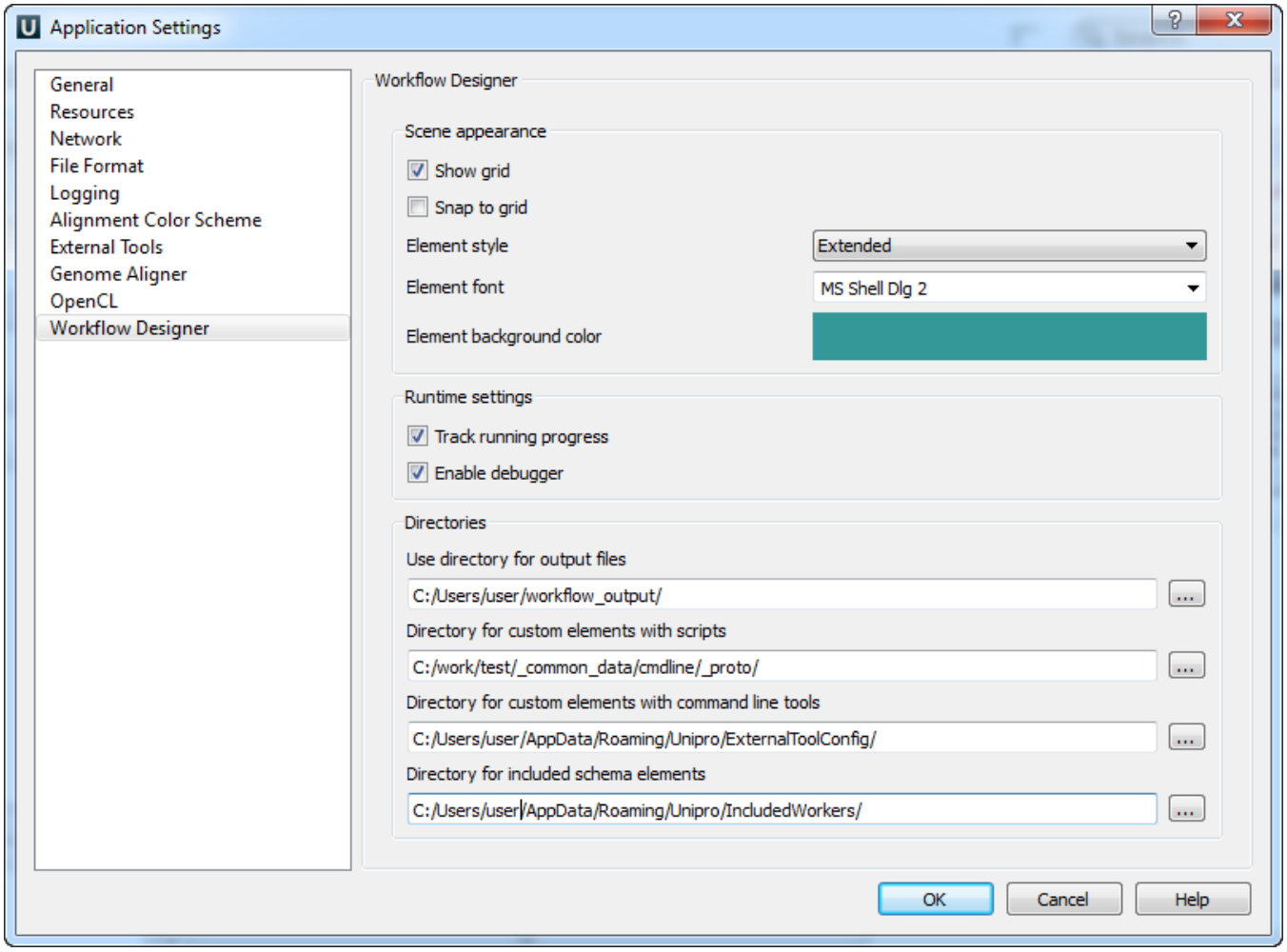
Context Menus

Some features are also available through context menus over corresponding areas, e.g.:



Application Settings

To change common Workflow Designer setting select the *Settings Preferences...* main menu item and select the *Workflow Designer* tab in the opened *Application Settings* dialog.



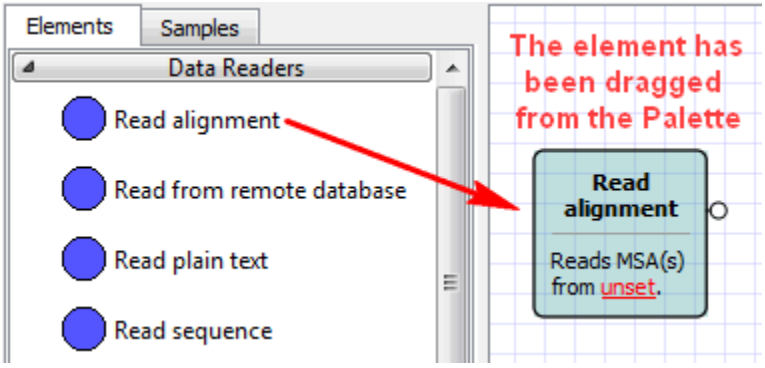
How to Create and Run Workflow

- Select *Tools –> Workflow Designer* or *File->New worflow* items in the main menu.

  Result: The Workflow Designer window appears.

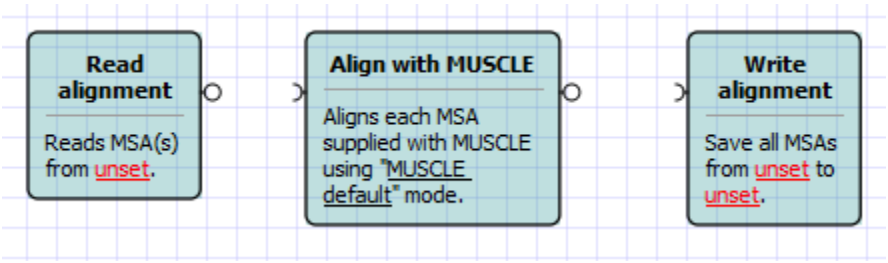
- On the *Elements* tab of the *Palette* find the *Read alignment* element. It is located in the *Data sources* group and drag it to the *Scene*.

  Result: The element is shown on the Scene.



- Repeat the previous step for the *Write Alignment* element from the *Data sinks* group and for the *Align with MUSCLE* element from the *Multiple sequence alignment* group.

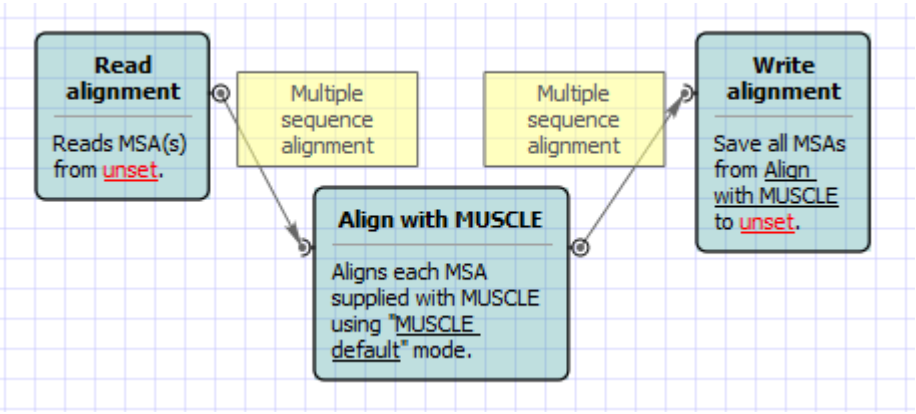
  Result: All three elements are on the Scene.



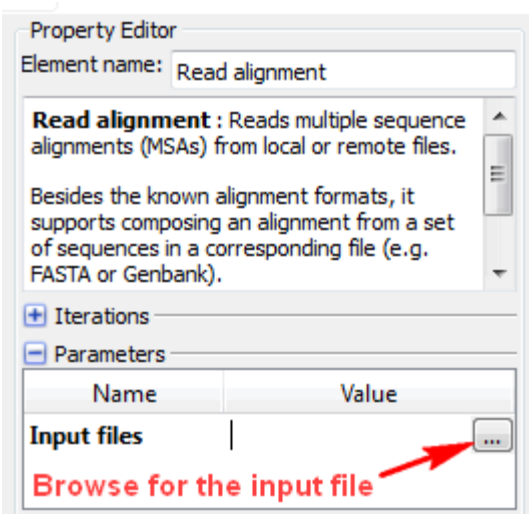
- Connect the elements:

  - Drag an arrow from the *output port* of the *Read alignment* element to the *Align with MUSCLE* element.
  - Drag an arrow from the output port of the *Align with MUSCLE* element to the *Write alignment* element.

  Result: The elements are connected with arrows.



- Select the *Read alignment* element. In the *Parameters* area of the *Property Editor* click on the *Value* column of the *Input files* parameter:



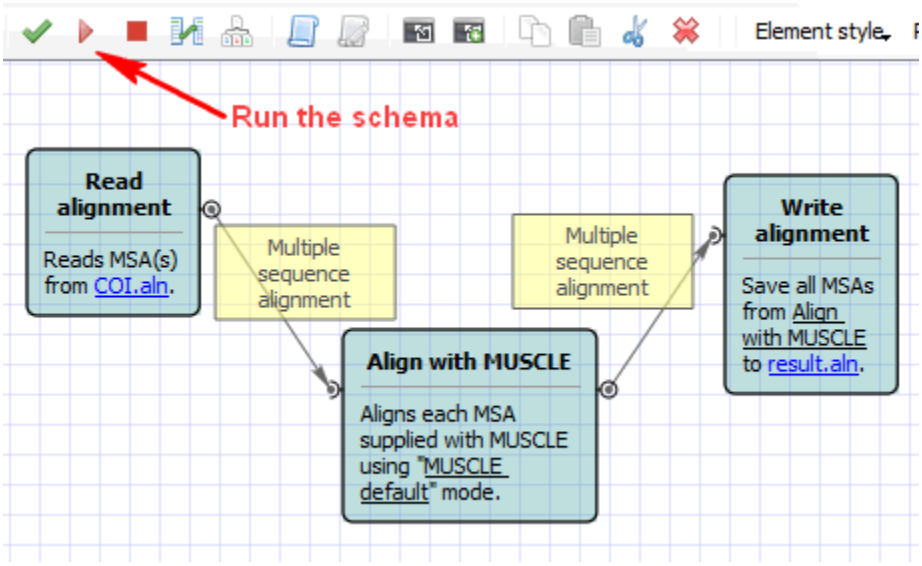
- And browse for an input file, e.g.Select the $UGENE\data\samples\CLUSTALW\COI.aln file.

  Result: The *Input files* value is set to the file's path.

- Select the *Write alignment* element and set the *Output file*, e.g. you can just enter result.aln.

  Result: All required workflow parameters are set.

- Click the *Run workflow* button on the toolbar.



  Result: After the workflow has run, a blue notification has pop up.

- Open the the result.aln file in UGENE.

  Result: The file has been opened. It contains the result of the alignment with MUSCLE.

How to Use Sample Workflows

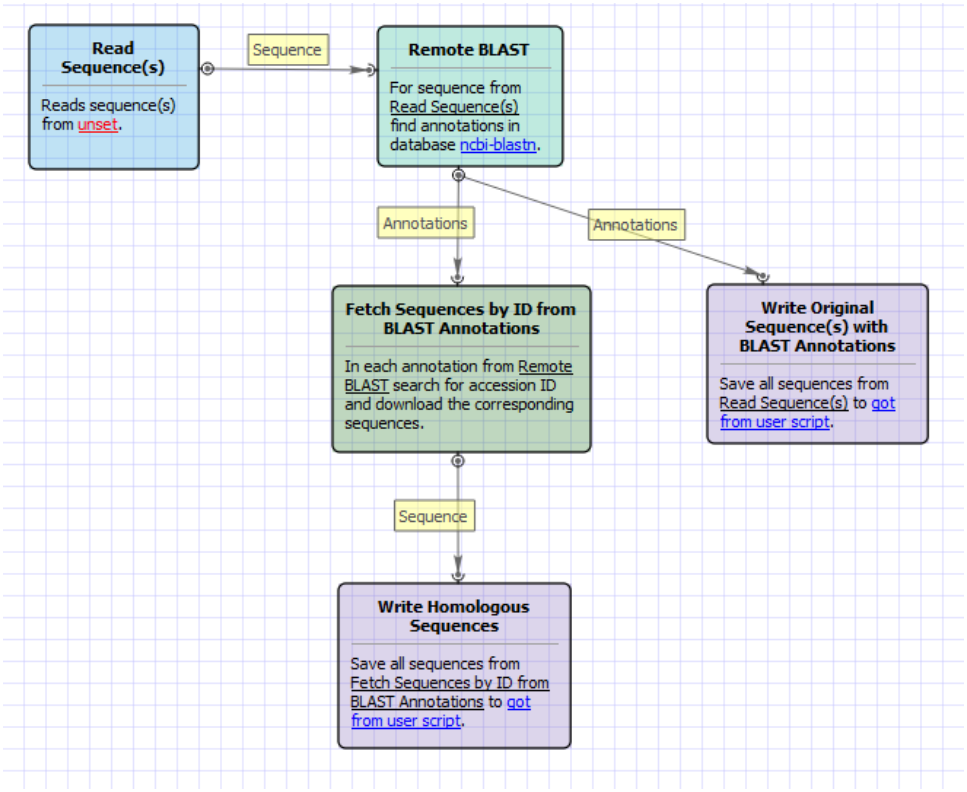
UGENE Workflow Designer contains a set of sample workflows that help a biologist to solve certain tasks for multiple input files or datasets at the same time. The list of samples can be found in the Workflow Samples section of the documentation.

To use a sample:

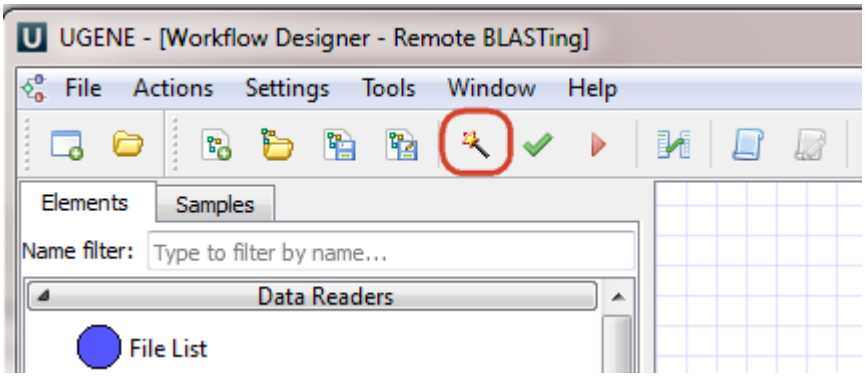
1. Start the Workflow Designer by selecting "Tools > Workflow Designer" in the main menu of the UGENE window.
   See also: the paragraph about launching the Workflow Designer.

2. Select the "Samples" tab on the Workflow Designer palette, i.e. on the left side of the opened window.
   See also: the tab is described in the Workflow Samples section.

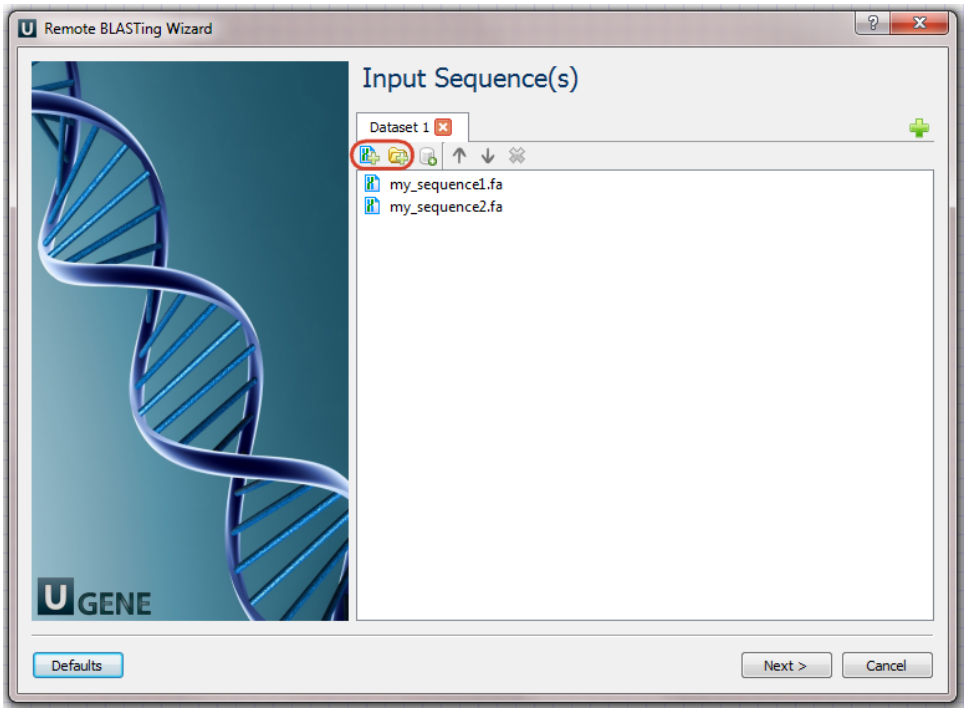
3. Double-click on the required sample.
   The workflow will be opened and shown on the Workflow Designer scene, i.e. the center area of the window.
   For example, a workflow for doing BLAST and getting the results from the NCBI server is shown below.



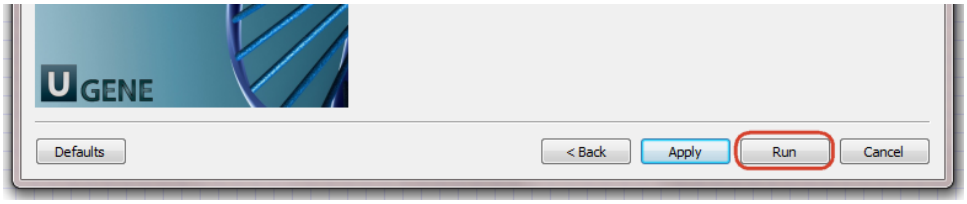
4. Select the wizard button on the Workflow Designer toolbar (the button is marked on the image below) to start the wizard for the workflow.
   Additional technical details: A wizard can be used to configure all the parameters for the workflow more easily. The other way to configure the parameters is by editing them in the Property Editor. A wizard is not available for a newly created workflow, but it can be added by editing the workflow file.



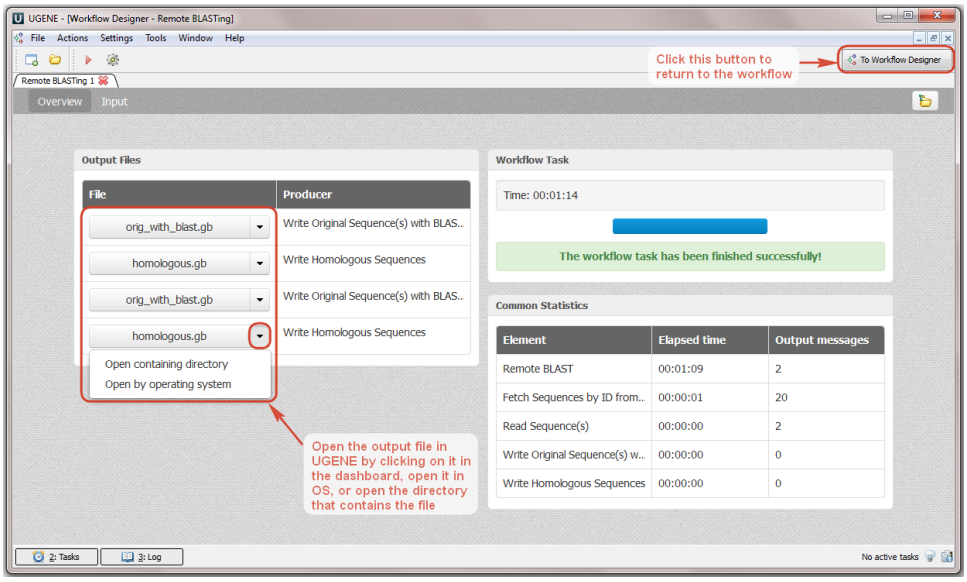
5. Input the required data. The input varies very much on the workflow that has been selected on step 3 (see above).
   For example, in case of the remote BLAST workflow, at least one sequence is expected to be input. On the image below two sequences were input for the workflow. Buttons that can be used to add different files or even folders with files are also marked on the image.



6. Optionally, modify the workflow parameters on other pages of the wizard.
7. Click the "Run" button on the last wizard page to run the workflow.
   For example:



8. Launching of the workflow opens the dashboard. Wait until the workflow is finished. The output files will be available in the corresponding section of the dashboard.
   For example, in case of the remote BLAST workflow, the dashboard will look as follows:



Manipulating Element

You can add new *workflow element* to the *Scene*, copy, cut, paste or delete it. Also you can select all elements currently presented on the Scene.

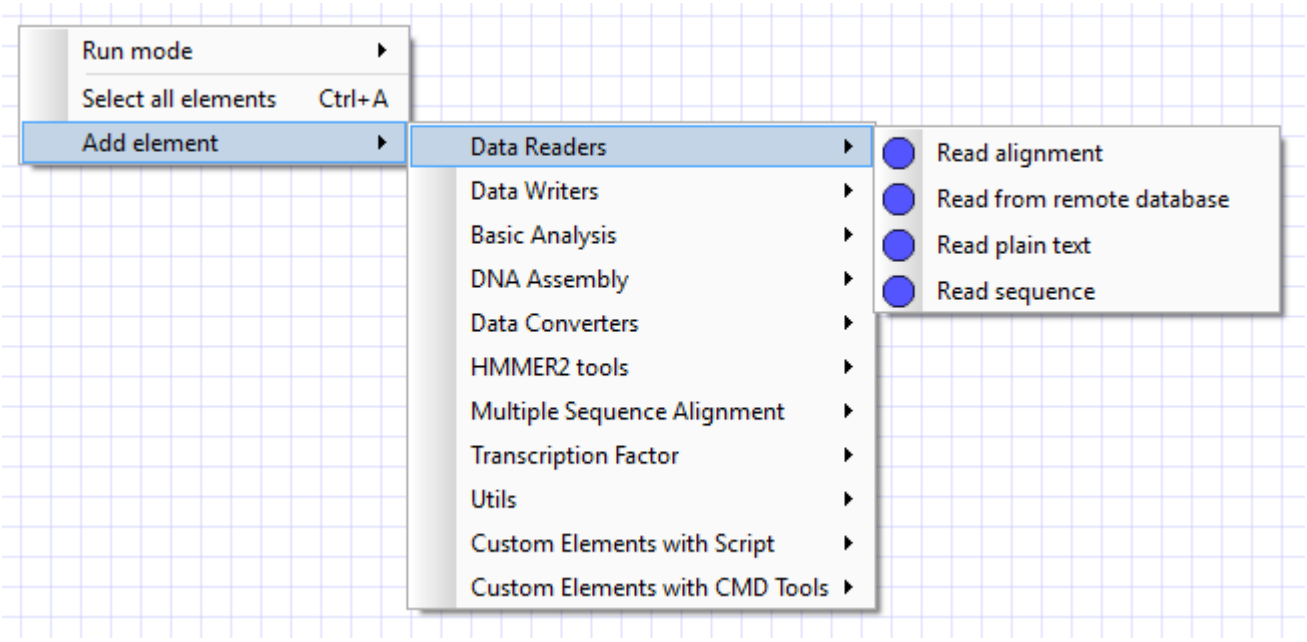
- Adding Element
- Copying Element
- Pasting Element
- Cutting Element
- Deleting Element
- Selecting All Elements on Scene

Adding Element

There are several ways to add an *element* to the *Scene*.

The easiest way is to drag the required element from the *Palette* to the Scene. Or you can just click on the element on the Palette and then click somewhere on the Scene.

Also you can select an element in the *Add item* submenu of the *Actions* main menu or of the Scene context menu, for example:



When the required element is selected click somewhere on the Scene to insert it.

Copying Element

To copy one or several *workflow elements* select them on the *Scene*. Note, that you can hold the Ctrl key to select several elements. Then choose the *Copy* item in the *Actions* main menu or in a selected element context menu.

The Ctrl+C hotkey is also available for this action.

Now you can *paste* these elements somewhere on the Scene.

Pasting Element

You can paste *workflow elements* that have been *cut* or *copied*.

To do it choose the *Paste* item in the *Actions* main menu or in the *Scene* context menu.

Or use the Ctrl+V hotkey to paste the elements.

Cutting Element

To cut one or several *workflow elements* select them on the *Scene*. Choose the *Cut* item in the *Actions* main menu or in a selected element context menu.

The Ctrl+X hotkey is also available for this action.

Now you can *paste* these elements.

Deleting Element

Select one or several *workflow elements* on the *Scene* that you want to delete. Then choose the *Delete* item in the *Actions* main menu or in a selected element context menu.

The hotkey for this action is Del.

Selecting All Elements on Scene

To select all *workflow elements* presented on the *Scene* choose the *Select all elements* in the *Actions* main menu or in the Scene context menu.

Or use the Ctrl+A hotkey.

Manipulating Workflow

You can create a new *workflow*, save it and then load it again.

The designed workflow can be dispalyed in a neat self-describing layout and exported to PDF document, rastor or vector image with publication-ready quality.

You can validate created or modified workflow before running it.

If you need, you can stop a workflow execution.

- Creating New Workflow
- Loading Workflow
- Saving Workflow
- Exporting Workflow as Image
- Validating Workflow
- Running Workflow
- Dashboard
- Stopping and Pausing Workflow

Creating New Workflow

To create a new *workflow* select the *File->New workflow*, *Actions New workflow* items in the main menu or *New workflow* toolbar button.

Or press Ctrl+N.

Loading Workflow

To load a workflow select the *Actions Load workflow* item in the main menu or *Load workflow* toolbar button.

Or press Ctrl+L.

Hint

You can load a workflow by dragging the workflow file (e.g. with .uwl extension) to the UGENE window.

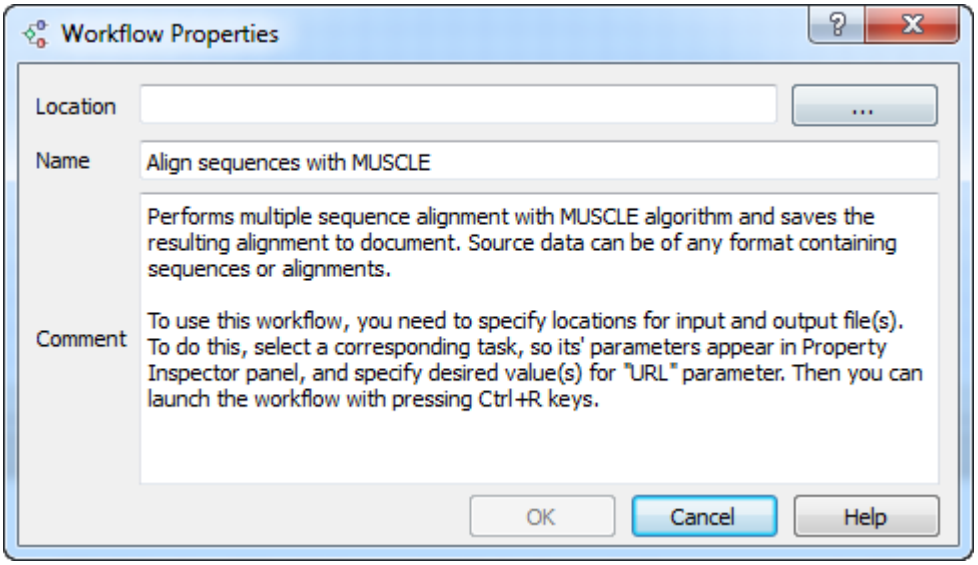
Saving Workflow

Choose *Actions Save workflow* item in the main menu or *Save workflow* toolbar button to save a workflow. The workflow is saved to a file of native UGENE format (with.uwl extension).

The format is human-readable, you can find it's description in chapter *Workflow File Format*.

There is Ctrl+S keyboard shortcut for this action.

If you save a workflow for the first time the *Workflow properties* dialog appears:



Here you can browse for the workflow file *Location* and specify the workflow *Name* and *Comment*.

Once a workflow has been saved, it can be *loaded*. If you modify the loaded workflow and save changes, then corresponding .uwl file is modified.

To save the workflow with different properties choose the *Actions Save workflow as* item in the main menu and specify the required settings in the *Workflow properties* dialog.

Exporting Workflow as Image

Workflow workflow can be exported as:

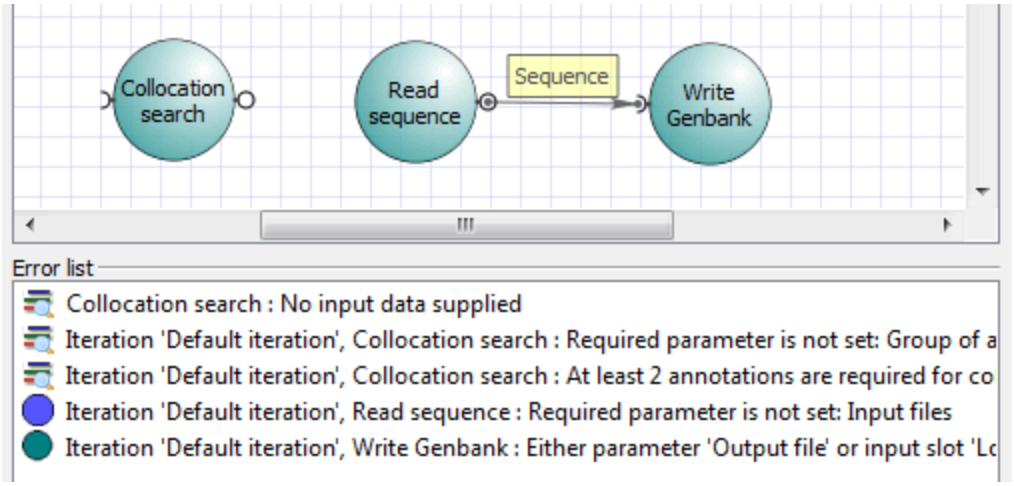
- Raster image (*.png, *.bmp, *.jpg, *.jpeg, *.ppm, *.xbm, *.xpm)
- Vector image (*.svg)
- Portable document (*.pdf, *.ps)

To export a workflow select the *Actions Export workflow as image* item in the main menu or use the Ctrl+Shift+S keyboard shortcut. *Export Image* dialog will appear. Enter a file name and choose the file type.

Validating Workflow

Before a workflow can be actually executed, it should be verified by the Workflow Designer. During the process of verification the Workflow Designer checks if there are errors in the dataflow logic or unspecified parameters and can provide a user with optimization or layout hints. If no errors were found, the workflow is valid to be *run*.

You can request workflow validation at any stage of workflow design. To do it choose the *Actions Validate workflow* item in the main menu or *Validate workflow* toolbar button or invoke it by pressing Ctrl+E. A list of identified issues and warnings if any, or a notification of validation success will appear.

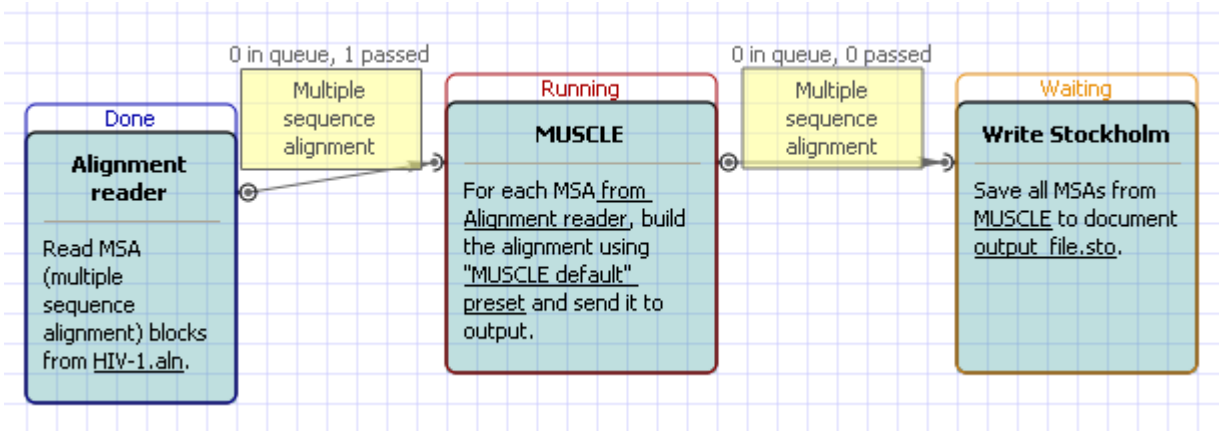


Double-clicking on items in the list selects the faulty element/iteration.

Running Workflow

Once you are satisfied with the designed workflow and have it configured, click the *Run workflow* button on the toolbar (alternatively, you can select the *Acti ons Run workflow* item in the main menu or launch it by pressing Ctrl+R). The workflow gets verified and scheduled for background execution. If you continue editing the workflow, this will not affect the launched execution. You can control the workflow execution via the *Task View*: watch progress, cancel it, etc. Upon completion, the Workflow Designer produces a *dashboard* with a summary report. The report displays status of each iteration execution and provides other details.

Note, that you can see the progress of a workflow execution in a Workflow Designer window by checking the *Track running progress on diagram* option in the *Applcation Settings* dialog:



Dashboard

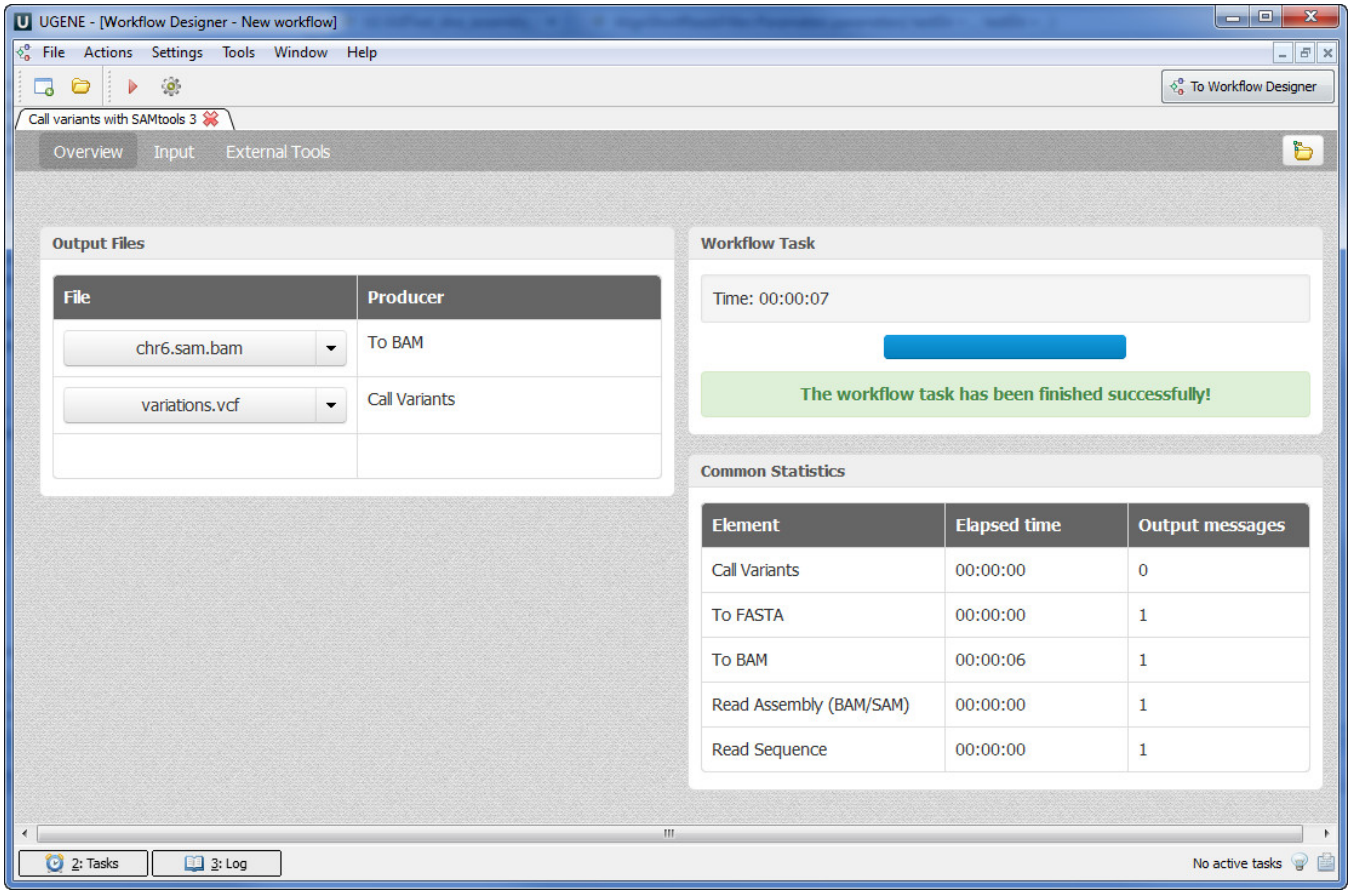
The dashboard is a central place to view the overall progress of a single workflow. Every dashboard contains two tabs:

- Overview tab
- Input tab

If a workflow uses external tools the following tab appears on dashboard:

- External Tools tab

The following picture shows the sketch of the the dashboard:



- Dashboard Window Components
- Using Dashboard

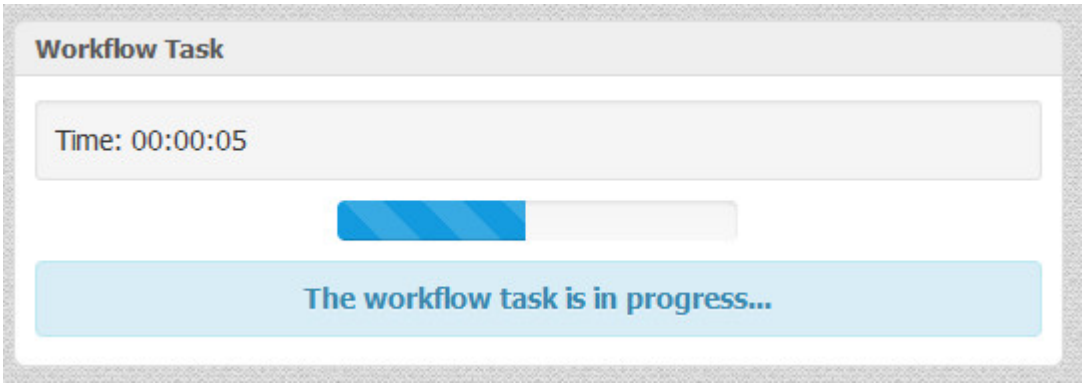
Dashboard Window Components

Overview tab

"Workflow Task" widget

It contains:

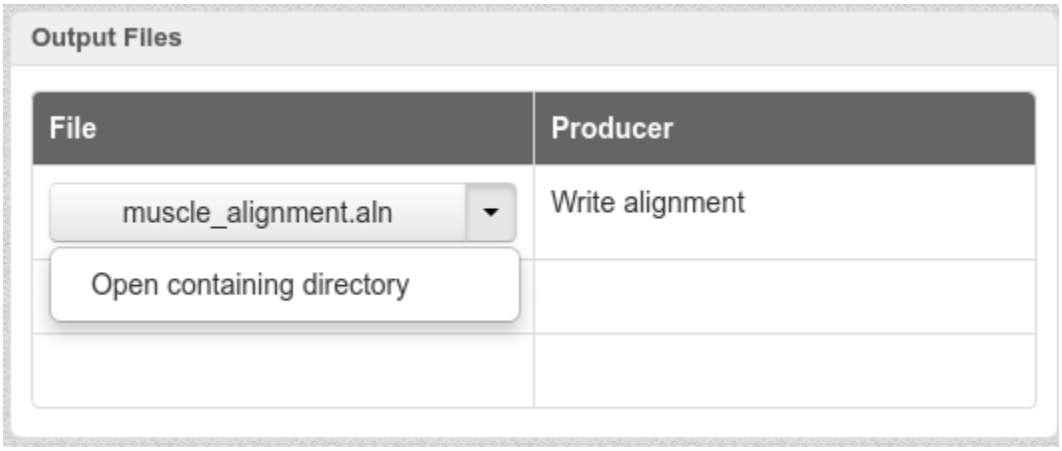
- the workflow working time;
- the workflow running progress;
- the workflow task status: failed, finished successfully, running and etc.;



"Output Files" widget

It contains a table with the information about all created output files. The table columns are:

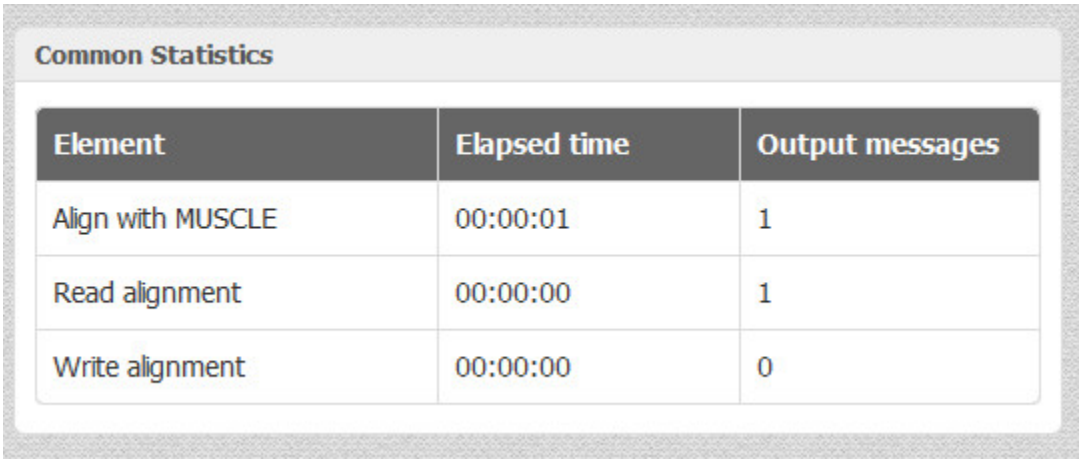
- clickable file name (with a help of the arrow on the right side of the file name you can open the file containing directory or open the file by operating system);
- the name of the workflow element that has produced the file;



"Common Statistics" widget

It contains a table with common statistic for each workflow element in the workflow. The table columns are:

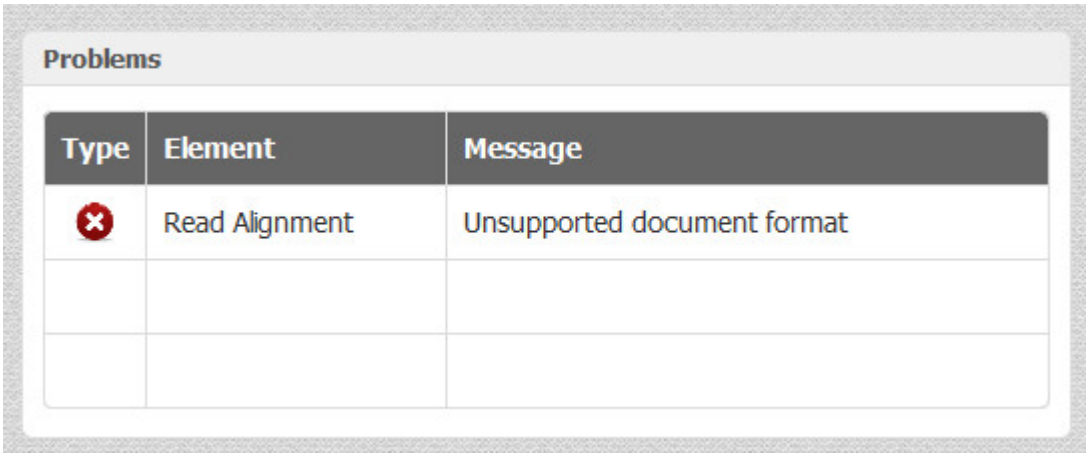
- name of the workflow element;
- time of the workflow element execution;
- the number of messages that has been retrieved;



"Problems" widget

It contains a table with problems. The table columns are:

- problems type (warning, error and etc.)
- name of the element with problem
- error message

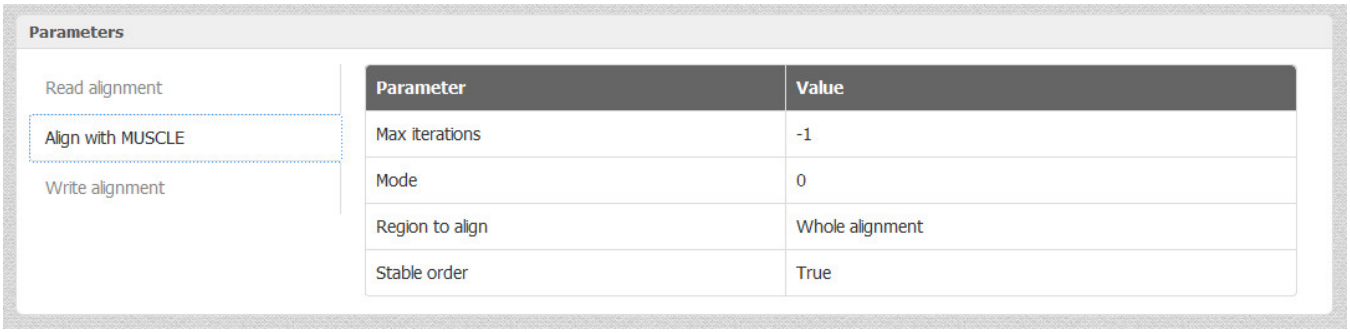


Input tab

"Parameters" widget

It contains a table with common statistic for each workflow element's parameter in the workflow. The table columns are:

- names of the workflow elements;
- names of the workflow parameters;
- values of the workflow parameters;
- clickable file name values of the worflow parameters (here you can open the file containing directory or open the file by operating system);

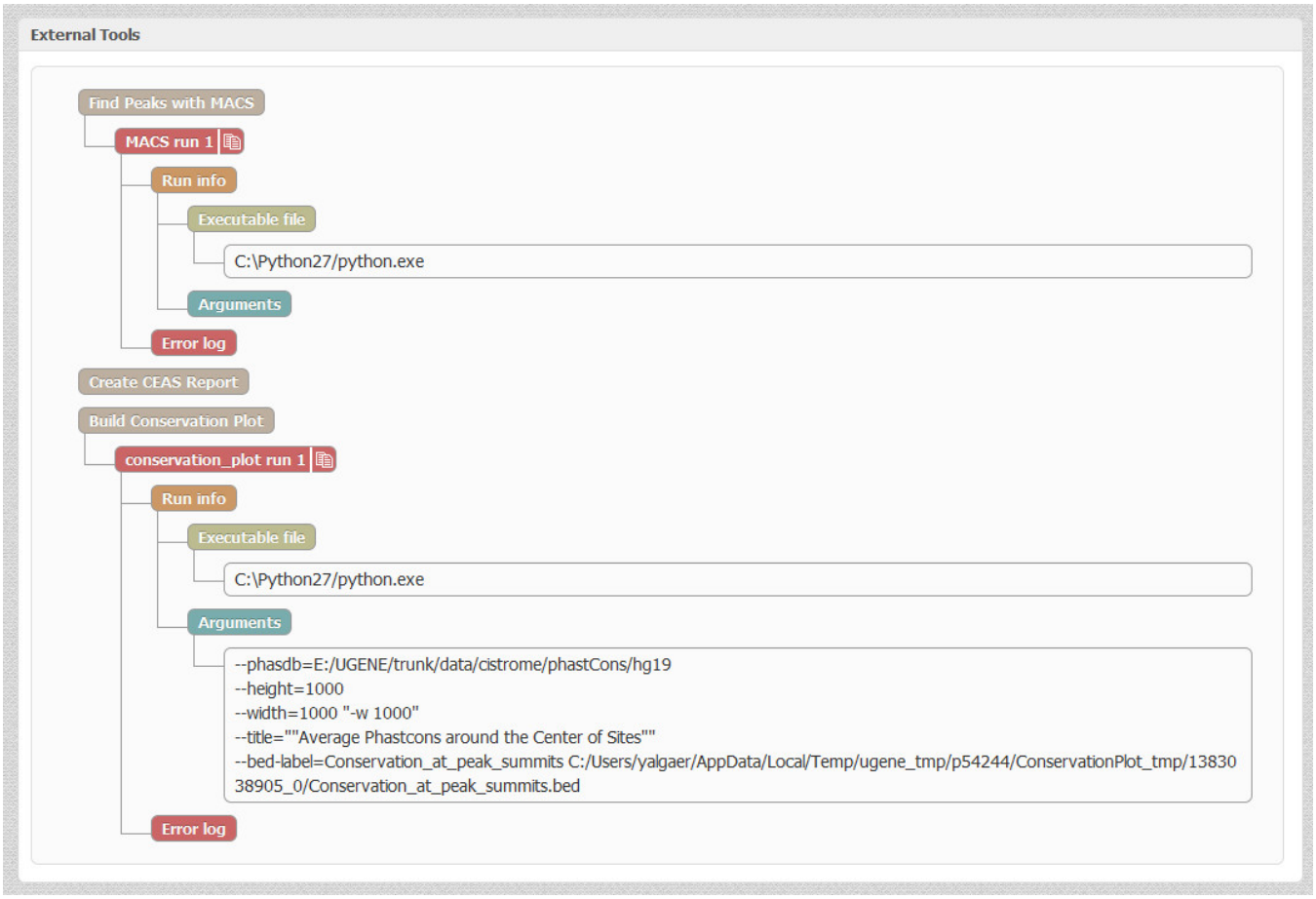


External Tools tab

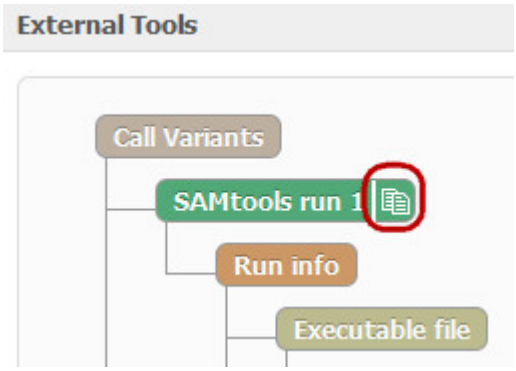
"External Tools" widget

It contains information about external tools. There are:

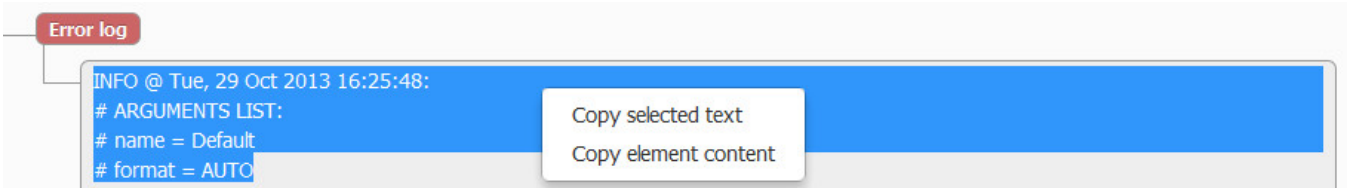
- names of the external tools;
- executable file of the external tool;
- arguments of the external tool;



To copy external tool run string click the following button:

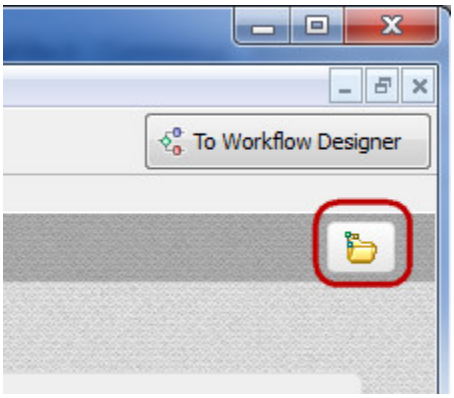


With a help of the context menu of this widget you can copy selected text from the dashboard or copy all text of the active element:

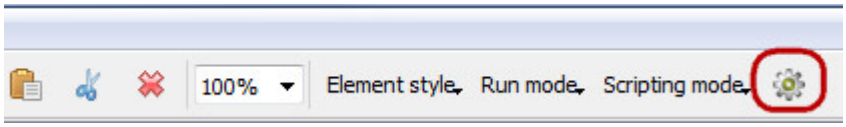


Using Dashboard

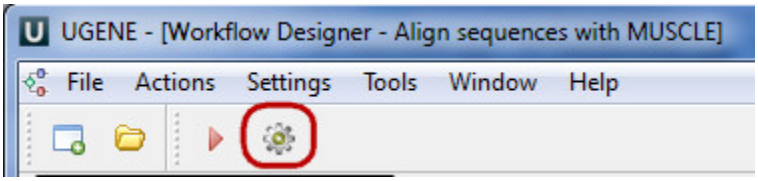
For each workflow which has been runned new dashboard will be opened. This dashboards will be saved in the *selected directory*. Also you will see this dashboard after UGENE will be runned again. Furthermore you can open the original workflow for your results by clicking on this button:



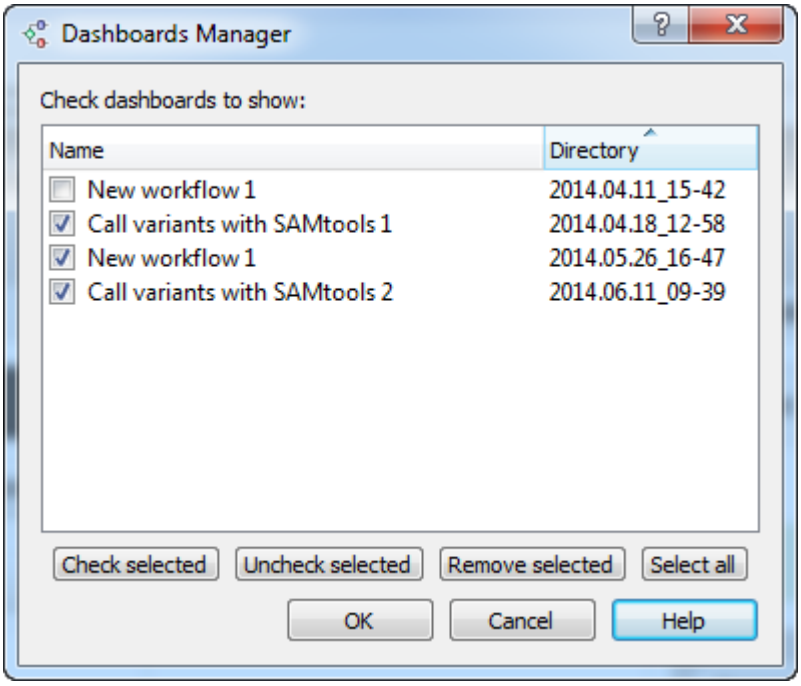
To remove or to load a dashboard click to the *Dashboards manager* button on the *Workflow Designer* main toolbar:



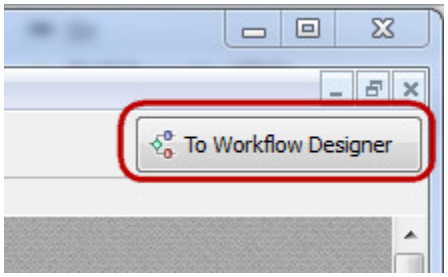
or on the *Dashboard* toolbar:



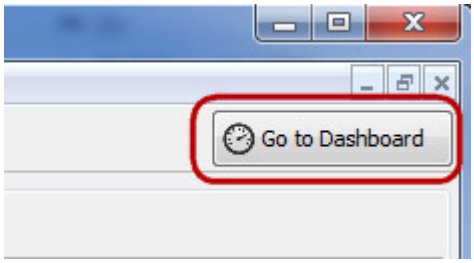
The following dialog appears:



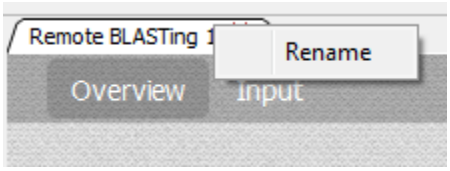
To see a dashboard select it and check it's checkbox. To remove a dashboard select it and click the *Remove selected* button. Click OK button. The selected and checked dashboards appears in the *Dashboard* main window. You can go back to the *Workflow Designer* main window from *Dashboard* wind ow by clicking on this button:



And go back to the *Dashboard* main window from *Workflow Designer* main window by clicking on this button:

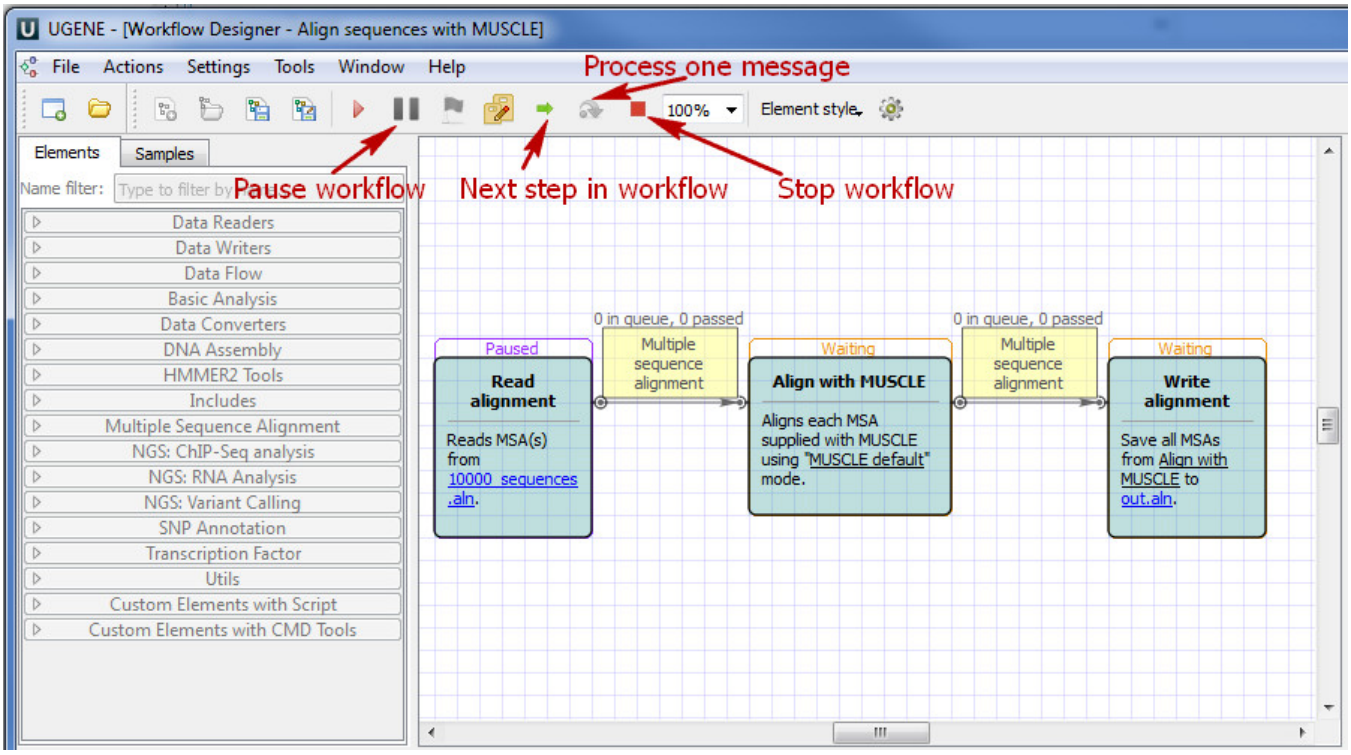


To rename a dashboard tab use the following context menu:



Stopping and Pausing Workflow

A workflow execution can be stopped, paused and run step by step. After you run workflow the following toolbar buttons appears:



With a help of these buttons you can:

Pause workflow - pause the runned workflow.

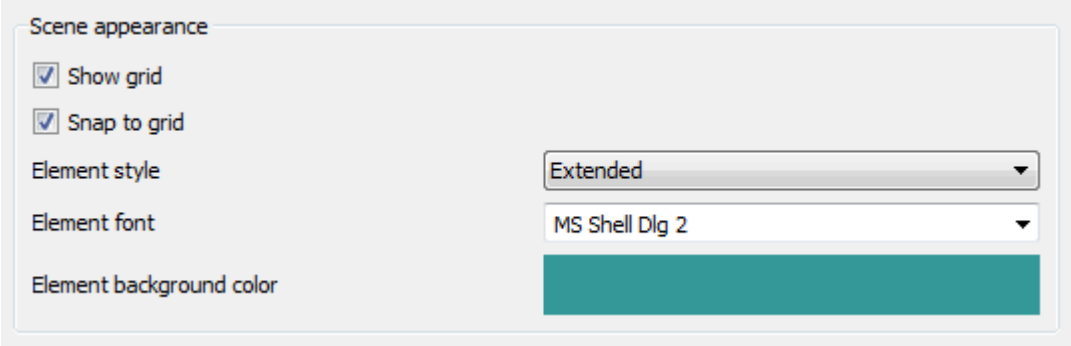
Next step in workflow - do the next step in workflow.

Process one message - do the first queue message step of the selected element in workflow. It is active if an element selected.

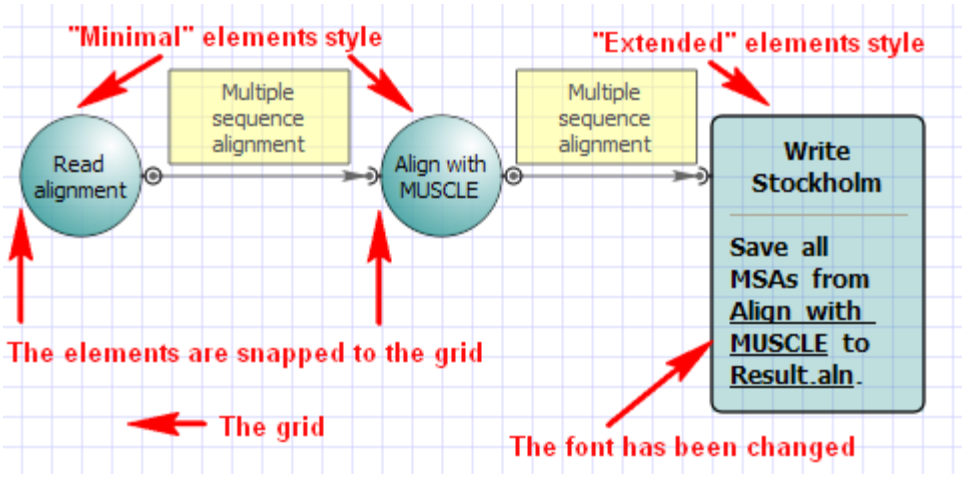
Stop workflow - cancel workflow process.

Changing Appearance

The default setting that influence the Workflow Designer appearance can be set in the *Application Settings* dialog. The parameters are shown in the image below:



The *Show grid*, *Snap to grid*, *Element style,* and *Element font* parameters affect the view of a workflow:



To change the appearance of a particular element use it's context menu submenus *Element properties* and *Element style*.

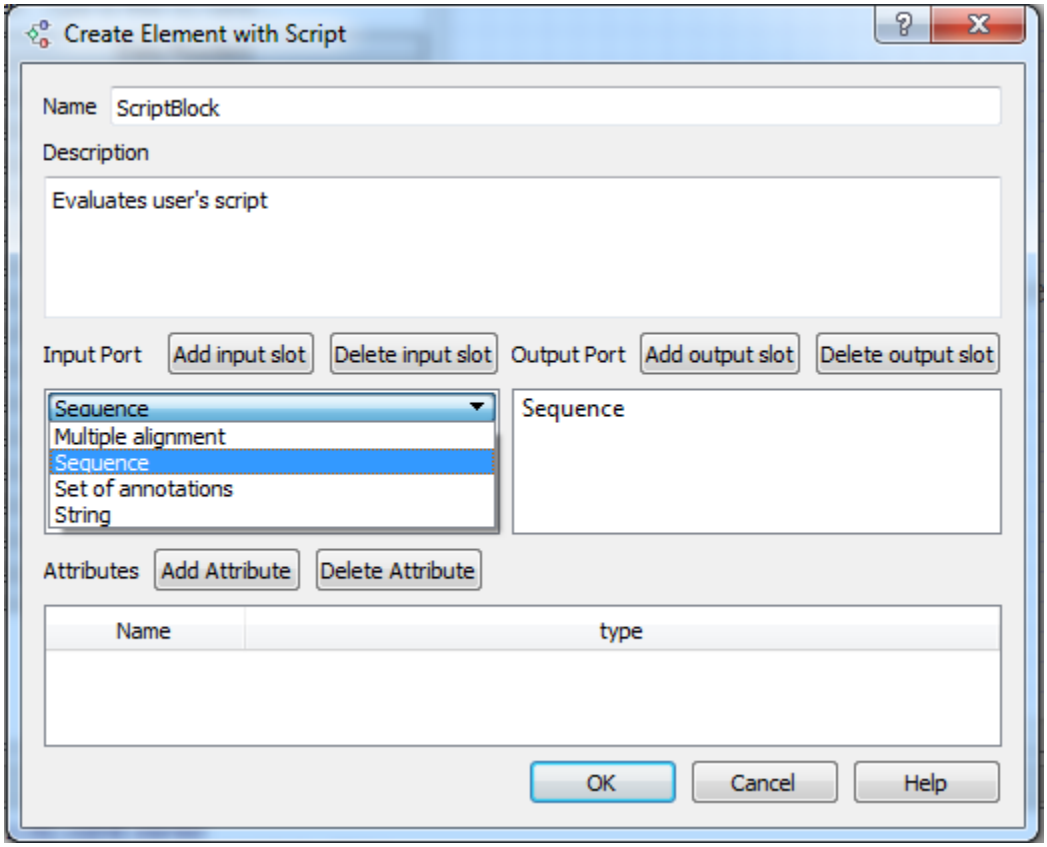
Another way to change an element style is to use the *Element style* submenu in the main menu.

Custom Elements with Scripts

It is possible to create custom algorithmic blocks using scripts in the Workflow Designer.

To create an element either select *Actions* *Create* *Element with Script* in the main menu.

The *Create Element with Script* dialog will appear:



Here you should set the name of the element, its description and input/output ports of the element. It is possible to create a port with several input/output slots.

There are 4 types of data for a slot available:

- Multiple alignments

- Sequence
- Set of annotations
- Files

You can also add an attribute. The following types are supported for attributes:

- String
- Number
- Boolean

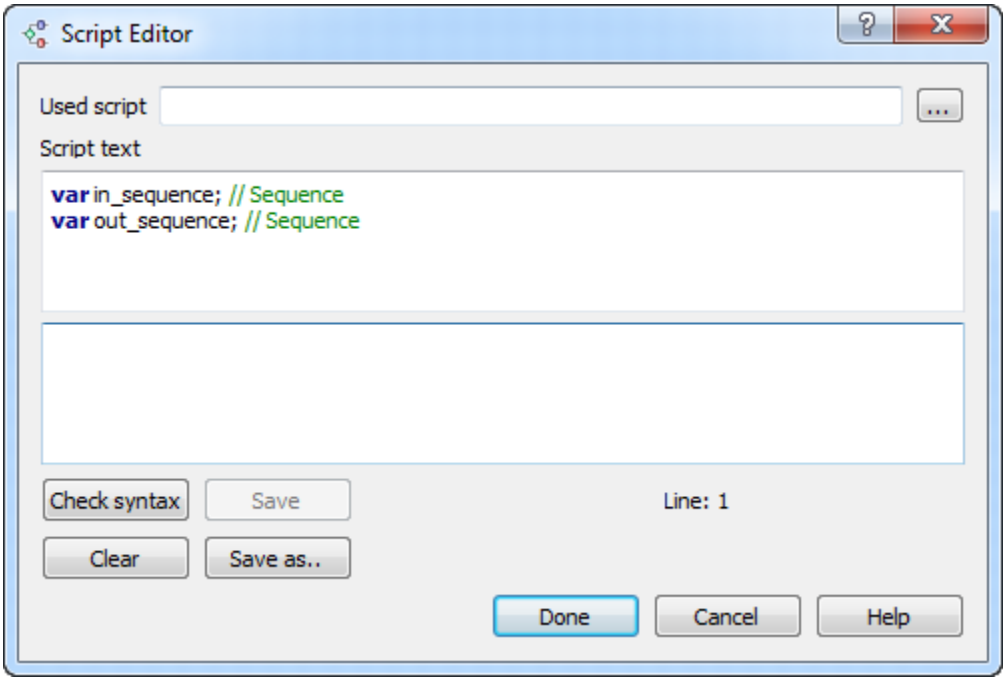
The element created is stored in a directory that can be set in the *Application Settings* dialog.

The element also becomes available in the *Custom Elements with Scripts* group on the *Palette*.

It is required to write a script for the element. Supported languages for the script are languages based on the ECMAScript (Javascript, QtScript).

To edit the script select the element on the *Scene* and either select *Actions Edit script of the element* in the main menu or use the *Edit script of the element* item in the context menu.

The *Script editor* dialog will appear:



As you can see there are predefined variables for the ports and the attributes in the script. The variables for the input slots begin with the “in_” prefix, variables for the output slots begin with the “out_” prefix. It is possible to load a script from a file (use the *Used script* field to do it).

For each supported data type UGENE provides a number of functions that can be used in the scripts.

- Functions Supported for Multiple Alignment Data
- Functions Supported for Sequence Data
- Functions Supported for Set of Annotations Data
- Functions Supported for Files
- Common Function

Functions Supported for Multiple Alignment Data

- **createAlignment** (Sequence seq1, Sequence seq2, ...) — returns the alignment created from the sequences.
- **addToAlignment** (MAlignment aln, Sequence seq, int row = -1) — adds the sequence to the specified row of the alignment. If the “row” parameter is not specified the sequence is added to the end of the alignment.
- **sequenceFromAlignment** (MAlignment aln, int row) — returns the sequence from the specified row of the alignment.
- **findInAlignment** (MAlignment aln, Sequence seq) — searches the alignment for the specified string. Return the number of the row if the sequence has been found or “-1” if it hasn’t been found.
- **findInAlignment** (MAlignment aln, QString name) — searches the alignment for a sequence with the specified name.
- **removeFromAlignment** (MAlignment aln, int row) — removes a sequence from the specified row of the alignment.
- **rowNum** (MAlignment aln) — returns the number of rows in the alignment.
- **columnNum** (MAlignment aln) — returns the length of the alignment.

- **alignmentAlphabetType** (MAlignment aln) — returns the alignment's alphabet.

Functions Supported for Sequence Data

- **subsequence** (Sequence seq, int beg, int end) - returns the subsequence between the "beg" and "end" parameters.
- **complement** (Sequence seq) - returns the complement sequence.
- **translate** (Sequence seq, int offset = 0) - returns one of the three sequence translations. Which one is returned is determined by the "offset" parameter.
- **size** (Sequence seq) - returns the length of the sequence.
- **getName** (Sequence seq) - returns the name of the sequence.
- **alphabetType** (Sequence seq) - returns the alphabet of the sequence.
- **charAt** (Sequence seq, int ind) - returns the symbol located in the "ind" position of the sequence.
- **hasQuality** (Sequence seq) - determines whether the sequence has the "Quality" parameter.
- **getMinimumQuality** (Sequence seq) - returns the minimum value of the "Quality".
- **isAmino** (Sequence seq) - returns true if it is amino acid sequence.
- **concatSequence** (Sequence1 seq1, Sequence2 seq2,...) - returns the one sequence consists of the all input sequences.
- **sequenceFromText** (QString " ") - returns the sequence consists of the input text.

Functions Supported for Set of Annotations Data

- **annotatedRegions** (Sequence seq, AnnotationTable anns, QString name) — returns subsequences of the annotations with the specified "name".
- **addQualifier** (AnnotationTable anns, QString qual, QString val, QString name = "") — sets the qualifier in the annotations with the specified "name" to the specified value. If the "name" is not specified, then all annotations are taken into account.
- **getLocation** (AnnotationTable anns, int ind) — returns the annotation location with the specified index.
- **filterByQualifier** (AnnotationsTable anns, QString qual, QString val) - returns the qualifier with the specified value.
- **hasAnnotationName** - (AnnotationsTable anns, QString " ") - returns the annotation with the specified name there is or there is not.

Functions Supported for Files

- **writeFile** (QString url, QString " ") - writes the specified text data to the file with specified url.
- **appendFile** (QString url, QString " ") - appends the specified text data to the end of the file with the specified url.
- **readFile** (QString url) - reads the file with the specified url.

Common Function

- **printToLog** (parameter) - prints the results to the *Log View*.

Custom Elements with External Tools

In UGENE you can create a custom workflow *element* that would launch any command line tool.

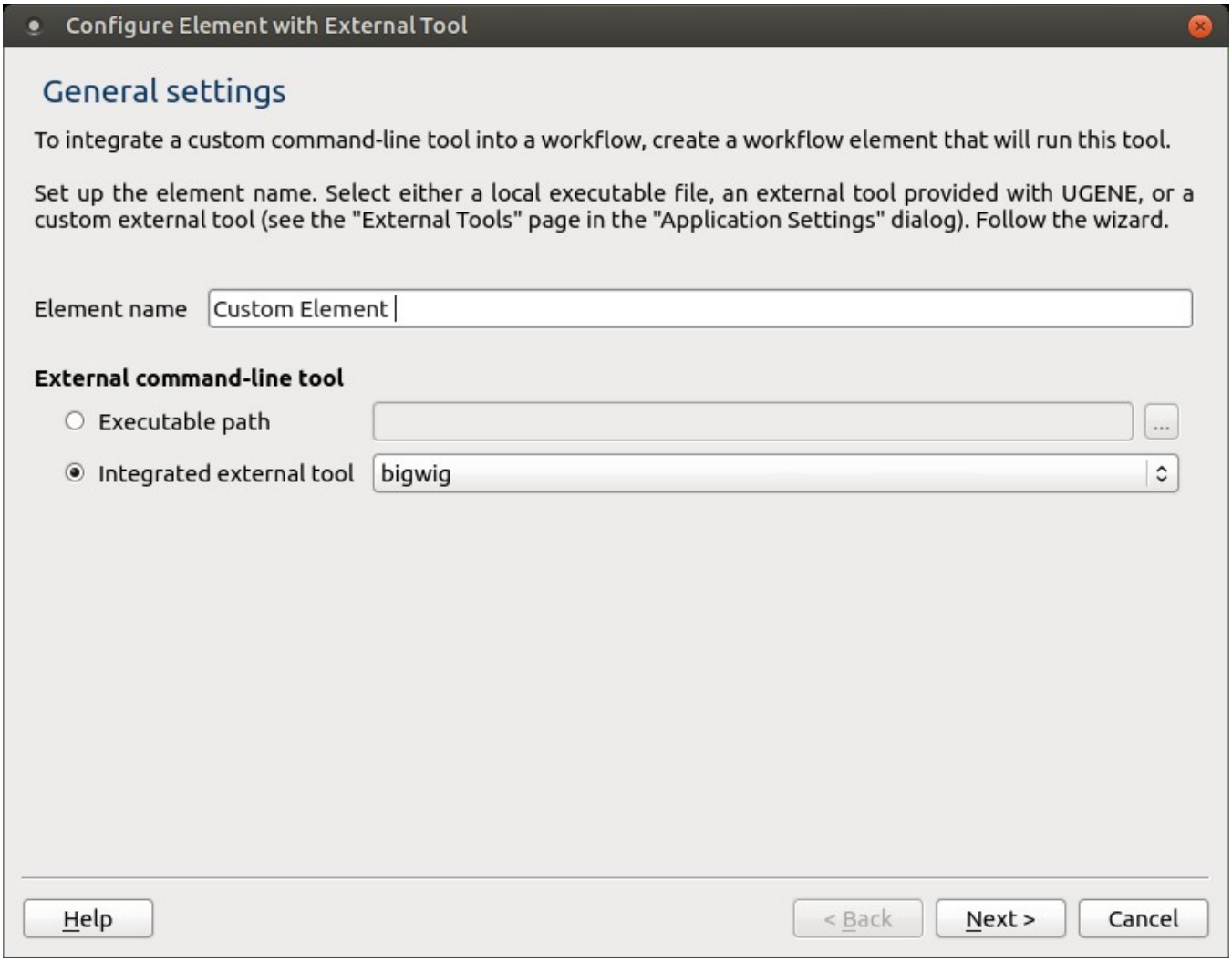
- Creating Element
- Editing Element
- Adding Existent Element
- Removing Element

Creating Element

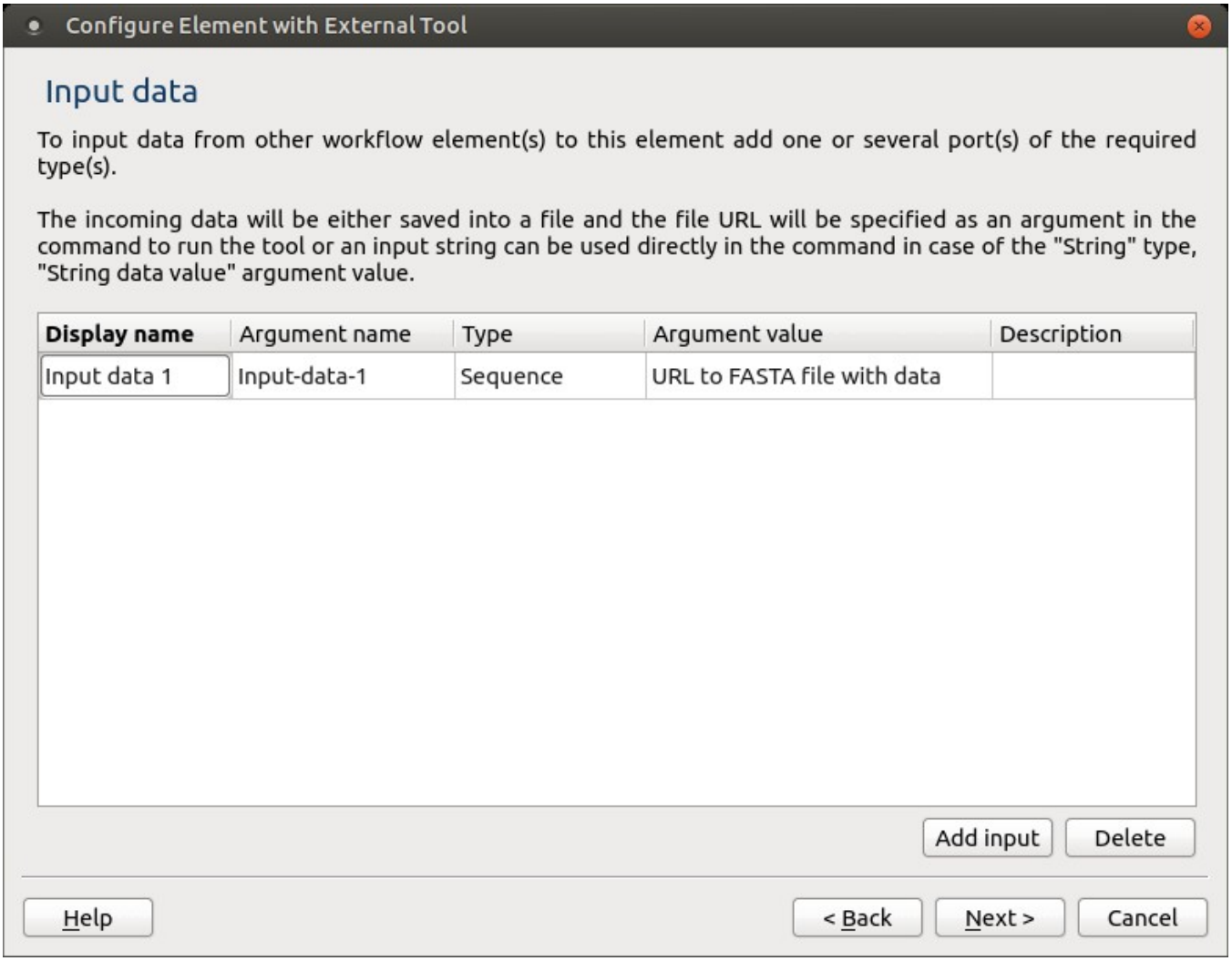
To create an element for a command line tool select either *Actions Create element with external tool* in the main menu or the following icon on the toolbar:



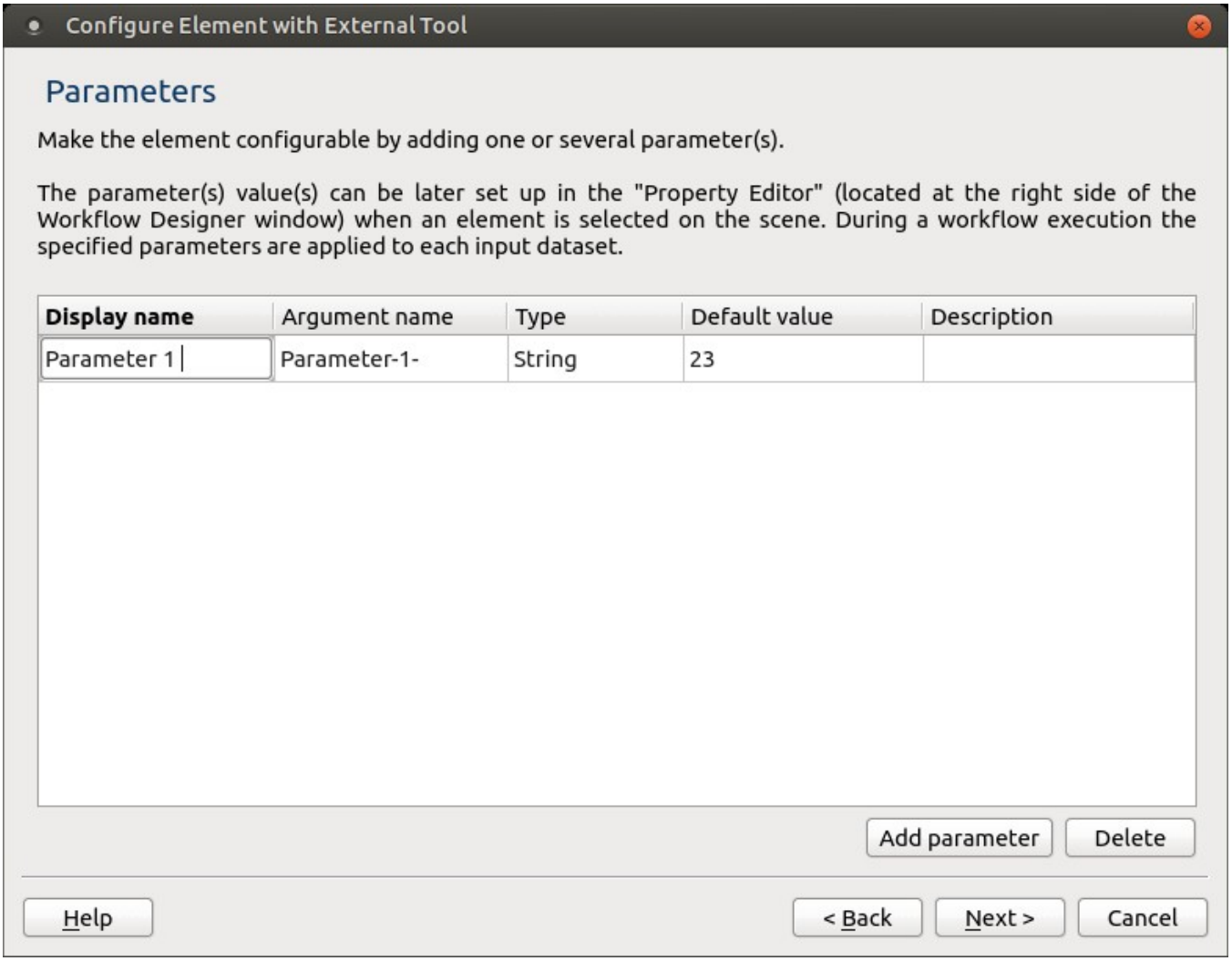
The *Configure Element with External Tool* wizard appears. On the first page of the wizard input a name and external command-line tool. Letters, numbers and underscores are allowed in the name.



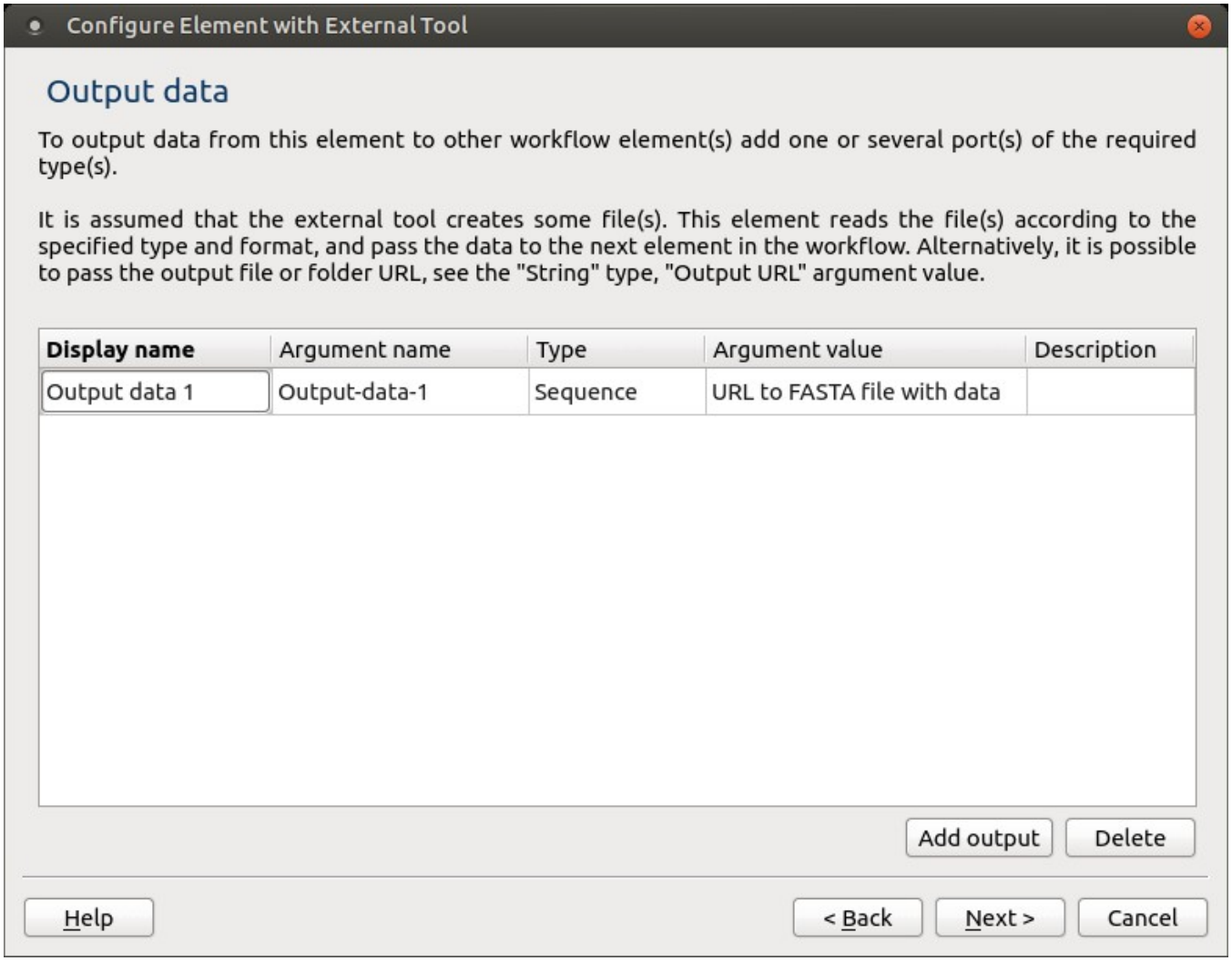
On the second page add the required input data:



On the third page of the wizard you can add parameters for the command line tool. Later you would be able to set values for the parameters in the Property Editor.



On the next page add the required output data:



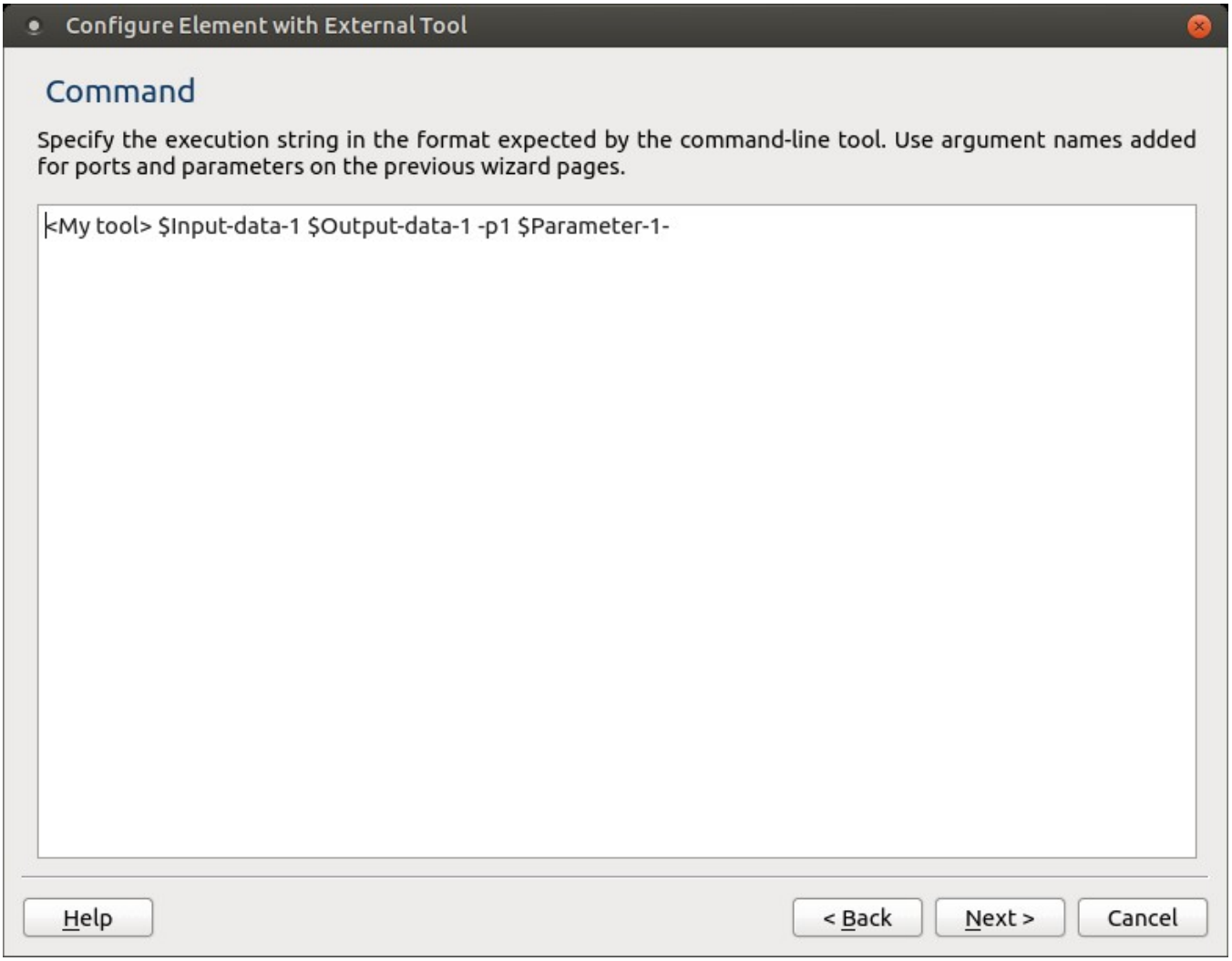
For each input or output you should:

- Input a name (letters, numbers, and underscores are allowed in the name).
- Select a type: multiple alignment, sequence, sequence with annotations, a set of annotations or string.
- Optionally input a description.

For each parameter added you should:

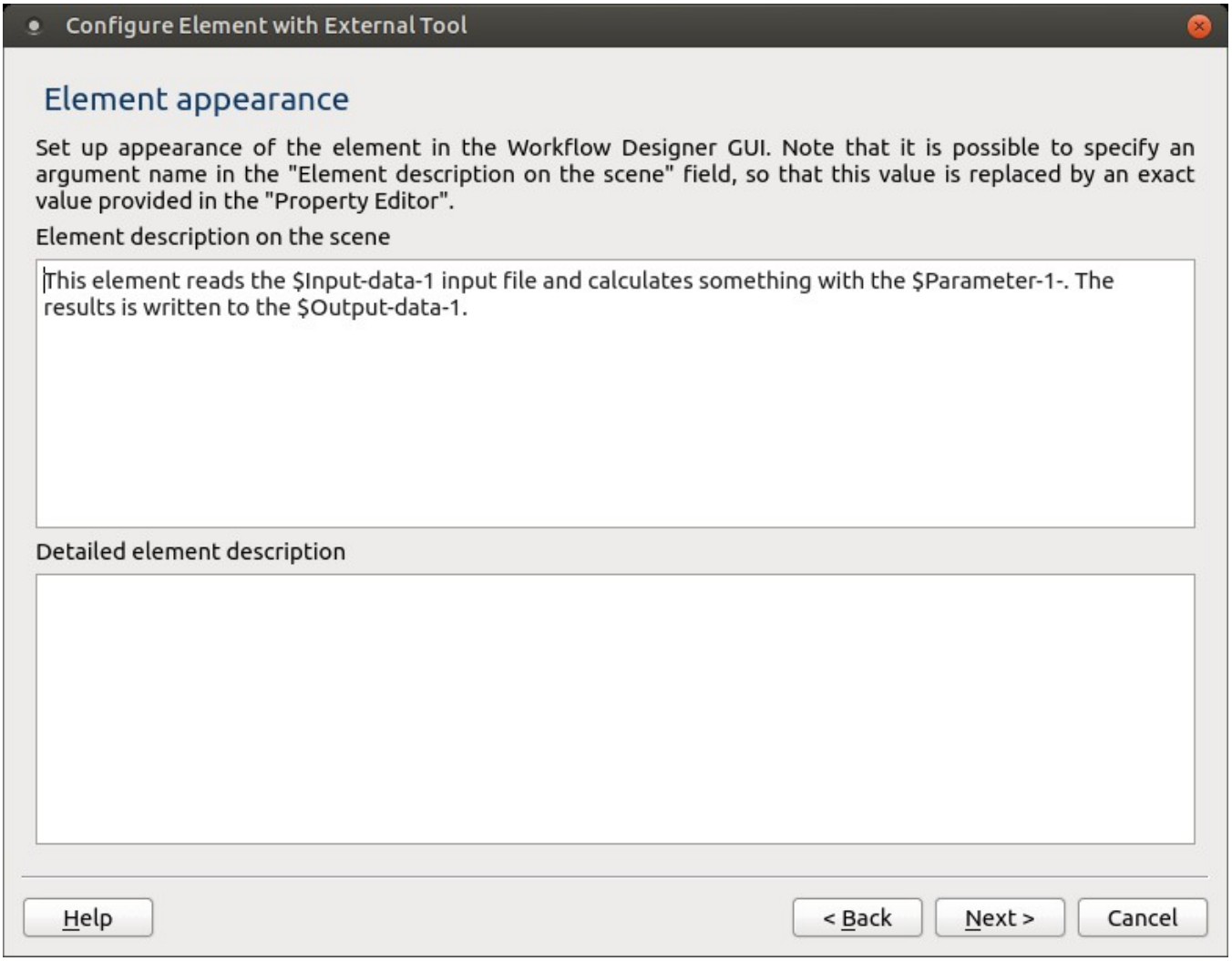
- Input a name (letters, numbers and underscores are allowed in the name).
- Select its type: boolean, number, string or URL.
- Optionally input the description.

On the next page of the wizard you should input the execution string, i.e. the command that would be executed.

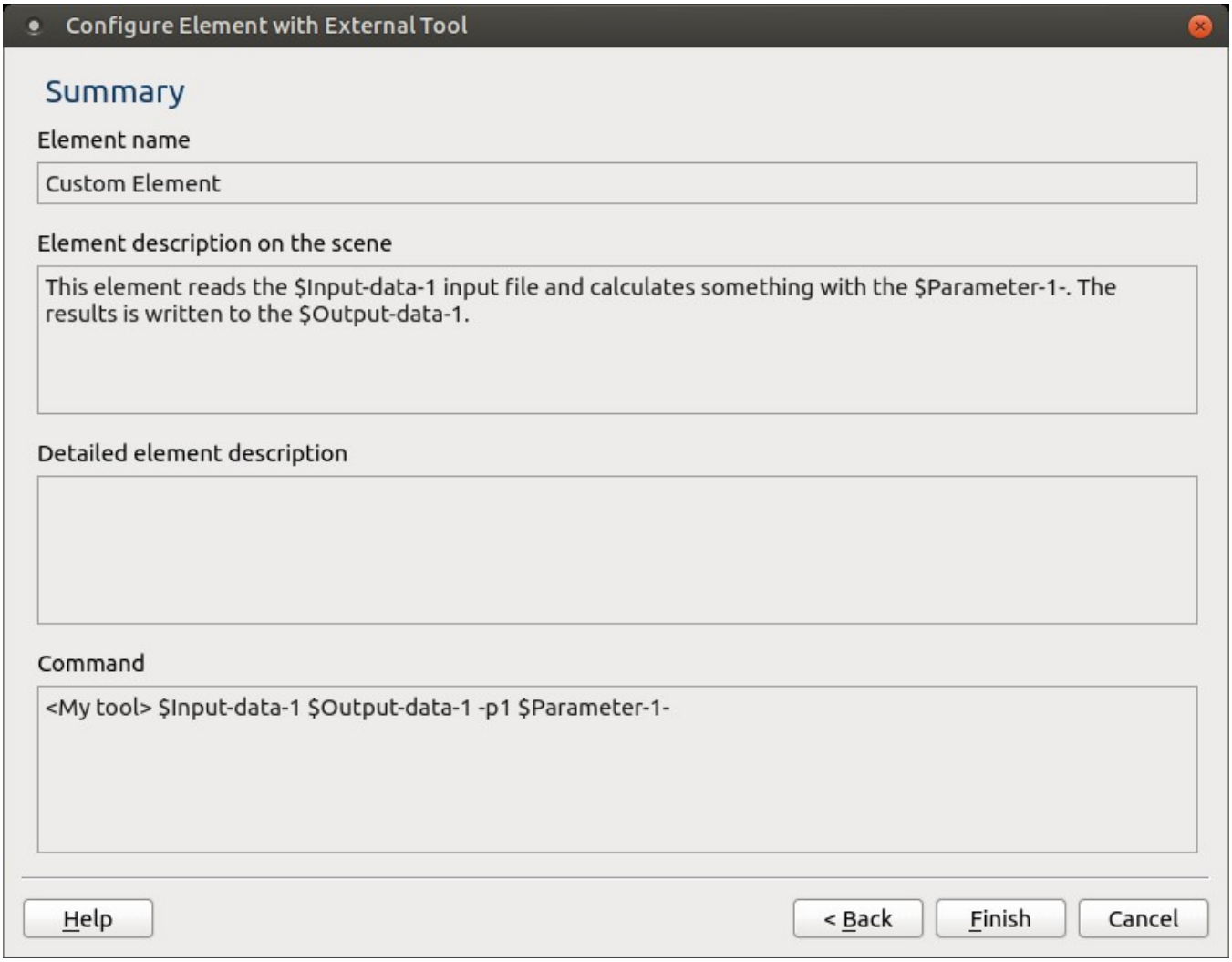


The signature of the execution string depends on the command that is launched. But the general rule is that input/output data and attributes have prefix $. You can set the parameterized description for the new element (the description that appears not in property editor but on the element itself). In the parameterized description, you also can use parameters substitution with prefix $. If the paths in the execution string contain spaces, they must be enclosed with quotes.

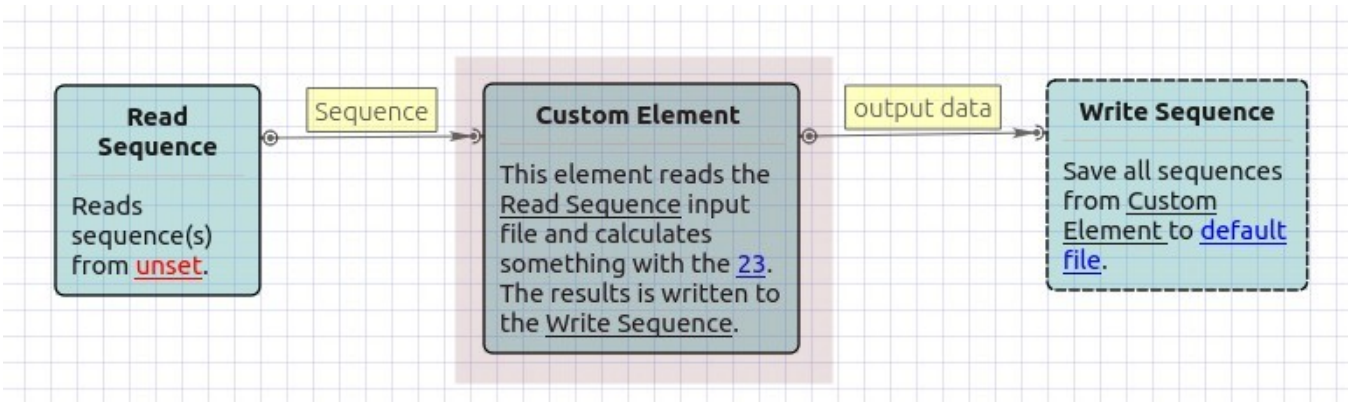
On the next wizard page, you can optionally input the description of the element. It would be shown on the element on the *Scene*. The description can be parameterized. This means that if you input e.g. an attribute name (with prefix $), the name on the element would be substituted with the value of the corresponding parameter. For example input the following parameters:



On the next wizard page, you can see the final element summary:



The created element is added to the Workflow Designer scene, and can be connected with other elements in a workflow:



The new element also becomes available in the "Custom Elements with External Tools" group on the "Elements" tab of the Workflow Designer palette (at the left part of the window), so that you can use it at any time in other workflow.

Editing Element

The element created appears in the *Custom Elements with External Tools* group on the *Palette*.

To edit an element select the *Edit* item in its context menu in the *Palette* or select the *Edit configuration* item in its context menu on the *Scene*. The creation element wizard would appear.

Adding Existent Element

The elements are stored in the files with the .etc extension.

The directory to store the elements can be set in the *Application Settings* dialog.

To add an element from a file to the *Workflow Designer* select either *Actions* *Add element with external tool* in the main menu or the following icon on the toolbar:



In the appeared dialog select the required .etc file. The element is added to the group on the *Palette* and appears on the *Scene*.

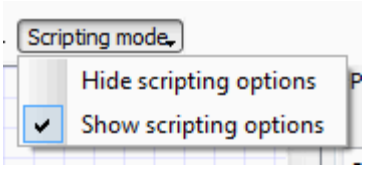
Removing Element

To remove an element right-click on it and select the *Remove* item in the element's context menu. The corresponding .etc file is also removed in this case.

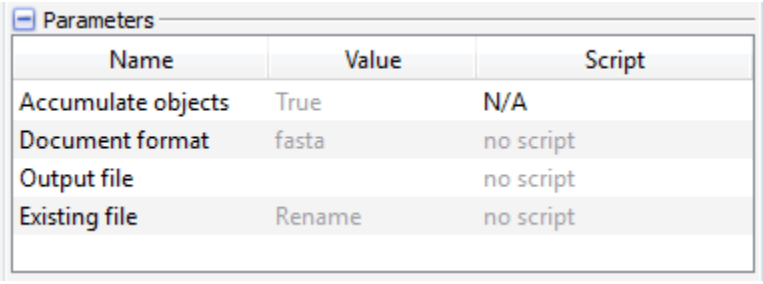
Using Script to Set Parameter Value

When you select an element the *Parameters* area of the *Property Editor* displays two columns: *Name* and *Value*.

Select the *Show scripting options* item in the *Scripting mode* menu on the toolbar or in the *Actions* main menu.



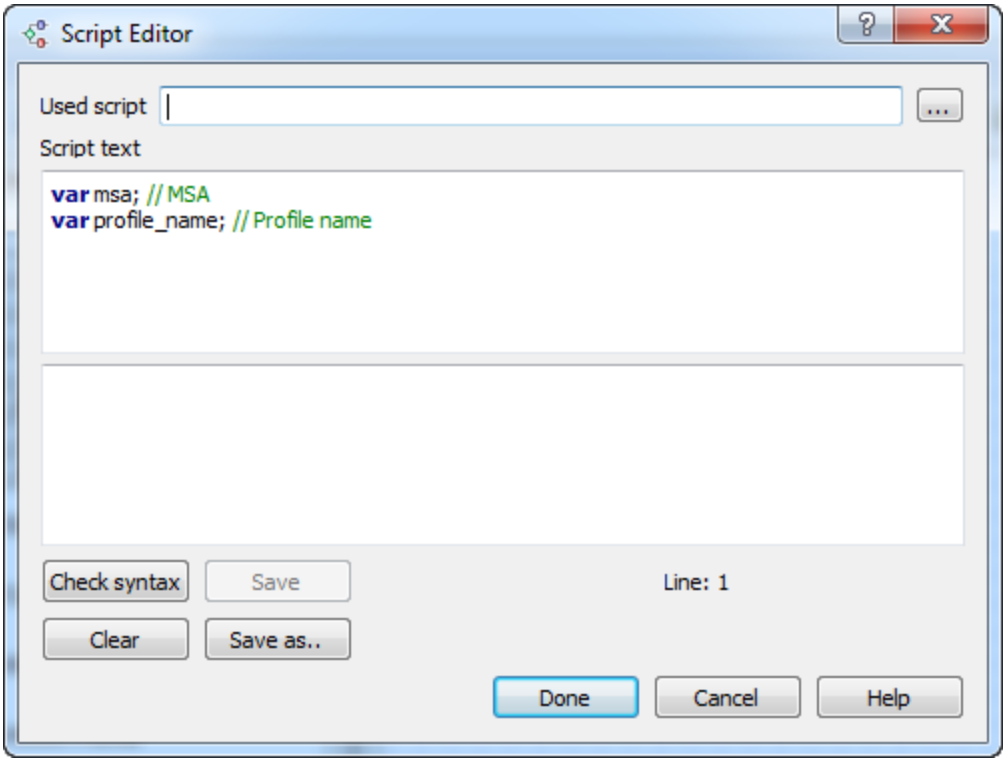
You can see that the third column *Script* has appeared in the *Parameters* area, for example:



A script value can either be:

- not available for a parameter (*N/A* value)
- not set (*no script*)
- set by user (*user script*)

To set a script value (when it is available) select the *user script* item in the *Script* column. The following dialog appears:



Here you can see the variables available from the dataflow and can write your script. Supported languages for the script are languages based on the ECMAScript (Javascript, QtScript).

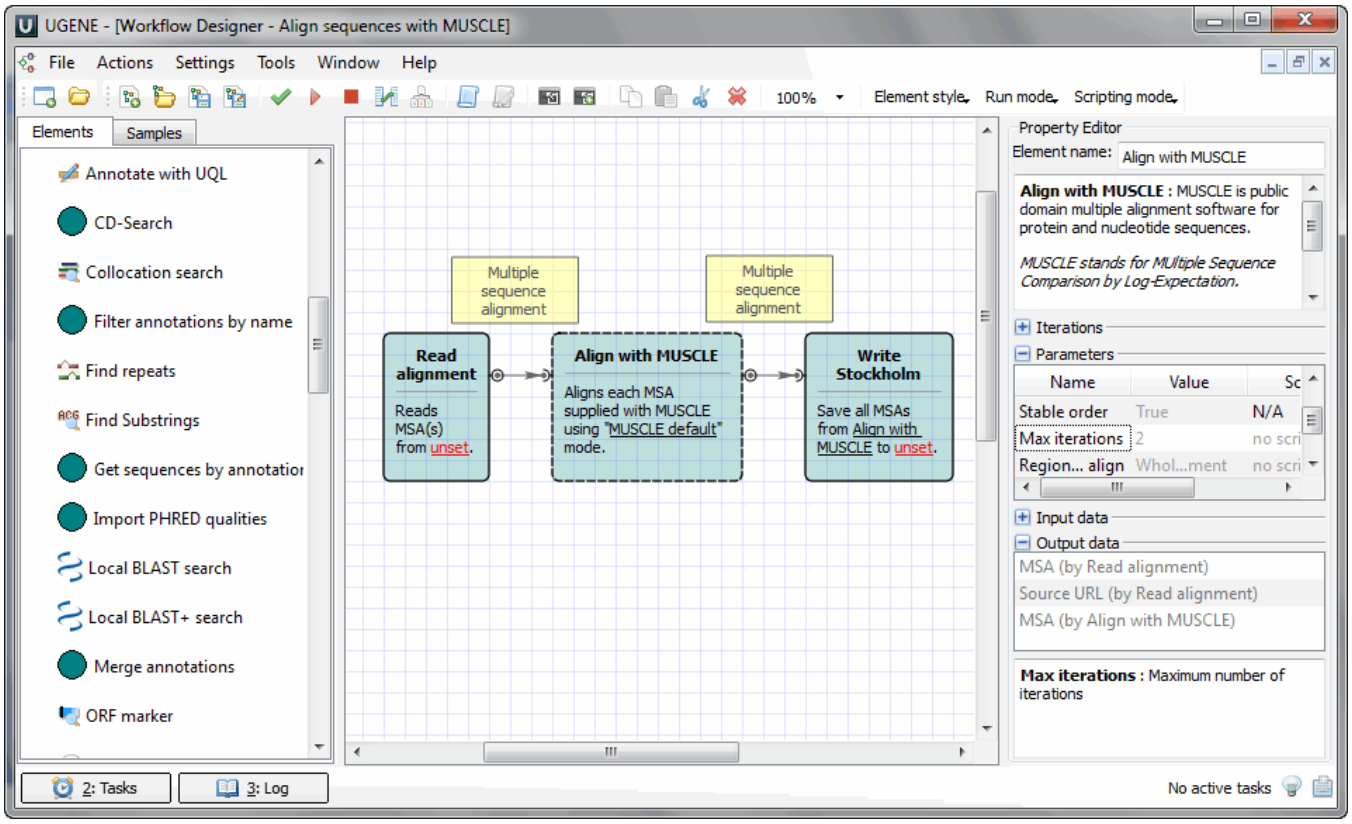
Running Workflow from the Command Line

UGENE provides command line interface (CLI). To learn more about UGENE CLI and commands available read main UGENE User Manual.

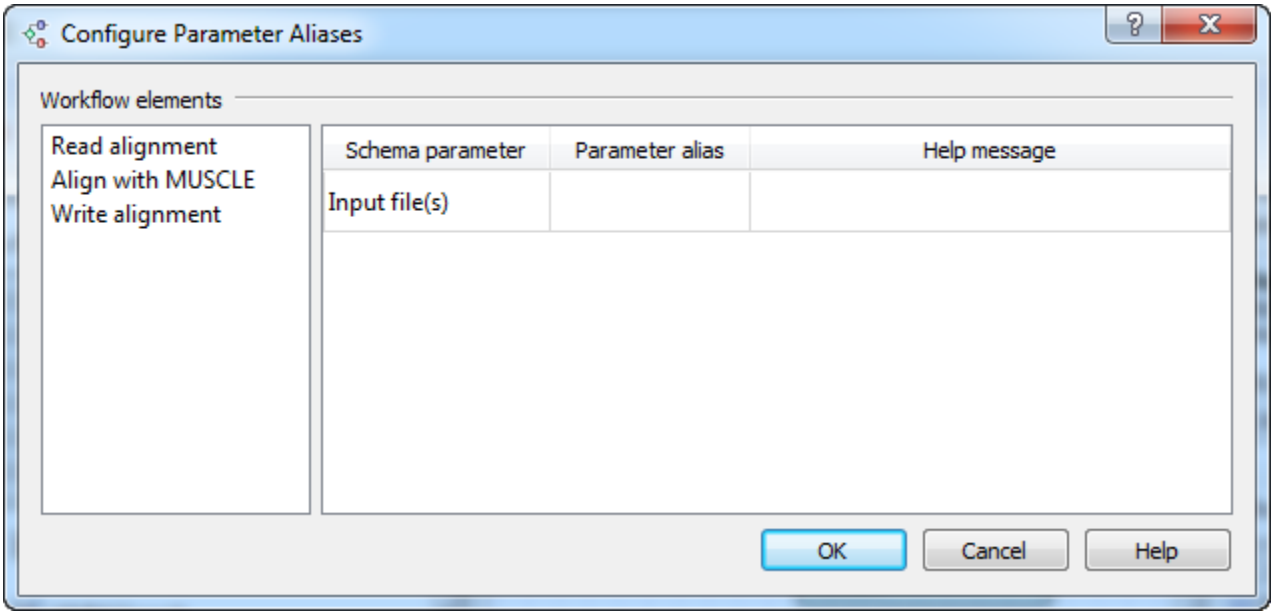
This chapter describes how you can create a new command using a *workflow*.

To run a workflow from the command line do the following:

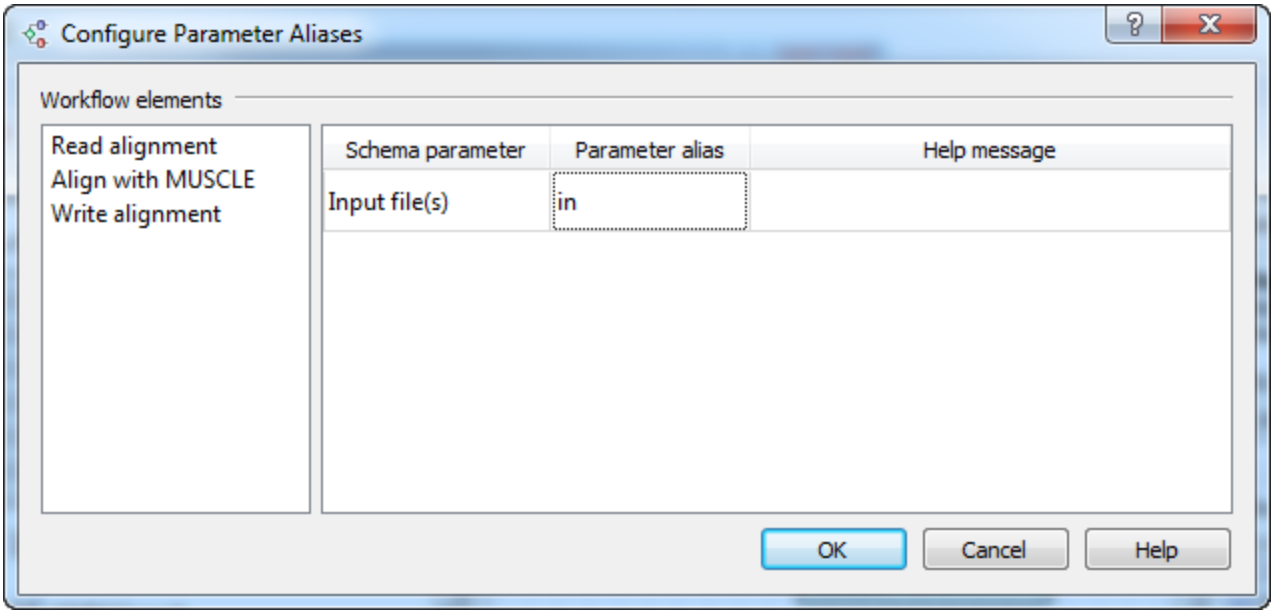
- Create the workflow in the Workflow Designer. For example on the image below the *Align sequences with MUSCLE* sample workflow is used:



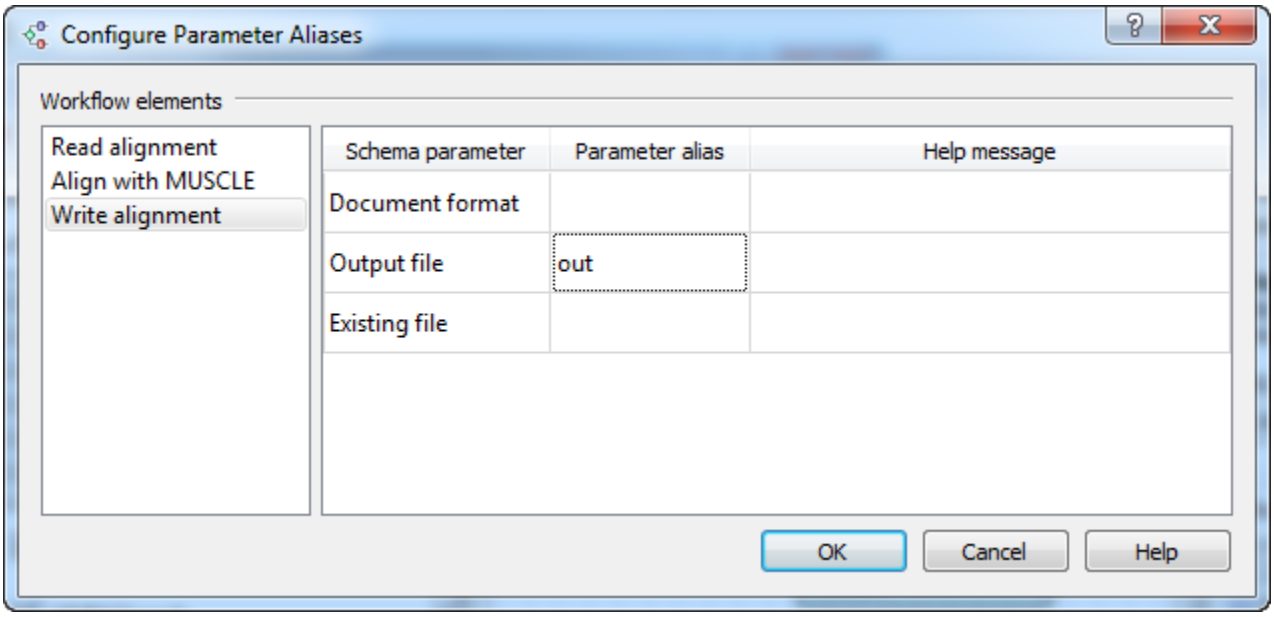
- Now you should configure aliases for those parameters and ports and slots that you are going to use from the command line. To do it select the *Actions Set parameter aliases...* item in the main menu or the *Set parameter aliases* toolbar button. The following dialog appears:



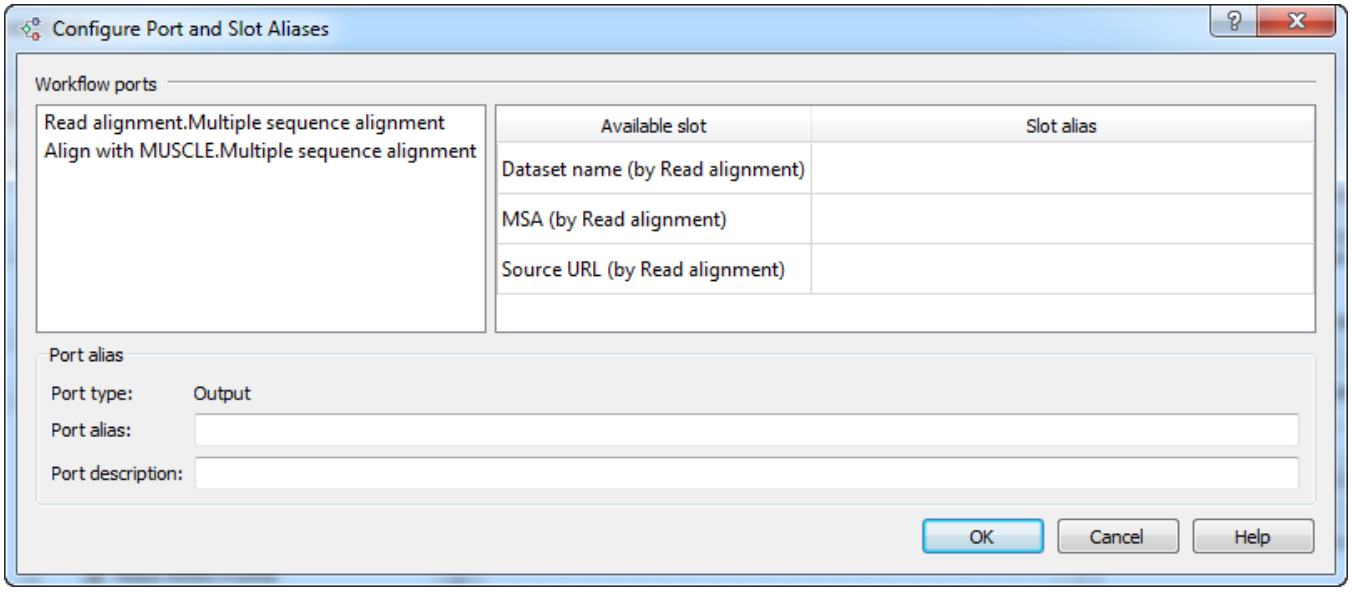
It contains the list of objects that corresponds to the *elements* of the workflow. For each object the list of parameters is available for which you can assign command line aliases. For example, assign alias **in** to parameter *Input file* (of the *Read alignment* element):



And alias **out** to parameter *Output file* (of the *Write Stockholm* element).



To select ports and slots aliases use the following dialog by the *Actions->Configure port and slot aliases* main menu item:



Press the *Ok* button to save aliases and close the dialog. When you create aliases you can import workflow to element by the *Actions->Import workflow to element* main menu item.

- *Save the workflow* to a file: if you follow the example, choose the *Actions Save workflow as...* item in the main menu, browse for the file location and enter **mySchema** as the workflow name. This name will be used to launch the workflow from the command line.
- Launch the workflow from the command line:

```
[path_to_ugene\]ugene --task={schema_name} [--{parameter1}={value1} [--{parameter2}={value2} ...]]
```

The run information will be saved into the text file. By default it is the working directory.

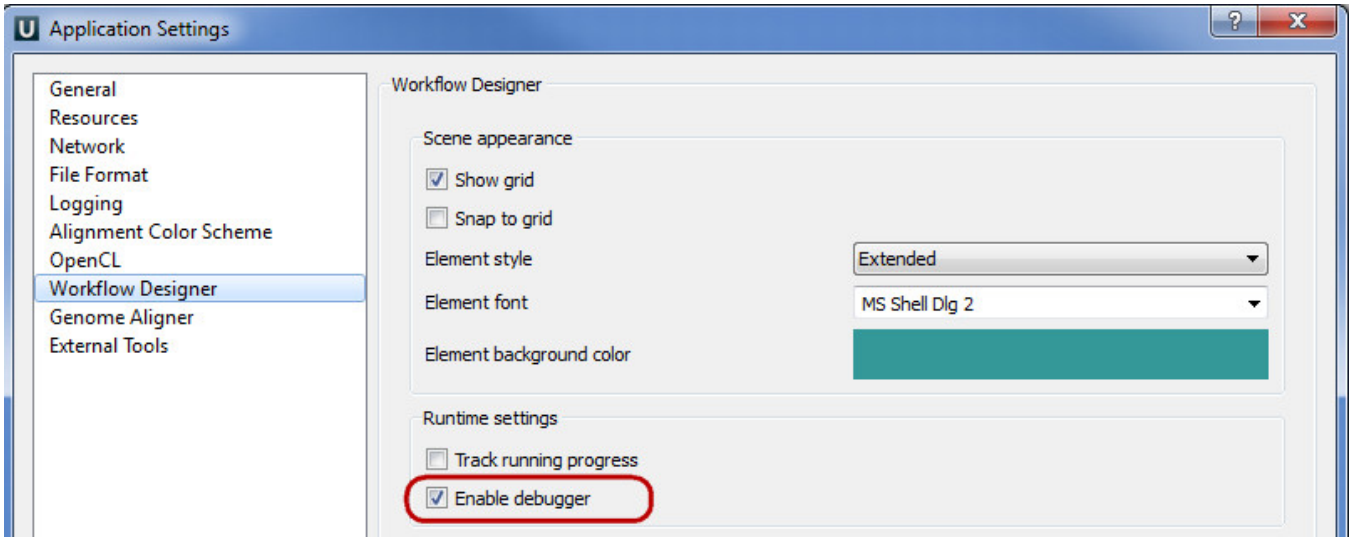
For example on Windows the command can look as follows:

```
ugene --task=C:\mySchema --in=C:\COI.aln --out=C:\COI.sto
```

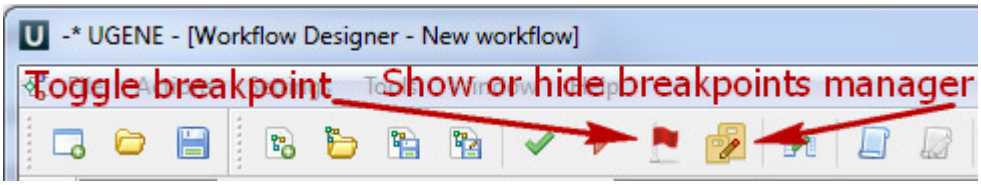

In this example the path to the directory with the UGENE executable is added to the system PATH variable.

Running Workflow in Debugging Mode

By default a *workflow* runs without debugging settings. To use it go to the *Application Settings* (Settings–>Preferences) and check the following checkbox and click *OK*:



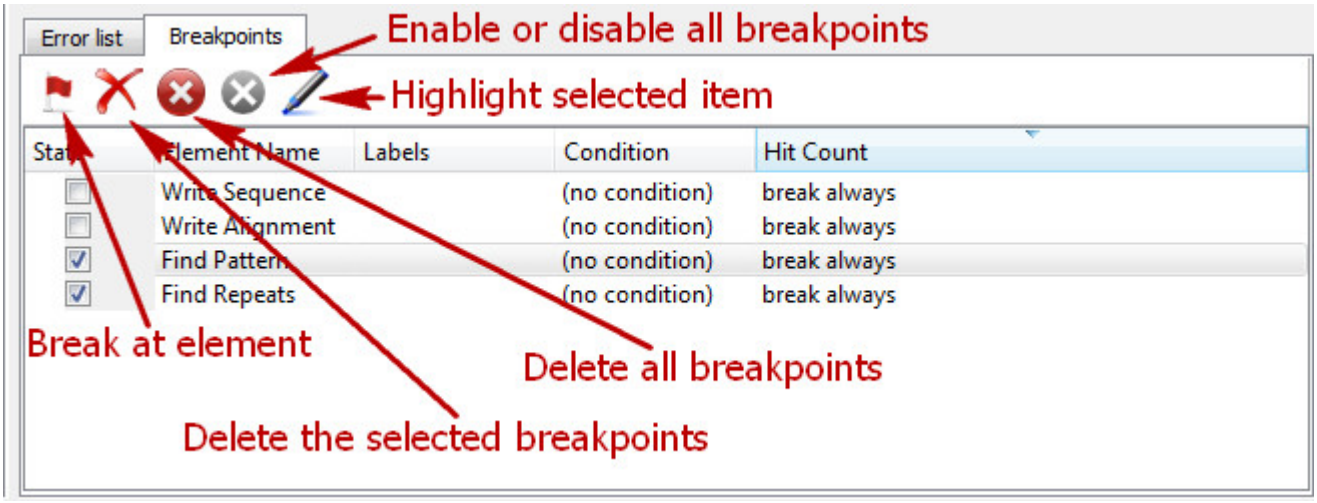
After that the two new buttons appears on the main toolbar:



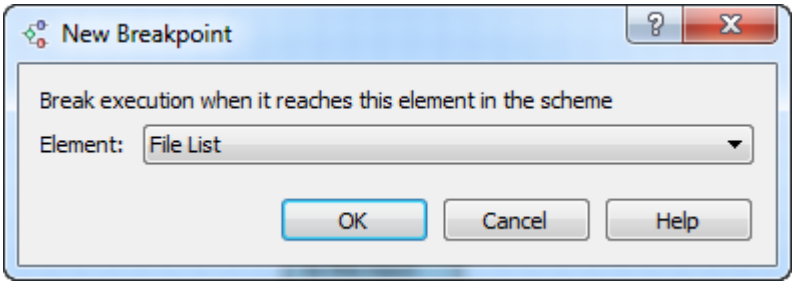
- Creating Breakpoints
- Manipulating Breakpoints

Creating Breakpoints

You can create a pause element in a workflow with a help of the *Toggle breakpoint* button or by the *Ctrl+B* shortcut. To do it select the element and press this button. If you press the *Show or hide breackpoint manager* the breakpoint manager appears:



Break at element - creates new breakpoint. If you press on this button the following dialog will appear. Choose the breakpoint element and click *OK* button.



Delete the selected breakpoints - this button deletes the selected breakpoint.

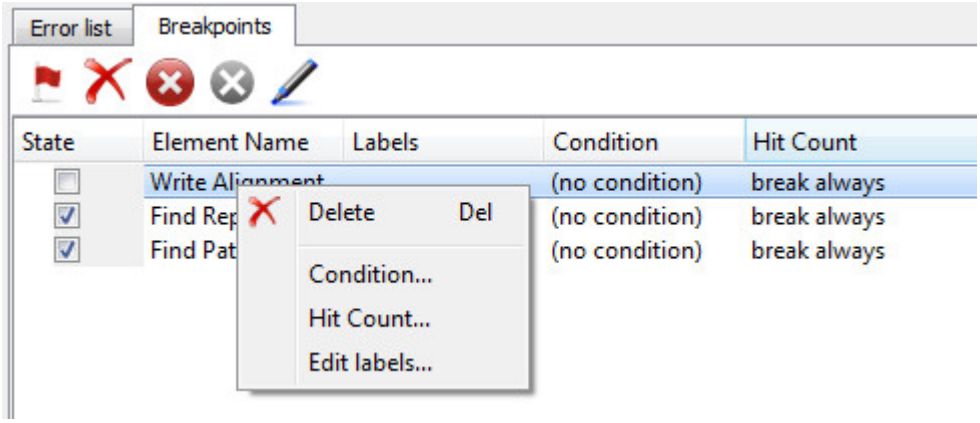
Delete all breakpoints - this button deletes all breakpoints.

Enable or disable all breakpoints - this button check or uncheck all breakpoints. Check on the breakpoint means that the breakpoint enable and will be used.

Highlight selected item - this button highlights the breakpoint element.

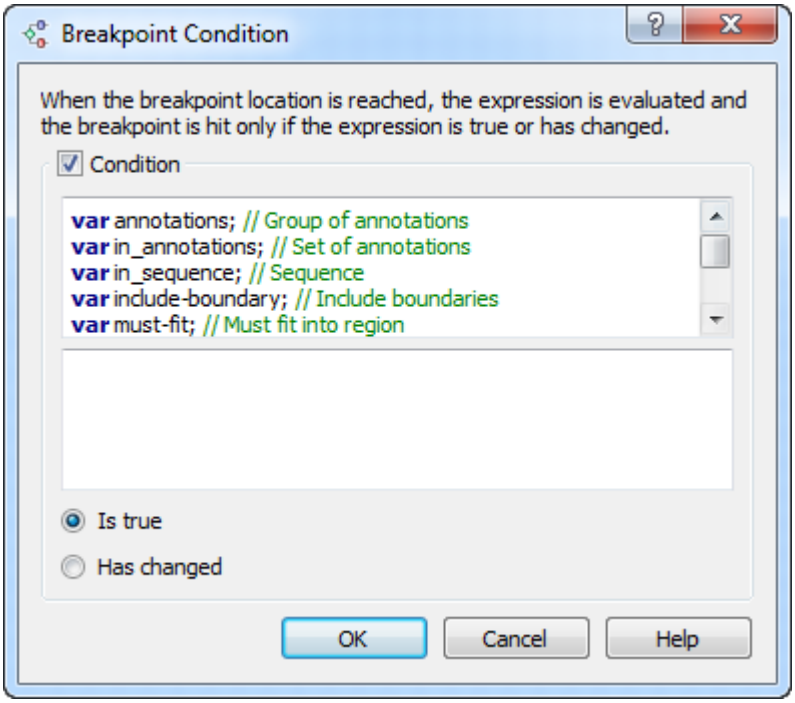
Manipulating Breakpoints

The following operations are available for each breakpoint:



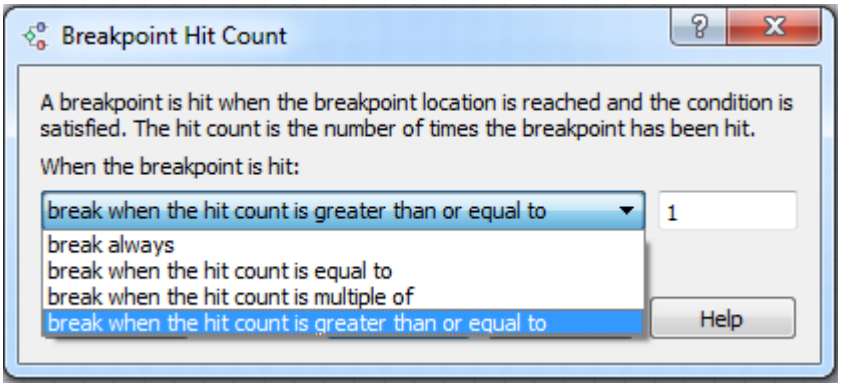
Delete - delete the selected breakpoint.

Condition - creates a breakpoint condition. Click on this menu item and the following dialog appears:



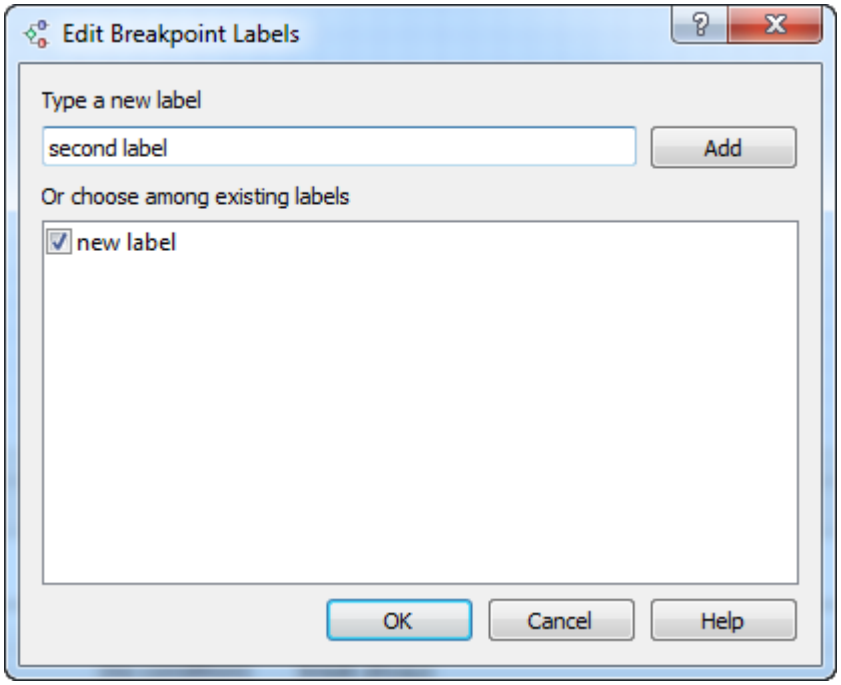
When the breakpoint location is reached, the expression is evaluated and the breakpoint is hit only if the expression is true or has changed.

Hit Count - breakpoint hit count. Click on this menu item and the following dialog appears:



A breakpoint is hit when the breakpoint location is reached and the condition is satisfied. The hit count is the number of times the breakpoint has been hit.

Edit labels - allows to add breakpoint labels. Click on this menu item and the following dialog appears:



Workflow File Format

Using the GUI is not the only way to create/edit a *workflow workflow*. A workflow is saved to a file with .uwl extension. The format of the file is human-readable. This chapter describes this format and explains how you can create/edit a workflow file using a text editor.

The best way to learn workflow workflow file format is to study an existent .uwl file. The file consists of the header and the body. Check the description of each part below.

- Header
- Body

Header

The header consists of the following key string:

```
#!UGENE_WORKFLOW
```

And multiline description of the workflow:

```
# Write here the description
# of your workflow.
```

Body

The body begins with the **workflow** keyword followed by the name of the workflow and curly braces:

```
workflow schema_name {

    # Description of the elements
    # Description of the dataflow
    # Description of the iterations
    # Metainformation (aliases and visual information)

}
```

- Elements
- Dataflow
- Metainformation

Elements

Each *element* used in the *workflow* must be described inside the body.An element description consists of the element name and a set of parameters enclosed in curly braces. A parameter and the value are separated by ':', different parameters are separated by ';':

```
element_name {

    parameter1:value1;
    parameter2:value2;
    ...
}
```

See, for example, a description of the *Read alignment* element:

```
read-msa {
    type:read-msa;
    name:"Read alignment";
    url-in:/home/user/pkinase.sto;
}
```

Note, that the values of the parameters for an element can also be presented in the *iterations* block.For all elements the following parameters are defined:

- **type** - specifies the type of the element.
- **name** - specifies the name of the element. It corresponds to the element's name in the GUI
- **.validator** - validates the element by the input validator type's parameters:
    - **type** - specifies the type of the validator.

For example this validator validate that the read sequence element has two or three datasets:

```
    read-sequence {
        type:read-sequence;
        name:"Read Sequence";
        .validator {
            type:datasets-count;
            min:2;
            max:3;
        }
    }
```

For *custom elements* there is special parameter:

- **script** - sets the script text of the element, for example:

```
dump-info {
    type:"Script-Dump sequence info"
    name:"Dump sequence info"
    script {
        out_text=getName(in_sequence) + ": " + size(in_sequence);
    }
}
```

The list of parameters available depend on an element. Refer to the *Workflow Elements* chapter to find out the parameters for a particular element.To *set a script value for a parameter* use the following form:

```
parameter_name {
    a script value
};
```

Dataflow

The description of the elements is followed by the description of their connections to each other, i.e. the dataflow. For ports connections the description starts with the **.actor-bindings** keyword and has the following format:

```
    .actor-bindings {
        element1_name.output_port1_name->element2_name.input_port2_name;
    }
```

This pair says that data from port*1* of *element1* will be transferred to *port2* of *element2*. For slots the following format without start keyword is used:

```
element1_name.slot1_name->element2_name.port2_name.slot2_name
```

This pair says that data from *slot1* of *element1* will be transferred to *slot2* of *port2* of *element2*. See, for example, the minimum description of a dataflow of a workflow, that aligns an input MSA and writes the result to a file in ClustalW format.

```
    .actor-bindings {
        read-msa.out-msa->muscle.in-msa
        muscle.out-msa->write-msa.in-msa
    }
    read-msa.msa->muscle.in-msa.msa
    muscle.msa->write-msa.in-msa.msa
```

Metainformation

A metainformation block sets visual parameters of the workflow and aliases for running it from the command line.

Each block starts with **.meta** keyword and consists of the aliases and visual blocks:

```
.meta {
    aliases {
        # The workflow aliases
    }
    visual {
        # Visual data for element1
        # Visual data for element2
        # ...
    }
}
```

Parameter Aliases

The block starts with the **parameter-aliases** keyword and has the following format:

```
parameter-aliases {
    element_name.parameter_name:value;
    ...
}
```

The value specified for an element parameter is used as the alias for this parameter when the workflow is *executed from the command line*.

See an example of setting workflow aliases:

```
.meta {
    parameter-aliases {
        read-msa.url-in:in;
```

```
        write-msa.url-out:out;
    }
    ...
}
```

Visual

The block starts with the **visual** keyword. It describes the appearance of the workflow in a Workflow Designer window, i.e. appearance of the workflow *elements* and *connections*:

```
visual {

    # Elements appearance
    element_name1 {
        element_appearance_parameter1:value1;
        element_appearance_parameter2:value2;
        ...
    }
    element_name2 {
        ...
    }
    ...


    # Connections appearance
    element1_name.port1_name->element2_name.port2_name {
        connection_appearance_parameter1:value3;
        ...
    }
    ...
}
```

To describe an element appearance the following parameters are used:

- **description** — description of the element in the *Property Editor*. It is in HTML format.
- **tooltip** — tooltip shown on the element.
- **pos** — position of the element, assuming that bottom right corner of the window is (0, 0) position.
- **style** — style of the element. The following values are available:
    - **ext** — for extended element style
    - **simple** — for minimal element style
- **bounds —** defines the bounds of the element rectangle in the extended style.
- **bg-color-ext** — color of the element in the extended style. The color must be specified in the RGBA format.
- **bg-color-simple** — color of the element in the minimal style.
- **port_name.angle** — position of the port on the element. Here the *port_name* must be replaced by the name of the port.

For now, the only parameter that describes a connection appearance is:

- **text-pos** — position of the text near the connection arrow.

For example:

```
visual {
    read-sequence {
        description:"";
        tooltip:"Reads sequences and annotations ...";
        pos:"-930 -885";
        style:ext;
        bg-color-ext:"0 128 128 64";
        bounds:"-30 -30 45 103";
        out-sequence.angle:272.309;
    }
    write-sequence {
        ...
    }
    read-sequence.out-sequence->write-sequence.in-sequence {
        text-pos:"-27.5 -24";
    }
```

Workflow Elements

This section contains detailed description of all workflow elements presented in the Workflow Designer.

For each element you can find:

- Description of the parameters used in the GUI
- Corresponding parameters names used in a workflow file
- Information about input and output ports

The type of a parameter can be one of the following:

string

A string.

numeric

A number.

boolean

A boolean data type. Available values are: true / false, 0 / 1 and yes / no.

A port's slot type can be one of the following:

sequence

Biological sequence

msa

Multiple sequence alignment

text

A text

annotation-table

Table of annotations

annotation-table-list

A list of different tables of annotations

ebwt-index

Bowtie index

hmm2-profile

A HMM profile of HMMER2 package

fmatrix

Frequency matrix

wmatrix

Weight matrix

sitecon-model

SITECON model

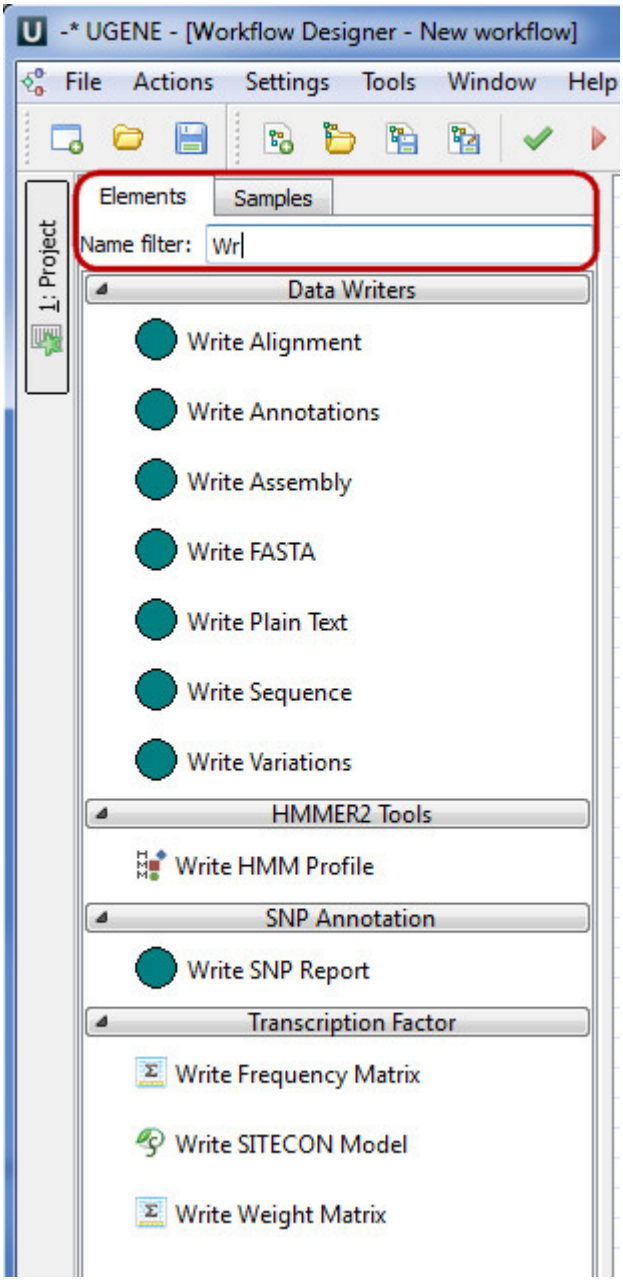
assembly

Assembly

variation

Variation track

To search an element use the name filter or press the *Ctrl+F* shortcut that moves you to the name filter also:



- Data Readers
  - Read Alignment Element
  - Read Annotations Element
  - Read FASTQ File with SE Reads Element
  - Read FASTQ Files with PE Reads Element
  - Read File URL(s) Element
  - Read NGS Reads Assembly Element
  - Read Plain Text Element
  - Read Sequence Element
  - Read Sequence from Remote Database Element
  - Read Variants Element
- Data Writers
  - Write Alignment Element
  - Write Annotations Element
  - Write FASTA Element
  - Write NGS Reads Assembly Element
  - Write Plain Text Element
  - Write Sequence Element
  - Write Variants Element
- Data Flow

    - Filter Element
    - Grouper Element
    - Multiplexer Element
    - Sequence Marker Element
- Basic Analysis
    - Amino Translations Element
    - Annotate with UQL Element
    - CD-Search Element
    - Collocation Search Element
    - Export PHRED Qualities Element
    - Fetch Sequences by ID From Annotation Element
    - Filter Annotation by Name Element
    - Filter Annotations by Qualifier
    - Find Correct Primer Pairs Element
    - Find Pattern Element
    - Find Repeats Element
    - Gene-by-gene approach report
    - Get Sequences by Annotations Element
    - Group Primer Pairs Element
    - Import PHRED Qualities Element
    - Intersect Annotations Element
    - Local BLAST+ Search Element
    - Merge Annotations Element
    - ORF Marker Element
    - Remote BLAST Element
    - Sequence Quality Trimmer Element
    - Smith-Waterman Search Element
- Data Converters
    - Convert bedGraph Files to bigWig Element
    - Convert Text to Sequence Element
    - File Format Conversion Element
    - Reverse Complement Element
    - Split Assembly into Sequences Element
- DNA Assembly
    - Assembly Sequences with CAP3
- HMMER2 Tools
    - HMM2 Build Element
    - HMM2 Search Element
    - Read HMM2 Profile Element
    - Write HMM2 Profile Element
- HMMER3 Tools
    - HMM3 Build Element
    - HMM3 Search Element
    - Read HMM3 Profile
    - Write HMM3 Profile
- Multiple Sequence Alignment
    - Align Profile to Profile with MUSCLE Element
    - Align with ClustalO Element
    - Align with ClustalW Element
    - Align with Kalign Element
    - Align with MAFFT Element
    - Align with MUSCLE Element
    - Align with T-Coffee Element
    - Extract Consensus from Alignment as Sequence
    - Extract Consensus from Alignment as Text
    - In Silico PCR Element
    - Join Sequences into Alignment Element
    - Map to Reference Element
    - Split Alignment into Sequences Element
- NGS: Basic Functions
    - CASAVA FASTQ Filter Element
    - Cut Adapter Element
    - Extract Consensus from Assembly Element
    - Extract Coverage from Assembly Element
    - FASTQ Merger Element
    - FASTQ Quality Trimmer Element
    - FastQC Quality Control Element
    - Filter BAM/SAM Files Element
    - Genome Coverage Element
    - Improve Reads with Trimmomatic Element
    - Merge BAM Files Element
    - Remove Duplicates in BAM Files Element
    - Slopbed Element
    - Sort BAM Files Element
- NGS: ChIP-Seq Analysis
    - Annotate Peaks with peak2gene Element
    - Build Conservation Plot Element
    - Collect Motifs with SeqPos Element
    - Conduct GO Element

    - Create CEAS Report Element
    - Find Peaks with MACS Element
- NGS: Map/Assemble Reads
    - Assemble Reads with SPAdes Element
    - Map Reads with Bowtie Element
    - Map Reads with Bowtie2 Element
    - Map Reads with BWA Element
    - Map Reads with BWA-MEM Element
    - Map Reads with UGENE Genome Aligner Element
    - Map RNA-Seq Reads with TopHat Element
- NGS: Metagenomics Classification
    - Build CLARK Database
    - Build DIAMOND Database Element
    - Build Kraken Database Element
    - Classification Report Element
    - Classify Sequences with CLARK Element
    - Classify Sequences with DIAMOND Element
    - Classify Sequences with Kraken Element
    - Classify Sequences with MetaPhlAn2 Element
    - Ensemble Classification Data Element
    - Filter by Classification Element
    - Improve Classification with WEVOTE Element
- NGS: RNA-Seq Analysis
    - Assemble Transcripts with StringTie Element
    - Assembly Transcripts with Cufflinks Element
    - Extract Transcript Sequences with gffread Element
    - Merge Assemblies with Cuffmerge Element
    - StringTie Gene Abudance Report Element
    - Test for Diff. Expression with Cuffdiff Element
- NGS: Variant Analysis
    - Call Variants with SAMtools Element
    - Change Chromosome Notation for VCF Element
    - Convert SnpEff Variations to Annotations Element
    - Create VCF Consensus Element
    - SnpEff Annotation and Filtration Element
- Transcription Factor
    - Build Frequency Matrix Element
    - Build SITECON Model Element
    - Build Weight Matrix Element
    - Convert Frequency Matrix Element
    - Read Frequency Matrix Element
    - Read SITECON Model Element
    - Read Weight Matrix Element
    - Search for TFBS with SITECON Element
    - Search for TFBS with Weight Matrix Element
    - Write Frequency Matrix Element
    - Write SITECON Model Element
    - Write Weight Matrix Element
- Utils
    - DNA Statistics Element
    - Generate DNA Element

Data Readers

Data Readers *elements* read data (from files, remote databases, etc.) and provide them to other elements in a *workflows*.

- Read Alignment Element
- Read Annotations Element
- Read FASTQ File with SE Reads Element
- Read FASTQ Files with PE Reads Element
- Read File URL(s) Element
- Read NGS Reads Assembly Element
- Read Plain Text Element
- Read Sequence Element
- Read Sequence from Remote Database Element
- Read Variants Element

Read Alignment Element

Input one or several files in one of the multiple sequence alignment formats, supported by UGENE (ClustalW, FASTA, etc.).

The element outputs message(s) with the alignment data.

See the list of all available formats here.

Element type: read-msa

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Input files	A semicolon-separated list of paths to the input files.		**url-in**	*string*

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *Multiple sequence alignment*

Name in Workflow File: out-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*
Source URL	**url**	*string*

Read Annotations Element

Input one or several files with annotations: a file may also contain a sequence (e.g. GenBank format) or contain annotations only (e.g. GTF format).

The element outputs message(s) with the annotations data.

See the list of all available formats here.

Element type: read-annotations

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Input file(s)	Input files.	Dataset 1;	**url-in**	*string*
Mode	If the file contains more than one annotation table, Split mode sends them "as is" to the output, while Merge appends all the annotation tables and outputs the sole merged annotation table. In Merge files is the same as Merge but it operates with all annotation tables from all files of one dataset.	Merge	**mode**	*numeric*

Input/Output Ports

The element has 1 *output port*:

Name in GUI: Annotations

Name in Workflow File: out-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table-list*
Dataset name	**dataset**	*string*
Source URL	**out-url**	*string*

Read FASTQ File with SE Reads Element

Input one or several files with NGS single-end reads in FASTQ format. The element outputs the file(s) URL(s).

Type: get-se-reads-list

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Input file(s)	Input files.	Dataset 1;	**url1**	*string*

Input/Output Ports

The element has 1 *output port*:

Name in GUI: Output file

Name in Workflow File: out

Slots:

Slot InGUI	Slot in Workflow File	Type
Source URL 1	**reads-url1**	*string*

Read FASTQ Files with PE Reads Element

Input one or several pairs of files with NGS paired-end reads in FASTQ format. The element outputs the corresponding pairs of URLs.

Element type: get-pe-reads-list

Parameters in GUI

Parameter	Description	Default value	Parameter in Workflow File	Type
Input file(s)	Input files.	Dataset 1;	**url1**	*string*
Input file(s)	Input files.	Dataset 2;	**url2**	*string*

Input/Output Ports

The element has 1 *output port*:

Name in GUI: Output file

Name in Workflow File: out

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL 1	**reads-url1**	*string*
Source URL 2	**reads-url2**	*string*

Read File URL(s) Element

Input one or several files in any format. The element outputs the file(s) URL(s).

Element type: get-file-list

Parameters in GUI

Parameter	Description	Default value	Parameter in Workflow File	Type
Input directory	Input directory.		**in-path**	*string*

	Specify whether to output absolute or relative paths of the files.	True	**absolute**	*boolean*
Recursive reading	Get files from all nested directories or just from the current one.	False	**recursive**	*boolean*
Include name filter	Filter files by the specified value. It can be, for example, a file name or a regular expression of the file name.		**include-name-filter**	*string*
Exclude name filter	Exclude files using the specified filter value. The value can be, for example, a file name or a regular expression of the file name.		**exclude-name-filter**	*string*

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *out-url*

Name in Workflow File: out-url

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**out-url**	*string*

Read NGS Reads Assembly Element

Input one or several files with assembled NGS reads in SAM, BAM, or UGENEDB format.

The element outputs message(s) with the assembled reads data.

See the list of all available formats here.

Element type: read-assembly

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Input file(s)	Input files.	Dataset 1;	**url-in**	*string*

Input/Output Ports

The element has 1 *output port*:

Name in GUI: Assembly

Name in Workflow File: out-assembly

Slots:

Slot In GUI	Slot in Workflow File	Type
Assembly data	**assembly**	*assembly*
Dataset name	**dataset**	*string*
Source URL	**out-url**	*string*

Read Plain Text Element

Input one or several text files. The element outputs text message(s), read from the file(s).

See the list of all available formats here.

Element type: read-text

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Input files (required)	Semicolon-separated list of paths to the input files.		**url-in**	*string*
Read by lines (required)	Specifies to read the input file line by line.	false	**read-by-lines**	*boolean*

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *Plain text*

Name in Workflow File: out-text

Slots:

Slot In GUI	Slot in Workflow File	Type
Plain text	**text**	*string*
Source URL	**url**	*string*

Read Sequence Element

Input one or several files with nucleotide or protein sequences.

A file may also contain annotations. Any format, supported by UGENE, is allowed (GenBank, FASTA, etc.).

The element outputs message(s) with the sequence and annotations data.

See the list of all available formats here.

Element type: read-sequence

Parameters in GUI

Parameter	Description	Default value	Parameter in Workflow File	Type
Input files	Semicolon-separated list of datasets to the input files.		**url-in**	*string*
Mode	If the file contains more than one sequence, “split” mode sends them as is to output, while “merge” appends all the sequences and outputs the merged sequence.	Split	**mode**	*numeric* Available values are: • 0 - for split mode • 1 - for merge mode
Merging gap	In the “merge” mode, inserts the specified number of gaps between the original sequences. This is helpful e.g. to avoid finding false positives at the merge boundaries.	10	**merge-gap**	*numeric*
Sequence count limit	Split mode only. Read only first N sequences from each file. Set 0 value for reading all sequences.	0	**sequence-count-limit**	*numeric*
Accession filter	Only reports a sequence with the specified accession (id).		**accept-accession**	*string*

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *Sequence*

Name in Workflow File: out-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*
Set of annotations	**annotations**	*annotation-table*
Source URL	**url**	*string*

Read Sequence from Remote Database Element

Download sequence(s) with the specified ID(s) from one of the remote databases: NCBI, Ensembl, PDB, etc.

The sequences are downloaded with the associated annotations in a file format, specific for the selected database.

The element outputs message(s) with the sequence and annotations data.

Element type: fetch-sequence

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Resource IDs (required)	Semicolon-separated list of resource IDs in the database.		**resource-id**	*string*
Database (required)	Name of the database to read from.	NCBI Genbank (DNA sequence)	**database**	*string* Available values are: • ncbi-dna (NCBI GenBank (DNA sequence)) • ncbi-protein (NCBI protein sequence database) • pdb (PDB) • swiss-plot (SWISS-PROT) • uniprot-swiss-prot (UniProtKB /Swiss-Prot) • uniprot-trembl (UniProtKB /TrEMBL)
Save file to directory	Directory to store a file loaded from the database.	default	**save-dir**	*string*
Read resource ID(s) from source	The source to read resource IDs from the list or a local file.	List of TDs	**ids-source**	*string*

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *Sequence*

Name in Workflow File: out-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*
Set of annotations	**annotations**	*annotation-table*

Read Variants Element

Input one or several files with variants in one of the formats, supported by UGENE (e.g. VCF).

The element outputs message(s) with the variants data.

See the list of all available formats here.

Element type: read-variations

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Input file(s)	Input file(s).	Dataset 1	**url-in**	*string*

Input/Output Ports

The element has 1 *output port*:

Name in GUI: Variation track

Name in Workflow File: out-variations

Slots:

Slot In GUI	Slot in Workflow File	Type
Dataset name	**dataset**	*string*
Source url	**url**	*string*
Variation track	**variation-track**	*variation*

Data Writers

Data Writers *elements* write data supplied from other elements in a workflow to a file or files.

- Write Alignment Element
- Write Annotations Element
- Write FASTA Element
- Write NGS Reads Assembly Element
- Write Plain Text Element
- Write Sequence Element
- Write Variants Element

Write Alignment Element

The element gets message(s) with alignment data and saves the data to the specified file(s) in one of the multiple sequence alignment formats, supported by UGENE (ClustalW, FASTA, etc.).

Element type: write-msa

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Data storage	Place to store workflow results: local file system or a database.		**data-storage**	*string*
Document format	Format of the output file.	clustal	**document-format**	*string* Available values are: • clustal • mega • msf • sam • srfasta • stockholm
Output file	Location of the output data file. If this parameter is set, then the “Location” slot is not taken into account.		**url-out**	*string*
Output file suffix	This suffix will be used for generating the output file name.		**url-suffix**	*string*
Existing file	If a target file already exists, you can specify how it should be handled: either overwritten, renamed or appended (if supported by file format).	Rename	**write-mode**	*numeric* Available values are:

				• 0 - for overwrite • 1 - for append • 2 - for rename

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Multiple sequence alignment*

Name in Workflow File: in-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*
Location	**url**	*string*

Write Annotations Element

The element gets message(s) with annotations data and saves the data to the specified file(s) in one of the appropriate formats (GenBank, GTF, etc.).

Element type: write-annotations

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Data storage	Place to store workflow results: local file system or a database.		**data-storage**	*string*
Output file	Location of the output data file. If this attribute is set, slot “Location” in port will not be used.		**url-out**	*string*
Output file suffix	This suffix will be used for generating the output file name.		**url-suffix**	*string*
Existing file	If a target file already exists, you can specify how it should be handled: either overwritten, renamed or appended (if supported by file format).	Rename	**write-mode**	*numeric* Available values are: • 0 - for overwrite • 1 - for append • 2 - for rename
Document format	Document format of output file.	genbank	**document-format**	*string* Available values are: • CSV • GenBank • GFF
Annotations name	Object name of the annotations.	unknown feature	**annotations-name**	*string*
CSV separator	String which separates values in CSV file(s).	”,” (comma)	**separator**	*string*
Write sequence name	Write sequence to CSV file(s).	False	**write_names**	*boolean*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input annotations*

Name in Workflow File: in-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table-list*
Sequence	**sequence**	*sequence*
Source URL	**url**	*string*

Write FASTA Element

The element gets message(s) with sequence data and saves the data to the specified file(s) in FASTA format.

Element type: write-fasta

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output file	Location of the output data file. If this attribute is set, then the “Location” slot is not taken into account.		**url-out**	*string*
Output file suffix	This suffix will be used for generating the output file name.		**url-suffix**	*string*
Existing file	If a target file already exists, you can specifyhow it should be handled: either overwritten, renamed or appended (if supported by file format).	Rename	**write-mode**	*numeric* Available values are: • 0 - for overwrite • 1 - for append • 2 - for rename
Accumulate objects	Accumulates all incoming data in one file or creates separate files for each input. In the latter case, an incremental numerical suffix is added to a file name.	True	**accumulate**	*boolean*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Sequence*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*
Location	**url**	*string*
FASTA header	**fasta-header**	*string*

Write NGS Reads Assembly Element

The element gets message(s) with assembled reads data and saves the data to the specified file(s) in one of the appropriate formats (SAM, BAM, or UGENEDB).

Element type: write-assembly

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Data storage	Place to store workflow results: local file system or a database.		**data-storage**	*string*
Document format	Document format of the output file.	bam	**document-format**	*string*
Build index (BAM only)	Build BAM index for the target BAM file. The file .bai will be created in the same directory.	True	**build-index**	*boolean*
Output file	Location of output data file. If this attribute is set, slot "Location" in port will not be used.		**out-url**	*string*
Output file suffix	This suffix will be used for generating the output file name.		**url-suffix**	*string*
Existing file	If a target file already exists, you can specify how it should be handled: either overwritten, renamed or appended (if supported by file format). If Rename option is chosen existing file will be renamed.	Rename	**write-mode**	*numeric*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Assembly*

Name in Workflow File: in-assembly

Slots:

Slot In GUI	Slot in Workflow File	Type
Assembly data	**assembly**	*assembly*
Location	**url**	*string*

Write Plain Text Element

The element gets message(s) with text data and saved the data to the specified text file(s).

Element type: write-text

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Data storage	Place to store workflow results: local file system or a database.		**data-storage**	*string*
Output file	Location of the output data file. If this attribute is set, then the “Location” slot is not taken into account.		**url-out**	*string*
Output file suffix	This suffix will be used for generating the output file name.		**url-suffix**	*string*
Existing file	If a target file already exists, you can specify how it should be handled: either overwritten, renamed or appended (if supported by file format).	Rename	**write-mode**	*numeric* Available values are: • 0 - for overwrite • 1 - for append • 2 - for rename
Accumulate objects	Accumulates all incoming data in one file or creates separate files for each input. In the latter case, an incremental numerical suffix is added to a file name.	True	**accumulate**	*boolean*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Plain text*

Name in Workflow File: in-text

Slots:

Slot In GUI	Slot in Workflow File	Type
Plain text	**text**	*string*
Location	**url**	*string*

Write Sequence Element

The element gets message(s) with sequence data and, optionally, associated annotations data and saves the data to the specified file(s) in one of the appropriate formats (GenBank, FASTA, etc.).

Element type: write-sequence

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Data storage	Place to store workflow results: local file system or a database.		**data-storage**	*string*
Output file	Location of the output data file. If this attribute is set, then the “Location” slot is not taken into account.		**url-out**	*string*
Output file suffix	This suffix will be used for generating the output file name.		**url-suffix**	*string*
Existing file	If a target file already exists, you can specify how it should be handled: either overwritten, renamed or appended (if supported by file format).	Rename	**write-mode**	*numeric* Available values are: • 0 - for overwrite • 1 - for append • 2 - for rename
Document format	Format of the output file.	fasta	**document-format**	*string* Available values are: • fasta • fastq • genbank • raw
Accumulate objects	Accumulates all incoming data in one file or creates separate files for each input. In the latter case, an incremental numerical suffix is added to a file name.	True	**accumulate**	*boolean*
Split sequence	Split each incoming sequence on several parts.	1	**split**	*numeric*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Sequence*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type

Sequence	**sequence**	*sequence*
Location	**url**	*string*
Set of annotations	**annotations**	*annotation-table-list*

Write Variants Element

The element gets message(s) with variations data and saves the data to the specified file(s) in one of the appropriate formats (e.g. VCF).

Element type: write-variations

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Data storage	Place to store workflow results: local file system or a database.		**data-storage**	*split*
Accumulate objects	Accumulate all incoming data in one file or create separate files for each input.In the latter case, an incremental numerical suffix is added to the file name.	True	**accumulate**	*boolean*
Document format	Document format of output file.	snp	**document-format**	*string*
Output file	Location of output data file. If this attribute is set, slot "Location" in port will not be used.		**out-url**	*string*
Output file suffix	This suffix will be used for generating the output file name.		**url-suffix**	*string*
Existing file	If a target file already exists, you can specify how it should be handled: either overwritten, renamed or appended (if supported by file format). If Rename option is chosen existing file will be renamed.	Rename	**write-mode**	*numeric*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Variation track

Name in Workflow File: in-variations

Slots:

Slot In GUI	Slot in Workflow File	Type
Location	**url**	*string*
Variation track	**variation-track**	*variation*

Data Flow

- Filter Element
- Grouper Element
- Multiplexer Element
- Sequence Marker Element

Filter Element

This element passes through only data that matches the input filter value (or values).

Element type: filter-by-values

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Filter by value(s)	Semicolon-separated list of values used to filter the input data.		**text**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input values*

Name in Workflow File: in-data

Slots:

Slot In GUI	Slot in Workflow File	Type
Input values	**text**	*string*

The element has 1 *output port*:

Name in GUI: *Passing values (by Filter)*

Name in Workflow File: filtered-data

Grouper Element

The element groups data supplied to the specified slot by the specified property (for example, by value). Additionally, it is possible to merge data from another slots associated with the specified one.

Element type: grouper

Parameters

To use the *Grouper* element connect the *Grouper's* input port to the required workflow element. Select the *Grouper* element on the *Scene* and specify *Grou p slot* and *Group operation* parameters in the *Parameters* area in the *Property Editor*. To merge associated data, it is possible to create as many *Output slot (s)* as required (see details below).

Group slot

The *Group slot* specifies a *slot* that is used to group the input data. The list of available values of the parameter depend on the slots of workflow elements which produce data in the workflow before the *Grouper* element. There is a special *Unset* value. When it is selected, only one group is created.

Group operation

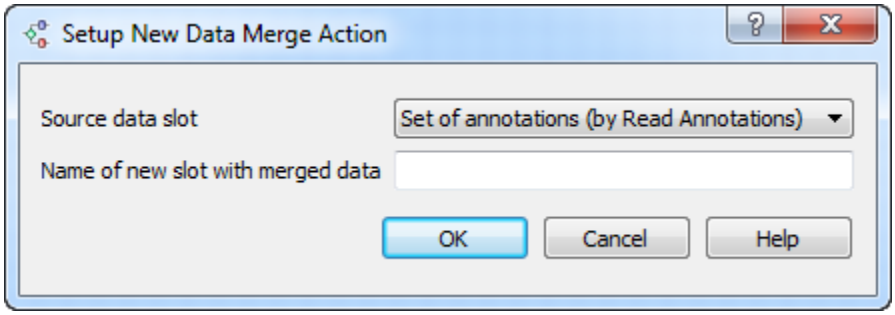
The *Group operation* specifies criteria to group data supplied to the *Group slot*. It can take the following values:

- *By value* - input data are compared by value (a group is created for each unique value, it can contain one or several identical values)
- *By identity* - input data are compared by internal data ID (all values are unique)
- *By name* - input data are compared by their names

By value group operation is available for group slots of types *Sequence*, *Set of annotations*, *MSA*, *Plain text*, *Source URL*. *By identity* and *By name* group operations are available for group slots of type *Sequence* only.

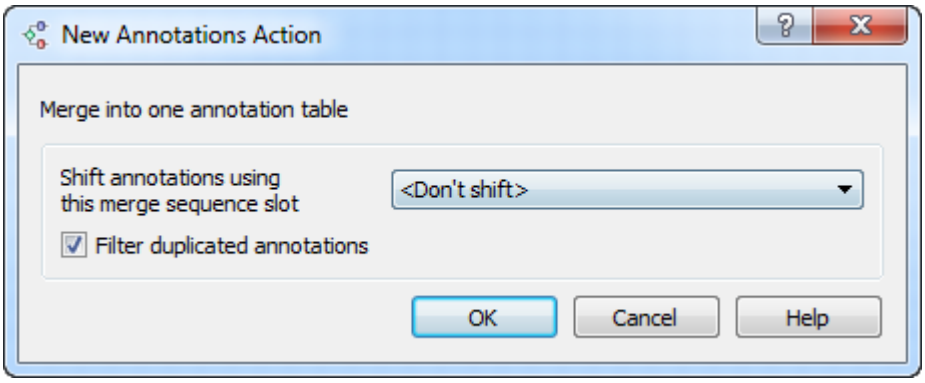
Output slots

When data supplied to the *Group slot* are divided into different groups the associated data are also got into a group. The possible associated data depend on the workflow. For example, a *Sequence Reader* element contains slots *Sequence* and *Set of annotations*. These data are **associated** as annotations belong to a sequence. Another example of associated data are sequence markers created by the *Sequence Marker* element. The associated data, therefore, can be additionally handled (i.e. merged) by the *Grouper*element. The action that can be performed on the associated data depends on their type. In any case to output handled associated data you need to create a new output slot in the*Grouper* element. To create it click the *Add* button in the *Gro uper's Parameters* area. The following dialog appears:



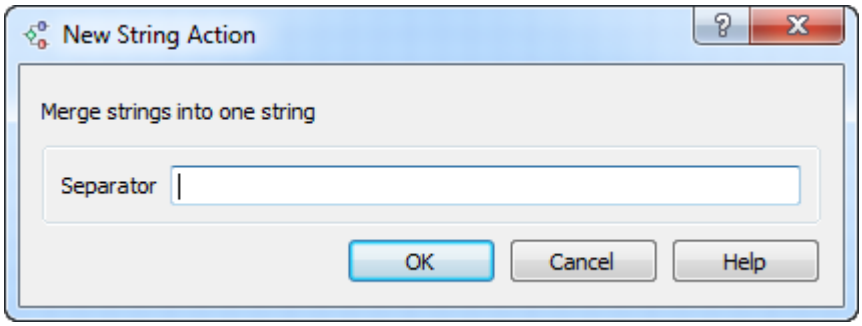
In the dialog you should select a *Source data slot* (i.e. a slot with the associated data) and input a name of the new slot. Click the OK button. A new dialog appears that specifies how the associated data should be merged. The view of the dialog and the available merge actions for different types of the *Source data slot* are the following:

- For a *Set of annotations* slot the *New Annotations Action* dialog appears:



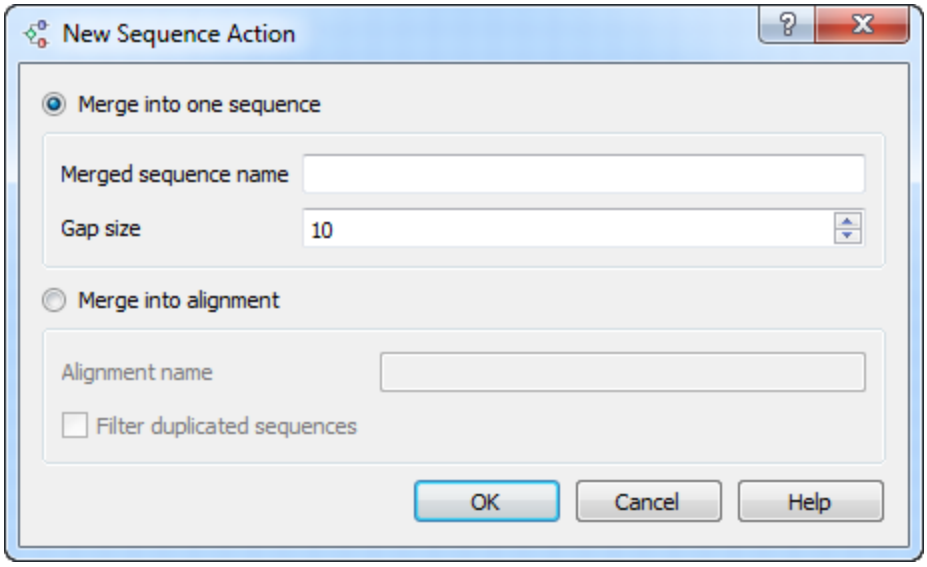
You can merge annotations into one annotation table and, optionally, filter duplicated annotations. Also, you can shift annotations. To do it, you need to create another output slot with type *Sequence* and *Merge into one sequence* option selected (see below). In other words you need to merge all sequences in a group into one sequence. In this case you select the corresponding sequence slot in the *New Annotations Action* dialog and each set of annotations in a group is shifted according to the corresponding sequence in the group. As the result you have one sequence and one set of annotations allocated on the whole sequence.

- For *Source URL* and *Plain text* slots the *New String Action* dialog appears:



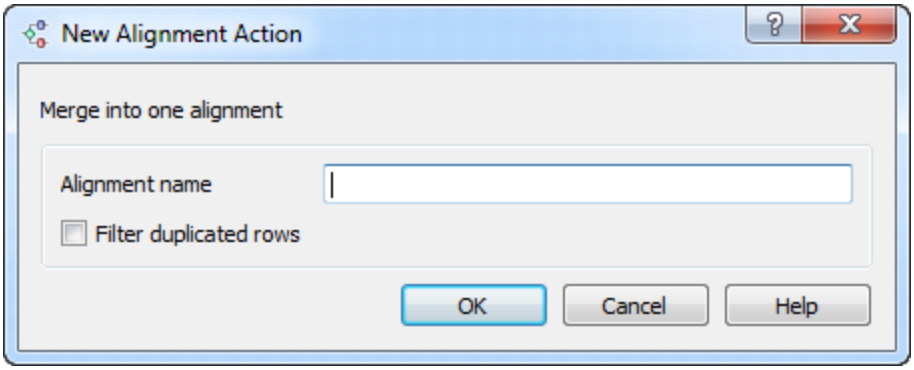
Using this dialog you can merge strings into one string. Optionally, you can specify an additional strings separator.

- For a *Sequence* slot the *New Sequence Action* dialog appears:



You can either merge all sequences in a group into one sequence or create a multiple sequence alignment. In the first case you need to specify the *Merged sequence name* and you can select the number of unknown characters between the merged sequences. In the second case you need to specify the alignment name. To filter duplicated sequence check the corresponding check box.

- For a *MSA* slot the *New Alignment Action* dialog appears:



Input the alignment name in this dialog. To filter duplicated rows check the corresponding check box.

To edit a created slot, select it in the *Parameters* area of the *Grouper* element and click the *Edit* button. To remove the slot, select it and click the *Remove* b utton.

Input/Output Ports

The element has 1 *input port* that can take any incoming data.

Name in GUI: *Input data flow*

Name in workflow File: input-data

The element has 1 *output port*.

Name in GUI: *Grouped output data flow*

Name in workflow File: output-data

Slots:

Slot In GUI	Slot in workflow File	Type
Group size	**group-size**	*string*

Also the port has one default slot of the grouped data and it may also have one or several customized output slots (see above).

Multiplexer Element

The element allows you to join two data flows into a single data flow, i.e. to join messages from two input ports into concatenated messages and send them to the output. The concatenation approach is determined by the *Multiplexing rule* parameter.

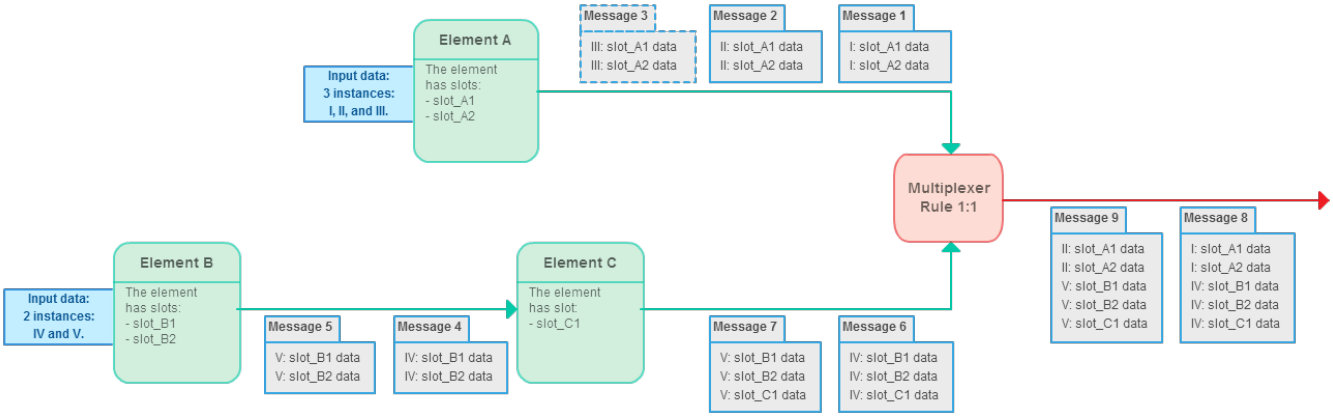
There are the following multiplexing rules:

- 1 to 1
- 1 to many

Rule: 1 to 1

This rule means that the multiplexer gets one message from the first input port and one message from the second input port, joins them into a single message, and transfers it to the output. This procedure is repeated while there are available messages in both input ports.

See *an example workflow* below:



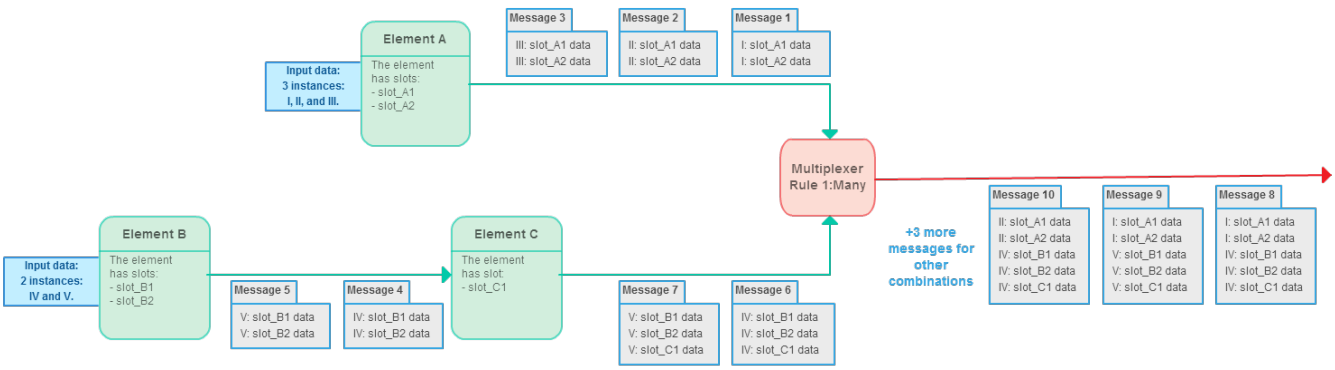
As you can see:

- There are elements **A**, **B**, **C**, and the Multiplexer.
- **A** and **B** are data readers.
- **A** gets three data objects as input. These objects are denoted as **I**, **II**, and **III**. **A** has two slots, so the input data objects may also have various data. For example, this may be "Sequence" and "Set of annotations" slots, and the data are read from three GenBank files.
- **B** gets two data objects as input. These objects are denoted as **IV** and **V**. **B** also has two slots in this example.
- **C** gets messages in the workflow from **B**. It has one output slot. For example, this may be a "Set of annotations" slot, i.e. additional annotations were calculated for input objects **IV** and **V**.
- Now in the Multiplexer element we have three messages from **A**, that correspond to the three input objects **I**, **II**, and **III**. And we have two messages from **B** and **C** elements, that correspond to the two input objects **IV** and **V** with additional information, calculated in **C**.
- The multiplexing rule is "1 to 1". This means that we only take into account messages that have a pair. Thus, "Message 3" is ignored in this case. However, the multiplexer concatenates the other messages. "Message 1" is concatenated with "Message 6", and "Message 8" is produced. "Message 2" is concatenated with "Message 7", and "Message 9" is produced.

Rule: 1 to many

This rule means that the multiplexer gets one message from the first input port, joins it with each message from the second input port, and transfers the joined messages to the output. This procedure is repeated for each message from the first input port.

See *an example workflow* below:



As you can see the conditions are the same as in the first "1 to 1" case, described above:

- As on the first image there are elements **A**, **B**, **C**, and the Multiplexer.
- **A** and **B** are data readers.
- **A** gets three data objects as input. These objects are denoted as **I**, **II**, and **III**. **A** has two slots.
- **B** gets two data objects as input. These objects are denoted as **IV** and **V**. **B** has two slots.
- **C** gets messages in the workflow from **B**. It has one output slot.
- The Multiplexer element receives three messages from **A** and two messages from **C**.

However, the multiplexing is done so that each message from **A** is concatenated from each message from **C**. As a result the following messages are produced:

- "Message 1" + "Message 6" = "Message 8"
- "Message 1" + "Message 7" = "Message 9"
- "Message 2" + "Message 6" = "Message 10"
- "Message 2" + "Message 7" = "Message 11"
- "Message 3" + "Message 6" = "Message 12"
- "Message 3" + "Message 7" = "Message 13"

Element type: multiplexer

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Multiplexing rule	Available values are: • 1 to 1 • 1 to many See the detailed description of the values above.	1 to 1	**multiplexing-rule**	*string*

Input/Output Ports

The *Multiplexer* element has ports, but it has not slots.

The element has 2 input port:

1. The first input port:
   - **Name in GUI:** *First input port*
   - **Name in Workflow File:** input-data-1
2. The second input port:
   - **Name in GUI:** *Second input port*
   - **Name in Workflow File:** input-data-2

The element has 1 output port:

- **Name in GUI:** *Multiplexed output*
- **Name in Workflow File:** output-data

Element in Samples

The element is used in the following workflow samples:

- Find Substrings in Sequences
- Merge Sequences and Annotations
- Search for TFBS

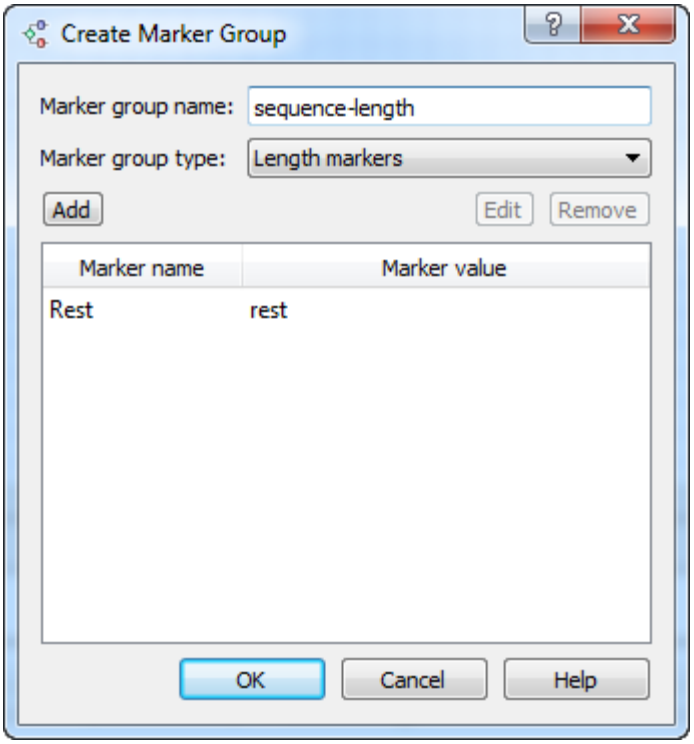
Sequence Marker Element

Adds one or several marks to the input sequence depending on the sequence properties. Use this element, for example, in conjunction with the *Filter* element.

Element type: mark-sequence

Parameters

To create a new marker group that would mark the input sequence, select the *Add* button in the *Parameters* area. The *Create Marker Group* dialog appears:



Choose a type of the marker group and input a marker group name. The following types are available:

Length markers — marks a sequence by length. The sequence is marked, for example, if its length is less or greater than the specified value.

Sequence name markers — marks a sequence by a sequence name.

Annotations count markers — marks a sequence by the number of annotations.

Qualifier integer value markers — marks a sequence by the number of integer qualifiers.

Qualifier text value markers — marks a sequence by the number of text qualifiers.

Qualifier float value markers — marks a sequence by the number of float qualifiers.

Text markers — marks a sequence by a file name. For example, if the name:

1. starts with the specified text;
2. ends with the specified text;
3. contains the specified text;
4. matches the specified regular expression .

Each marker group can contain more than one marker. Use the *Add*, *Edit* and *Remove* buttons in the dialog to create, modify and delete markers in the marker group.

To edit the created marker group, select the group in the *Parameters* area and click *Edit*:



To remove a marker group select it in the list and click *Remove*.

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Sequence*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*
Location	**url**	*string*
Set of annotations	**annotations**	*annotation-table-list*

The element has 1 *output port*.

Name in GUI: *Marked sequence*

Name in Workflow File: out-marked-seq

Slots:

Each created marker group adds a text slot with the following properties:

Slot In GUI	Slot in Workflow File	Type
Name of the marker group	Name of the marker group	*string*

Basic Analysis

- Amino Translations Element
- Annotate with UQL Element
- CD-Search Element
- Collocation Search Element
- Export PHRED Qualities Element
- Fetch Sequences by ID From Annotation Element
- Filter Annotation by Name Element
- Filter Annotations by Qualifier
- Find Correct Primer Pairs Element
- Find Pattern Element
- Find Repeats Element
- Gene-by-gene approach report
- Get Sequences by Annotations Element
- Group Primer Pairs Element
- Import PHRED Qualities Element
- Intersect Annotations Element
- Local BLAST+ Search Element
- Merge Annotations Element
- ORF Marker Element
- Remote BLAST Element
- Sequence Quality Trimmer Element
- Smith-Waterman Search Element

Amino Translations Element

Translates a sequence into it's amino translation or translations.

Element type: sequence-translation

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Translate from	Specifies position that should be used to translate the sequence from: first, second, third or all (three output amino sequences would be generated).	all	**pos-2-translate**	*string* Available values are: • all • first • second • third
Auto selected genetic code	Specifies that genetic code should be selected automatically.	True	**auto-translation**	*boolean*
Genetic code	Genetic code that should be used to translate the input nucleotide sequence.	The Standard	**genetic-code**	*string*

		Genetic Code		

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input Data*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

And 1 *output port*:

Name in GUI: *Amino sequence*

Name in Workflow File: out-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*
Plain text	**text**	*string*

Annotate with UQL Element

Analyzes a nucleotide sequence with a UGENE Query Language (UQL) workflow. The workflow specifies a set of features to search for and their positional relationship.

To learn more about UQL workflows read UGENE Query Designer Manual.

Element type: query

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Workflow (required)	UQL workflow file.		**schema**	*string*
Merge	Merges regions of each result into a single annotation.	False	**merge**	*boolean*
Offset	If the *Merge* parameter is set to *True*, adds left and right offsets of the specified length to the annotation.	0	**offset**	*numeric*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input sequences*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

And 1 *output port*:

Name in GUI: *Result annotations*

Name in Workflow File: out-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table*

CD-Search Element

Finds conserved domains in protein sequences. In case conserved domains database is downloaded the search can be executed on local machine. The search can be submitted to the NCBI for remote execution.

Element type: cd-search

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Annotate as	Name of the result annotations marking found conserved domains.	CDD result	**result-name**	*string*
Database	Currently, CD-Search is offered with the following search databases: • CDD - this is a superset including NCBI-curated domains and data imported from Pfam, SMART, COG, PRK, and TIGRFAM. • Pfam - a mirror of a recent Pfam-A database of curated seed alignments. Pfam version numbers do change with incremental updates. As with SMART, families describing very short motifs or peptides may be missing from the mirror. An HMM-based search engine is offered on the Pfam site. • SMART - a mirror of a recent SMART set of domain alignments. Note that some SMART families may be missing from the mirror due to update delays or because they describe very short conserved peptides and/or motifs, which would be difficult to detect using the CD-Search service. You may want to try the HMM-based search service offered on the SMART site. Note also that some SMART domains are not mirrored in CD because they represent “superfamilies” encompassing several individual, but related, domains; the corresponding seed alignments may not be available from the source database in these cases. Note also that SMART version numbers do not change with incremental updates of the source database (and the mirrored CD-Search database). • TIGRFAM - a mirror of a recent TIGRFAM set of domain alignments. An HMM-based search engine is offered on the TIGRFAM site. • COG - a mirror of the current COG database of orthologous protein families focusing on prokaryotes. Seed alignments have been generated by an automated process. An alternative search engine, “Cognitor”, which runs protein-BLAST against a database of COG-assigned sequences, is offered on the COG site. • KOG - a eukaryotic counterpart to the COG database. KOGs are not included in the CDD superset, but are searchable as a separate data set.	C D D Available values are: • CDD • Pfam • TIGRFAM • COG • KOG • Prk • SMART	**db-name**	*string*
Database directory	Specifies database directory for local search.		**db-path**	*string*
Local search	Perform the search on local machine or submit the search to NCBI for remote execution.	True	**local-search**	*boolean*
Expect value	Modifies the E-value threshold used for filtering results. False positive results should be very rare with the default setting of 0.01, results with E-values in the range of 1 and above should be considered putative false positives.		**e-val**	*numeric*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input sequence*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

And 1 *output port*:

Name in GUI: *Annotations*

Name in Workflow File: out-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table*

Collocation Search Element

Finds groups of specified annotations in each supplied set of annotations, stores found regions as annotations.

Element type: collocated-annotation-search

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Result type	Copy original annotations or annotate found regions with new ones.	Create new annotations	**result-type**	*string*
Result annotation (required)	Name of the result annotation to mark found collocations.	misc_feature	**result-name**	*string*
Include boundaries	Include most left and most right boundary annotations regions into result or exclude them.	True	**annotations**	*string*
Group of annotations (required)	List of annotation names to search. Found regions will contain all the named annotations.		**include-boundary**	*boolean*
Region size	Effectively this is the maximum allowed distance between the interesting annotations in a group.	1000	**region-size**	*numeric*
Must fit into region	Specifies whether the interesting annotations should entirely fit into the specified region to form a group.	False	**must-fit**	*boolean*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input data*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*
Set of annotations	**annotations**	*annotation-table-list*

And 1 *output port*:

Name in GUI: *Group annotations*

Name in Workflow File: out-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table*

Export PHRED Qualities Element

Export corresponding PHRED quality scores from input sequences.

Element type: export-phred-qualities

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
PHRED output	Path to file with PHRED quality scores.		**url-out**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: DNA sequences

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*string*

Fetch Sequences by ID From Annotation Element

Parses annotations to find any IDs and fetches corresponding sequences.

Element type: fetch-sequence

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Save file to directory	The directory to store sequence files loaded from a database.	default	**save-dir**	*string*
NCBI database	The database to read from.	nucleotide Available values are: • nucleotide • protein	**database**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input annotations*

Name in Workflow File: in-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table*

And 1 *output port*:

Name in GUI: *Sequence*

Name in Workflow File: out-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table*
Sequence	**sequence**	*sequence*

Filter Annotation by Name Element

Filters annotations by name.

Element type: filter-annotations

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Annotation names	List of annotation names, separated by spaces, that will be accepted or filtered.		**annotation-names**	*string*
Annotation names file	File with annotation names, separated with whitespaces which will be accepted or filtered.		**annotation-names-file**	*string*
Accept or filter	Selects the name filter: accept specified names or accept all except specified.	True	**accept-or-filter**	*boolean*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input annotations*

Name in Workflow File: in-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table*

And 1 *output port*:

Name in GUI: *Result annotations*

Name in Workflow File: out-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table*

Filter Annotations by Qualifier

Filters annotations by qualifier.

Element type: filter-annotations-by-qualifier

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Qualifier name	Name of the qualifier to use for filtering.		**qualifier-name**	*string*
Qualifier value	Text value of the qualifier to apply as filtering criteria.		**qualifier-value**	*string*
Accept or filter	Selects the name filter: accept specified names or accept all except specified.	True	**accept-or-filter**	*boolean*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input annotations*

Name in Workflow File: in-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table*

And 1 *output port*:

Name in GUI: *Result annotations*

Name in Workflow File: out-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table*

Find Correct Primer Pairs Element

Find correct primer pairs, which consist of valid primers without dimers.

Element type: find-primers

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output report file	Path to the report output file.		**output-file**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input sequences

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	sequence	*sequence*

Find Pattern Element

Searches regions in a sequence similar to a pattern sequence. Outputs a set of annotations.

Element type: search

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Annotate as	Name of the result annotation.	misc_feature	**result-name**	*string*
Pattern(s)	Semicolon-separated list of patterns to search for.		**pattern**	*string*
Pattern file	Load pattern from file in any sequence format or in newline-delimited format.		**pattern_file**	*string*
Use pattern name	If patterns are loaded from a file, use names of pattern sequences as annotation names. The name from the parameters is used by default.	False	**use-names**	*boolean*
Max Mismatches	Maximum number of mismatches between a substring and a pattern.	0	**max-mismatches-num**	*numeric*
Search in	Specifies which strands should be searched: direct, complementary or both.	both strands	**strand**	*numeric* Available values are: • 0 - for searching in both strands • 1 - for searching in direct strand • 2 - for searching in complement strand
Allow Insertions /Deletions	Takes into account possibility of insertions/deletions when searching. By default substitutions are only considered.	False	**allow-ins-del**	*boolean*
Support ambiguous bases	Performs correct handling of ambiguous bases. When this option is activated insertions and deletions are not considered.	False	**ambiguous**	*boolean*
Search in Translation	Translates a supplied nucleotide sequence to protein and searches in the translated sequence.	False	**amino**	*boolean*
Qualifier name for pattern name	Name of qualifier in result annotations which is containing a pattern name.	pattern_name	**pattern-name-qual**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input data*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*
Plain text	**text**	*string*

And 1 *output port*:

Name in GUI: *Pattern annotations*

Name in Workflow File: out-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table*

Find Repeats Element

Finds repeats in each supplied sequence, stores found regions as annotations.

Element type: repeats-search

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Annotate as (required)	Name of the result annotation to mark found repeats.	repeat_unit	**result-name**	*string*
Algorithm	Control over variations of the algorithm.	Auto	**algorithm**	*numeric* Available values are: • 0 - algorithm choosed automaticly • 1 - for diagonal algorithm • 2 - for suffix index algorithm
Filter nested	Filters nested repeats.	True	**filter-nested**	*boolean*
Identity	Repeats identity in percent.	100	**identity**	*numeric*
Inverted	Specifies to search for inverted repeats.	False	**inverted**	*boolean*
Max distance	Maximum distance between the repeats.	5000	**max-distance**	*numeric*
Min distance	Minimum distance between the repeats.	0	**min-distance**	*numeric*
Min length	Minimum length of the repeats.	5	**min-length**	*numeric*
Parallel threads	Number of parallel threads used for the task.	Auto	**threads**	*numeric* 0 - for using autodetected threads number

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input sequence*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

And 1 *output port*:

Name in GUI: *Repeat annotations*

Name in Workflow File: out-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table*

Gene-by-gene approach report

Output a table of genes found in a reference sequence.

Element type: genebygene-report-id

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output file	File to store a report.		**output-file**	*string*
Annotation name	Annotation name used to compare genes and reference genomes.	blast-result	**annotation_name**	*string*
Existing file	If a target report already exists you should specify how to handle that. Merge two table in one. Overwrite or Rename existing file.	Merge	**existing**	*string*
Identity cutoff	Identity between gene sequence length and annotation length in percent. BLAST identity (if specified) is checked after	90.0000%	**identity**	*numeric*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Gene by gene report data

Name in Workflow File: in-data

Slots:

Slot In GUI	Slot in Workflow File	Type
Input annotations	**gene-ann**	*ann-table-list*
Input sequences	**gene-seq**	*seq*

Get Sequences by Annotations Element

Extracts annotated regions from the input sequence.

Element type: extract-annotated-sequence

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Translate	Translates the annotated regions if the corresponding annotation marks a protein subsequence.	False	**translate**	*boolean*
Complement	Complements the annotated regions if the corresponding annotation is located on the complement strand.	False	**complement**	*boolean*
Split joined	Split joined annotations to single region annotations.	False	**split-joined-annotations**	*boolean*
Extend left	Extends the resulted regions to left.	0	**extend-left**	*numeric*
Extend right	Extends the resulted regions to right.	0	**extend-right**	*numeric*
Gap length	Inserts a gap of a specified length between the merged locations of the annotation.	0	**merge-gap-length**	*numeric*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input sequence*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

Set of annotations	**annotations**	*annotation-table*

And 1 *output port*:

Name in GUI: *Annotated regions*

Name in Workflow File: out-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

Group Primer Pairs Element

Select groups of primer pairs, which can be simultaneously used in one reaction tube.

The primers must be supplied in the following order: pair1_direct_primer, pair1_reverse_primer, pair2_direct_primer, pair2_reverse_primer, etc.

Element type: primers-grouper

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output report file	Path to the report output file.		**output-file**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Primer pairs*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

Import PHRED Qualities Element

Adds corresponding PHRED quality scores to the sequences. Use this element to convert .fasta and .qual pair to fastq format.

Element type: import-phred-qualities

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
PHRED input (required)	Path to a file with PHRED quality scores.		**url-in**	*string*
Quality format	Format to encode quality scores.	Sanger	**quality-format**	*string* Available values are: • Sanger • Illumina 1.3+ • Solexa/Illumina 1.0

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *DNA sequences*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type

Sequence	**sequence**	*sequence*

And 1 *output port*:

Name in GUI: *DNA sequences with imported quailities*

Name in Workflow File: out-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

Intersect Annotations Element

Intersects two sets of annotations denoted as A and B.

Element type: intersect-annotations

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Result annotations	Select one of the following: • Shared interval to report intervals shared between overlapped annotations from set A and set B. • Overlapped annotations from A to report annotations from set A that have an overlap with annotations from set B. • Non-overlapped annotations from A to report annotations from set A that have NO overlap with annotations from set B.	Overlapped annotations from set A	**report**	*numeric*
Unique overlaps	If the parameter value is "True", write original A entry once if any overlaps found in B. In other words, just report the fact at least one overlap was found in B. The minimum overlap number is ignored in this case. If the parameter value is "False", the A annotation is reported for every overlap found.	True	**unique**	*boolean*
Minimum overlap	Minimum overlap required as a fraction of an annotation from set A. By default, even 1 bp overlap between annotations from set A and set B is taken into account. Yet sometimes you may want to restrict reported overlaps to cases where the annotations in B overlaps at least X% (e.g. 50%) of the A annotation. This parameter is only available if the parameter "Unique overlaps" is "False".	0.0000001%	**minimum-overlap**	*numeric*

Input/Output Ports

Name in GUI: Annotations AThe element has 2 *input ports*:

Name in Workflow File: input-annotations-a

Slots:

Slot In GUI	Slot in Workflow File	Type
Annotations A	**annotations**	*annotation-table*

Name in GUI: Annotations B

Name in Workflow File: input-annotations-b

Slots:

Slot In GUI	Slot in Workflow File	Type
Annotations B	**annotations**	*annotation-table*

And 1 *output port*:

Name in GUI: Annotations

Name in Workflow File: output-intersect-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Annotations	**annotations**	*annotation-table*

Local BLAST+ Search Element

Finds annotations for DNA sequence in a local BLAST database.

Element type: blast-plus

BLAST+ is used as an external tool from UGENE and it must be installed on your system. To learn more about the external tools, please, read main UGENE User Manual.

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Search type	Selects the type of the BLAST searches.	blastn	**blast-type**	*string* Available values are: • blastn • blastp • blastx • tblastn • tblastx
Database path	Path to the database files.		**db-path**	*string*
Database name	Base name for BLAST DB files.		**db-name**	*string*
Tool path	Path to the BLAST executable.	default	**tool-path**	*string*
Temporary directory	Directory for temporary files.	default	**temp-dir**	*string*
Expected value	Expectation threshold value.	10	**e-val**	*numeric*
Culling limit	If the query range of a hit is enveloped by that of at least this many higher-scoring hits, delete the hit	0	**max-hits**	*numeric*
Annotate as	Name of the result annotations.	blast_result	**result-name**	*string*
Gapped alignment	Perform gapped alignment.	use	**gapped-aln**	*boolean*
Gap costs	Cost to create and extend a gap in an alignment.	2 2	**gap-costs**	*string*
Match scores	Reward and penalty for matching and mismatching bases.	1 -3	**match-scores**	*string*
BLAST output	Location of BLAST output file.		**blast-output**	*string*
BLAST output type	Type of BLAST output file.	XML (-outfmt 5)	**type-output**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input sequence*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

And 1 *output port*:

Name in GUI: *Annotations*

Name in Workflow File: out-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table*

Merge Annotations Element

Writes all supplied sequences to file(s) in FASTQ format.

Element type: write-fastq

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output file (required)	Location of the output data file. If this attribute is set, then the “Location” slot is not taken into account.		**url-out**	*string*
Existing file	If a target file already exists, you can specify how it should be handled: either overwritten, renamed or appended (if supported by file format).	Rename	**write-mode**	*numeric* Available values are: • 0 - for overwrite • 1 - for append • 2 - for rename
Accumulate objects	Accumulates all incoming data in one file or creates separate files for each input. In the latter case, an incremental numerical suffix is added to a file name.	True	**accumulate**	*boolean*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Sequence*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*
Location	**url**	*string*

ORF Marker Element

Finds Open Reading Frames (ORFs) in each supplied nucleotide sequence, stores found regions as annotations.

Element type: orf-search

Parameters in GUI

Parameter	Description	Default value	Parameter in Workflow File	Type

Annotate as (required)	Name of the result annotations.	ORF	**result-name**	*string*
Search in	Specifies which strands should be searched: direct, complement or both.	both strands	**strand**	*numeric* Available values are: • 0 - for searching in both strands • 1 - for searching in direct strand • 2 - for searching in complement strand
Min length	Ignores ORFs shorter than the specified length.	100	**min-length**	*numeric*
Genetic code	Specifies which genetic code should be used for translating the input nucleotide sequence.	The Standard Genetic Code	**genetic-code**	*string* Available values are: • NCBI-GenBank #1 • NCBI-GenBank #2 • etc.
Require init codon	Allows or not ORFs starting with any codon other than terminator.	True	**require-init-codon**	*boolean*
Require stop codon	Ignores boundary ORFs which last beyound the search region (i.e. have no stop codon within the range).	False	**require-stop-codon**	*boolean*
Allow alternative codons	Allows ORFs starting with alternative initiation codons, accordingly to the current translation table.	False	**allow-alternative-codons**	*boolean*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input sequence*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

And 1 *output port*:

Name in GUI: *ORF annotations*

Name in Workflow File: out-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table*

Remote BLAST Element

Finds annotations for the supplied DNA sequence in the NCBI remote database.

Element type: **blast**

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Database	Selects the database to search through. Available databases are blastn, blastp and cdd.	ncbi-blastn	**db**	*string* Available values are: • ncbi-blastn

				• ncbi-blastp • ncbi-cdd
Database	Select the database to search through.		**db**	*string*
Expected value	This parameter specifies the statistical significance threshold of reporting matches against the database sequences.	10	**e-val**	*string*
Results limit	The maximum number of results.	10	**hits**	*numeric*
Megablast	Use megablast.	False	**megablast**	*boolean*
Short sequence	Optimizes search for short sequences.	False	**short-sequence**	*boolean*
Entrez query	Enter an Entrez query to limit search.		**entrez-query**	*string*
Annotate as	Name of the result annotations.		**result-name**	*string*
BLAST output	Location of the BLAST output file. This parameter insignificant for cdd search.		**blast-output**	*string*
Gap costs	Cost to create and extend a gap in an alignment.	2 2	**gap-costs**	*string*
Match scores	Reward and penalty for matching and mismatching bases.	1 -3	**match-scores**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input sequence*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

And 1 *output port*:

Name in GUI: *Annotations*

Name in Workflow File: out-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table*

Sequence Quality Trimmer Element

Scans each input sequence from the end to find the first position where the quality is greater or equal to the minimum quality threshold.

Then it trims the sequence to that position.

If a whole sequence has quality less than the threshold or the length of the output sequence less than the minimum length threshold then the sequence is skipped.

Elemet type: SequenceQualityTrim

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Trimming quality threshold	Quality threshold for trimming.	30	**qual-id**	*numeric*
Min length	Too short reads are discarded by the filter.	0	**len-id**	*numeric*
Trim both ends	Trim both ends of a read or not. Usually, you need to set **True** for **Sanger** sequencing and **False** for **NGS**	True	**both-ends**	*boolean*

Input/Output Ports

The element has 1 *input port.*

Name in GUI: *Input data*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

And 1 *output port*:

Name in GUI: *Output data*

Name in Workflow File: out-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

Smith-Waterman Search Element

Searches regions in a sequence similar to a pattern sequence. Outputs a set of annotations.

Under the hood is the well-known Smith-Waterman algorithm for performing local sequence alignment.

Element type: ssearch

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Substitution Matrix	Describes the rate at which one character in a sequence changes to other character states over time.	Auto	**matrix**	*string* Available values are: • Auto - for auto detecting matrix • blosum60 • dna • rna • ...
Algorithm	Version of the Smith-Waterman algorithm. You can use the optimized versions of the algorithm (SSE, CUDA and OpenCL) if your hardware supports these capabilities.	OPENCL	**algorithm**	*string* Available values are: • Classic 2 • SSE2 • OpenCL • CUDA
Filter Results	Specifies either to filter the intersected results or to return all the results.	filter-intersections	**filter-strategy**	*string* Available values are: • filter-intersections • none
Min Score	Minimal percent similarity between a sequence and a pattern.	90%	**min-score**	*numeric*
Search in	Specifies which strands should be searched: direct, complementary or both.	both strands	**strand**	*numeric* Available values are: • 0 - for searching in both strands

				• 1 - for searching in direct strand • 2 - for searching in complement strand
Search in Translation	Translates a supplied nucleotide sequence to protein and searches in the translated sequence.	False	**amino**	*boolean*
Gap Open Score	Penalty for opening a gap.	-10.0	**gap-open-score**	*numeric*
Gap Extension Score	Penalty for extending a gap.	-1.0	**gap-ext-score**	*numeric*
Use Pattern Names	Use a pattern name as an annotation name.	True	**use-names**	*boolean*
Annotate as	Name of the result annotations.	misc_feature	**result-name**	*string*
Qualifier name for pattern name	Name of qualifier in result annotations which is containing a pattern name.	pattern name	**pattern-name-qual**	*string*

Input/Output Ports

The element has 2 *input ports.* The first input port:

Name in GUI: *Input data*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

The second input port:

Name in GUI: *Pattern data*

Name in Workflow File: pattern

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

And 1 *output port*:

Name in GUI: *Pattern annotations*

Name in Workflow File: out-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table*

Data Converters

- Convert bedGraph Files to bigWig Element
- Convert Text to Sequence Element
- File Format Conversion Element
- Reverse Complement Element
- Split Assembly into Sequences Element

Convert bedGraph Files to bigWig Element

Convert bedGraph files to bigWig.

Element type: bgtbw-bam

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	Select an output directory. Custom - specify the output directory in the 'Custom directory' parameter. Workflow - internal workflow directory. Input file - the directory of the input file.	Input file	**out-mode**	*numeric*
Custom directory	Specify the output directory.		**custom-dir**	*string*
Genome	File with genome length.	human. hg18	**genome**	*string*
Output name	A name of an output file. If default of empty value is provided the output name is the name of the first file with additional extention.		**out-name**	*string*
Block size	Number of items to bundle in r-tree (-blockSize).	256	**bs**	*numeric*
Items per slot	Number of data points bundled at lowest level (-itemsPerSlot).	1024	**its**	*numeric*
Uncompressed	If set, do not use compression.(-unc).	False	**unc**	*boolean*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: BedGrapgh files

Name in Workflow File: in-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

And 1 *output port*:

Name in GUI: BigWig files

Name in Workflow File: out-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

Convert Text to Sequence Element

Converts the input text to a sequence.

Element type: convert-text-to-sequence

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Sequence name (required)	Result sequence name.	*Sequence*	**sequence-name**	*string*
Sequence alphabet	Alphabet of the sequence. Chooose *Auto* to auto-detect the alphabet or one of the following values: • *All symbols* • *Extended DNA* • *Extended RNA* • *Standard DNA* • *Standard RNA* • *Standard amino*	*Auto*	**alphabet**	*string*

Skip unknown symbols	If *True*, ignores all symbols that are not presented in the sequence alphabet selected.	*True*	**skip-unknown**	*boolean*
Replace unknown symbols with	Replaces all unknown symbols with the specified symbol.	*N*	**replace-unknown-with**	*string* (1 character)

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input text*

Name in Workflow File: in-text

Slots:

Slot In GUI	Slot in Workflow File	Type
Plain text	**text**	*string*

And 1 *output port*:

Name in GUI: *Output sequence*

Name in Workflow File: out-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

File Format Conversion Element

Converts the file to the selected format if it is not excluded.

Element type: files-conversion

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Document format	Document format of the output file.		**document-format**	*string*
Excluded formats	The input file won't be converted to any of the selected formats.		**excluded-formats**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: File

Name in Workflow File: in-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**input-url**	*string*

And 1 *output port*:

Name in GUI: File

Name in Workflow File: out-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**output-url**	*string*

Reverse Complement Element

Converts input sequence into its reverse, complement or reverse-complement counterpart.

Element type: reverse-complement

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Operation type	Selects either to produce the reverse, complement, or reverse-complement sequence.	Reverse Complement	**op-type**	*string* Available values are: • reverse-complement • complement • reverse

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input sequence*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

And 1 *output port*:

Name in GUI: *Output sequence*

Name in Workflow File: out-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

Split Assembly into Sequences Element

Splits assembly into sequences(reads).

Element type: reverse-complement

Input/Output Ports

The element has 1 *input port*:

Name in GUI: in-assembly

Name in Workflow File: in-assembly

Slots:

Slot In GUI	Slot in Workflow File	Type
Assembly data	**assembly**	*assembly*

And 1 *output port*:

Name in GUI: out-sequence

Name in Workflow File: out-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**seq**	*string*

DNA Assembly

- Assembly Sequences with CAP3

Assembly Sequences with CAP3

CAP3 is a contig assembly program. It allows assembly long DNA reads (up to 1000 bp).

Binaries can be downloaded from http://seq.cs.iastate.edu/cap3.html Huang, X. and Madan, A. (1999) CAP3: A DNA Sequence Assembly Program, Genome Research, 9: 868-877.

Element type: cap3

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output file	Write assembly results to this output file in ACE format..	result.ace	**out-file**	*string*
Quality cutoff for clipping	Base quality cutoff for clipping (-c).	12	**clipping-cutoff**	*numeric*
Clipping range	Set a number which unit is base. It will get the refGenes in n bases from peak center. (--distance).	100	**clipping-range**	*numeric*
Quality cutoff for differeneces	Base quality cutoff for differences (-b).	20	**diff-cutoff**	*numeric*
Maximum difference score	Max qscore sum at differences (-d). If an overlap contains lots of differences at bases of high quality, then the overlap is removed. The difference score is calculated as follows. If the overlap contains a difference at bases of quality values q1 and q2, then the score at the difference is max(0, min(q1, q2) - b), where b is Quality cutoff for differences. The difference score of an overlap is the sum of scores at each difference.	200	**diff-max-qscore**	*numeric*
Match score factor	Match score factor (-m) is one of the parameters that affects similarity score of an overlap. See Overlap similarity score cutoff description for details.	2	**match-score-factor**	*numeric*
Mismatch score factor	Mismatch score factor (-n) is one of the parameters that affects similarity score of an overlap. See Overlap similarity score cutoff description for details.	-5	**mismatch-score-factor**	*numeric*
Gap penalty factor	Gap penalty factor (-g) is one of the parameters that affects similarity score of an overlap. See Overlap similarity score cutoff description for details.	6	**gap-penalty-factor**	*numeric*
Overlap similarity score cutoff	If the similarity score of an overlap is less than the overlap similarity score cutoff (-s), then the overlap is removed. The similarity score of an overlapping alignment is defined using base quality values as follows. A match at bases of quality values q1 and q2 is given a score of m * min(q1,q2), where m is Match score factor. A mismatch at bases of quality values q1 and q2 is given a score of n * min(q1,q2), where n is Mismatch score factor. A base of quality value q1 in a gap is given a score of -g * min(q1,q2), where q2 is the quality value of the base in the other sequence right before the gap and g is Gap penalty factor. The score of a gap is the sum of scores of each base in the gap minus a gap open penalty. The similarity score of an overlapping alignment is the sum of scores of each match, each mismatch, and each gap.	900	**overlap-sim-score-cutoff**	*numeric*
Overlap length cutoff	An overlap is taken into account only if the length of the overlap in bp is no less than the specified value (parameter -o of CAP3).	40	**overlap-length-cutoff**	*numeric*
Overlap percent identity cutoff	An overlap is taken into account only if the percent identity of the overlap is no less than the specified value (parameter -p of CAP3).	90	**overlap-perc-id-cutoff**	*numeric*
Max number of word matches	This parameter allows one to trade off the efficiency of the program for its accuracy (parameter -t of CAP3). For a read f, CAP3 computes overlaps between read f and other reads by considering short word matches between read f and other reads. A word match is examined to see if it can be extended into a long overlap. If read f has overlaps with many other reads, then read f has many short word matches with many other reads. This parameter gives an upper limit, for any word, on the number of word matches between read f and other reads that are considered by CAP3. Using a large value for this parameter allows CAP3 to consider more word matches between read f and other reads, which can find more overlaps for read f, but slows down the program. Using a small value for this parameter has the opposite effect.	300	**max-num-word-matches**	*numeric*

Band expansion size	CAP3 determines a minimum band of diagonals for an overlapping alignment between two sequence reads. The band is expanded by a number of bases specified by this value (parameter -a of CAP3).	20	**band-exp-size**	*numeric*
Max gap length in an overlap	The maximum length of gaps allowed in any overlap (-f). I.e. overlaps with longer gaps are rejected. Note that a small value for this parameter may cause the program to remove true overlaps and to produce incorrect results. The parameter may be used to split reads from alternative splicing forms into separate contigs.	20	**max-gap-in-overlap**	*numeric*
Assembly reverse reads	Specifies whether to consider reads in reverse orientation for assembly (originally, parameter -r of CAP3).	True	**assembly-reverse**	*boolean*
CAP3 tool path	The path to the CAP3 external tool in UGENE.	default	**path**	*string*
Temporary directory	The directory for temporary files.	default	**tmp-dir**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input sequences

Name in Workflow File: in-data

Slots:

Slot In GUI	Slot in Workflow File	Type
Dataset name	**dataset**	*string*
Input URL(s)	**in.url**	*string*

HMMER2 Tools

- HMM2 Build Element
- HMM2 Search Element
- Read HMM2 Profile Element
- Write HMM2 Profile Element

HMM2 Build Element

Builds a HMM profile from a multiple sequence alignment. The HMM profile is a statistical model which captures position-specific information about how conserved each column of the alignment is, and which residues are likely.

Element type: hmm2-build

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Profile name	Descriptive name of the HMM profile.		**profile-name**	*string*
HMM strategy	Specifies the kind of alignments you want to allow.	hmmls	**strategy**	*numeric* Available values are: • 0 - for hmms • 1 - for hmmls

				• 2 - for hmmfs • 3 - for hmmsw
Calibrate profile	Enables/disables optional profile calibration. An empirical HMM calibration costs time but it only has to be done once per model, and can greatly increase the sensitivity of a database search.	True	**calibrate**	*boolean*
Parallel calibration	Number of parallel threads that the calibration will run in.	1	**calibration-threads**	*numeric*
Standard deviation	Standard deviation of the synthetic sequence length. A positive number. Note that the Gaussian is left-truncated so that no sequences have lengths.	200.0	**deviation**	*numeric*
Fixed length of samples	Fixes the length of the random sequences to, where is a positive (and reasonably sized) integer. The default is instead to generate sequences with a variety of different lengths, controlled by a Gaussian (normal) distribution.	0	**fix-samples-length**	*numeric*
Mean length of samples	Mean length of the synthetic sequences, positive real number.	325	**mean-samples-length**	*numeric*
Number of samples	Number of synthetic sequences. If is less than about 1000, the fit to the EVD may fail Higher numbers of will give better determined EVD parameters. The default is 5000; it was empirically chosen as a tradeoff between accuracy and computation time.	5000	**samples-num**	*numeric*
Random seed	The random seed, where is a positive integer. The default is to use time() to generate a different seed for each run, which means that two different runs of hmmcalibrate on the same HMM will give slightly different results. You can use this option to generate reproducible results for different hmmcalibrate runs on the same HMM.	0	**seed**	*numeric*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input MSA*

Name in Workflow File: in-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*

And 1 *output port*:

Name in GUI: *HMM profile*

Name in Workflow File: out-hmm2

Slots:

Slot In GUI	Slot in Workflow File	Type
HMM profile	**hmm2-profile**	*hmm2-profile*

HMM2 Search Element

Searches each input sequence for significantly similar sequence matches to all specified HMM profiles. In case several profiles were supplied, searches with all profiles one by one and output united set of annotations for each sequence

Element type: hmm2-search

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Result annotation	Name of the result annotations.	hmm_signal	**result-name**	*string*
Filter by high E-value	E-value filtering can be used to exclude low-probability hits from result.	1e-1	**e-val**	*numeric*

Number of seqs	Calculates the E-value scores as if we had seen a sequence database of sequences.	1	**seqs-num**	*numeric*
Filter by low score	Score based filtering is an alternative to E-value filtering to exclude low-probability hits from result.	-1000000000.0	**score**	*numeric*

Input/Output Ports

The element has 2 *input port*. The first gets the input sequence:

Name in GUI: *Input sequence*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

The second input port gets the HMM profile:

Name in GUI: *HMM profile*

Name in Workflow File: in-hmm2

Slots:

Slot In GUI	Slot in Workflow File	Type
HMM profile	**hmm2-profile**	*hmm2-profile*

And 1 *output port*:

Name in GUI: *HMM annotations*

Name in Workflow File: out-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table*

Read HMM2 Profile Element

Reads HMM profiles from file(s). The files can be local or Internet URLs.

Element type: hmm2-read-profile

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Input files (required)	Semicolon-separated list of paths to the input files.		**url-in**	*string*

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *HMM profile*

Name in Workflow File: out-hmm2

Slots:

Slot In GUI	Slot in Workflow File	Type
HMM profile	**hmm2-profile**	*hmm2-profile*

Write HMM2 Profile Element

Saves all input HMM profiles to the specified locations.

Element type: hmm2-write-profile

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output file (required)	Location of the output data file. If this attribute is set, the “Location” slot is not taken into account.		**url-out**	*string*
Existing file	If a target file already exists, you can specify how it should be handled: either overwritten, renamed or appended (if supported by file format).	Rename	**write-mode**	*numeric* Available values are: • 0 - for overwrite • 1 - for append • 2 - for rename

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *HMM profile*

Name in Workflow File: in-hmm2

Slots:

Slot In GUI	Slot in Workflow File	Type
HMM profile	**hmm2-profile**	*hmm2-profile*
Location	**url**	*string*

HMMER3 Tools

- HMM3 Build Element
- HMM3 Search Element
- Read HMM3 Profile
- Write HMM3 Profile

HMM3 Build Element

Builds a HMM3 profile from a multiple sequence alignment. The HMM3 profile is a statistical model which captures position-specific information about how conserved each column of the alignment is, and which residues are likely.

Element type: hmm3-build

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Random seed	Random generator seed. 0 - means that one-time arbitrary seed will be used.	0	**seed**	*numeric*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input MSA*

Name in Workflow File: in-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*

And 1 *output port*:

Name in GUI: *HMM3 profile*

Name in Workflow File: out-hmm3

Slots:

Slot In GUI	Slot in Workflow File	Type
HMM profile	**hmm3-profile**	*hmm3-profile*

HMM3 Search Element

Searches each input sequence for significantly similar sequence matches to all specified HMM profiles. In case several profiles were supplied, searches with all profiles one by one and output united set of annotations for each sequence.

Element type: hmm3-search

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Result annotation	Name of the result annotations.	hmm_signal	**result-name**	*string*
Seed	Random generator seed. 0 - means that one-time arbitrary seed will be used.	0	**seed**	*numeric*
Filter by high E-value	E-value filtering can be used to exclude low-probability hits from the result.	1e-1	**seqs-num**	*numeric*
Filter by low score	Score based filtering is an alternative to E-value filtering to exclude low-probability hits from the result.	0.01	**score**	*numeric*

Input/Output Ports

The element has 2 *input port*. The first gets the input sequence:

Name in GUI: *Input sequence*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

The second input port gets the HMM profile:

Name in GUI: *HMM3 profile*

Name in Workflow File: in-hmm3

Slots:

Slot In GUI	Slot in Workflow File	Type
HMM profile	**hmm3-profile**	*hmm3-profile*

And 1 *output port*:

Name in GUI: *HMM3 annotations*

Name in Workflow File: out-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table*

Read HMM3 Profile

Reads HMM3 profiles from file(s). The files can be local or Internet URLs.

Element type: hmm3-read-profile

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Input files (required)	Semicolon-separated list of paths to the input files.		**url-in**	*string*

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *HMM3 profile*

Name in Workflow File: out-hmm3

Slots:

Slot In GUI	Slot in Workflow File	Type
HMM profile	**hmm3-profile**	*hmm3-profile*

Write HMM3 Profile

Saves all input HMM3 profiles to the specified locations.

Element type: hmm3-write-profile

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output file	Location of the output data file. If this attribute is set, the “Location” slot is not taken into account.		**url-out**	*string*
Existing file	If a target file already exists, you can specify how it should be handled: either overwritten, renamed or appended (if supported by file format). If Rename option is chosen existing file will be renamed.	Rename	**write-mode**	*numeric* Available values are: • 0 - for overwrite • 1 - for append • 2 - for rename

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *HMM3 profile*

Name in Workflow File: in-hmm3

Slots:

Slot In GUI	Slot in Workflow File	Type
HMM profile	**hmm3-profile**	*hmm3-profile*
Location	**url**	*string*

Multiple Sequence Alignment

- Align Profile to Profile with MUSCLE Element
- Align with ClustalO Element
- Align with ClustalW Element
- Align with Kalign Element
- Align with MAFFT Element
- Align with MUSCLE Element
- Align with T-Coffee Element
- Extract Consensus from Alignment as Sequence
- Extract Consensus from Alignment as Text
- In Silico PCR Element
- Join Sequences into Alignment Element
- Map to Reference Element
- Split Alignment into Sequences Element

Align Profile to Profile with MUSCLE Element

Aligns second profile to master profile with MUSCLE aligner.

Type: align-profile-to-profile

Input/Output Ports

The element has 1 *input port*:

Name in GUI: in-profiles

Name in Workflow File: in-profiles

Slots:

Slot In GUI	Slot in Workflow File	Type
Master profile	master-msa	*malignment*
Second profile	second-msa	*malignment*

And 1 *output port*:

Name in GUI: out-msa

Name in Workflow File: out-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	msa	*malignment*

Align with ClustalO Element

Aligns multiple sequence alignments (MSAs) supplied with ClustalO.

Element type: ClustalO

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Number of iterations	Number of (combined guide-tree/HMM) iterations.	1	**num-iterations**	*numeric*
Number of guidetree iterations	Maximum number guidetree iterations.	0	**max-guidetree-iterations**	*numeric*
Number of HMM iterations	Maximum number of HMM iterations.	0	**max-hmm-iterations**	*numeric*
Set auto options	Set options automatically (might overwrite some of your options).	False	**set-auto**	*boolean*
Tool path	Path to the ClustalO tool. The default path can be set in the UGENE application settings.	Default	**path**	*string*
Temporary directory	Directory to store temporary files.	Default	**temp-dir**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input MSA

Name in Workflow File: in-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*malignment*

And 1 *output port*:

Name in GUI: ClustalO result MSA

Name in Workflow File: out-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*malignment*

Align with ClustalW Element

Aligns multiple sequence alignments (MSAs) supplied with ClustalW.

ClustalW is a general-purpose multiple sequence alignment program for DNA or proteins. Visit http://www.clustal.org/ to learn more about it.

Clustal is used as an external tool from UGENE and it must be installed on your system. To learn more about the external tools, please, read the main UGENE User Manual.

Element type: clustalw

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Weight matrix	For proteins, it is a scoring table which describes the similarity of each amino acid to each other. For DNA it is the scores assigned to matches and mismatches.	default	**matrix**	*numeric* Available values are: • 0 - for IUB • 1 - for ClustalW • 2 - for BLOSUM • 3 - for PAM • 4 - for GONNET • 5 - for ID • -1 - for default matrix
End gaps	The penalty for closing a gap.	False	**close-gap-penalty**	*boolean*
Gap distance	The gap separation penalty. Tries to decrease the chances of gaps being too close to each other.	4.42	**gap-distance**	*numeric*
Gap extension penalty	The penalty for extending a gap.	8.52	**gap-ext-penalty**	*numeric*
Gap open penalty	The penalty for opening a gap.	53.90	**gap-open-penalty**	*numeric*
Hydrophilic gaps off	Hydrophilic gap penalties are used to increase the chances of a gap within a run (5 or more residues) of hydrophilic amino acids.	False	**no-hydrophilic-gaps**	*boolean*
Residue-specific gaps	Residue-specific penalties are amino specific gap penalties that reduce or increase the gap opening penalties at each position in the alignment.	False	**no-residue-specific-gaps**	*boolean*

off				
Iteration type	Alignment improvement iteration type.	None	**iteration-type**	*numeric* Available values are: • 0 - for None • 1 - for Tree • 2 - for Alignment
Number of iterations	The maximum number of iterations to perform.	3	**iterations-max-num**	*numeric*
Tool path (required)	Path to the ClustalW tool. The default path can be set in the UGENE Application Settings.	default	**path**	*string*
Temporary directory	Directory to store temporary files.	default	**temp-dir**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input MSA*

Name in Workflow File: in-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*

And 1 *output port*:

Name in GUI: *ClustalW result MSA*

Name in Workflow File: out-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*

Align with Kalign Element

Aligns multiple sequence alignments (MSAs) supplied with Kalign. Kalign is a fast and accurate multiple sequence alignment tool. The original version of the tool can be found onhttp://msa.sbc.su.se.

Element type: kalign

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Gap extension penalty	The penalty for extending a gap.	8.52	**gap-ext-penalty**	*numeric*
Gap open penalty	The penalty for opening/closing a gap. Half the value will be subtracted from the alignment score when opening, and half when closing a gap.	54.90	**gap-open-penalty**	*numeric*
Terminal gap penalty	The penalty to extend gaps from the N/C terminal of protein or 5'/3' terminal of nucleotide sequences.	4.42	**terminal-gap-penalty**	*numeric*
Bonus score	A bonus score that is added to each pair of aligned residues.	0.02	**bonus-score**	*numeric*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input MSA*

Name in Workflow File: in-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*

And 1 *output port*:

Name in GUI: *Kalign result MSA*

Name in Workflow File: out-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*

Align with MAFFT Element

Originally, MAFFT is a multiple sequence alignment program for unix-like operating systems. Currently, Windows version is also available.

MAFFT is used as an external tool from UGENE and it must be installed on your system. To learn more about the external tools, please, read main UGENE User Manual.

MAFFT is used as an external tool from UGENE and it must be installed on your system. To learn more about the external tools, please, read main UGENE User Manual.

Element type: mafft

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Offset	Works like gap extension penalty.	0	**gap-ext-penalty**	*numeric*
Gap open penalty	Gap open penalty.	1.53	**gap-open-penalty**	*numeric*
Max iteration	Maximum number of iterative refinement.	0	**iterations-max-num**	*numeric*
Tool path (default)	Path to the ClustalW tool. The default path can be set in the UGENE application settings.	default	**path**	*string*
Temporary directory	Directory to store temporary files.	default	**temp-dir**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input MSA*

Name in Workflow File: in-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*

And 1 *output port*:

Name in GUI: *Multiple sequence alignment*

Name in Workflow File: out-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*

Align with MUSCLE Element

MUSCLE is public domain multiple alignment software for protein and nucleotide sequences. MUSCLE stands for MUltiple Sequence Comparison by Log-Expectation.

Element type: muscle

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Mode	Selector of preset configurations, that give you the choice of optimizing accuracy, speed, or some compromise between the two. The default favors accuracy.	MUSCLE default	**mode**	*numeric* Availables values are: • 0 - for MUSCLE default • 1 - for Large alignment • 2 - for Refine only
Stable order	Do not rearrange aligned sequences (-stable switch of MUSCLE). Otherwise, MUSCLE re-arranges sequences so that similar sequences are adjacent in the output file. This makes the alignment easier to evaluate by eye.	True	**stable**	*boolean*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input MSA*

Name in Workflow File: in-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*

And 1 *output port*:

Name in GUI: *Multiple sequence alignment*

Name in Workflow File: out-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*

Align with T-Coffee Element

T-Coffee is a multiple sequence alignment package.

T-Coffee is used as an external tool from UGENE and it must be installed on your system. To learn more about the external tools, please, read main UGENE User Manual.

Element type: tcoffee

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Gap extension penalty	Gap Extension Penalty. Positive values give rewards to gaps and prevent the alignment of unrelated segments.	0	**gap-ext-penalty**	*numeric*
Gap open penalty	Gap open penalty. Must be negative, best matches get a score of 1000.	-50	**gap-open-penalty**	*numeric*
Max iteration	Number of iteration on the progressive alignment. 0 - no iteration, -1 - Nseq iterations.	0	**iterations-max-num**	*numeric*
Tool path (required)	Path to the ClustalW tool. The default path can be set in the UGENE Application Settings.	default	**path**	*string*
Temporary directory	Directory to store temporary files.	default	**temp-dir**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input MSA*

Name in Workflow File: in-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*

And 1 *output port*:

Name in GUI: *Multiple sequence alignment*

Name in Workflow File: out-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*

Extract Consensus from Alignment as Sequence

Extract the consensus sequence from the incoming multiple sequence alignment.

Element type: extract-msa-consensus-sequence

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Algorithm	The algorithm of consensus extracting.		**algorithm**	*string*
Threshold	The threshold of the algorithm.	100	**threshold**	*numeric*
Keep gaps	Set this parameter if the result consensus must keep the gaps.	True	**keep-gaps**	*boolean*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *in-msa*

Name in Workflow File: in-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*

And 1 *output port*:

Name in GUI: *out-sequence*

Name in Workflow File: out-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*seq*

Extract Consensus from Alignment as Text

Extract the consensus string from the incoming multiple sequence alignment.

Element type: extract-msa-consensus-string

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Algorithm	The algorithm of consensus extracting.		**algorithm**	*string*
Threshold	The threshold of the algorithm.	100	**threshold**	*numeric*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *in-msa*

Name in Workflow File: in-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*

And 1 *output port*:

Name in GUI: *out-sequence*

Name in Workflow File: out-text

Slots:

Slot In GUI	Slot in Workflow File	Type
Plain text	**text**	*string*

In Silico PCR Element

Simulates PCR for input sequences and primer pairs. Creates the table with the PCR statistics.

Element type: in-silico-pcr

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Primers URL	A URL to the input file with primer pairs.		**primers-url**	*string*

Report URL	A URL to the output file with the PCR report.		**report-url**	*string*
Mismatches	Number of allowed mismatches.	3	**mismatches**	*numeric*
Min perfect match	Number of bases that match exactly on 3' end of primers.	15	**perfect-match**	*numeric*
Max product size	Maximum size of amplified region.	5000	**max-product**	*numeric*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input sequence

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

And 1 *output port*:

Name in GUI: PCR product

Name in Workflow File: out

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table*
Sequence	**sequence**	*sequence*

Join Sequences into Alignment Element

Creates a multiple sequence alignment from sequences.

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input sequences*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

And 1 *output port*:

Name in GUI: *Result alignment*

Name in Workflow File: out-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*

Map to Reference Element

Align input sequences (e.g. Sanger reads) to the reference sequence.

Element type: align-to-reference

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Reference URL	A URL to the file with a reference sequence.		**reference**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input sequence

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	sequence	*sequence*

And 1 *output port*:

Name in GUI: Aligned data

Name in Workflow File: out

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	annotations	*ann_table*
MSA	msa	*malignment*
Sequence	sequence	*sequence*

Split Alignment into Sequences Element

Splits an input alignment into sequences.

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input alignment*

Name in Workflow File: in-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*

And 1 *output port*:

Name in GUI: *Output sequences*

Name in Workflow File:

Slots: out-sequence

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

NGS: Basic Functions

- CASAVA FASTQ Filter Element
- Cut Adapter Element
- Extract Consensus from Assembly Element

- Extract Coverage from Assembly Element
- FASTQ Merger Element
- FASTQ Quality Trimmer Element
- FastQC Quality Control Element
- Filter BAM/SAM Files Element
- Genome Coverage Element
- Improve Reads with Trimmomatic Element
- Merge BAM Files Element
- Remove Duplicates in BAM Files Element
- Slopbed Element
- Sort BAM Files Element

CASAVA FASTQ Filter Element

Reads in FASTQ file produced by CASAVA 1.8 contain 'N' or 'Y' as a part of an idetifier. 'Y' if a read if filtered, 'N' if the read if the read is not filtered. The workflow cleans up the filtered reads. For example: @HWI-ST880:181:D1WRUACXX:8:1102:4905:2125 1:N:0:TAAGGG CTTACATAACTACTGACCATGCTCTCTCTTGTCTGTCTCTTATACACATCT + 1114422223223242 32AAFFHIJJJJJJIHIIF111CGGFHIG???FGB @HWI-ST880:181:D1WRUACXX:8:1102:7303:2101 1:Y:0:TAAGGG TCCTTACTGTCTGAGCAATGGGATTCCATCTTTTACGATCTAGACATGGCT + 11++4222322.

Element type: CASAVAFilter

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	Select an output directory. Custom - specify the output directory in the 'Custom directory' parameter. Workflow - internal workflow directory. Input file - the directory of the input file.	Input file	**out-mode**	*numeric*
Custom directory	Specify the output directory.		**custom-dir**	*string*
Output file name	A name of an output file. If default of empty value is provided the output name is the name of the first file with additional extention.		**out-name**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input File

Name in Workflow File: in-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

And 1 *output port*:

Name in GUI: Output File

Name in Workflow File: out-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

Cut Adapter Element

Removes adapter sequences.

Element type: CutAdaptFastq

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	Select an output directory. Custom - specify the output directory in the 'Custom directory' parameter. Workflow - internal workflow directory. Input file - the directory of the input file.	Input file	**out-mode**	*string*
Output file name	A name of an output file. If default of empty value is provided the output name is the name of the first file with additional extention.		**out-name**	*string*
FASTA file with 3' adapters	A FASTA file with one or multiple sequences of adapter that were ligated to the 3' end. The adapter itself and anything that follows is trimmed. If the adapter sequence ends with the '$ character, the adapter is anchored to the end of the read and only found if it is a suffix of the read.		**adapters-url**	*string*
FASTA file with 5' adapters	A FASTA file with one or multiple sequences of adapters that were ligated to the 5' end. If the adapter sequence starts with the character '^', the adapter is 'anchored'. An anchored adapter must appear in its entirety at the 5' end of the read (it is a prefix of the read). A non-anchored adapter may appear partially at the 5' end, or it may occur within the read. If it is found within a read, the sequence preceding the adapter is also trimmed. In all cases, the adapter itself is trimmed.		**front-url**	*string*
FASTA file with 5' and 3' adapters	A FASTA file with one or multiple sequences of adapters that were ligated to the 5' end or 3' end.		**anywhere-url**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input File

Name in Workflow File: in-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

And 1 *output port*:

Name in GUI: Output File

Name in Workflow File: out-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

Extract Consensus from Assembly Element

Extract the consensus sequence from the incoming assembly.

Element type: extract-consensus

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Algorithm	The algorithm of consensus extracting.	Default	**algorithm**	*string*
Keep gaps	Set this parameter if the result consensus must keep the gaps.	True	**keep-gaps**	*boolean*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: in-assembly

Name in Workflow File: in-assembly

Slots:

Slot In GUI	Slot in Workflow File	Type
Assembly data	**assembly**	*assembly*

And 1 *outut port*:

Name in GUI: out-sequence

Name in Workflow File: out-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*string*

Extract Coverage from Assembly Element

Extract the coverage and bases count from the incoming assembly.

Element type: extract-assembly-coverage

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output file	Location of output data file. If this attribute is set, slot "Location" in port will not be used.	assembly_coverage.txt	**url-out**	*string*
Export	Data type to export.	coverage	**export-type**	*string*
Treshold	The minimum coverage value to export.	1	**threshold**	*numeric*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: in-assembly

Name in Workflow File: in-assembly

Slots:

Slot In GUI	Slot in Workflow File	Type
Assembly data	**assembly**	*assembly*

FASTQ Merger Element

Merges input sequences to one output file.

Element type: MergeFastq

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	Select an output directory. Custom - specify the output directory in the 'Custom directory' parameter. Workflow - internal workflow directory. Input file - the directory of the input file.		**out-mode**	*string*
Output file name	A name of an output file. If default of empty value is provided the output name is the name of the first file with additional extention.		**out-name**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input File

Name in Workflow File: in-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

The element has 1 *output port*:

Name in GUI: Output File

Name in Workflow File: out-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

FASTQ Quality Trimmer Element

The workflow scans each input sequence from the end to find the first position where the quality is greater or equal to the minimum quality threshold. Then it trims the sequence to that position. If a the whole sequence has quality less than the threshold or the length of the output sequence less than the minimum length threshold then the sequence is skipped.

Element type: QualityTrim

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	Select an output directory. Custom - specify the output directory in the 'Custom directory' parameter. Workflow - internal workflow directory. Input file - the directory of the input file.	Input file	**out-mode**	*numeric*
Custom directory	Specify the output directory.		**custom-dir**	*string*
Output file name	A name of an output file. If default of empty value is provided the output name is the name of the first file with additional extention.		**out-name**	*string*
Quality threshold	Quality threshold for trimming.	30	**qual-id**	*numeric*
Min Length	Too short reads are discarded by the filter.	0	**len-id**	*numeric*
Trim both ends	Trim the both ends of a read or not. Usually, you need to set True for Sanger sequencing and False for NGS	True	**both-ends**	*boolean*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input File

Name in Workflow File: in-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

And 1 *output port*:

Name in GUI: Output File

Name in Workflow File: out-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

FastQC Quality Control Element

Builds quality control reports.

Element type: fastqc

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output file	Specify the output file name.	Input file	**out-file**	*string*
List of adapters	Specifies a non-default file which contains the list of adapter sequences which will be explicity searched against the library. The file must contain sets of named adapters in the form name[tab]sequence. Lines prefixed with a hash will be ignored.		**adapter**	*string*
List of contaminants	Specifies a non-default file which contains the list of contaminants to screen overrepresented sequences against. The file must contain sets of named contaminants in the form name[tab]sequence. Lines prefixed with a hash will be ignored.		**contaminants**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Short reads

Name in Workflow File: in-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

Filter BAM/SAM Files Element

Filters BAM/SAM files using SAMTools view.

Element type: filter-bam

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	Select an output directory. Custom - specify the output directory in the 'Custom directory' parameter. Workflow - internal workflow directory. Input file - the directory of the input file.		**out-mode**	*numeric*
Custom directory	Custom output directory.		**custom-dir**	*string*
Output name	A name of an output BAM/SAM file. If default of empty value is provided the output name is the name of the first BAM/SAM file with .filtered extention.		**out-name**	*string*
Output format	Format of an output assembly file.	bam	**out-format**	*string*
Region	Regions to filter. For BAM output only. chr2 to output the whole chr2. chr2:1000 to output regions of chr 2 starting from 1000. chr2:1000-2000 to ouput regions of chr2 between 1000 and 2000 including the end point. To input multiple regions use the space seprator (e.g. chr1 chr2 chr3:1000-2000).		**region**	*string*
MAPQ	Minimum MAPQ quality score.	0	**mapq**	*numeric*

threshold				
Skip flag	Skip alignment with the selected items. Select the items in the combobox to configure bit flag. Do not select the items to avoid filtration by this parameter.		**flag**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: BAM/SAM File

Name in Workflow File: in-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**input-url**	*string*

And 1 *output port*:

Name in GUI: Filtered BAM/SAM files

Name in Workflow File: out-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**output-url**	*string*

Genome Coverage Element

Calculates genome coverage using bedtools genomecov.

Element type: genomecov

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	Select an output directory. Custom - specify the output directory in the 'Custom directory' parameter. Workflow - internal workflow directory. Input file - the directory of the input file.	Input file	**out-mode**	*numeric*
Custom directory	Specify the output directory.		**custom-dir**	*string*
Output file name	A name of an output file. If default of empty value is provided the output name is the name of the first file with additional extention.		**out-name**	*string*
Genome	In order to prevent the extension of intervals beyond chromosome boundaries, bedtools slop requires a genome file defining the length of each chromosome or contig (-g).	human. hg18	**genome**	*string*
Report mode	Histogram () - Compute a histogram of coverage. Per-base (0-based) (-dz) - Compute the depth of feature coverage for each base on each chromosome (0-based). Per-base (1-based) (-d) - Compute the depth of feature coverage for each base on each chromosome (1-based) BEDGRAPH (-bg) - Produces genome-wide coverage output in BEDGRAPH format. BEDGRAPH (including uncoveded) (-bga) - Produces genome-wide coverage output in BEDGRAPH format (including uncovered).	Histogram	**mode-id**	*numeric*
Split	Treat âsplitâ BAM or BED12 entries as distinct BED intervals when computing coverage. For BAM files, this uses the CIGAR âNâ and âDâ operations to infer the blocks for computing coverage. For BED12 files, this uses the BlockCount, BlockStarts, and BlockEnds fields (i.e., columns 10,11,12) (-split).	False	**split-id**	*boolean*
Strand	Calculate coverage of intervals from a specific strand. With BED files, requires at least 6 columns (strand is column 6) (-strand).	False	**strand-id**	*boolean*
5 prime	Calculate coverage of 5â positions (instead of entire interval) (-5).	False	**prime5-id**	*boolean*
3 prime	Calculate coverage of 3â positions (instead of entire interval) (-3).	False	**prime3-id**	*boolean*
Max	Combine all positions with a depth >= max into a single bin in the histogram (-max).	21474836	**max-id**	*numeric*

		47		
Scale	Scale the coverage by a constant factor.Each coverage value is multiplied by this factor before being reported. Useful for normalizing coverage by, e.g., reads per million (RPM). Default is 1.0; i.e., unscaled (-scale).	1.00000	**scale-id**	*numeric*
Trackline	Adds a UCSC/Genome-Browser track line definition in the first line of the output (-trackline).	False	**trackline-id**	*boolean*
Trackopts	Writes additional track line definition parameters in the first line (-trackopts).		**trackopts-id**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input File

Name in Workflow File: in-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

And 1 *output port*:

Name in GUI: Output File

Name in Workflow File: out-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

Improve Reads with Trimmomatic Element

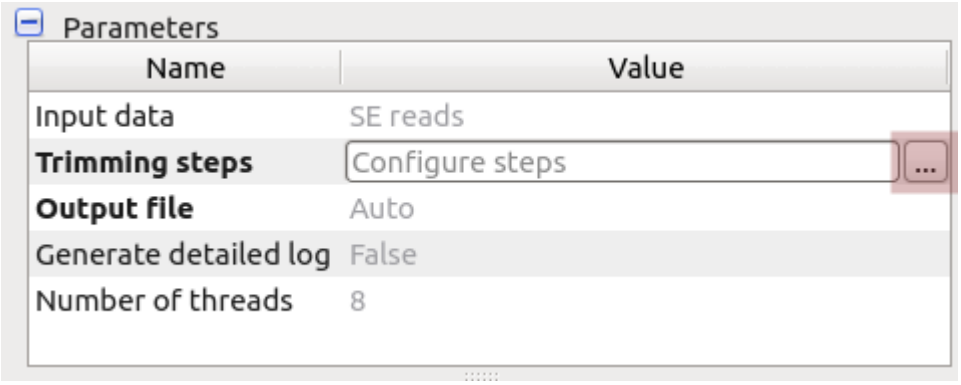
Trimmomatic is a fast, multithreaded command line tool that can be used to trim and crop Illumina (FASTQ) data as well as to remove adapters.

Element type: trimmomatic

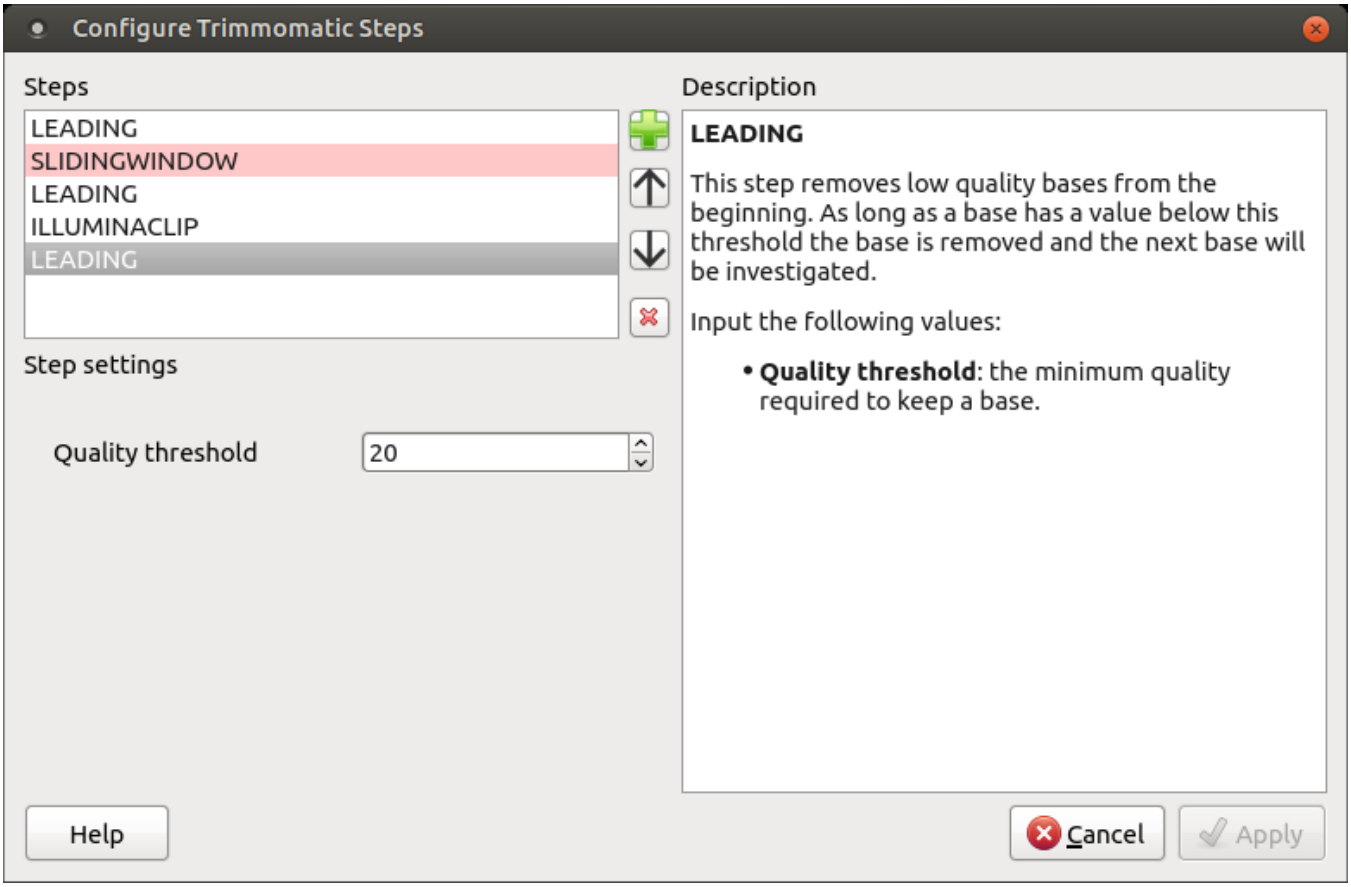
Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Input data	Set the type of the input reads: single-end (SE) or paired-end (PE). One or two slots of the input port are used depending on the value of the parameter. Pass URL(s) to data to these slots. Note that the paired-end mode will use additional information contained in paired reads to better find an adapter or PCR primer fragments introduced by the library preparation process.	SE reads	**input-data**	*string*
Trimming steps	Configure trimming steps that should be performed by Trimmomatic.	configure steps	**trimming-steps**	*string*
Output file	Specify the output file name.	auto	**output-url**	*string*
Generate detailed log	Select "True"to generate a file with log of all read trimmings, indicating the following details (-trimlog): • thread name • the surviving sequence length • the location of the first surviving base, aka. the amount trimmed from the start • the location of the last surviving base in the original read • the amount trimmed from the end	False	**generate-log**	*bool*
Number of threads	Use multiple threads (-threads).	8	**threads**	*numeric*

To configure trimming steps use the following button:



The following dialog will appear:



Click the *Add new step* button and select a step. The following options are available:

- ILLUMINACLIP: Cutadapterandotherillumina-specific sequences from the read.
- SLIDINGWINDOW: Perform a sliding window trimming, cutting once the average quality within the window falls below a threshold.
- LEADING: Cut bases off the start of a read, if below a threshold quality.
- TRAILING: Cut bases off the end of a read, if below a threshold quality.
- CROP: Cut the read to a specified length.
- HEADCROP: Cut the specified number of bases from the start of the read.
- MINLEN: Drop the read if it is below a specified length.
- AVGQUAL: Drop the read if the average quality is below the specified level.
- TOPHRED33: Convert quality scores to Phred-33.
- TOPHRED64: Convert quality scores to Phred-64.

Each step has the own parameters:

AVGQUAL

This step drops a read if the average quality is below the specified level.

Input the following values:

- Quality threshold: the minimum average quality required to keep a read.

CROP

This step removes bases regardless of quality from the end of thread, so that the readhas maximally the specified length after this step has been performed. Steps performed after CROP might of course further shorten the read.

Input the following values:

- Length: the number of bases to keep, from the start of the read.

HEADCROP

This step removes the specified number of bases, regardless of quality, from the beginning of the read.

Input the following values:

- Length: the number of bases to remove from the start of the read.

ILLUMINACLIP

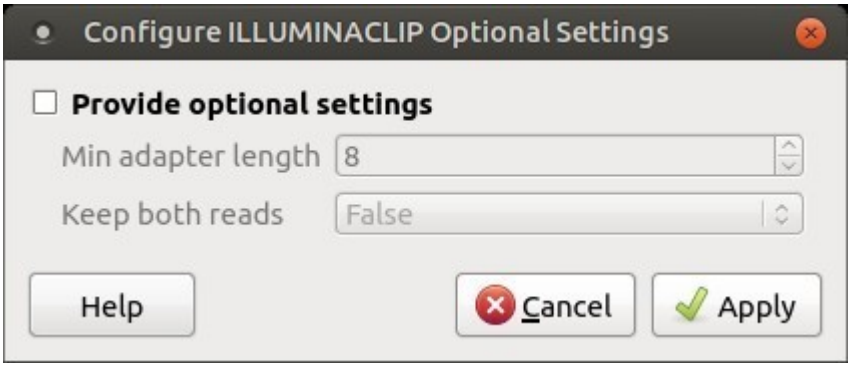
This step is used to find and remove Illumina adapters.

Trimmomatic first compares short sections of an adapter and a read. If they match enough, the entire alignment between the read and adapter is scored. For paired-end reads, the "palindrome" approach is also used to improve the result. See Trimmomatic manual for details.

Input the following values:

- Adapter sequences: a FASTA file with the adapter sequences. Files for TruSeq2 (GAII machines), TruSeq3 (HiSeq and MiSeq machines) and Nextera kits for SE and PE reads are now available by default. The naming of the various sequences within the specified file determines how they are used.
- Seed mismatches: the maximum mismatch count in short sections which will still allow a full match to be performed.
- Simple clip threshold: a threshold for simple alignment mode. Values between 7 and 15 are recommended. A perfect match of a 12 base sequence will score just over 7, while 25 bases are needed to score 15.
- Palindrome clip threshold: a threshold for palindrome alignment mode. For palindromic matches, a longer alignment is possible. Therefore the threshold can be in the range of 30. Even though this threshold is very high (requiring a match of almost 50 bases) Trimmomatic is still able to identify very, very short adapter fragments.

There are also two optional parameters for palindrome mode: Min adapter length and Keep both reads. Use the following dialog. To call the dialog press the *Optional* button.



LEADING

This step removes low-quality bases from the beginning. As long as a base has a value below this threshold the base is removed and the next base will be investigated.

Input the following values:

- Quality threshold: the minimum quality required to keep a base.

MAXINFO

This step performs an adaptive quality trim, balancing the benefits of retaining longer reads against the costs of retaining bases with errors. See Trimmomatic manual for details.

Input the following values:

- Target length: the read length which is likely to allow the location of the read within the target sequence. Extremely short reads, which can be placed into many different locations, provide little value. Typically, the length would be in the order of 40 bases, however, the value also depends on the size and complexity of the target sequence.

- Strictness: the balance between preserving as much read length as possible vs. removal of incorrect bases. A low value of this parameter (0.8) favours read correctness.

MINLEN

This step removes reads that fall below the specified minimum length. If required, it should normally be after all other processing steps. Reads removed by this step will be counted and included in the "dropped reads" count.

Input the following values:

- Length: the minimum length of reads to be kept.

SLIDINGWINDOW

This step performs a sliding window trimming, cutting once the average quality within the window falls below a threshold. By considering multiple bases, a single poor quality base will not cause the removal of high-quality data later in the read.

Input the following values:

- Window size: the number of bases to an average across.
- Quality threshold: the average quality required.

TOPHRED33

This step (re)encodes the quality part of the FASTQ file to base 33.

TOPHRED64

This step (re)encodes the quality part of the FASTQ file to base 64.

TRAILING

This step removes low-quality bases from the end. As long as a base has a value below this threshold the base is removed and the next base (i.e. the preceding one) will be investigated. This approach can be used removing the special Illumina " low-quality segment" regions (which are marked with a quality score of 2), but SLIDINGWINDOW or MAXINFO are recommended instead.

Input the following values:

- Quality threshold: the minimum quality required to keep a base.

To remove a step use the *Remove selected step* button. The pink highlighting means the required parameter has not been set.

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input FASTQ file(s)

Name in Workflow File: in

Slots:

Slot In GUI	Slot in Workflow File	Type
Input FASTQ URL	**reads-url1**	*string*

And 1 *output port*:

Name in GUI: Improved FASTQ file(s)

Name in Workflow File: out-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Output FASTQ URL	**reads-url1**	*string*

Merge BAM Files Element

Merge BAM files using SAMTools merge.

Element type: merge-bam

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	Select an output directory. Custom - specify the output directory in the 'Custom directory' parameter. Workflow - internal workflow directory. Input file - the directory of the input file.		**out-mode**	*numeric*
Custom directory	Custom output directory.		**custom-dir**	*string*
Output BAM name	A name of an output BAM file. If default of empty value is provided the output name is the name of the first BAM file with .merged.bam extention.		**out-name**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: BAM File

Name in Workflow File: in-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**input-url**	*string*

And 1 *output port*:

Name in GUI: Merged BAM files

Name in Workflow File: out-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**output-url**	*string*

Remove Duplicates in BAM Files Element

Remove PCR duplicates of BAM files using SAMTools rmdup.

Element type: rmdup-bam

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	Select an output directory. Custom - specify the output directory in the 'Custom directory' parameter. Workflow - internal workflow directory. Input file - the directory of the input file.	Input file	**out-mode**	*numeric*
Output BAM name	A name of an output file. If default of empty value is provided the output name is the name of the first file with additional extention.		**out-name**	*string*
Remove for single-end reads	Remove duplicate for single-end reads. By default, the command works for paired-end reads only (-s).	False	**remove-single-end**	*boolean*
Treat as single-end	Treat paired-end reads and single-end reads (-S).	False	**treat_reads**	*boolean*

Input/Output Ports

The element has 1 *input port*:
Name in GUI: Input File

Name in Workflow File: in-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

And 1 *output port*:

Name in GUI: Output File

Name in Workflow File: out-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

Slopbed Element

Increases the size of each feature in files using bedtools slop.

Element type: slopbed

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	Select an output directory. Custom - specify the output directory in the 'Custom directory' parameter. Workflow - internal workflow directory. Input file - the directory of the input file.	Input file	**out-mode**	*numeric*
Custom directory	Specify the output directory.		**custom-dir**	*string*
Output file name	A name of an output file. If default of empty value is provided the output name is the name of the first file with additional extention.		**out-name**	*string*
Genome	In order to prevent the extension of intervals beyond chromosome boundaries, bedtools slop requires a genome file defining the length of each chromosome or contig (-g).	human. hg18	**genome-id**	*string*
Each direction increase	Increase the BED/GFF/VCF entry by the same number base pairs in each direction. If this parameter is used -l and -l are ignored. Enter 0 to disable (-b).	0	**b-id**	*numeric*
Substract from start	The number of base pairs to subtract from the start coordinate. Enter 0 to disable (-l).	0	**l-id**	*numeric*
Add to end	The number of base pairs to add to the end coordinate. Enter 0 to disable (-r).	0	**r-id**	*numeric*
Strand-based	Define -l and -r based on strand. For example. if used, -l 500 for a negative-stranded feature, it will add 500 bp to the end coordinate (-s).	False	**s-id**	*boolean*
As fraction	Define -l and -r as a fraction of the featureâs length. E.g. if used on a 1000bp feature, -l 0.50, will add 500 bp âupstreamâ (-pct).	False	**pct-id**	*boolean*
Print header	Print the header from the input file prior to results (-header).	False	**header-id**	*boolean*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input File

Name in Workflow File: in-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

And 1 *output port*:

Name in GUI: Output File

Name in Workflow File: out-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

Sort BAM Files Element

Sort BAM Files using SAMTools Sort.

Element type: Sort-bam

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	Select an output directory. Custom - specify the output directory in the 'Custom directory' parameter. Workflow - internal workflow directory. Input file - the directory of the input file.	Input file	**out-mode**	*numeric*
Custom directory	Specify the output directory.		**custom-dir**	*string*
Output BAM name	A name of an output file. If default of empty value is provided the output name is the name of the first file with additional extention.		**out-name**	*string*
Build index	Build index for the sorted file with SAMTools index.	human. hg18	**index**	*boolean*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: BAM File

Name in Workflow File: in-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

And 1 *output port*:

Name in GUI: Sorted BAM File

Name in Workflow File: out-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

NGS: ChIP-Seq Analysis

- Annotate Peaks with peak2gene Element
- Build Conservation Plot Element
- Collect Motifs with SeqPos Element
- Conduct GO Element
- Create CEAS Report Element
- Find Peaks with MACS Element

Annotate Peaks with peak2gene Element

Gets refGenes near the ChIP regions identified by a peak-caller.

Element type: peak2gene-id

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Genome file	Select a genome file (sqlite3 file) to search refGenes. (--genome).	hg19	**genome**	*string*
Output file	Select which type of genes need to output. up for genes upstream to peak summit, down for genes downstream to peak summit, all for both up and down. (--op).	all	**outpos**	*string*
Official gene symbols	Output official gene symbol instead of refseq name. (--symbol).	False	**symbol**	*boolean*

Distance	Set a number which unit is base. It will get the refGenes in n bases from peak center. (--distance).	3000	**distance**	*numeric*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Peak2gene data

Name in Workflow File: in-data

Slots:

Slot In GUI	Slot in Workflow File	Type
Treatment features	**_treat-ann**	*ann-table-list*

And 1 *output port*:

Name in GUI: Peak2gene output data

Name in Workflow File: out-data

Slots:

Slot In GUI	Slot in Workflow File	Type
Gene regions	**gene-annotation**	*ann-table-list*
Peak regions	**peak-annotation**	*ann-table-list*

Build Conservation Plot Element

Plots the PhastCons score profiles.

Element type: conservation_plot-id

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output file	File to store phastcons results (BMP).		**output-file**	*string*
Title	Title of the figure (--title).	Average Phastcons around the Center of Sites	**title**	*string*
Label	Label of data in the figure (--bed-label).	Conservation_at_peak_summits	**label**	*string*
Assembly version	The directory to store phastcons scores (--phasdb).	hg19	**assembly_version**	*string*
Window width	Window width centered at middle of regions (-w).	1000	**windos_s**	*numeric*
Height	Height of plot (--height).	1000	**height**	*numeric*
Width	Width of plot (--width).	1000	**width**	*numeric*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: conservation_plot data

Name in Workflow File: in-data

Slots:

Slot In GUI	Slot in Workflow File	Type
Input regions	**cp_treat-ann**	*ann-table-list*

Collect Motifs with SeqPos Element

Finds motifs enriched in a set of regions.

Element type: seqpos-id

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	The directory to store seqpos results.		**output-dir**	*string*
Genome assembly version	UCSC database version (GENOME).	hg19	**assembly**	*string*
Output file name	Name of the output file which stores new motifs found during a de novo search (-n).	Default	**out_name**	*string*
De novo motifs	Run de novo motif search (-d).	False	**de_novo**	*boolean*
Motif database	Known motif collections. (-m). Warning: computation time increases with selecting additional databases. It is recommended to use cistrome.xml. It is a comprehensive collection of motifs from the other databases with similar motifs deleted.	cistrome.xml	**motif_db**	*string*
Region width	Width of the region to be scanned for motifs; depends on a resolution of assay (-w).	600	**reg_width**	*numeric*
Pvalue cutoff	Pvalue cutoff for the motif significance (-p).	0.001	**p_val**	*numeric*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: SeqPos data

Name in Workflow File: in-data

Slots:

Slot In GUI	Slot in Workflow File	Type
Input regions	**cp_treat-ann**	*ann-table-list*

Conduct GO Element

Given a list of genes, using Bioconductor (GO, GOstats) and DAVID at NIH.

Element type: conduct-go-id

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	The directory to store Conduct GO results.		**output-dir**	*string*
Title	Title is used to name the output files - so make it meaningful.	Default	**title**	*string*
Gene Universe	Select a gene universe.	hgu133a.db	**gene-universe**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Conduct GO data

Name in Workflow File: in-data

Slots:

Slot In GUI	Slot in Workflow File	Type
Target genes	**in-ann**	*ann-table-list*

Create CEAS Report Element

It provides summary statistics on ChIP enrichment in important genomic regions such as individual chromosomes, promoters, gene bodies or exons, and infers the genes most likely to be regulated by the binding factor under study.

Element type: ceas-report

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output report file	Path to the report output file. Result for CEAS analysis.		**image-file**	*string*
Output annotations file	Name of tab-delimited output text file, containing a row of annotations for every RefSeq gene. (file is not generated if no peak location data is supplied).		**anns-file**	*string*
Gene annotations table	Path to gene annotation table (e.g. a refGene table in sqlite3 db format (--gt).	hg19	**anns-table**	*string*
Span size	Span from TSS and TTS in the gene-centered annotation (base pairs). ChIP regions within this range from TSS and TTS are considered when calculating the coverage rates in promoter and downstream (--span).	3000	**span**	*numeric*
Wiggle profiling resolution	Wiggle profiling resolution. WARNING: Value smaller than the wig interval (resolution) may cause aliasing error. (--pf-res).	50	**profiling-resolution**	*numeric*
Promoter /downstream interval	Promoter/downstream intervals for ChIP region annotation are three values or a single value can be given. If a single value is given, it will be segmented into three equal fractions (e.g. 3000 is equivalent to 1000,2000,3000) (--rel-dist).	3000	**promoter-sizes**	*numeric*
BiPromoter ranges	Bidirectional-promoter sizes for ChIP region annotation. It's two values or a single value can be given. If a single value is given, it will be segmented into two equal fractions (e.g. 5000 is equivalent to 2500,5000) (--bisizes).	5000	**promoter-bisizes**	*string*
Relative distance	Relative distance to TSS/TTS in WIGGLE file profiling (--rel-dist).	3000	**relative-distance**	*string*
Gene group files	Gene groups of particular interest in wig profiling. Each gene group file must have gene names in the 1st column. The file names are separated by commas (--gn-groups).		**group-files**	*string*
Gene group names	Set this parameter empty for using default values. The names of the gene groups from "Gene group files" parameter. These names appear in the legends of the wig profiling plots. Values range: comma-separated list of strings. Default value: 'Group 1, Group 2,...Group n' (--gn-group-names).		**group-names**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: CEAS data

Name in Workflow File: in-data

Slots:

Slot In GUI	Slot in Workflow File	Type
Enrichment signal	**enrichment-signal**	*ann-table-list*
Peak regions	**peak-regions**	*string*

Find Peaks with MACS Element

Performs peak calling for ChIP-Seq data.

Element type: macs-id

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	Directory to save MACS output files.		**output-dir**	**output-dir**
Name	The name string of the experiment. MACS will use this string NAME to create output files like 'NAME_peaks.xls', 'NAME_negative_peaks.xls', 'NAME_peaks.bed', 'NAME_summits.bed', 'NAME_model.r' and so on. So please avoid any confliction between these filenames and your existing files (--name).		**file-names**	**file-names**
Wiggle	If this flag is on, MACS will store the fragment pileup in wiggle format for the whole genome data instead of for	hg19	**wiggle-**	**wiggle-**

output	every chromosomes (--wig) (--single-profile).		**output**	**output**
Wiggle space	By default, the resolution for saving wiggle files is 10 bps,i.e., MACS will save the raw tag count every 10 bps. You can change it along with '--wig' option (--space).	3000	**wiggle-space**	**wiggle-space**
Genome size (Mbp)	Homo sapience - 2700 Mbp Mus musculus - 1870 Mbp Caenorhabditis elegans - 90 Mbp Drosophila melanogaster - 120 Mbp It's the mappable genome size or effective genome size which is defined as the genome size which can be sequenced. Because of the repetitive features on the chromosomes, the actual mappable genome size will be smaller than the original size, about 90% or 70% of the genome size (--gsize).	50	**genome-size**	**genome-size**
P-value	P-value cutoff. Default is 0.00001, for looser results, try 0.001 instead (--pvalue).	3000	**p-value**	**p-value**
Tag size (optional)	Length of reads. Determined from first 10 reads if not specified (input 0) (--tsize).	5000	**tag-size**	**tag-size**
Keep duplicates	It controls the MACS behavior towards duplicate tags at the exact same location -- the same coordination and the same strand. The default auto option makes MACS calculate the maximum tags at the exact same location based on binomal distribution using 1e-5 as pvalue cutoff; and the all option keeps every tags. If an integer is given, at most this number of tags will be kept at the same location (--keep-dup).	3000	**keep-duplicates**	**keep-duplicates**
Use model	Whether or not to use MACS paired peaks model (--nomodel).		**use-model**	**use-model**
Model fold	Select the regions within MFOLD range of high-confidence enrichment ratio against. Model fold is available when Use model is true, which is the foldchange to chose paired peaks to build paired peaks model. Users need to set a lower(smaller) and upper(larger) number for fold change so that MACS will only use the peaks within these foldchange range to build model (--mfold).		**model-fold**	**model-fold**
Shift size	An arbitrary shift value used as a half of the fragment size when model is not built. Shift size is available when Use model is false, which will represent the HALF of the fragment size of your sample. If your sonication and size selection size is 300 bps, after you trim out nearly 100 bps adapters, the fragment size is about 200 bps, so you can specify 100 here (--shiftsize).		**shift-size**	**shift-size**
Band width	The band width which is used to scan the genome for model building. You can set this parameter as the sonication fragment size expected from wet experiment. Used only while building the shifting model (--bw).		**band-width**	**band-width**
Use lambda	Whether to use local lambda model which can use the local bias at peak regions to throw out false positives (--nolambda).		**use-lambda**	**use-lambda**
Small nearby region	The small nearby region in basepairs to calculate dynamic lambda. This is used to capture the bias near the peak summit region. Invalid if there is no control data (--slocal).		**small-nearby**	**small-nearby**
Large nearby region	The large nearby region in basepairs to calculate dynamic lambda. This is used to capture the surround bias (--llocal).		**large-nearby**	**large-nearby**
Auto bimodal	Whether turn on the auto pair model process.If set, when MACS failed to build paired model, it will use the nomodelsettings, the Shift size parameter to shift and extend each tags (--on-auto).		**auto_bimodal**	**auto_bimodal**
Scale to large	When set, scale the small sample up to the bigger sample.By default, the bigger dataset will be scaled down towards the smaller dataset,which will lead to smaller p/qvalues and more specific results.Keep in mind that scaling down will bring down background noise more (--to-large).		**scale_large**	**scale_large**

Input/Output Ports

The element has 1 *input port*:

Name in GUI: MACS data

Name in Workflow File: in-data

Slots:

Slot In GUI	Slot in Workflow File	Type
Treatment features	**_treatment-ann**	*ann-table-list*
Control features	**control-ann**	*ann-table-list*

And 1 *output port*:

Name in GUI: MACS output data

Name in Workflow File: out-data

Slots:

Slot In GUI	Slot in Workflow File	Type
Peak regions	**peak-regions**	*ann-table-*

		list
Peak summits	**peak-summits**	*ann-table-list*
Treatment fragments pileup	**wiggle-treat**	*string*

NGS: Map/Assemble Reads

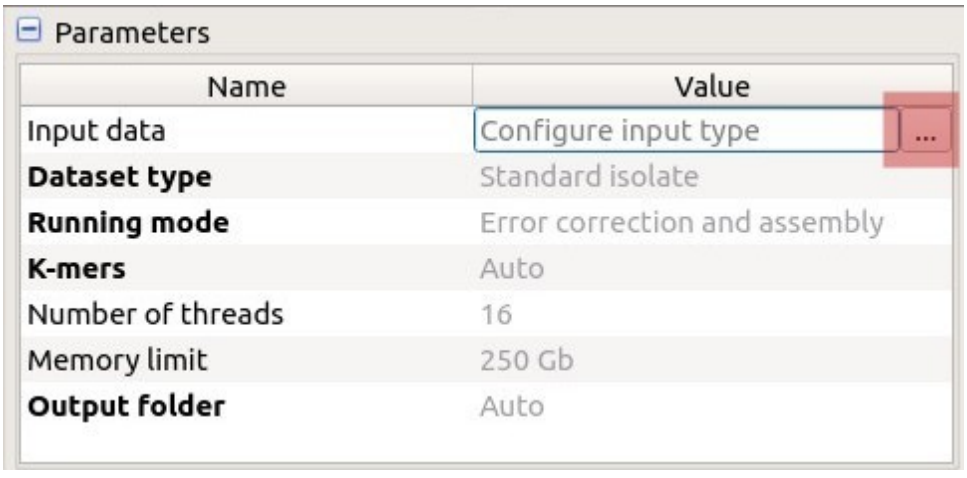
- Assemble Reads with SPAdes Element
- Map Reads with Bowtie Element
- Map Reads with Bowtie2 Element
- Map Reads with BWA Element
- Map Reads with BWA-MEM Element
- Map Reads with UGENE Genome Aligner Element
- Map RNA-Seq Reads with TopHat Element

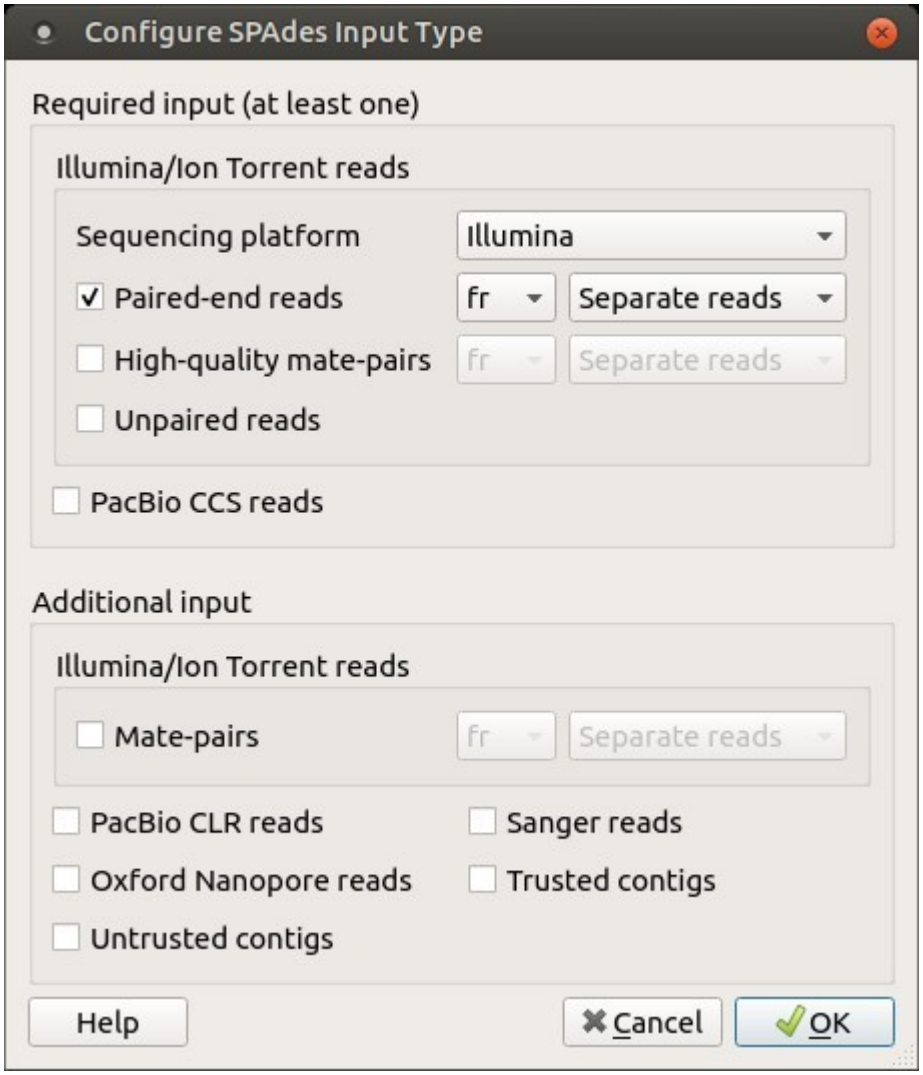
Assemble Reads with SPAdes Element

Performers assembly of input short reads.

Element type: spades-id

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Input data	Select the type of input for SPAdes. URL(s) to the input files of the selected type(s) should be provided to the corresponding port(s) of the workflow element. At least one library of the following types is required: • Illumina paired-end/high-quality mate-pairs/unpaired reads • IonTorrent paired-end/high-quality mate-pairs/unpaired reads • PacBio CCS reads (at least 5 reads coverage is recommended) It is strongly suggested to provide multiple paired-end and mate-pair libraries according to their insert size (from smallest to longest). Additionally, one may input Oxford Nanopore reads, Sanger reads, contigs generated by other assembler(s), etc. Note that Illumina and IonTorrent libraries should not be assembled together. All other types of input data are compatible. It is also possible to set up reads orientation (forward-reverse (fr), reverse-forward (rf), forward-forward (ff)) and specify whether paired reads are separate or interlaced. Illumina, IonTorrent or PacBio CCS reads should be provided in FASTQ format. Illumina or PacBio read may also be provided in FASTA format. Error correction should be skipped in this case (see the "Running mode" parameter). Sanger, Oxford Nanopore, and PacBio CLR reads can be provided in both formats since SPAdes does not run error correction for these types of data. To configure input data use the following button:  		**output-dir**	*string*

	The following dialog will appear: 			
Dataset type	Input dataset type.	Multi Cell	**dataset-type**	*string*
Running mode	Running mode.	Error correction and assembly	**running-mode**	*string*
K-mers	k-mersizes (-k).	auto	**k-mer**	*numeri*
Number of threads	Number of threads (-t).	16	**threads**	*numeri*
Memory limit (Gb)	Memory limit (-m).	250	**memlimit**	*numeri*
Output folder	Folder to save Spades output files.	Auto		

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Spades data

Name in Workflow File: in-data

Slots:

Slot In GUI	Slot in Workflow File	Type
URL of a file with right pair reads	url	*string*
URL of a file with reads	url	*string*

And 1 *output port*:

Name in GUI: SPAdes output data

Name in Workflow File: out-data

Slots:

Slot In GUI	Slot in Workflow File	Type
Scaffolds URL	url	*string*
Contig URL	url	*string*

Map Reads with Bowtie Element

Performs alignment of short reads with Bowtie.

Element type: align-reads-with-bowtie

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	Directory to save Bowtie output files.		**output-dir**	*string*
Reference genome	Path to an indexed reference genome.		**reference**	*string*
Output file name	Base name of the output file. 'out.sam' by default.	out.sam	**outname**	*string*
Library	Is this library mate-paired?	single-end	**library**	*string*
Mode	When the -n option is specified (which is the default), bowtie determines which alignments are valid according to the following policy, which is similar to Maq's default policy. In -v mode, alignments may have no more than V mismatches, where V may be a number from 0 through 3 set using the -v option. Quality values are ignored. The -v option is mutually exclusive with the -n option.	-n mode	**mismatches_type**	*string*
Mismatches number	Mismatches number.	2	**mismatches_number**	*numeric*
Mismatches number	Maximum permitted total of quality values at all mismatched read positions throughout the entire alignment, not just in the seed. The default is 70. Like Maq, bowtie rounds quality values to the nearest 10 and saturates at 30; rounding can be disabled with --nomaqround.	70	**maqerr**	*numeric*
Seed length	The seed length; i.e., the number of bases on the high-quality end of the read to which the -n ceiling applies. The lowest permitted setting is 5 and the default is 28. bowtie is faster for larger values of -l.	28	**seedLen**	*numeric*
Maximum of backtracks	The maximum insert size for valid paired-end alignments. E.g. if -X 100 is specified and a paired-end alignment consists of two 20-bp alignments in the proper orientation with a 60-bp gap between them, that alignment is considered valid (as long as -I is also satisfied). A 61-bp gap would not be valid in that case. If trimming options -3 or -5 are also used, the -X constraint is applied with respect to the untrimmed mates, not the trimmed mates. Default: 250.	800	**maxbts**	*numeric*
Best hits	The number of megabytes of memory a given thread is given to store path descriptors in --best mode. Best-first search must keep track of many paths at once to ensure it is always extending the path with the lowest cumulative cost. Bowtie tries to minimize the memory impact of the descriptors, but they can still grow very large in some cases. If you receive an error message saying that chunk memory has been exhausted in --best mode, try adjusting this parameter up to dedicate more memory to the descriptors. Default: 64.	64	**chunkmbs**	*numeric*
Seed	Use as the seed for pseudo-random number generator.	0	**seed**	*numeric*
Colorspace	When -C is specified, read sequences are treated as colors. Colors may be encoded either as numbers (0=blue, 1=green, 2=orange, 3=red) or as characters A/C/G/T (A=blue, C=green, G=orange, T=red).	False	**colorspace**	*boolean*
No Maq rounding	Maq accepts quality values in the Phred quality scale, but internally rounds values to the nearest 10, with a maximum of 30. By default, bowtie also rounds this way. --nomaqround prevents this rounding in bowtie.	False	**nomaqround**	*boolean*
No forward orientation	If --nofw is specified, bowtie will not attempt to align against the forward reference strand.	False	**nofw**	*boolean*
No reverse-complement orientation	If --norc is specified, bowtie will not attempt to align against the reverse-complement reference strand.	False	**norc**	*boolean*

Try as hard	Try as hard as possible to find valid alignments when they exist, including paired-end alignments. This is equivalent to specifying very high values for the --maxbts and --pairtries options. This mode is generally much slower than the default settings, but can be useful for certain problems. This mode is slower when (a) the reference is very repetitive, (b) the reads are low quality, or (c) not many reads have valid alignments.	False	**tryhard**	*boolean*
Best alignments	Make Bowtie guarantee that reported singleton alignments are best in terms of stratum (i.e. number of mismatches, or mismatches in the seed in the case of -n mode) and in terms of the quality values at the mismatched position(s). bowtie is somewhat slower when --best is specified.	False	**best**	*boolean*
All alignment	Report all valid alignments per read or pair.	False	**all**	*boolean*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Bowtie data

Name in Workflow File: in-data

Slots:

Slot In GUI	Slot in Workflow File	Type
URL of a file with mate reads	**readsurl**	*string*
URL of a file with reads	**readspairedurl**	*string*

And 1 *output port*:

Name in GUI: Bowtie output data

Name in Workflow File: out-data

Slots:

Slot In GUI	Slot in Workflow File	Type
Assembly URL	**assembly-out**	*string*

Map Reads with Bowtie2 Element

Performs alignment of short reads with Bowtie2.

Element type: align-reads-with-bowtie2

Parameters in GUI

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	Directory to save Bowtie2 output files.		**output-dir**	*string*
Reference genome	Path to an indexed reference genome.		**reference**	*string*
Output file name	Base name of the output file. 'out.sam' by default.	out.sam	**outname**	*string*
Library	Is this library mate-paired?	single-end	**library**	*string*
Mode	When the -n option is specified (which is the default), bowtie determines which alignments are valid according to the following policy, which is similar to Maq's default policy. In -v mode, alignments may have no more than V mismatches, where V may be a number from 0 through 3 set using the -v option. Quality values are ignored. The -v option is mutually exclusive with the -n option.	--end-to-end	**mode**	*string*
Number of mismatches	Sets the number of mismatches to allowed in a seed alignment. Can be set to 0 or 1. Setting this higher makes alignment slower (often much slower) but increases sensitivity.	0	**mismatches_number**	*numeric*
Seed length (--L)	Sets the length of the seed substrings to align. Smaller values make alignment slower but more senstive.	20	**seed_len**	*numeric*
Add columns to allow gaps (--dpad)	"Pads" dynamic programming problems by the specified number of columns on either side to allow gaps.	15	**dpad**	*numeric*
Disallow gaps (--gbar)	Disallow gaps within a specified number of positions of the beginning or end of the read.	4	**gbar**	*numeric*

Seed (--seed)	Use as the seed for pseudo-random number generator.	0	**seed**	*numeric*
Threads	Launch specified number of parallel search threads. Threads will run on separate processors/cores and synchronize when parsing reads and outputting alignments. Searching for alignments is highly parallel, and speedup is close to linear.	1	**threads**	*numeric*
No unpaired alignments (--no-mixed)	If Bowtie2 cannot find a paired-end alignment for a pair, by default it will go on to look for unpaired alignments for the constituent mates. This is called "mixed mode." To disable mixed mode, set this option. Bowtie2 runs a little faster in the mixed mode, but will only consider the alignment status of pairs per se, not individual mates.	False	**nomixed**	*boolean*
No discordant alignments (--no-discordant)	By default, Bowtie2 looks for discordant alignments if it cannot find any concordant alignments. A discordant alignment is an alignment where both mates align uniquely, but that does not satisfy the paired-end constraints. This option disables that behavior.	False	**nodiscordant**	*boolean*
No forward orientation (--nofw)	If --nofw is specified, bowtie will not attempt to align against the forward reference strand.	False	**nofw**	*boolean*
No reverse-complement orientation (--norc)	If --norc is specified, bowtie will not attempt to align against the reverse-complement reference strand.	False	**norc**	*boolean*
No overlapping mates (--no-overlap)	If one mate alignment overlaps the other at all, consider that to be non-concordant. Default: mates can overlap in a concordant alignment.	False	**nooverlap**	*boolean*
No mates containing one another (--no-contain)	If one mate alignment contains the other, consider that to be non-concordant. Default: a mate can contain the other in a concordant alignment.	False	**nocontain**	*boolean*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Bowtie2 data

Name in Workflow File: in-data

Slots:

Slot In GUI	Slot in Workflow File	Type
URL of a file with mate reads	**readsurl**	*string*
URL of a file with reads	**readspairedurl**	*string*

And 1 *output port*:

Name in GUI: Bowtie2 output data

Name in Workflow File: out-data

Slots:

Slot In GUI	Slot in Workflow File	Type
Assembly URL	**assembly-out**	*string*

Map Reads with BWA Element

Performs alignment of short reads with BWA.

Element type: align-reads-with-bwa

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	Directory to save BWA-MEM output files.		**output-dir**	*string*
Reference genome	Path to an indexed reference genome.		**reference**	*string*
Output file name	Base name of the output file. 'out.sam' by default.	out.sam	**outname**	*string*
Library	Is this library mate-paired?	single-end	**library**	*string*

Use missing prob	Use missing prob instead maximum edit distance.	True	**use-miss-prob**	*boolean*
Missing prob	Missing prob (-n).	0.04	**missing-prob**	*numeric*
Seed length	Seed length (-l).	32	**seed-length**	*numeric*
Max gap opens	Max gap opens (-o).	1	**max-gap**	*numeric*
Index algorithm	Index algorithm (-a).	is	**index-alg**	*string*
Best hits	Best hits (-R).	30	**best-hits**	*numeric*
Long-scaled gap penalty for long deletions	Long-scaled gap penalty for long deletions (-L).	False	**scaled-gap**	*boolean*
Non iterative mode	Non iterative mode (-N).	False	**non-iterative**	*boolean*
Enable long gaps	Enable long gaps.	True	**enable-long-gaps**	*boolean*
Max gap extensions	Max gap extensions (-e).	0	**gap-extensions**	*numeric*
Indel offset	Indel offset (-i).	5	**indel-offset**	*numeric*
Max long deletions extensions	Max long deletions extensions(-d).	10	**long-deletions**	*numeric*
Barcode length	Barcode length (-B).	0	**barcode-length**	*numeric*
Max queue entries	Max queue entries (-m).	2000000	**max-queue**	*numeric*
Threads	Threads (-t).	4	**num-threads**	*numeric*
Max seed differencies	Max seed differencies (-k).	2	**max-seed**	*numeric*
Mismatch penalty	Mismatch penalty (-M).	3	**mistmatch-penalty**	*numeric*
Gap open penalty	Gap open penalty (-O).	11	**gap-open-penalty**	*numeric*
Gap extension penalty	Gap extension penalty; a gap of size k cost (-E).	4	**gap-ext-penalty**	*numeric*
Quality threshold	Quolity threshold (-q).	0	**quality-threshold**	*numeric*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: BWA data

Name in Workflow File: in-data

Slots:

Slot In GUI	Slot in Workflow File	Type
URL of a file with mate reads	**readsurl**	*string*
URL of a file with reads	**readspairedurl**	*string*

And 1 *output port*:

Name in GUI: BWA output data

Name in Workflow File: out-data

Slots:

Slot In GUI	Slot in Workflow File	Type
Assembly URL	**assembly-out**	*string*

Map Reads with BWA-MEM Element

Performs alignment of short reads with BWA-MEM.

Element type: bwamem-id

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	Directory to save BWA-MEM output files.		**output-dir**	*string*

Reference genome	Path to an indexed reference genome.		**reference**	*string*
Output file name	Base name of the output file. 'out.sam' by default.	out.sam	**outname**	*string*
Library	Is this library mate-paired?	single-end	**library**	*string*
Number of threads	Number of threads (-t).	1	**threads**	*numeric*
Min seed length	Path to an indexed reference genome (-k).	19	**min-seed**	*numeric*
Index algorithm	Index algorithm (-a).	autodetect	**index-alg**	*string*
Band width	Bandwidth for banded alignment (-w).	100	**band-width**	*numeric*
Dropoff	Off-diagonal X-dropoff (-d).	100	**dropoff**	*numeric*
Internal seed length	Look for internal seeds inside a seed longer than {-k} (-r).	1.50000	**seed-lookup**	*numeric*
Skip seed threshold	Skip seeds with more than INT occurrences (-c).	10000	**seed-threshold**	*numeric*
Drop chain threshold	Drop chains shorter than FLOAT fraction of the longest overlapping chain (-D).	0.5	**drop-chains**	*numeric*
Rounds of made rescues	Perform at most INT rounds of mate rescues for each read (-m).	100	**mate-rescue**	*numeric*
Skip mate rescue	Skip mate rescue (-S).	False	**skip-mate-rescues**	*boolean*
Skip pairing	Skip pairing; mate rescue performed unless -S also in use (-P).	False	**skip-pairing**	*boolean*
Matching score	Score for a sequence match (-A).	1	**match-score**	*numeric*
Mismatch penalty	Penalty for a mismatch (-B).	4	**mistmatch-penalty**	*numeric*
Gap open penalty	Gap open penalty (-O).	6	**gap-open-penalty**	*numeric*
Gap extension penalty	Gap extension penalty; a gap of size k cost {-O} (-E).	1	**gap-ext-penalty**	*numeric*
Penalty for clipping	Penalty for clipping (-L).	5	**clipping-penalty**	*numeric*
Penalty unpaired	Penalty for an unpaired read pair (-U).	17	**inpaired-panalty**	*numeric*
Score threshold	Minimum score to output (-T).	30	**score-threshold**	*numeric*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: BWA data

Name in Workflow File: in-data

Slots:

Slot In GUI	Slot in Workflow File	Type
URL of a file with mate reads	**readsurl**	*string*
URL of a file with reads	**readspairedurl**	*string*

And 1 *output port*:

Name in GUI: BWA-MEM output data

Name in Workflow File: out-data

Slots:

Slot In GUI	Slot in Workflow File	Type
Assembly URL	**assembly-out**	*string*

Map Reads with UGENE Genome Aligner Element

Unique UGENE algorithm for aligning short reads to reference genome.

Element type: genome-aligner

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type

Output file name	Base name of the output file. 'out.sam' by default.	out.sam	**outname**	*string*
Reference genome	Path to an indexed reference genome.		**reference**	*string*
Is absolute mismatches values?	**true** - absolute mismatches mode is used **false** - percentage mismatches mode is used You can choose absolute or percentage mismatches values mode.	True	**if-absolute-mismatches-value**	*boolean*
Absolute mismatches	Number of mismatches allowed while aligning reads.	0	**absolute-mismatches**	*numeric*
Align reverse complement reads	Set this option to align both direct and reverse complement reads.	False	**reverse**	*boolean*
Use "best"-mode	Report only the best alignment for each read (in terms of mismatches).	True	**best**	*boolean*
Omit reads with qualities lower than	Omit reads with qualities lower than the specified value. Reads that have no qualities are not omitted. Set "0" to switch off this option.	0	**quality-threshold**	*numeric*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Genome aligner data

Name in Workflow File: in-data

Slots:

Slot In GUI	Slot in Workflow File	Type
URL of a file with mate reads	**readsurl**	*string*
URL of a file with reads	**readspairedurl**	*string*

And 1 *output port*:

Name in GUI: Genome aligner output data

Name in Workflow File: out-data

Slots:

Slot In GUI	Slot in Workflow File	Type
Assembly URL	**assembly-out**	*string*

Map RNA-Seq Reads with TopHat Element

TopHat is a program for mapping RNA-Seq reads to a long reference sequence. It uses Bowtie or Bowtie2 to map the reads and then analyzes the mapping results to identify splice junctions between exons.

Provide URL(s) to FASTA or FASTQ file(s) with NGS RNA-Seq reads to the input port of the element, set up the reference sequence in the parameters. The result is saved to the specified BAM file, URL to the file is passed to the output port. Several UCSC BED tracks are also produced: junctions, insertions, and deletions.

Element type: tophat

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Reference input type	Select "Sequence" to input a reference genome as a sequence file. Note that any sequence file format, supported by UGENE, is allowed (FASTA, GenBank, etc.). The index will be generated automatically in this case. Select "Index" to input already generated index files, specific for the tool.	Index	**reference-input-type**	*string*
Bowtie index folder	The folder with the Bowtie index for the reference sequence.		**bowtie-index-dir**	*string*
Bowtie index basename	The basename of the Bowtie index for the reference sequence.		**bowtie-index-basename**	*string*

Output folder	The base name of the output folder. It could be modified with a suffix.		**out-dir**	
Mate inner distance	The expected (mean) inner distance between mate pairs.	50	**mate-inner-distance**	*numeric*
Mate standard deviation	The standard deviation for the distribution on inner distances between mate pairs.	20	**mate-standard-deviation**	*numeric*
Library type	Specifies RNA-Seq protocol.	fr-unstranded	**library-type**	*numeric*
No novel junctions	Only look for reads across junctions indicated in the supplied GFF or junctions file. This parameter is ignored if R aw junctions or Known transcript file is not set.	False	**no-novel-junctions**	*boolean*
Raw junctions	The list of raw junctions.		**raw-junctions**	*string*
Known transcript file	A set of gene model annotations and/or known transcripts.		**known-transcript**	*string*
Max multihits	Instructs TopHat to allow up to this many alignments to the reference for a given read, and suppresses all alignments for reads with more than this many alignments.	20	**max-multihits**	*numeric*
Segment length	Each read is cut up into segments, each at least this long. These segments are mapped independently.	25	**segment-length**	*numeric*
Fusion search	Turn on fusion mapping.	False	**fusion-search**	*boolean*
Transcriptome only	Only align the reads to the transcriptome and report only those mappings as genomic mappings.	False	**transcriptome-only**	*boolean*
Transcriptome max hits	Maximum number of mappings allowed for a read, when aligned to the transcriptome (any reads found with more than this number of mappings will be discarded).	60	**transcriptome-max-hits**	*numeric*
Prefilter multihits	When mapping reads on the transcriptome, some repetitive or low complexity reads that would be discarded in the context of the genome may appear to align to the transcript sequences and thus may end up reported as mapped to those genes only. This option directs TopHat to first align the reads to the whole genome in order to determine and exclude such multi-mapped reads (according to the value of the Max multihits option).	False	**prefilter-multihits**	*boolean*
Min anchor length	The anchor length. TopHat will report junctions spanned by reads with at least this many bases on each side of the junction. Note that individual spliced alignments may span a junction with fewer than this many bases on one side. However, every junction involved in spliced alignments is supported by at least one read with this many bases on each side.	8	**min-anchor-length**	*numeric*
Splice mismatches	The maximum number of mismatches that may appear in the anchor region of a spliced alignment.	0	**splice-mismatches**	*numeric*
Read mismatches	Final read alignments having more than these many mismatches are discarded.	2	**read-mismatches**	*numeric*
Segment mismatches	Read segments are mapped independently, allowing up to this many mismatches in each segment alignment.	2	**segment-mismatches**	*numeric*
Solexa 1.3 quals	As of the Illumina GA pipeline version 1.3, quality scores are encoded in Phred-scaled base-64. Use this option for FASTQ files from pipeline 1.3 or later.	False	**solexa-1-3-quals**	*boolean*
Bowtie version	Specifies which Bowtie version should be used.	Bowtie2	**bowtie-version**	*numeric*
Bowtie -n mode	TopHat uses -v in Bowtie for initial read mapping (the default), but with this option, -n is used instead. Read segments are always mapped using -v option.	Use -v mode	**bowtie-n-mode**	*numeric*
Bowtie tool path	The path to the Bowtie external tool.	default	**bowtie-tool-path**	*string*
SAMtools tool path	The path to the SAMtools tool. Note that the tool is available in the UGENE External Tool Package.	default	**samtools-tool-path**	*string*
TopHat tool path	The path to the TopHat external tool in UGENE.	default	**path**	*string*
Temporary folder	The directory for temporary files.	default	**temp-dir**	*string*
Samples map	The map which divides all input datasets into samples. Every sample has the unique name.			

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input reads

Name in Workflow File: in-assembly

Slots:

Slot In GUI	Slot in Workflow File	Type
Dataset name	**dataset**	*string*
Input reads	**first.in**	*assembly*
Input reads url	**in-url**	*string*
Input paired reads url	**paired-url**	*string*
Input paired reads	**second.in**	*assembly*

And 1 *output port*:

Name in GUI: TopHat output

Name in Workflow File: out-assembly

Slots:

Slot In GUI	Slot in Workflow File	Type
Accepted hits	**accepted.hits**	*assembly*
Accepted hits url	**hits-url**	*string*

NGS: Metagenomics Classification

- Build CLARK Database
- Build DIAMOND Database Element
- Build Kraken Database Element
- Classification Report Element
- Classify Sequences with CLARK Element
- Classify Sequences with DIAMOND Element
- Classify Sequences with Kraken Element
- Classify Sequences with MetaPhlAn2 Element
- Ensemble Classification Data Element
- Filter by Classification Element
- Improve Classification with WEVOTE Element

Build CLARK Database

Build a CLARK database from a set of reference sequences ("targets"). NCBI taxonomy data are used to map the accession number found in each reference sequence to its taxonomy ID.

Element type: clark-build

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Database	A folder that should be used to store the database files.		**database**	*string*
Genomic library	Genomes that should be used to build the database ("targets"). The genomes should be specified in FASTA format. There should be one FASTA file per reference sequence. A sequence header must contain an accession number (i.e., >accession.number ... or >gi|number|ref|accession.number| ...).		**taxonomy**	*url-datasets*
Taxonomy rank	Set the taxonomy rank for the database. CLARK classifies metagenomic samples by using only one taxonomy rank. So as a general rule, consider first the genus or species rank, then if a high proportion of reads cannot be classified, reset your targets definition at a higher taxonomy rank (e.g., family or phylum).	Species	**taxonomy-rank**	*number*

Input/Output Ports

The element has 1 *output port*:

Name in GUI: Output CLARK database

Name in Workflow File: out

Slots:

SlotInGUI	Slot in Workflow File	Type
Output URL	**url**	*string*

Build DIAMOND Database Element

Build a DIAMOND formatted database from a FASTA input file.

Element type: diamond-build

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Database	A name of the binary DIAMOND database file that should be created.		**database**	*string*
Genomic library	Genomes that should be used to build the database.		**genomic-library**	*url-datasets*

Input/Output Ports

The element has 1 *output port*:

Name in GUI: Output DIAMOND database

Name in Workflow File: out

Slots:

SlotInGUI	Slot in Workflow File	Type
Output URL	**url**	*string*

Build Kraken Database Element

Build a Kraken database from a genomic library or shrink a Kraken database.

Element type: kraken-build

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Mode	Select "Build" to create a new database from a genomic library (--build). Select "Shrink" to shrink an existing database to have only specified number of k-mers (--shrink).	Build	**mode**	*string*
Database	Name of the output Kraken database (corresponds to --db that is used with --build, and to --new-db that is used with --shrink).		**database**	*string*
Genomic library	Genomes that should be used to build the database. The genomes should be specified in FASTA format. The sequence IDs must contain either a GI number or a taxonomy ID.		**genomic-library**	*url-datasets*
K-mer length	K-mer length in bp (--kmer-len).	31	**k-mer-length**	*number*
Minimizer length	Minimizer length in bp (--minimizer-len). The minimizers serve to keep k-mers that are adjacent in query sequences close to each other in the database, which allows Kraken to exploit the CPU cache. Changing the value of the parameter can significantly affect the speed of Kraken, and neither increasing nor decreasing of the value will guarantee faster or slower speed.	15	**minimizer-length**	*number*
Maximum database size	By default, a full database build is done. To shrink the database before the full build, input the size of the database in Mb (this corresponds to the --max-db-size parameter, but Mb is used instead of Gb). The size is specified together for the database and the index.	No limit	**maximum-database-size**	*number*
Clean	Remove unneeded files from a built database to reduce the disk usage (--clean).	True	**clean**	*bool*

Work on disk	Performs most operations on disk rather than in RAM (this will slow down build in most cases).	False	**work-on-disk**	*bool*
Jellyfiah hash size	The "kraken-build" tool uses the "jellyfish" tool. This parameter specifies the hash size for Jellyfish. Supply a smaller hash size to Jellyfish, if you encounter problems with allocating enough memory during the build process (--jellyfish-hash-size). By default, the parameter is not used.	Skip	**jellyfish-hash-size**	*number*
Number of threads	Use multiple threads (--threads).	8	**threads**	*number*

Input/Output Ports

The element has 1 *output port*:

Name in GUI: Output Kraken database

Name in Workflow File: out

Slots:

SlotInGUI	Slot in Workflow File	Type
Output URL	**url**	*string*

Classification Report Element

Based on the input taxonomy classification data the element generates a detailed report and saves it in a tab-delimited text format.

Element type: classification-report

Parameters

Parameter	Description	Defaultvalue	Parameter in Workflow File	Type
Output file	Specify the output text file name.		**output-url**	*string*
All taxa	By default, taxa with no sequences (reads or contigs) assigned are not included into the output. This option specifies to include all taxa. This may be useful when an output from several samples is compared. Set "Sort by" to "Tax ID" in this case.	False	**all-taxa**	*bool*
Sort by	It is possible to sort rows in the output file in two ways: • by the number of reads, covered by the clade rooted at the taxon(i.e. "clade_num" for this taxID) • by taxIDs The second option may be useful when an output from different samples is compared.	Tax ID	**sort-by**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input taxonomy data

Name in Workflow File: in

Slots:

SlotInGUI	Slot in Workflow File	Type
Taxonomy classification data	**tax-data**	*tax-classification*

Classify Sequences with CLARK Element

CLARK (CLAssifier based on Reduced K-mers) is a tool for supervised sequence classification based on discriminative k-mers.

UGENE provides the GUI for CLARK and CLARK-l variants of the CLARK framework for solving the problem of the assignment of metagenomic reads to known genomes.

Element type: clark-classify

Parameters

Parameter	Description	Defaultvalue	Parameter in Workflow	Type

			File	
Input data	To classify single-end (SE) reads or contigs, received by reads de novo assembly, set this parameter to "SE reads or contigs". To classify paired-end (PE) reads, set the value to "PE reads".	SE reads or contigs	**sequencing-reads**	*string*
Classification tool	Use CLARK-l on workstations with limited memory (i.e., "l" for light), this software tool provides precise classification on small metagenomes. It works with a sparse or "light" database (up to 4 GB of RAM) while still performing ultra accurate and fast results.	CLARK-l	**tool-variant**	*number*
Database	A path to the folder with the CLARK database files (-D). It is assumed that "targets.txt" file is located in this folder (the file is passed to the "classify_metagenome.sh" script from the CLARK package via parameter -T).		**database**	*string*
Minimum k-mer frequency	Minimum of k-mer frequency/occurrence for the discriminative k-mers (-t). For example, for 1 (or, 2), the program will discard any discriminative k-mer that appear only once (or, less than twice).	0	**k-min-freq**	*number*
Mode	Set the mode of the execution (-m): • "Full" to get detailed results, confidence scores and other statistics. • "Default" to get results summary and perform best trade-off between classification speed, accuracy and RAM usage. • "Express" to get results summary with the highest speed possible.	Default	**mode**	*bool*
Gap	"Gap" or number of non-overlapping k-mers to pass when creating the database (-). Increase the value if it is required to reduce the RAM usage. Note that this will degrade the sensitivity.	4	**gap**	*number*
Load database into memory	Request the loading of database file by memory mapped-file (--ldm). This option accelerates the loading time but it will require an additional amount of RAM significant. This option also allows to load the database in multithreaded-task (see also the "Number of threads" parameter).	False	**preload**	*bool*
Number of threads	Use multiple threads for the classification and, with the "Load database into memory" option enabled, for the loading of the database into RAM (-n).	8	**threads**	*number*
Output file	Specify the output file name.	auto	**output-url**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input sequences:

URL(s) to FASTQ or FASTA file(s) should be provided. In casethe of SE reads or contigs use the "Input URL 1" slot only.

In case of PE reads input "left" reads to "Input URL 1", "right" reads to "Input URL 2". See also the "Input data" parameter of the element.

Name in Workflow File: in

Slots:

SlotInGUI	Slot in Workflow File	Type
Input URL 1	**url**	*string*

The element has 1 *output port*:

Name in GUI: CLARK Classification:

A map of sequence names with the associated taxonomy IDs, classified by CLARK.

Name in Workflow File: out

Slots:

SlotInGUI	Slot in Workflow File	Type
Taxonomy classification data	tax-data	*tax-classification*

Classify Sequences with DIAMOND Element

In general, DIAMOND is a sequence aligner for protein and translated DNA searches similar to the NCBI BLAST software tools. However, it provides a speedup of BLAST ranging up to x20,000. Using this workflow element one can use DIAMOND for taxonomic classification of short DNA reads and longer sequences such as contigs. The lowest common ancestor (LCA) algorithm is used for the classification.

Element type: diamond-classify

Parameters

Parameter	Description	Defaultvalue	Parameter in Workflow File	Type
Database	Input a binary DIAMOND database file.		**database**	*string*
Genetic code	Genetic code used for translation of query sequences (--query-gencode).	The standard genetic code	**genetic-code**	*number*
Sensitive mode	The sensitive modes (--sensitive, --more-sensitive) are generally recommended for aligning longer sequences. The default mode is mainly designed for short read alignment, i.e. finding significant matches of >50 bits on 30-40aa fragments.	Default	**sensitive-mode**	*string*
Top alignments percentage	DIAMOND uses the lowest common ancestor (LCA) algorithm for taxonomy classification of the input sequences. This parameter specifies what alignments should be taken into account during the calculations (--top). For example, the default value "10" means to take top 10% of the best hits (i.e. sort all query/subject-alignments by a score, take top 10% of the alignments with the best score, calculate the lowest common ancestor for them).	10%	**top-alignments-percentage**	*number*
Frameshift	Penalty for frameshift in DNA-vs-protein alignments. Values around 15 are reasonable for this parameter. Enabling this feature will have the aligner tolerate missing bases in DNA sequences and is most recommended for long, error-prone sequences like MinION reads.	Skipped	**frame-shift**	*number*
Expected value	Maximum expected value to report an alignment (--evalue/-e).	0.0010	**e-value**	*number*
Matrix	Scoring matrix (--matrix).	BLOSUM62	**matrix**	*string*
Gap open penalty	Gap open penalty (--gapopen).	Default	**gap-open**	*number*
Gap extension penalty	Gap extension penalty (--gapextend).	Default	**gap-extend**	*number*
Block size	Block size in billions of sequence letters to be processed at a time (--block-size). This is the main parameter for controlling the program's memory usage. Bigger numbers will increase the use of memory and temporary disk space, but also improve performance. The program can be expected to use roughly six times this number of memory (in GB).	0.5	**block-size**	*number*
Index chunks	The number of chunks for processing the seed index (--index-chunks). This option can be additionally used to tune the performance. It is recommended to set this to 1 on a high memory server, which will increase performance and memory usage, but not the usage of temporary disk space.	4	**index-chunks**	*number*
Number of threads	Number of CPU threads (--treads).	8	**threads**	*number*
Output file	Specify the output file name. The output file is a tab-delimited file with the following fields: * Query ID * NCBI taxonomy ID (0 if unclassified) * E-value of the best alignment with a known taxonomy ID found for the query (0 if unclassified)	auto	**output-url**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input sequences:

URL(s) to FASTQ or FASTA file(s) should be provided.

The input files may contain single-end reads, contigs, or "left" reads in case of the paired-end sequencing (see "Input data" parameter of the element).

Name in Workflow File: in

Slots:

SlotInGUI	Slot in Workflow File	Type
Input URL	**url**	*string*

The element has 1 *output port*:

Name in GUI: DIAMOND Classification:

A list of sequence names with the associated taxonomy IDs, classified by DIAMOND.

Name in Workflow File: out

Slots:

SlotInGUI	Slot in Workflow File	Type
Taxonomy classification data	tax-data	*tax-classification*

Classify Sequences with Kraken Element

Kraken is a taxonomic sequence classifier that assigns taxonomic labels to short DNA reads. It does this by examining the k-mers within a read and querying a database with those.

Element type: kraken-classify

Parameters

Parameter	Description	Defaultvalue	Parameter in Workflow File	Type
Input data	To classify single-end (SE) reads or contigs, received by reads de novo assembly, set this parameter to "SE reads or contigs". To classify paired-end (PE) reads, set the value to "PE reads". One or two slots of the input port are used depending on the value of the parameter. Pass URL(s) to data to these slots. The input files should be in FASTA or FASTQ formats.	SE reads or contigs	**input-data**	*string*
Database	A path to the folder with the Kraken database files.		**database**	*string*
Quick operation	Stop classification of an input read after the certain number of hits. The value can be specified in the "Minimum number of hits" parameter.	False	**quick-operation**	*bool*
Load database into memory	Load the Kraken database into RAM (--preload). This can be useful to improve the speed. The database size should be less than the RAM size. The other option to improve the speed is to store the database on ramdisk. Set this parameter to "False" in this case.	True	**preload**	*bool*
Number of threads	Use multiple threads (--threads).	8	**threads**	*number*
Output file	Specify the output file name.	auto	**output-url**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input sequences:

URL(s) to FASTQ or FASTA file(s) should be provided. In case of SE reads or contigs use the "Input URL 1" slot only.

In case of PE reads input "left" reads to "Input URL 1", "right" reads to "Input URL 2". See also the "Input data" parameter of the element.

Name in Workflow File: in

Slots:

SlotInGUI	Slot in Workflow File	Type
Input URL	**url**	*string*

The element has 1 *output port*:

Name in GUI: Kraken Classification:

A map of sequence names with the associated taxonomy IDs, classified by Kraken.

Name in Workflow File: out

Slots:

SlotInGUI	Slot in Workflow File	Type
Taxonomy classification data	tax-data	*tax-classification*

Classify Sequences with MetaPhlAn2 Element

MetaPhlAn2 (METAgenomic PHyLogenetic ANalysis) is a tool for profiling the composition of microbial communities (bacteria, archaea, eukaryotes, and viruses) from whole-metagenome shotgun sequencing data.

Element type: metaphlan2-classify

Parameters

Parameter	Description	Defaultvalue	Parameter in Workflow File	Type
Input data	To classify single-end (SE) reads or contigs, received by reads de novo assembly, set this parameter to "SE reads or contigs". To classify paired-end (PE) reads, set the value to "PE reads".	SE reads or contigs	input-data	*string*
Input file format	Set type of an input file (--input-type). Each input file will usually contain a lot of sequences that should be classified.	FASTA	input-format	*string*
Database	A path to a folder with MetaPhlAn2 database: BowTie2 index files, built from reference genomes, and *.pkl file (--mpa-pkl, --bowtie2db). By default, "mpa_v20_m200" database is provided (if it has been downloaded). The database was built on ~1M unique clade-specific marker genes identified from ~17,000 reference genomes (~13,500 bacterial and archaeal, ~3,500 viral, and ~110 eukaryotic).		database	*string*
Number of threads	The number of CPUs to use for parallelizing the mapping (--nproc).	8	threads	*number*
Analysis type	Specify the type of analysis to perform: • Relative abundance - profiling of metagenomes in terms of relative abundances (corresponds to "-t rel_ab") • Relative abundance with reads statistics - profiling of metagenomes in terms of relative abundances and estimate the number of reads coming from each clade ("-t rel_ab_w_read_stats") • Reads mapping - mapping from reads to clades, the output contains reads that hit a marker only ("-t reads_map") • Clade profiles - normalized marker counts for clades with at least a non - null marker("-t clade_profiles") • Marker abundance table - normalized marker counts: only when > 0.0 and optionally normalized by metagenome size ("-t marker_ab_table"), see also "Normalize by metagenome size" parameter • Marker presence table - list of markers present in the sample ("-t marker_pres_table"), see also "Presence threshold" parameter	Relative abundance	analysis-type	*string*
Tax level	The taxonomic level for the relative abundance output: all, kingdoms (Bacteria and Archaea) only, phyla only, etc. (--tax_lev).	All	tax-level	*string*
Bowtie2 output file	The file for saving the output of BowTie2 (--bowtie2out). In the case of PE reads one file is created per each pair of files.	Auto	bowtie2-output-url	*string*
Output file	MetaPhlAn2 output depends on the "Analysis type" parameter. By default, it is a tab-delimited file with the predicted taxon relative abundances.	Auto	output-url	*string*
Normalize by metagenome size	The parameter is present only when "Analysis type" is equal to "Marker abundance table". It is a combo box with values "Skip" (default) and "Normalize". If "Normalize" is selected, the total number of reads in the original metagenome is taken into account for normalization: UGENE calculates the number of reads in an input FASTA/FASTQ file and passes "--nreads" parameter to MetaPhlAn2.		normalize-by-size	*boolean*
Presence threshold	The parameter is present only when "Analysis type" is equal to the "Marker presence table". It is an INT value >= 0. The default value is 1. Specify a threshold for calling a marker.		presence-threshold	*number*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input sequences:

URL(s) to FASTQ or FASTA file(s) should be provided. In the case of SE reads or contigs use the "Input URL 1" slot only. In case of PE reads input "left" reads to "Input URL 1", "right" reads to "Input URL 2".See also the "Input data" parameter of the element

Name in Workflow File: in

Slots:

SlotInGUI	Slot in Workflow File	Type
Input URL	**url**	*string*

Ensemble Classification Data Element

The element ensembles data, produced by classification tools (Kraken, CLARK, DIAMOND), into a single file in CSV format. This file can be used as input for the WEVOTE classifier.

Element type: ensemble-classification

Parameters

Parameter	Description	Defaultvalue	Parameter in Workflow File	Type
Number of tools	Specify the number of classification tools. The corresponding data should be provided using the input ports.	2	number-tools	*string*
Output file	Specify the output file. The classification data are stored in CSV format with the following columns: 1. a sequence name 2. taxID from the first tool 3. taxID from the second tool 4. optionally, taxID from the third tool	ensemble.cvs	out-file	*string*

Input/Output Ports

The element has 3 identical *input ports*:

Name in GUI: Input taxonomy data:

An input slot for taxonomy classification data.

Name in Workflow File: tax-data1, tax-data2, tax-data3

Slots:

SlotInGUI	Slot in Workflow File	Type
Input tax data 1	**tax-data**	*tax-classification*
Input tax data 2	**tax-data**	*tax-classification*
Input tax data 3	**tax-data**	*tax-classification*

The element has 1 *output port*:

Name in GUI: Ensembled classification:

URL to the CSV file with ensembled classification data.

Name in Workflow File: out

Slots:

SlotInGUI	Slot in Workflow File	Type
Output URL	url	*string*

Filter by Classification Element

The filter takes files with NGS reads or contigs, classified by one of the tools: Kraken, CLARK, DIAMOND, WEVOTE.

For each input file, it outputs a file with unspecific sequences (i.e. sequences not classified by the tools, taxID = 0) and/or one or several files with sequences that belong to the specific taxonomic group(s).

Element type: classification-filter

Parameters

Parameter	Description	Defaultvalue	Parameter in Workflow File	Type
Input data	To filter single-end (SE) reads or contigs, received by reads de novo assembly, set this parameter to "SE reads or contigs". Use the "Input URL 1" slot of the input port. To filter paired-end (PE) reads, set the value to "PE reads". Use the ""Input URL 1" and "Input URL 2" slots of the input port to input the NGS reads data. Also, input the classification data, received from Kraken, CLARK, or DIAMOND, to the "Taxonomy classification data" input slot. Either one or two slots of the output port are used depending on the input data.	SE reads or contigs	sequencing-reads	*string*
Save unspecific sequences	Select "True" to put all unspecific input sequences (i. e. sequences with tax ID = 0) into a separate file. Select "False" to skip unspecific sequences. At least one specific taxon should be selected in the "Save sequences with taxID" parameter in this case.	True	save-unspecific-sequences	*bool*
Save sequences with taxID	Select a taxID to put all sequences that belong to this taxonomic group (i. e. the specified taxID and all children in the taxonomy tree) into a separate file.		tax-ids	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input sequences and tax IDs:

The following input should be provided:

- URL(s) to FASTQ or FASTA file(s).
- Corresponding taxonomy classification of sequences in the files.

To process single-end reads or contigs, pass the URL(s) to the "Input URL 1" slot.

To process paired-end reads, pass the URL(s) to files with the "left" and "right" reads to the "Input URL 1" and "Input URL 2" slots correspondingly.

The taxonomy classification data are received by one of the classification tools (Kraken, CLARK, or DIAMOND) and should correspond to the input files.

Name in Workflow File: in

Slots:

SlotInGUI	Slot in Workflow File	Type
Input URL	**url**	*string*
Taxonomy data	**tax-data**	*tax-classification*

The element has 1 *output port*:

Name in GUI: Output file(s):

The port outputs URLs to files with NGS reads, classified by taxon IDs: one file per each specified taxon ID per each input file (or the pair of files in case of PE reads).

Either one (for SE reads or contigs) or two (for PE reads) output slots are used depending on the input data. See also the "Input data" parameter of the element.

Name in Workflow File: out

Slots:

SlotInGUI	Slot in Workflow File	Type
Output URL 1	**url**	*string*
Output URL 2	**url**	*string*

Improve Classification with WEVOTE Element

WEVOTE (WEighted VOting Taxonomic idEntification) is a metagenome shortgun sequencing DNA reads classifier based on an ensemble of other classification methods (Kraken, CLARK, etc.).

Element type: wevote-classify

Parameters

Parameter	Description	Defaultvalue	Parameter in Workflow File	Type
Penalty	Score penalty for disagreements (-k)	2	**penalty**	*number*
Number of agreed tools	Specify the minimum number of tools agreed on WEVOTE decision (-a).	0	**number-of-agreed-tools**	*number*
Score threshold	Score threshold (-s)	0	**score-threshold**	*number*
Number of threads	Use multiple threads (-n).	8	**threads**	*number*
Output file	Specify the output text file name.	auto	**output-url**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input classification CSV file:

Input a CSV file in the following format: 1) a sequence name 2) taxID from the first tool 3) taxID from the second tool 4) etc.

Name in Workflow File: in

Slots:

SlotInGUI	Slot in Workflow File	Type
Input URL	**url**	*string*

The element has 1 *output port*:

Name in GUI: WEVOTE Classification:

A map of sequence names with the associated taxonomy IDs.

Name in Workflow File: out

Slots:

SlotInGUI	Slot in Workflow File	Type
Taxonomy classification data	**tax-data**	*tax-classification*

NGS: RNA-Seq Analysis

- Assemble Transcripts with StringTie Element
- Assembly Transcripts with Cufflinks Element
- Extract Transcript Sequences with gffread Element
- Merge Assemblies with Cuffmerge Element
- StringTie Gene Abudance Report Element
- Test for Diff. Expression with Cuffdiff Element

Assemble Transcripts with StringTie Element

StringTie is a fast and highly efficient assembler of RNA-Seq alignments into potential transcripts. It uses a novel network flow algorithm as well as an optional de novo assembly step to assemble and quantitate full-length transcripts representing multiple splice variants for each gene locus.

Element type: stringtie

Parameters

Parameter	Description	Defaultvalue	Parameter in Workflow	Type

			File	
Reference annotations	Use the reference annotation file (in GTF or GFF3 format) to guide the assembly process (-G). The output will include expressed reference transcripts as well as any novel transcripts that are assembled.		**reference-annotations**	*string*
Reads orientation	Select the NGS libraries type: unstranded, stranded fr-secondstrand (--fr), or stranded fr-firststand (--rf).	Unstranded	**reads-orientation**	*string*
Label	Use the specified string as the prefix for the name of the output transcripts (-l).	STRG	**label**	*string*
Min isoform fraction	Specify the minimum isoform abundance of the predicted transcripts as a fraction of the most abundant transcript assembled at a given locus (-f). Lower abundance transcripts are often artifacts of incompletely spliced precursors of processed transcripts.	0.1	**min-isoform-fraction**	*numeric*
Min assembled transcript length	Specify the minimum length for the predicted transcripts (-m).	200	**min-isoform-fraction**	*numeric*
Min anchor length for junctions	Junctions that don't have spliced reads that align them with at least this amount of bases on both sides are filtered out (-a).	10	**min-anchor-length**	*numeric*
Min junction coverage	There should be at least this many spliced reads that align across a junction (-j). This number can be fractional since some reads align in more than one place. A read that aligns in n places will contribute 1/n to the junction coverage.	1	**min-junction-coverage**	*numeric*
Trim transcripts based on coverage	By default StringTie adjusts the predicted transcript's start and/or stop coordinates based on sudden drops in coverage of the assembled transcript. Set this parameter to "False" to disable the trimming at the ends of the assembled transcripts (-t).	True	**trim-transcripts**	*bool*
Min coverage for assembled transcripts	Specifies the minimum read coverage allowed for the predicted transcripts (-c). A transcript with a lower coverage than this value is not shown in the output. This number can be fractional since some reads align in more than one place. A read that aligns in n places will contribute 1/n to the coverage.	2.5	**min-coverage**	*numeric*
Min locus gap separation	Reads that are mapped closer than this distance are merged together in the same processing bundle (-g).	50 bp	**min-locus-gap**	*numeric*
Fraction covered by multi-hit reads	Specify the maximum fraction of multiple-location-mapped reads that are allowed to be present at a given locus (-M). A read that aligns in n places will contribute 1/n to the coverage.	0.95	**multi-hit-fraction**	*numeric*
Skip assembling for sequences	Ignore all read alignments (and thus do not attempt to perform transcript assembly) on the specified reference sequences (-x). The value can be a single reference sequence name (e.g. "chrM") or a comma-delimited list of sequence names (e.g. "chrM,chrX,chrY"). This can speed up StringTie especially in the case of excluding the mitochondrial genome, whose genes may have very high coverage in some cases, even though they may be of no interest for a particular RNA-Seq analysis. The reference sequence names are case sensitive, they must match identically the names of chromosomes/contigs of the target genome against which the RNA-Seq reads were aligned in the first place.		**skip-sequences**	*string*
Multi-mapping correction	Enables or disables (-u) multi-mapping correction.	Enabled	**multi-mapping-correction**	*bool*
Verbose log	Enable detailed logging, if required (-v). The messages will be written to the UGENE log (enabling of "DETAILS" and "TRACE" logging may be required) and to the dashboard.	False	**verbose-log**	*bool*
Number **of threads**	Specify the number of processing threads (CPUs) to use for transcript assembly (-p).	8	**threads**	*numeric*
Output transcripts file	StringTie's primary output GTF file with assembled transcripts.	Auto	**transcripts-output-url**	*string*
Enable gene abundance output	Select "True" to generate gene abundances output (-A). The output is written to a tab-delimited text file. Also, the file URL is passed to an output slot of the workflow element.	False	**gene-abundance-output**	*bool*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input BAM file(s)

Name in Workflow File: in

Slots:

Slot in GUI	Slot in Workflow File	Type
Source URL	url	*string*

And 1 *output port*:

Name in GUI: StringTie output data

Name in Workflow File: out

Slots:

Slot in GUI	Slot in Workflow File	Type
Output URL	url	*string*

Assembly Transcripts with Cufflinks Element

Cufflinks accept aligned RNA-Seq reads and assemble the alignments into a parsimonious set of transcripts. Cufflinks then estimate the relative abundances of these transcripts based on how many reads support each one, taking into account biases in library preparation protocols.

Element type: cufflinks

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	Directory to save MACS output files.		**out-dir**	*string*
Reference annotation	Tells Cufflinks to use the supplied reference annotation to estimate isoform expression. Cufflinks will not assemble novel transcripts and the program will ignore alignments not structurally compatible with any reference transcript.		**ref-annotation**	*string*
RABT annotation	Tells Cufflinks to use the supplied reference annotation to guide Reference Annotation Based Transcript (RABT) assembly. Reference transcripts will be tiled with faux-reads to provide additional information in an assembly. The output will include all reference transcripts as well as any novel genes and isoforms that are assembled.		**rabt-annotation**	*string*
Library type	Specifies RNA-Seq protocol.	Standart Illumina	**library-type**	*numeric*
Mask file	Ignore all reads that could have come from transcripts in this file. It is recommended to include any annotated rRNA, mitochondrial transcripts other abundant transcripts you wish to ignore in your analysis in this file. Due to variable efficiency of mRNA enrichment methods and rRNA depletion kits, masking these transcripts often improves the overall robustness of transcript abundance estimates.		**mask-file**	*string*
Multi-read correct	Tells Cufflinks to do an initial estimation procedure to more accurately weight reads mapping to multiple locations in the genome.	False	**multi-read-correct**	*boolean*
Min isoform fraction	After calculating isoform abundance for a gene, Cufflinks filters out transcripts that it believes are very low abundance, because isoforms expressed at extremely low levels often cannot reliably be assembled, and may even be artifacts of incompletely spliced precursors of processed transcripts. This parameter is also used to filter out introns that have far fewer spliced alignments supporting them.	0.1	**min-isoform-fraction**	*numeric*
Frag bias correct	Providing Cufflinks with a multifasta file via this option instructs it to run the bias detection and correction algorithm which can significantly improve the accuracy of transcript abundance estimates.		**frag-bias-correct**	*string*
Pre-mRNA fraction	Some RNA-Seq protocols produce a significant amount of reads that originate from incompletely spliced transcripts, and these reads can confound the assembly of fully spliced mRNAs. Cufflinks use this parameter to filter out alignments that lie within the intronic intervals implied by the spliced alignments. The minimum depth of coverage in the intronic region covered by the alignment is divided by the number of spliced reads, and if the result is lower than this parameter value, the intronic alignments are ignored.	0.15	**pre-mrna-fraction**	*numeric*
Cufflinks tool path	The path to the Cufflinks external tool in UGENE.	default	**path**	*string*
Temporary directory	The directory for temporary files.	default	**tmp-dir**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input reads

Name in Workflow File: in-assembly

Slots:

Slot In GUI	Slot in Workflow File	Type
Assembly data	assembly	*assembly*
Source url	url	*string*

And 1 *output port*:

Name in GUI: Output annotations

Name in Workflow File: out-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Isoform-level expression values	isolevel.slot	*ann_table*

Extract Transcript Sequences with gffread Element

Extract transcript sequences from the genomic sequence(s) with gffread.

Element type: gffread

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output sequences	The url to the output file with the extracted sequences.		**url-out**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input transcripts

Name in Workflow File: in-data

Slots:

Slot In GUI	Slot in Workflow File	Type
Genomic sequence url	**genome**	*string*
Transcripts url	**transcripts**	*string*

And 1 *output port*:

Name in GUI: Extracted sequences url

Name in Workflow File: extracted-data

Slots:

Slot In GUI	Slot in Workflow File	Type
sequences	**sequences**	*string*

Merge Assemblies with Cuffmerge Element

Cuffmerge merges together several assemblies. It also handles running Cuffcompare for you and automatically filters a number of transfrags that are probably artifacts. If you have a reference file available, you can provide it to Cuffmerge in order to gracefully merge input (e.g. novel) isoforms and known isoforms and maximize overall assembly quality.

Element type: cuffmerge

Parameters

Parameter	Description	Default value	Parameter in	Type

			Workflow File	
Output directory	Directory to save MACS output files.		**out-dir**	*string*
Reference annotation	Merge the input assemblies together with this reference annotation.		**ref-annotation**	*string*
Reference sequence	The genomic DNA sequences for the reference. It is used to assist in classifying transfrags and excluding artifacts (e.g. repeats). For example, transcripts consisting mostly of lower-case bases are classified as repeats.		**ref-seq**	*string*
Minimum isoform fraction	Discard isoforms with abundance below this.	0.05	**min-isoform-fraction**	*numeric*
Cuffcompare tool path	The path to the Cuffcompare external tool in UGENE.	default	**cuffcompare-tool-path**	*string*
Cuffmerge tool path	The path to the Cuffmerge external tool in UGENE.	default	**path**	*string*
Temporary directory	The directory for temporary files.	default	**tmp-dir**	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Set of annotations

Name in Workflow File: in-assembly

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**in-annotations**	*ann_table*

And 1 *output port*:

Name in GUI: Set of annotations

Name in Workflow File: out-assembly

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**out-annotations**	*ann_table*

StringTie Gene Abudance Report Element

The element summarizes gene abundance output of StringTie and saves the result into a common tab-delimited text file. The first two columns of the file are "Gene ID" and "Gene name". Each other column contains "FPKM" values for the genes from an input gene abundance file.

Element type: stringtie-gene-abundance-report

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output file	Specify the name of the output tab-delimited text file.		output-url	*string*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Input StringTie gene abundance file(s) url

Name in Workflow File: in

Slots:

Slot in GUI	Slot in Workflow File	Type
Input URL	url	*string*

Test for Diff. Expression with Cuffdiff Element

Cuffdiff takes a transcript file as input, along with two or more fragment alignments (e.g. in SAM format) for two or more samples. It produces a number of output files that contain test results for changes in expression at the level of transcripts, primary transcripts, and genes. It also tracks changes in the relative abundance of transcripts sharing a common transcription start site, and in the relative abundances of the primary transcripts of each gene. Tracking the former allows one to see changes in splicing, and the latter lets one see changes in relative promoter use within a gene.

Element type: cuffdiff

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	Directory to save MACS output files.		**out-dir**	*string*
Time series analysis	If set to True, instructs Cuffdiff to analyze the provided samples as a time series, rather than testing for differences between all pairs of samples. Samples should be provided in increasing time order.	False	**time-series-analysis**	*boolean*
Upper quartile norm	If set to True, normalizes by the upper quartile of the number of fragments mapping to individual loci instead of the total number of sequenced fragments. This can improve robustness of differential expression calls for less abundant genes and transcripts.	False	**upper-quartile-norm**	*boolean*
Hits norm	Instructs how to count all fragments. Total specifies to count all fragments, including those not compatible with any reference transcript, towards the number of mapped fragments used in the FPKM denominator. Compatible specifies to use only compatible fragments. Selecting Compatible is generally recommended in Cuffdiff to reduce certain types of bias caused by differential amounts of ribosomal reads which can create the impression of falsely differentially expressed genes..	Compatible	**hits-norm**	*numeric*
Frag bias correct	Providing the sequences your reads were mapped to instructs Cuffdiff to run bias detection and correction algorithm which can significantly improve accuracy of transcript abundance estimates..		**frag-bias-correct**	*string*
Multi read correct	Do an initial estimation procedure to more accurately weight reads mapping to multiple locations in the genome.	False	**multi-read-correct**	*boolean*
Library type	Specifies RNA-Seq protocol.	Standard Illumina	**library-type**	*numeric*
Mask file	Ignore all reads that could have come from transcripts in this file. It is recommended to include any annotated rRNA, mitochondrial transcripts other abundant transcripts you wish to ignore in your analysis in this file. Due to variable efficiency of mRNA enrichment methods and rRNA depletion kits, masking these transcripts often improves the overall robustness of transcript abundance estimates..		**mask-file**	*numeric*
Min alignment count	The minimum number of alignments in a locus for needed to conduct significance testing on changes in that locus observed between samples. If no testing is performed, changes in the locus are deemed not significant, and the locus' observed changes don't contribute to correction for multiple testing..	10	**min-alignment-count**	*string*
FDR	The allowed false discovery rate used in testing.	0.05	**fdr**	*numeric*
Max MLE iterations	Sets the number of iterations allowed during maximum likelihood estimation of abundances.	5000	**max-mle-iterations**	*numeric*
Emit count tables	Include information about the fragment counts, fragment count variances, and fitted variance model into the report.	False	**emit-count-tables**	*boolean*
Cuffdiff tool path	The path to the Cuffdiff external tool in UGENE.	defaul	**path**	*string*
Temporary directory	The directory for temporary files.	default	**temp-dir**	*string*

Input/Output Ports

The element has 2 *input port*:

Name in GUI: Annotations

Name in Workflow File: in-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**in-annotations**	*ann_table*

Name in GUI: Assembly

Name in Workflow File: in-assembly

Slots:

Slot In GUI	Slot in Workflow File	Type
Assembly data	**assembly**	*assembly*
Source url	**url**	*string*

NGS: Variant Analysis

- Call Variants with SAMtools Element
- Change Chromosome Notation for VCF Element
- Convert SnpEff Variations to Annotations Element
- Create VCF Consensus Element
- SnpEff Annotation and Filtration Element

Call Variants with SAMtools Element

Calls SNPs and INDELS with SAMtools mpileup and bcftools.

Element type: call_variants

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output variants file	The url to the file with the extracted variations.			
Reference	Specify a file with the reference sequence. The sequence will be used as reference for all datasets with NGS assemblies.			
Use reference from	Specify "File" to set a single reference sequence for all input NGS assemblies. The reference should be set in the "Reference" parameter. Specify "Input port" to be able to set different references for difference NGS assemblies. The references should be input via the "Input sequences" port (e.g. use datasets in the "Read Sequence" element).	File		
Illumina-1.3+ encoding	Assume the quality is in the Illumina 1.3+ encoding (mpileup)(-6).	False	**illumina13-encoding**	*boolean*
Count anomalous read pairs	Do not skip anomalous read pairs in variant calling (mpileup)(-A).	False	**use_orphan**	*boolean*
Disable BAQ computation	Disable probabilistic realignment for the computation of base alignment quality (BAQ). BAQ is the Phred-scaled probability of a read base being misaligned. Applying this option greatly helps to reduce false SNPs caused by misalignments (mpileup)(-B).	False	**disable_baq**	*boolean*
Mapping quality downgrading coefficient	Coefficient for downgrading mapping quality for reads containing excessive mismatches. Given a read with a phred-scaled probability q of being generated from the mapped position, the new mapping quality is about sqrt ((INT-q)/INT)*INT. A zero value disables this functionality; if enabled, the recommended value for BWA is 50 (mpileup)(-C).	0	**capq_thres**	*numeric*
Max number of reads per input BAM	At a position, read maximally the number of reads per input BAM (mpileup)(-d).	250	**max_depth**	*numeric*
Extended BAQ computation	Extended BAQ computation. This option helps sensitivity especially for MNPs, but may hurt specificity a little bit (mpileup)(-E).	False	**ext_baq**	*boolean*
BED or position list file	BED or position list file containing a list of regions or sites where pileup or BCF should be generated. (mpileup)(-l).		**bed**	*string*
Pileup region	Only generate pileup in region STR (mpileup)(-r).		**reg**	*string*
Minimum mapping quality	Minimum mapping quality for an alignment to be used (mpileup)(-q).	0	**min_mq**	*numeric*
Minimum base quality	Minimum base quality for a base to be considered (mpileup)(-Q).	13	**min_baseq**	*numeric*
Gap	Phred-scaled gap extension sequencing error probability. Reducing INT leads to longer indels (mpileup)(-e).	20	**extQ**	*numeric*

extension error				
Homopolymer errors coefficient	Coefficient for modeling homopolymer errors. Given an l-long homopolymer run, the sequencing error of an indel of size s is modeled as INT*s/l. (mpileup)(-h).	100	**tandemQ**	*numeric*
No INDELs	Do not perform INDEL calling (mpileup)(-I).	False	**no_indel**	*boolean*
Max INDEL depth	Skip INDEL calling if the average per-sample depth is above INT (mpileup)(-L).	250	**max_indel_depth**	*numeric*
Gap open error	Phred-scaled gap open sequencing error probability. Reducing INT leads to more indel calls (mpileup)(-o).	40	**openQ**	*numeric*
List of platforms for indels	Comma dilimited list of platforms (determined by @RG-PL) from which indel candidates are obtained.It is recommended to collect indel candidates from sequencing technologies that have low indel error rate such as ILLUMINA. (mpileup)(-P).		**pl_list**	*string*
Retain all possible alternate	Retain all possible alternate alleles at variant sites. By default, the view command discards unlikely alleles. (bcf view)(-A).	False	**keepalt**	*boolean*
Indicate PL	Indicate PL is generated by r921 or before (ordering is different) (bcf view)(-F).	False	**fix_pl**	*boolean*
No genotype information	Suppress all individual genotype information (bcf view)(-G).	False	**no_geno**	*boolean*
A/C/G/T only	Skip sites where the REF field is not A/C/G/T (bcf view)(-N).	False	**acgt_only**	*boolean*
List of sites	List of sites at which information are outputted (bcf view)(-l).		**bcf_bed**	*string*
QCALL likelihood	Output the QCALL likelihood format (bcf view)(-Q).	False	**qcall**	*boolean*
List of samples	List of samples to use. The first column in the input gives the sample names and the second gives the ploidy, which can only be 1 or 2. When the 2nd column is absent, the sample ploidy is assumed to be 2. In the output, the ordering of samples will be identical to the one in FILE (bcf view)(-s).		**samples**	*string*
Min samples fraction	skip loci where the fraction of samples covered by reads is below FLOAT (bcf view)(-d).	0	**min_smpl_frac**	*numeric*
Per-sample genotypes	Call per-sample genotypes at variant sites. (bcf view)(-g).	True	**call_gt**	*boolean*
INDEL-to-SNP Ratio	Ratio of INDEL-to-SNP mutation rate. (bcf view)(-i).	-1	**indel_frac**	*numeric*
Max P(ref|D)	A site is considered to be a variant if P(ref|D)	0.5	**pref**	*numeric*
Prior allele frequency spectrum	If STR can be full, cond2, flat or the file consisting of error output from a previous variant calling run (bcf view)(-P).	full	**ptype**	*string*
Mutation rate	Scaled mutation rate for variant calling (bcf view)(-t).	0.001	**theta**	*numeric*
Pair/trio calling	Enable pair/trio calling. For trio calling, option -s is usually needed to be applied to configure the trio members and their ordering. In the file supplied to the option -s, the first sample must be the child, the second the father and the third the mother. The valid values of STR are pair, trioauto, trioxd and trioxs, where pair calls differences between two input samples, and trioxd (trioxs)specifies that the input is from the X chromosome non-PAR regions and the child is a female (male) (bcf view)(-T).		**ccall**	*string*
N group-1 samples	Number of group-1 samples. This option is used for dividing the samples into two groups for contrast SNP calling or association test. When this option is in use, the followingVCF INFO will be outputted: PC2, PCHI2 and QCHI2 (bcf view)(-1).	0	**n1**	*numeric*
N permutations	Number of permutations for association test (effective only with -1) (bcf view)(-U).	0	**n_perm**	*numeric*
Min P(chi^2)	Only perform permutations for P(chi^2).	0.01	**min_perm_p**	*numeric*
Minimum RMS quality	Minimum RMS mapping quality for SNPs (varFilter) (-Q).	10	**min-qual**	*numeric*
Minimum read depth	Minimum read depth (varFilter) (-d).	2	**min-dep**	*numeric*
Maximum read depth	Maximum read depth (varFilter) (-D).	10000000	**max-dep**	*numeric*
Alternate bases	Minimum number of alternate bases (varFilter) (-a).	2	**min-alt-bases**	*numeric*
Gap size	SNP within INT bp around a gap to be filtered (varFilter) (-w).	3	**gap-size**	*numeric*
Window size	Window size for filtering adjacent gaps (varFilter) (-W).	10	**window"**	*numeric*
Strand bias	Minimum P-value for strand bias (given PV4) (varFilter) (-1).	0.0001	**min-strand**	*numeric*

BaseQ bias	Minimum P-value for baseQ bias (varFilter) (-2).	1e-100	**min-baseQ**	*string*
MapQ bias	Minimum P-value for mapQ bias (varFilter) (-3).	0	**min-mapQ**	*numeric*
End distance bias	Minimum P-value for end distance bias (varFilter) (-4).	0.0001	**min-end-distance**	*numeric*
HWE	Minimum P-value for HWE (plus F).	0.0001	**min-hwe**	*numeric*
Log filtered	Print filtered variants into the log (varFilter) (-p).	False	**print-filtered**	*boolean*

Input/Output Ports

The element has 2 *input ports*:

Name in GUI: Input assembly

Name in Workflow File: in-assembly

Slots:

Slot In GUI	Slot in Workflow File	Type
Dataset name	**dataset**	*string*
Source url	**url**	*string*

Name in GUI: Input sequences

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Source url	**url**	*string*

And 1 *output port*:

Name in GUI: Output variations

Name in Workflow File: out-variations

Slots:

Slot In GUI	Slot in Workflow File	Type
Variation track	**variation-track**	*variation*

Change Chromosome Notation for VCF Element

Changes chromosome notation for each variant from the input, VCF or other variation files.

Element type: rename-chromosome-in-variation

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Replace prefixes	Input the list of chromosome prefixes that you would like to replace. For example "NC_000". Separate different prefixes by semicolons.		prefixes-to-replace	*string*
Replace by	Input the prefix that should be set instead, for example "chr".		prefix-replace-with	*string*

Input/Output Ports

The element has 1 *input ports*:

Name in GUI: Input file URL

Name in Workflow File: in-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

And 1 *output port*:

Name in GUI: Output file URL

Name in Workflow File: out-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Produced URL	**url**	*string*

Convert SnpEff Variations to Annotations Element

Parses information, added to variations by SnpEff, into standard annotations.

Element type: convert-snpeff-variations-to-annotations

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output file	Location of output data file. If this attribute is set, slot "Location" in port will not be used.		url-out	*string*
Document format	Document format of the output file.	genbank	document-format	*string*

Input/Output Ports

The element has 1 *input ports*:

Name in GUI: Input file URL

Name in Workflow File: in-variations-url

Slots:

Slot In GUI	Slot in Workflow File	Type
Source URL	**url**	*string*

Create VCF Consensus Element

Apply VCF variants to a fasta file to create a consensus sequence.

Element type: vcf-consensus

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output FASTA consensus	The URL to the output file with the resulting consensus.		consensus-url	*string*

Input/Output Ports

The element has 1 *input ports*:

Name in GUI: Input FASTA and VCF

Name in Workflow File: in-data

Slots:

Slot In GUI	Slot in Workflow File	Type
Fasta url	**fasta**	*string*
VCF url	**vcf**	*string*

And 1 *output port*:

Name in GUI: Fasta consensus URL

Name in Workflow File: out-consensus

Slots:

Slot In GUI	Slot in Workflow File	Type
out-consensus	**out-consensus**	*string*

SnpEff Annotation and Filtration Element

Annotates and filters variations with SnpEff.

Element type: seff

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output directory	Select an output directory. Custom - specify the output directory in the 'Custom directory' parameter. Workflow - internal workflow directory. Input file - the directory of the input file.	Input file	**out-mode**	*string*
Input format	Select the input format of variations.	VCF	**inp-format**	*string*
Output format	Select the format of annotated output files.	VCF	**out-format**	*string*
Genome	Select the target genome from the list of SnpEff databases. Genome data will be downloaded if it is not found. The list of databases depends on the SnpEff external tool version.	Homo sapiens	**genome**	*string*
Upstream /downstream length	Upstream and downstream interval size. Eliminate any upstream and downstream effect by using 0 length.	No upstream /downstream interval (0 bases)	**updown-length**	*numeric*
Cannonical transcripts	Use only cannonical transcripts.	False	**canon**	*boolean*
HGVS nomenclature	Annotate using HGVS nomenclature.	False	**hgvs**	*boolean*
Annotate loss of function	Annotate Loss of function (LOF) and Nonsense mediated decay (NMD).	False	**lof**	*boolean*
Annotate TFBSs motifs	Annotate transcription factor binding site motifs (only available for latest GRCh37).	False	**motif**	*boolean*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: Variations

Name in Workflow File: in-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source url	**url**	*string*

And 1 *output port*:

Name in GUI: Annotated variations

Name in Workflow File: out-file

Slots:

Slot In GUI	Slot in Workflow File	Type
Source url	**url**	*variation*

Transcription Factor

- Build Frequency Matrix Element
- Build SITECON Model Element
- Build Weight Matrix Element
- Convert Frequency Matrix Element
- Read Frequency Matrix Element
- Read SITECON Model Element
- Read Weight Matrix Element
- Search for TFBS with SITECON Element
- Search for TFBS with Weight Matrix Element
- Write Frequency Matrix Element
- Write SITECON Model Element
- Write Weight Matrix Element

Build Frequency Matrix Element

Builds a frequency matrix. Frequency matrices are used for probabilistic recognition of transcription factor binding sites.

Element type: fmatrix-build

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Matrix type	Dinucleic matrices are more detailed, while mononucleic one are more useful for small input data sets.	Mononucleic	**type**	*boolean* Available values are: • true - for Dinucleic • false - for Mononucleic

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input alignment*

Name in Workflow File: in-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*

And 1 *output port*:

Name in GUI: *Frequency matrix*

Name in Workflow File: out-fmatrix

Slots:

Slot In GUI	Slot in Workflow File	Type
Frequency matrix	**fmatrix**	*fmatrix*

Build SITECON Model Element

Builds statistical profile for SITECON. The SITECON is a program for probabilistic recognition of transcription factor binding sites.

Element type: sitecon-build

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Weight algorithm	Optional feature, in most cases applying no weight will fit. In some cases choosing algorithm 2 will increase the recognition quality.	None	**weight-algorithm**	*boolean* Available values are: • 0 - for None • 1 - for Algorithm2
Window size, bp	Window is used to pick out the most important alignment region and is located at the center of the alignment. Must be: windows size is not greater than TFBS alignment length, recommended: windows size is not greater than 50 bp.	40	**window-size**	*numeric*
Calibration length	Length of random synthetic sequences used to calibrate the profile. Should not be less than window size.	1M	**calibrate-length**	*numeric*
Random seed	The random seed, where is a positive integer. You can use this option to generate reproducible results for different runs on the same data.	0	**seed**	*numeric*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input alignment*

Name in Workflow File: in-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*
Origin	**url**	*string*

And 1 *output port*:

Name in GUI: *Sitecon model*

Name in Workflow File: out-sitecon

Slots:

Slot In GUI	Slot in Workflow File	Type
Sitecon model	**sitecon-model**	*sitecon-model*

Build Weight Matrix Element

Builds a weight matrix. Weight matrices are used for probabilistic recognition of transcription factor binding sites.

Element type: wmatrix-build

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Matrix type (required)	Dinucleic matrices are more detailed, while mononucleic one is more useful for small input data sets.	Mononucleic	**type**	*boolean* Available values are: • true - for Dinucleic • false - for Mononucleic
Weight algorithm	Different weight algorithms use different functions to build weight matrices. It allows us to get better precision on different data sets. Log-odds, NLG, and Match algorithms are sensitive to input matrices with zero values, so some of them may not work on those matrices.	Berg and Von Hippel	**weight-algorithm**	*string* Available values are: • Berg and von Hippel • Log-odds • Match • NLG

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input alignment*

Name in Workflow File: in-msa

Slots:

Slot In GUI	Slot in Workflow File	Type
MSA	**msa**	*msa*

And 1 *output port*:

Name in GUI: *Weight matrix*

Name in Workflow File: out-wmatrix

Slots:

Slot In GUI	Slot in Workflow File	Type

Weight matrix	wmatrix	*wmatrix*

Convert Frequency Matrix Element

Converts a frequency matrix to a weight matrix. Weight matrices are used for probabilistic recognition of transcription factor binding sites.

Element type: fmatrix-to-wmatrix

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Matrix type (required)	Dinucleic matrices are more detailed, while mononucleic one is more useful for small input data sets.	Mononucleic	**type**	*boolean* Available values are: • true - for Dinucleic • false - for Mononucleic
Weight algorithm	Different weight algorithms use different functions to build weight matrices. It allows us to get better precision on different data sets. Log-odds, NLG and Match algorithms are sensitive to input matrices with zero values, so some of them may not work on those matrices.	Berg and Von Hippel	**weight-algorithm**	*string* Available values are: • Berg and von Hippel • Log-odds • Match • NLG

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Frequency matrix*

Name in Workflow File: in-fmatrix

Slots:

Slot In GUI	Slot in Workflow File	Type
Frequency matrix	**fmatrix**	*fmatrix*

And 1 *output port*:

Name in GUI: *Weight matrix*

Name in Workflow File: out-wmatrix

Slots:

Slot In GUI	Slot in Workflow File	Type
Weight matrix	**wmatrix**	*wmatrix*

Read Frequency Matrix Element

Reads frequency matrices from file(s). The files can be local or Internet URLs.

Element type: fmatrix-read

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Input files (required)	Semicolon-separated list of paths to the input files.		**url-in**	*string*

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *Frequency matrix*

Name in Workflow File: out-fmatrix

Slots:

Slot In GUI	Slot in Workflow File	Type
Frequency matrix	**fmatrix**	*fmatrix*

Read SITECON Model Element

Reads SITECON profiles from file(s). The files can be local or Internet URLs.

Element type: sitecon-read

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Input files (required)	A semicolon-separated list of paths to the input files.		**url-in**	*string*

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *Sitecon model*

Name in Workflow File: out-sitecon

Slots:

Slot In GUI	Slot in Workflow File	Type
Sitecon model	**sitecon-model**	*sitecon-model*

Read Weight Matrix Element

Reads weight matrices from file(s). The files can be local or Internet URLs.

Element type: wmatrix-read

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Input files (required)	Semicolon-separated list of paths to the input files.		**url-in**	*string*

Input/Output Ports

And 1 *output port*:

Name in GUI: *Weight matrix*

Name in Workflow File: out-wmatrix

Slots:

Slot In GUI	Slot in Workflow File	Type
Weight matrix	**wmatrix**	*wmatrix*

Search for TFBS with SITECON Element

Searches each input sequence for transcription factor binding sites significantly similar to specified SITECON profiles. In case several profiles were supplied, searches with all profiles one by one and outputs merged set of annotations for each sequence.

Element type: sitecon-search

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Result annotation	Name of the result annotations.	misc_feature	**result-name**	*string*
Search in	Specifies which strands should be searched: direct, complement or both.	both strands	**strand**	*numeric* Available values are: • 0 - for searching in both strands • 1 - for searching in direct strand • 2 - for searching in complement strand
Min score	Recognition quality threshold, should be less than 100%. Choosing too low threshold will lead to recognition of too many TFBS recognised with too low trustworthiness. Choosing too high threshold may result in no TFBS recognised.	85	**min-score**	*numeric*
Min err1	Alternative setting for filtering results, minimal value of Error type I. Note that all thresholds (by score, by err1 and by err2) are applied when filtering results.	0.0	**err1**	*numeric*
Max err2	Alternative setting for filtering results, max value of Error type II. Note that all thresholds (by score, by err1 and by err2) are applied when filtering results.	0.001	**err2**	*numeric*

Input/Output Ports

The element has 2 *input ports*. The first port:

Name in GUI: *Sequence*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

The second input port gets the SITECON model:

Name in GUI: *Sitecon model*

Name in Workflow File: in-sitecon

Slots:

Slot In GUI	Slot in Workflow File	Type
Sitecon model	**sitecon-model**	*sitecon-model*

And there is 1 *output port*:

Name in GUI: *Sitecon annotations*

Name in Workflow File: out-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table*

Search for TFBS with Weight Matrix Element

Searches each input sequence for transcription factor binding sites significantly similar to specified weight matrices. In case several profiles were supplied, searches with all profiles one by one and outputs merged set of annotations for each sequence.

Element type: wmatrix-search

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Result annotation	Name of the result annotations.	misc_feature	**result-name**	*string*
Search in	Specifies which strands should be searched: direct, complement or both.	both strands	**strand**	*numeric* Available values are: • 0 - for searching in both strands • 1 - for searching in direct strand • 2 - for searching in complement strand
Min score	Minimum score to detect transcription factor binding site in percents.	85	**min-score**	*numeric*

Input/Output Ports

The element has 2 *input ports*. The first port:

Name in GUI: *Sequence*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

The second input port gets the SITECON model:

Name in GUI: *Weight matrix*

Name in Workflow File: in-wmatrix

Slots:

Slot In GUI	Slot in Workflow File	Type
Weight matrix	**wmatrix**	*wmatrix*

And there is 1 *output port*:

Name in GUI: *Weight matrix annotations*

Name in Workflow File: out-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table*

Write Frequency Matrix Element

Saves all input frequency matrices to specified location.

Element type: fmatrix-write

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output file (required)	Location of the output data file. If this attribute is set, the “Location” slot is not taken into account.		**url-out**	*string*
Existing file	If a target file already exists, you can specify how it should be handled: either overwritten, renamed or appended (if supported by file format).	Rename	**write-mode**	*numeric* Available values are: • 0 - for overwrite • 1 - for append • 2 - for rename

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Frequency matrix*

Name in Workflow File: in-fmatrix

Slots:

Slot In GUI	Slot in Workflow File	Type
Frequency matrix	**fmatrix**	*fmatrix*
Source URL	**url**	*string*

Write SITECON Model Element

Saves all input SITECON profiles to a specified location.

Type: sitecon-write

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output file (required)	Location of the output data file. If this attribute is set, the “Location” slot is not taken into account.		**url-out**	*string*

Existing file	If a target file already exists, you can specify how it should be handled: either overwritten, renamed or appended (if supported by file format).	Rename	**write-mode**	*numeric* Available values are: • 0 - for overwrite • 1 - for append • 2 - for rename

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Sitecon model*

Name in Workflow File: in-sitecon

Slots:

Slot In GUI	Slot in Workflow File	Type
Sitecon model	**sitecon-model**	*sitecon-model*
Source URL	**url**	*string*

Write Weight Matrix Element

Saves all input weight matrices to specified location.

Element type: wmatrix-write

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Output file (required)	Location of the output data file. If this attribute is set, the “Location” slot is not taken into account.		**url-out**	*string*
Existing file	If a target file already exists, you can specify how it should be handled: either overwritten, renamed or appended (if supported by file format).	Rename	**write-mode**	*numeric* Available values are: • 0 - for overwrite • 1 - for append • 2 - for rename

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Weight matrix*

Name in Workflow File: in-wmatrix

Slots:

Slot In GUI	Slot in Workflow File	Type
Weight matrix	**wmatrix**	*wmatrix*
Source URL	**url**	*string*

Utils

- DNA Statistics Element
- Generate DNA Element

DNA Statistics Element

Evaluates statistic for DNA sequences.

Element type: dna-stats

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
GC-content	Evaluate GC-content.	True	**gc-content**	*boolean*
GC1-content	Evaluate GC1-content.	True	**gc1-content**	*boolean*
GC2-content	Evaluate GC2-content.	True	**gc2-content**	*boolean*
GC3-content	Evaluate GC3-content.	True	**gc3-content**	*boolean*

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input sequence*

Name in Workflow File: in-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

The element has 1 *output port*:

Name in GUI: *Result annotation*

Name in Workflow File: out-annotations

Slots:

Slot In GUI	Slot in Workflow File	Type
Set of annotations	**annotations**	*annotation-table-list*

Generate DNA Element

Generates random DNA sequences with given nucleotide content that can be specified manually or evaluated from the reference file.

Element type: generate-dna

Parameters

Parameter	Description	Default value	Parameter in Workflow File	Type
Length	Length of the resulted sequence or sequences.	1000 bp	**length**	*numeric*
Count	Number of sequences to generate.	1	**count**	*numeric*
Seed	Value to initialize the random generator. By default (seed = -1) the generator is initialized with the system time.	-1	**seed**	*numeric*
Content	Specifies how the nucleotide content of the sequence(s) should be generated. It can be either taken from the reference file (see the *Reference*parameter), or input manually.	manual	**content**	*string*
Algorithm	Algorithm for generating random sequence(s). Two algorithms are available: GC Content and GC Skew. If you choose GC Content, then parameters *A*,*C*, *G*, *T* are used to generate the sequence. Otherwise, the *GC Skew*par ameter is used to generate the sequence(s).	GC Content	**algorithm**	*string* Availabl e values are: • gc - co

				• nt ent • gc - sk ew
Window size	The DNA sequence generation is divided into windows of the specified size. In each window the bases ratio, defined by other parameters, is kept.	1000	**window-size**	*numeric*
Reference	Path to the reference file (could be a sequence or an alignment).		**reference-url**	*string* Availabl e values are: • m an ual • ref er en ce
A	Adenine content.	25%	**percent-a**	*numeric*
C	Cytosine content.	25%	**percent-c**	*numeric*
G	Guanine content.	25%	**percent-g**	*numeric*
T	Thymine content.	25%	**percent-t**	*numeric*
GC Skew	GC Skew is calculated as (G - C) / (G + C), where G is the number of G's in the window, and C is the number of C's.	0.25	**gc-skew**	*numeric*

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *Sequences*

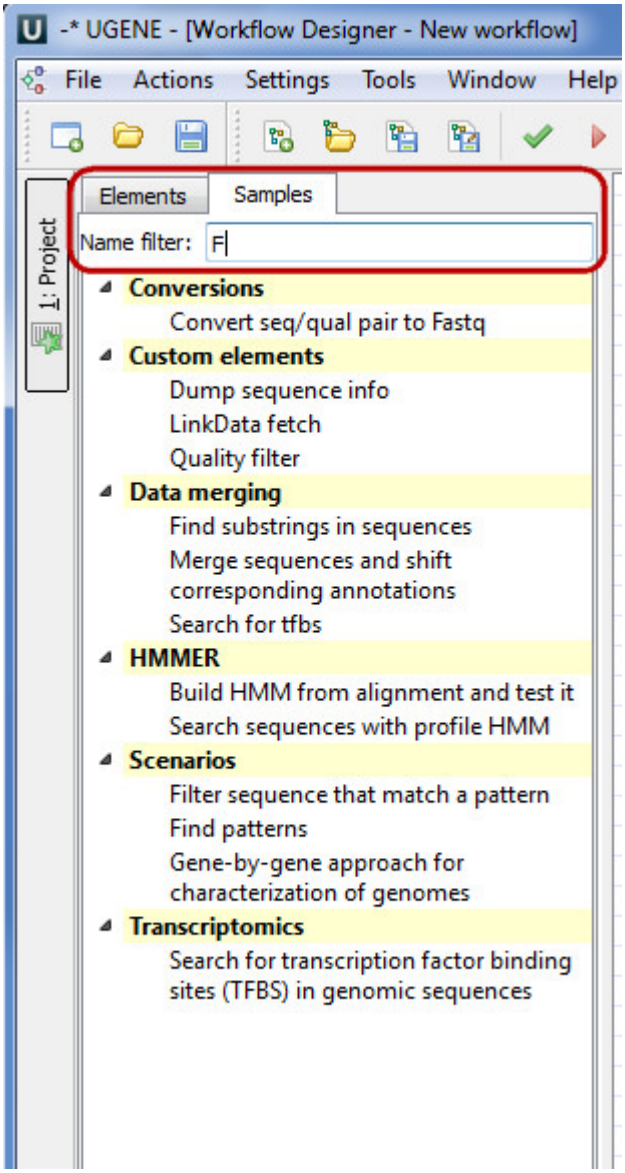
Name in Workflow File: out-sequence

Slots:

Slot In GUI	Slot in Workflow File	Type
Sequence	**sequence**	*sequence*

Workflow Samples

This section contains detailed description of workflow samples presented in the Workflow Designer. To search a sample use the name filter or press the *Ctr l+F* shortcut that moves you to the name filter also:



- Alignment
  - Align Sequences with MUSCLE
  - Extract Consensus as Sequence
  - Extract Consensus as Text
- Conversions
  - Convert "seq/qual" Pair to FASTQ
  - Convert Alignments to ClustalW
  - Convert UQL Schema Results to Alignment
  - Convert Sequence to Genbank
- Custom Elements
  - CASAVA FASTQ Filter
  - FASTQ Trimmer
  - Dump Sequence Info
  - LinkData Fetch
  - Quality Filter
- Data Marking
  - Marking by Annotation Number
  - Marking by Length
- Data Merging
  - Find Substrings in Sequences
  - Merge Sequences and Shift Corresponding Annotations
  - Search for TFBS
- HMMER
  - Build HMM from Alignment and test it
  - Search Sequences with Profile HMM
- NGS

- ChIP-Seq Coverage
- ChIP-seq Analysis with Cistrome Tools
- Extract Consensus from Assembly
- Extract Coverage from Assembly
- Extract Transcript Sequences
- Quality Control by FastQC
- De novo Assemble Illumina PE Reads
- De novo Assemble Illumina PE and Nanopore Reads
- De novo Assemble Illumina SE Reads
- De Novo Assembly and Contigs Classification
- Parallel NGS Reads Classification
- Serial NGS Reads Classification
- RNA-Seq Analysis with TopHat and StringTie
- RNA-seq Analysis with Tuxedo Tools
- Variation Annotation with SnpEff
- Call Variants with SAMtools
- Variant Calling and Effect Prediction
- Raw ChIP-Seq Data Processing
- Raw DNA-Seq Data Processing
- Raw RNA-Seq Data Processing
- Get Unmappet Reads

- Sanger Sequencing
  - Trim and Align Sanger Reads
- Scenarios
  - Filter Sequence That Match a Pattern
  - Search for Inverted Repeats
  - Find Patterns
  - Gene-by-gene Approach for Characterization of Genomes
  - Group Primer Pairs
  - Intersect Annotations
  - Filter out Short Sequences
  - Merge Sequences and Annotations
  - In Silico PCR Sample
  - Remote BLASTing
  - Get Amino Translations of a Sequence
- Transcriptomics
  - Search for Transcription Factor Binding Sites (TFBS) in Genomic Sequences

Alignment

- Align Sequences with MUSCLE
- Extract Consensus as Sequence
- Extract Consensus as Text

Align Sequences with MUSCLE

This workflow performs multiple sequence alignment with MUSCLE algorithm and saves the resulting alignment to Stockholm document. Source data can be of any format containing sequences or alignments.

How to Use This Sample

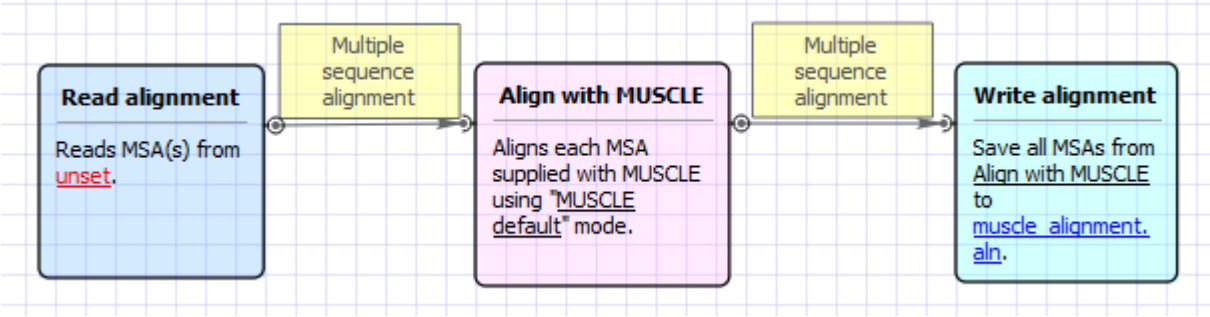
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Align Sequences with MUSCLE" can be found in the "Alignment" section of the Workflow Designer samples.

Workflow Image

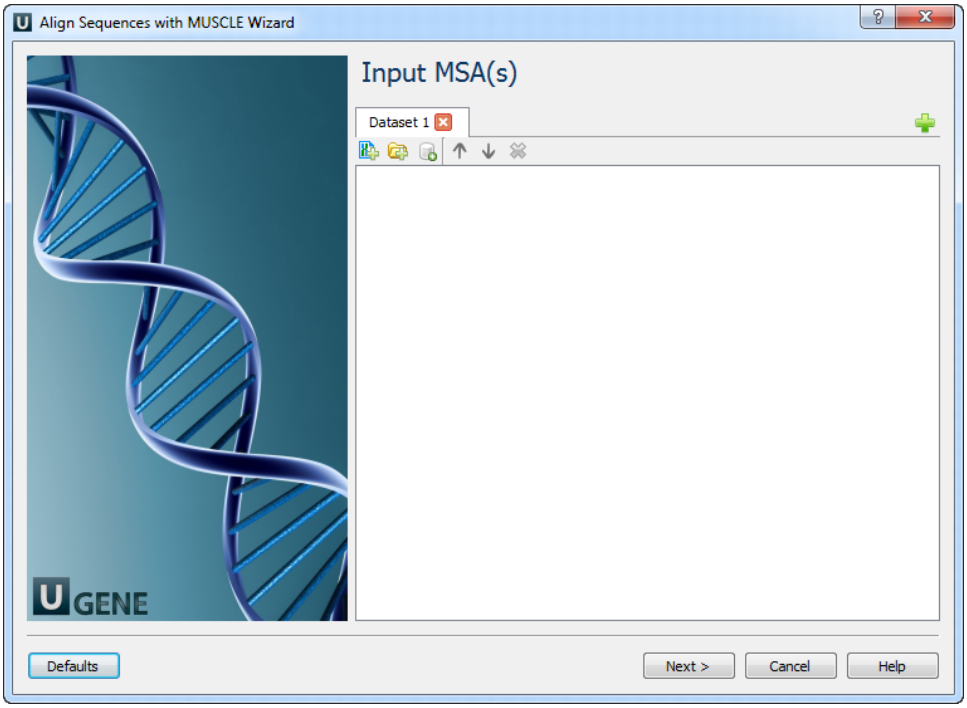
The workflow looks as follows:



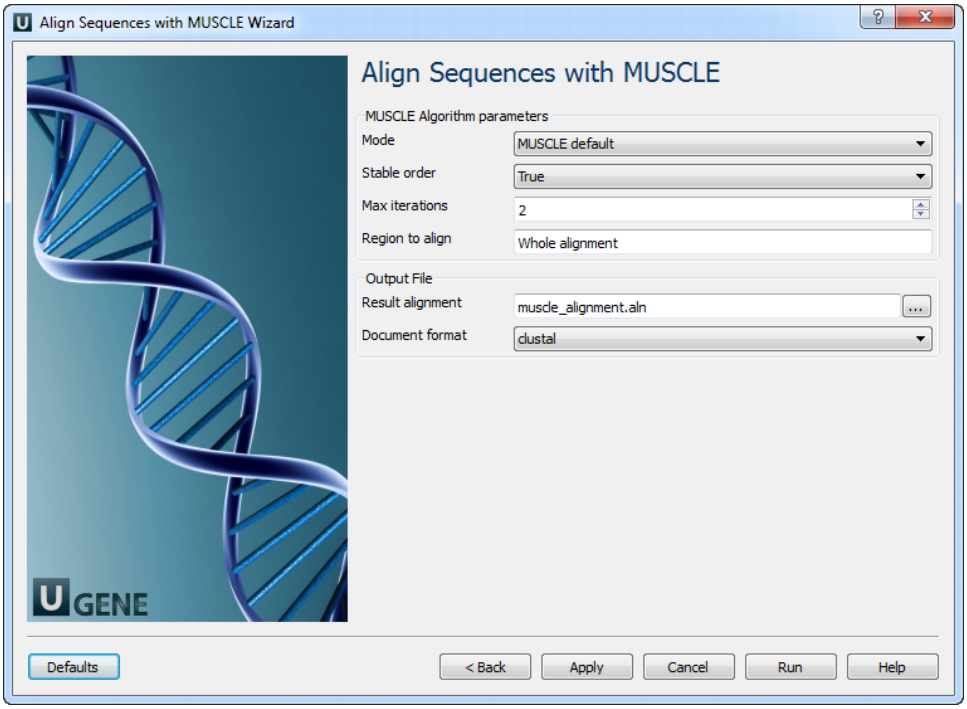
Workflow Wizard

The wizard has 2 pages.

1. Input MSA(s): On this page you must input multiple alignments file(s).



2. Align Sequences with MUSCLE: On this page you can modify MUSCLE and output parameters.



The following parameters are available:

Mode	Selector of preset configurations, that give you the choice of optimizing accuracy, speed, or some compromise between the two. The default favors accuracy.
Stable order	Do not rearrange aligned sequences (-stable switch of MUSCLE). Otherwise, MUSCLE re-arranges sequences so that similar sequences are adjacent in the output file. This makes the alignment easier to evaluate by eye.
Max iterations	Maximum number of iterations.
Region to align	Whole alignment or column range e.g. 1..100.
Result alignment	Location of output data file. If this attribute is set, slot "Location" in port will not be used.
Document format	Document format of output file.

Extract Consensus as Sequence

For each input multiple alignment the workflow calculates the consensus and saves it to a fasta file, named according to the name of the input alignment.

The "strict" algorithm with the "threshold" parameter set to "100%" is used by default to calculate the consensus. It means that the consensus will only contain characters that are the same in ALL sequences of the alignment. Decreasing the threshold will result in taking into account only the specified percentage number of the sequences, i.e. if the threshold is "80%" and 82% of the sequences have "A" at a certain column position, the consensus will also contain "A" at this position.

Also, you may select another algorithm to calculate the consensus. The algorithm, proposed by Victor Levitsky, uses the extended DNA alphabet. The greater the "threshold" value selected for this algorithm, the more rare characters are taken into account. The specified value must be between 50% and 100%.

Finally, there is "Keep gaps" parameter that specifies whether the result sequence must contain gaps, or they must be skipped. By default, the gaps are kept in the result consensus sequence.

How to Use This Sample

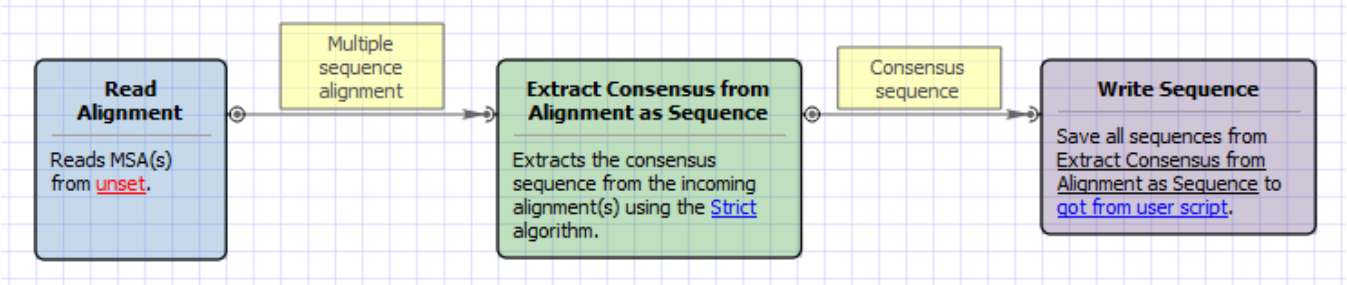
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Extract Consensus as Sequence" can be found in the "Alignment" section of the Workflow Designer samples.

Workflow Image

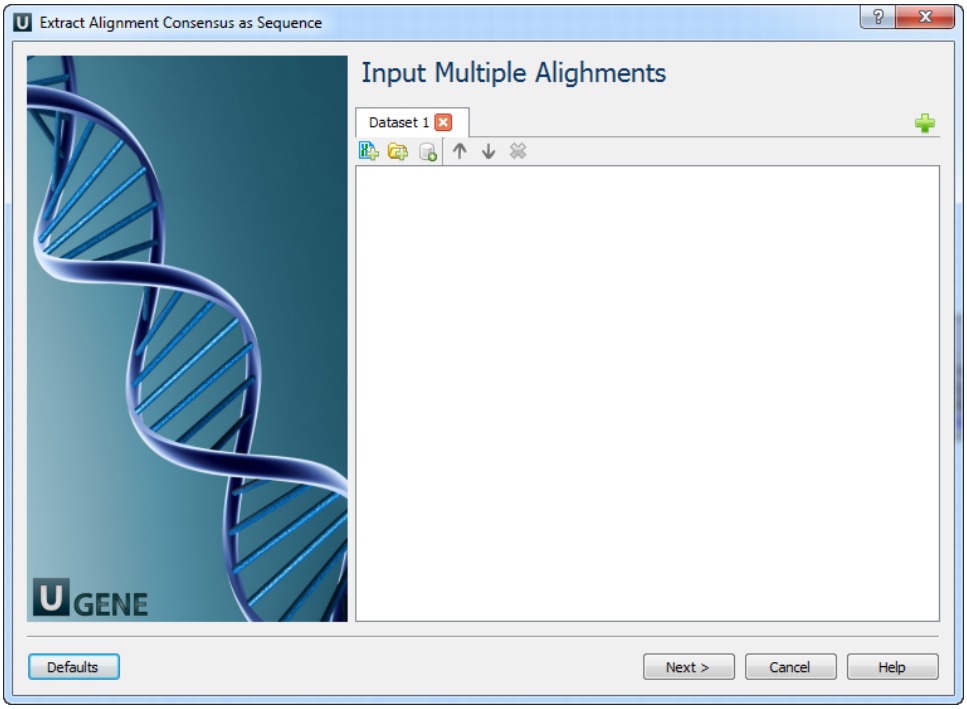
The workflow looks as follows:



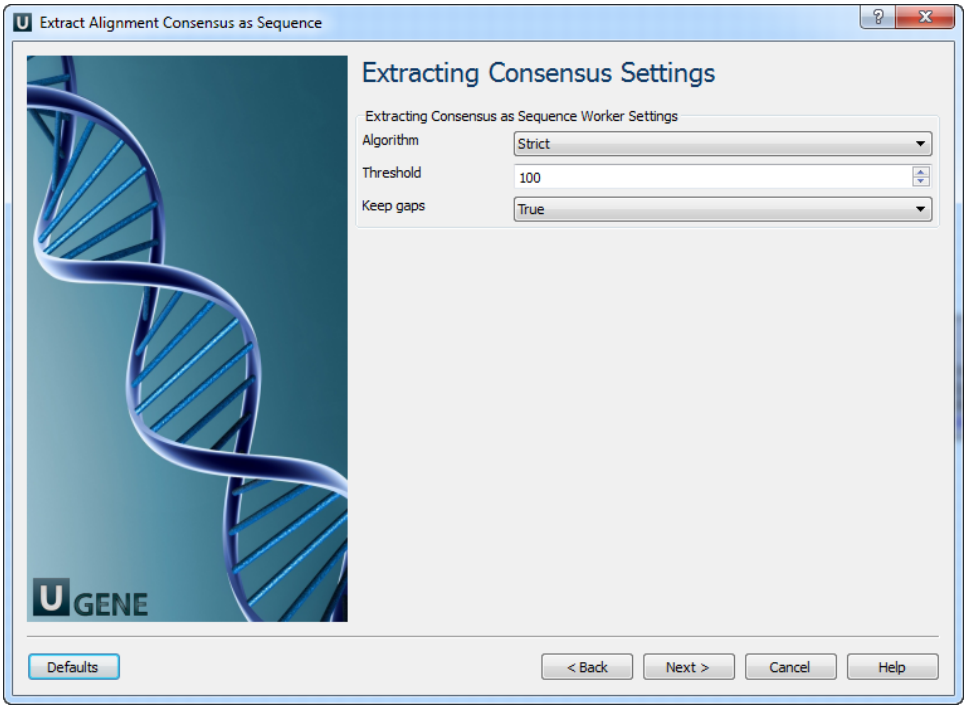
Workflow Wizard

The wizard has 3 pages.

1. Input Multiple Alignments: On this page you must input multiple alignments file(s).



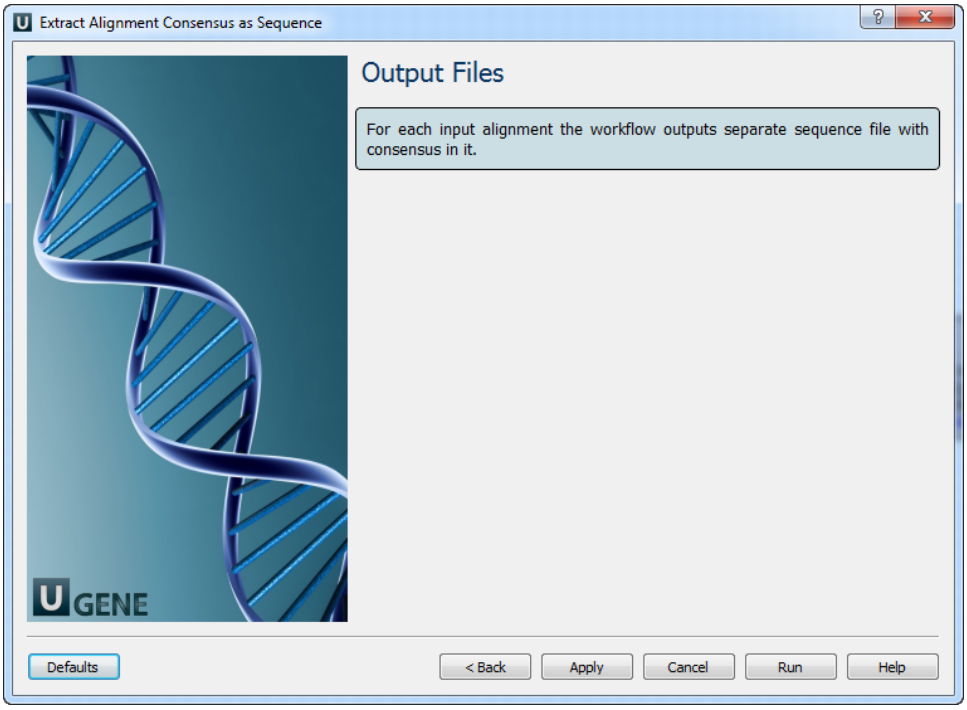
2. Extracting Consensus Settings: On this page you can modify extracting settings.



The following parameters are available:

Algorithm	The algorithm of consensus extracting.
Threshold	The threshold of the algorithm.
Keep gaps	Set this parameter if the result consensus must keep the gaps.

3. Output Files: For each input alignment the workflow outputs separate sequence file with consensus in it.



Extract Consensus as Text

For each input multiple alignment the workflow calculates the consensus and saves it to a text file, named according to the name of the input alignment.

The JalView algorithm (denoted as "default") is used by default to calculate the consensus. For each column of the alignment it returns either "+", if there are 2 characters with high frequency in this column, or a character in uppercase or lowercase. The case of the character depends on the percentage value of the character in the column and the "threshold" value.

Alternatively, you can use the ClustalW algorithm to calculate the consensus:

- If all characters in a column are exactly the same, the algorithm sets asterisk value ("*") to the corresponding position of the consensus.
- A colon value (":") indicates conservation between groups of strongly similar properties, i.e. the scoring value is greater than 0.5 in the Gonnet PAM 250 matrix (see documentation for details).
- If the scoring value is less than 0.5, the period (".") value is inserted.
- Otherwise, the algorithm inserts space (" ").

The "threshold" parameter is not applied to this algorithm.

How to Use This Sample

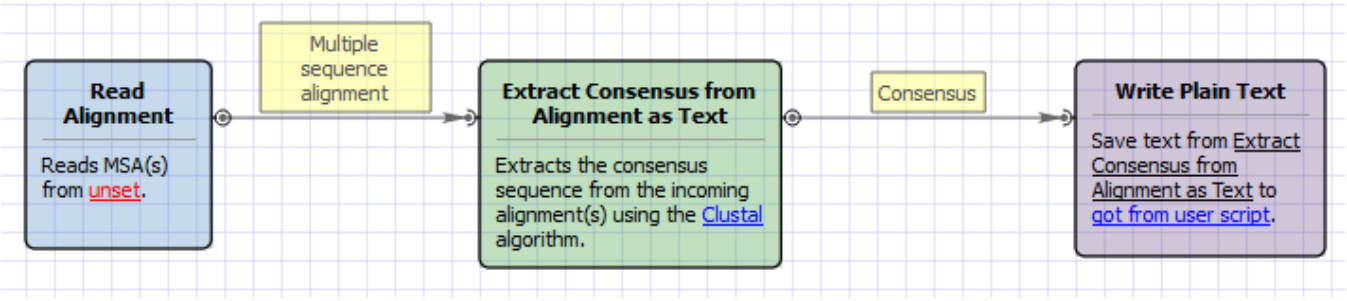
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Extract Consensus as Text" can be found in the "Alignment" section of the Workflow Designer samples.

Workflow Image

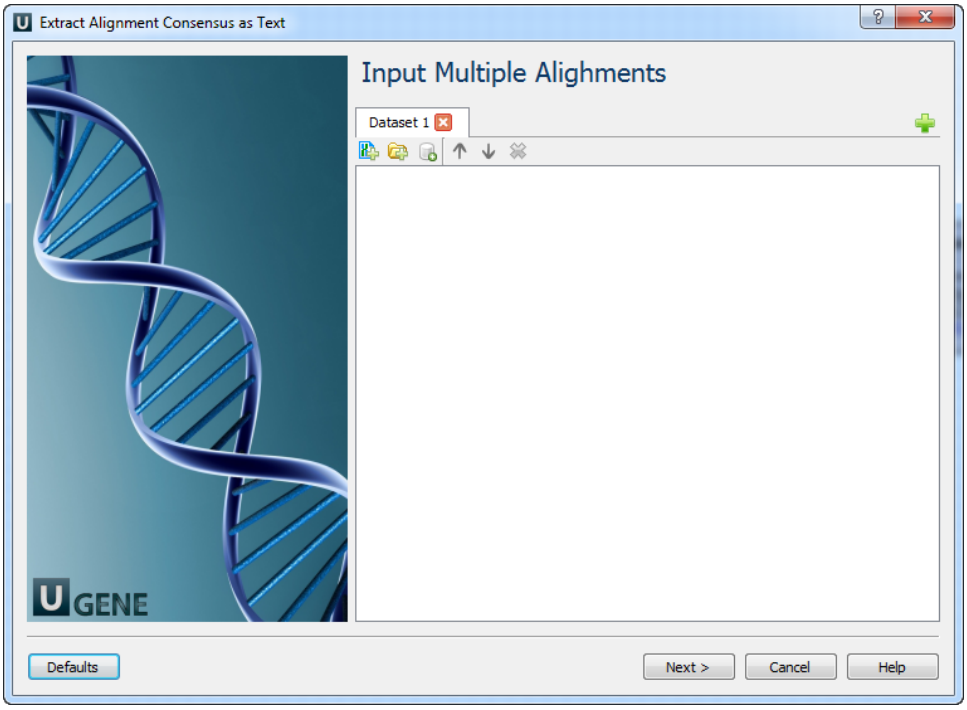
The workflow looks as follows:



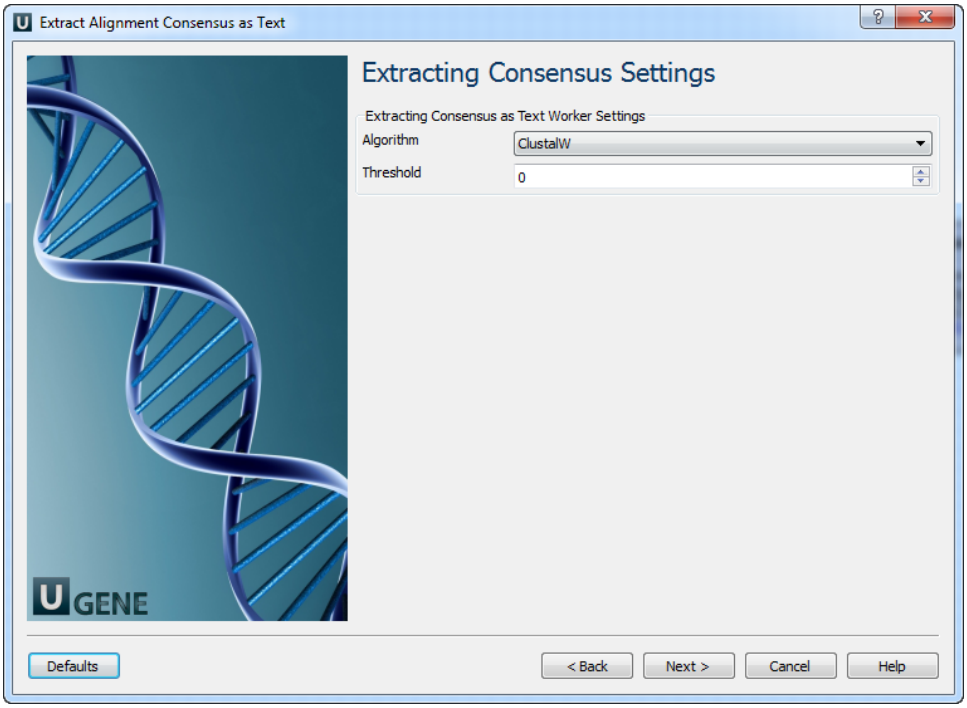
Workflow Wizard

The wizard has 3 pages.

1. Input Multiple Alignments: On this page you must input multiple alignments file(s).



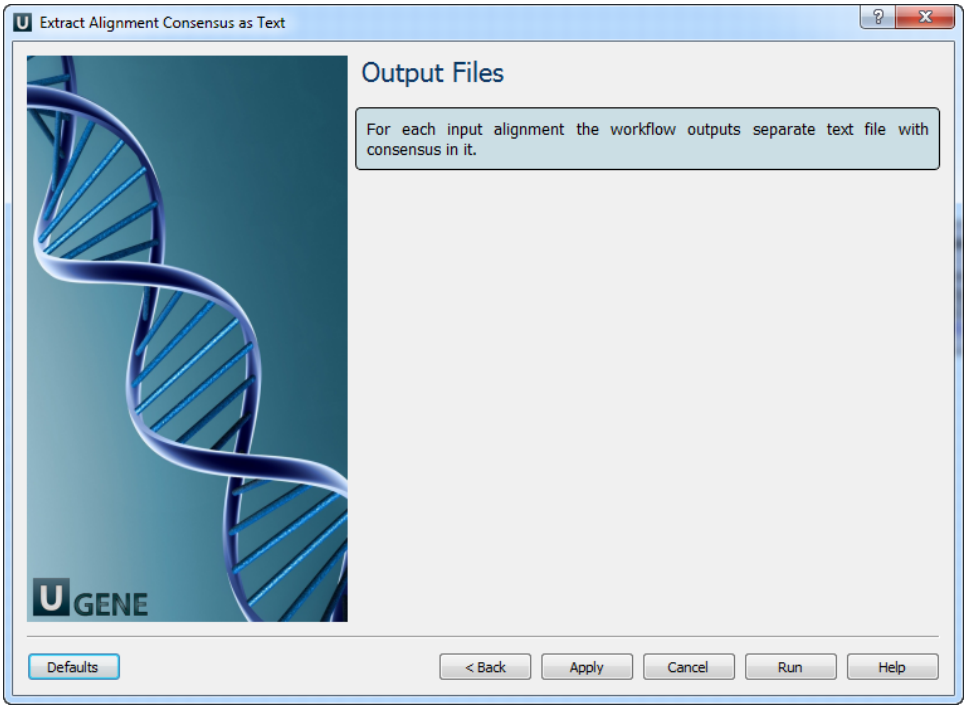
2. Extracting Consensus Settings: On this page you can modify extracting settings.



The following parameters are available:

Algorithm	The algorithm of consensus extracting.
Threshold	The threshold of the algorithm.

3. Output Files: For each input alignment the workflow outputs separate sequence file with consensus in it.



Conversions

- Convert "seq/qual" Pair to FASTQ
- Convert Alignments to ClustalW
- Convert UQL Schema Results to Alignment
- Convert Sequence to Genbank

Convert "seq/qual" Pair to FASTQ

This workflow allows to add PHRED quality scores to the sequence and save output to Fastq. For example, one can read a Fasta file, import PHRED quality values from corresponding qualities file and export the result to Fastq.

How to Use This Sample

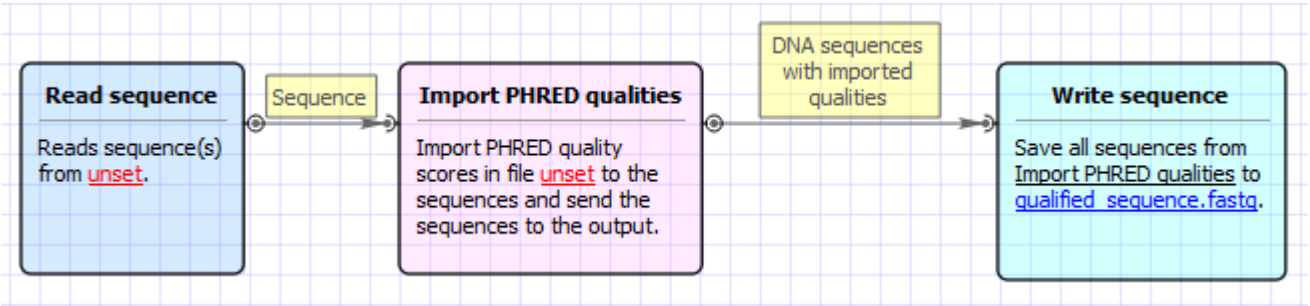
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Convert "seq/qual" Pair to FASTQ" can be found in the "Conversions" section of the Workflow Designer samples.

Workflow Image

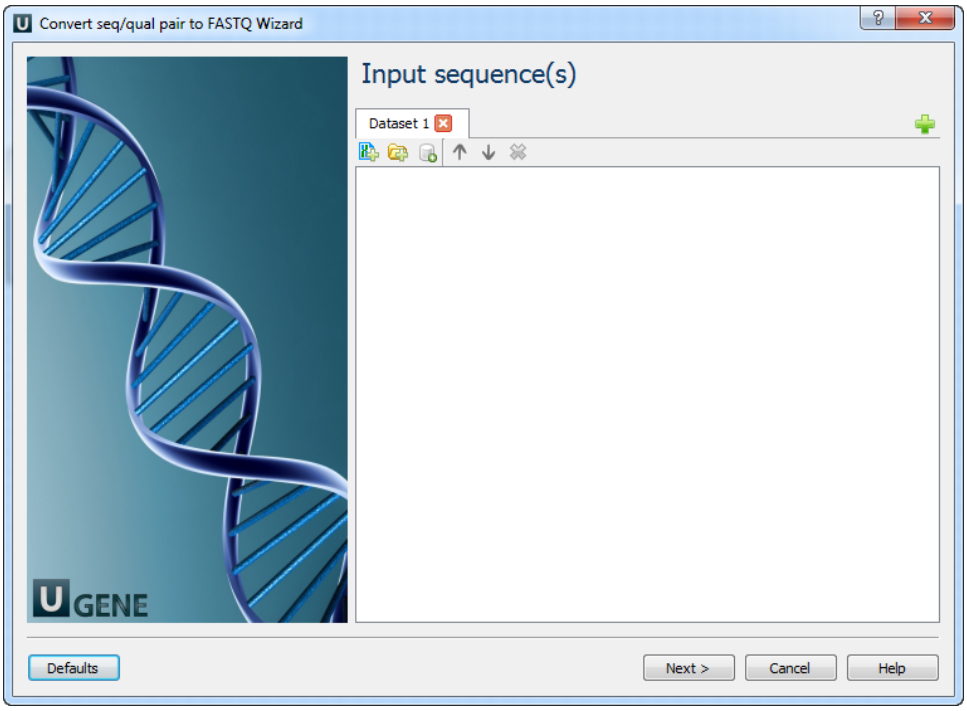
The workflow looks as follows:



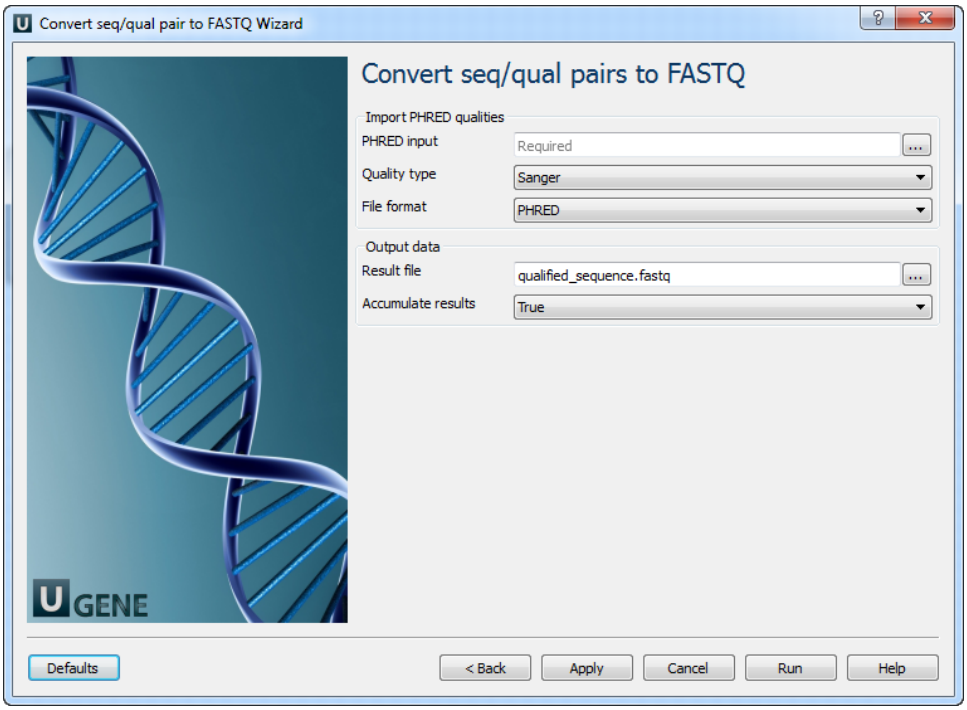
Workflow Wizard

The wizard has 2 pages.

1. Input Sequence(s): On this page you must input sequences(s).



2. Convert "seq/qual" Pair to FASTQ: On this page you can modify converting and output settings.



The following parameters are available:

PHRED input	Path to file with PHRED quality scores.
Quality type	Choose method to encode quality scores.
File format	Quality values can be in specialized FASTA-like PHRED qual format or encoded similar as in FASTQ files.
Result file	Location of output data file. If this attribute is set, slot "Location" in port will not be used.

Accumulate results	Accumulate all incoming data in one file or create separate files for each input.In the latter case, an incremental numerical suffix is added to the file name.

Convert Alignments to ClustalW

This workflow converts multiple alignment file(s) of any format to ClustalW document(s). If source file is a sequence format (e.g. FASTA), all contained sequences are added to the result alignment. Yet no real alignment is performed, this particular workflow illustrates pure data format conversion.

How to Use This Sample

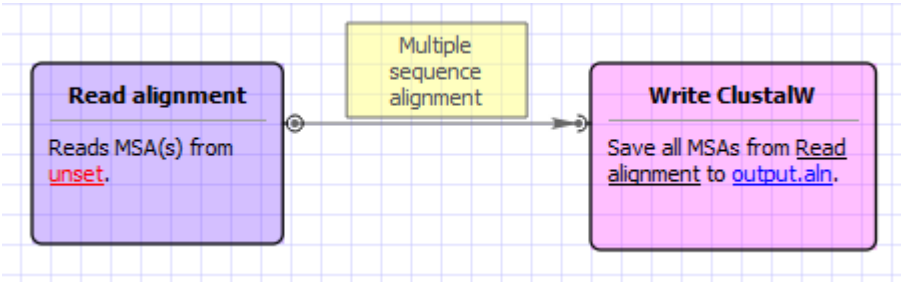
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Convert Alignments to ClustalW" can be found in the "Conversions" section of the Workflow Designer samples.

Workflow Image

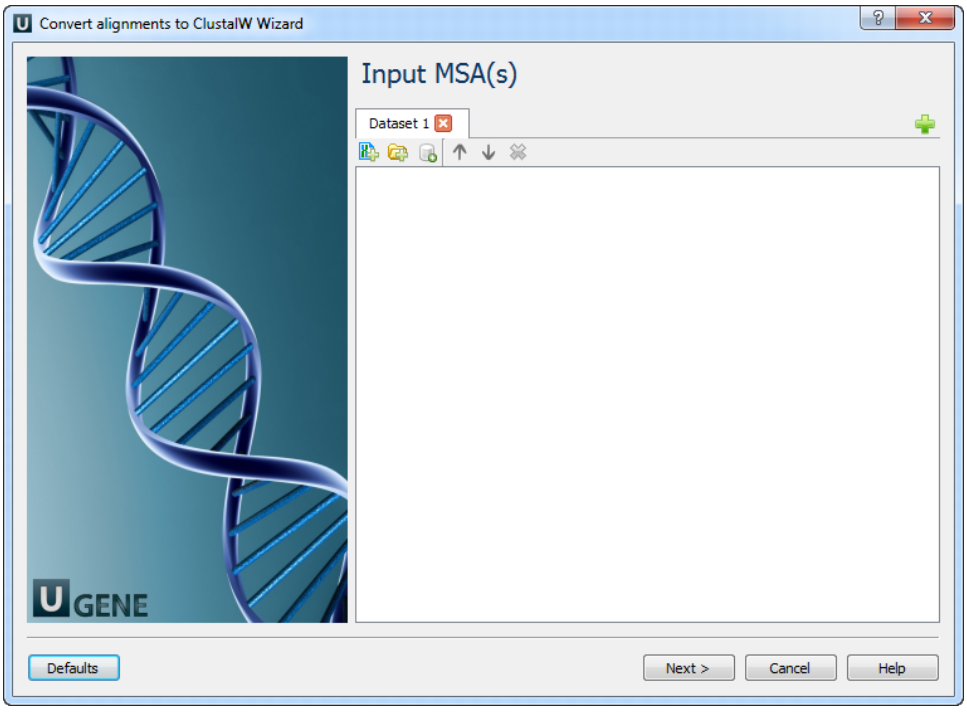
The workflow looks as follows:



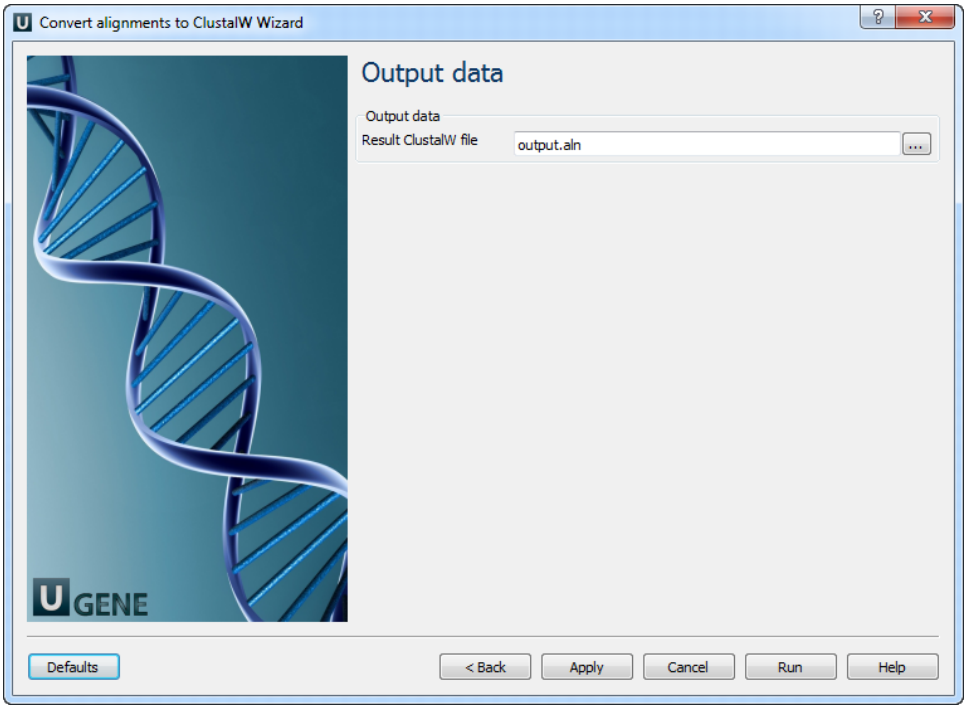
Workflow Wizard

The wizard has 2 pages.

1. Input MSA(s): On this page you must input MSA(s).



2. Output data: On this page you can modify output settings.



Convert UQL Schema Results to Alignment

This schema allows to analyze sequence with Query and save results as alignment of selected features.

How to Use This Sample

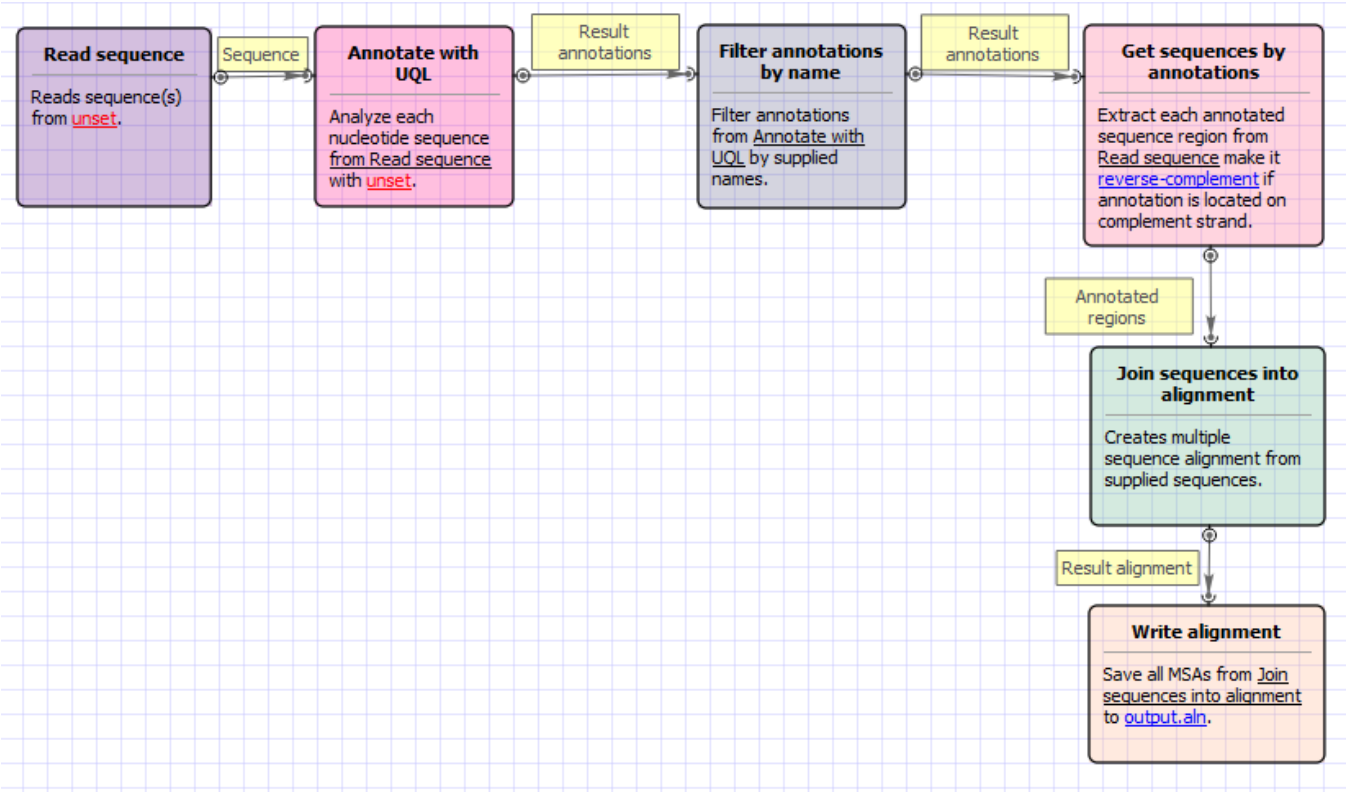
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Convert UQL Schema Results to Alignment" can be found in the "Conversions" section of the Workflow Designer samples.

Workflow Image

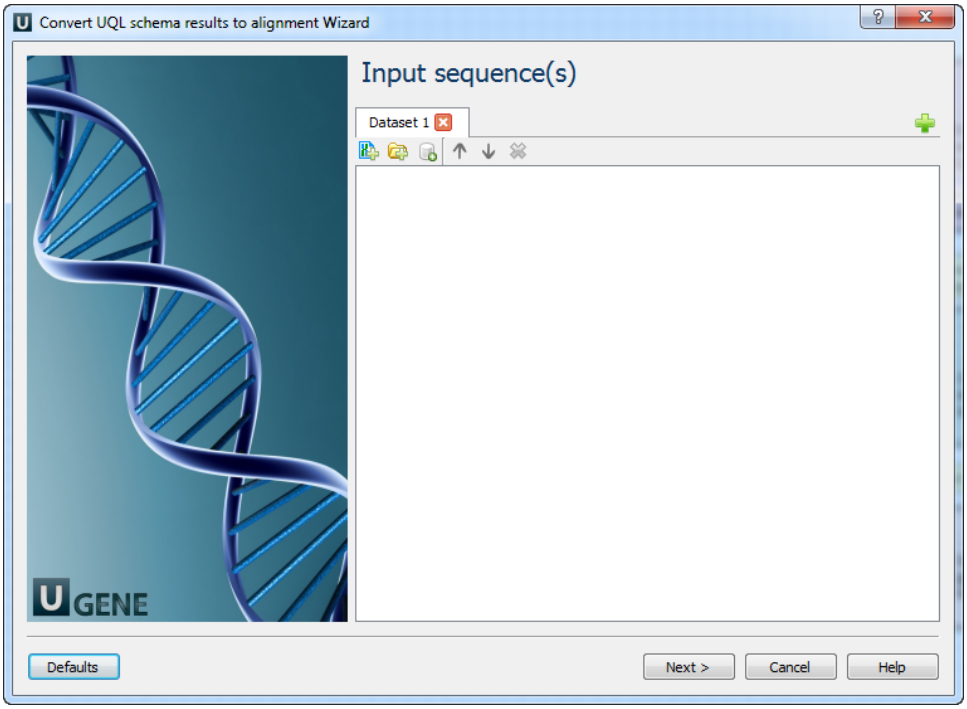
The workflow looks as follows:



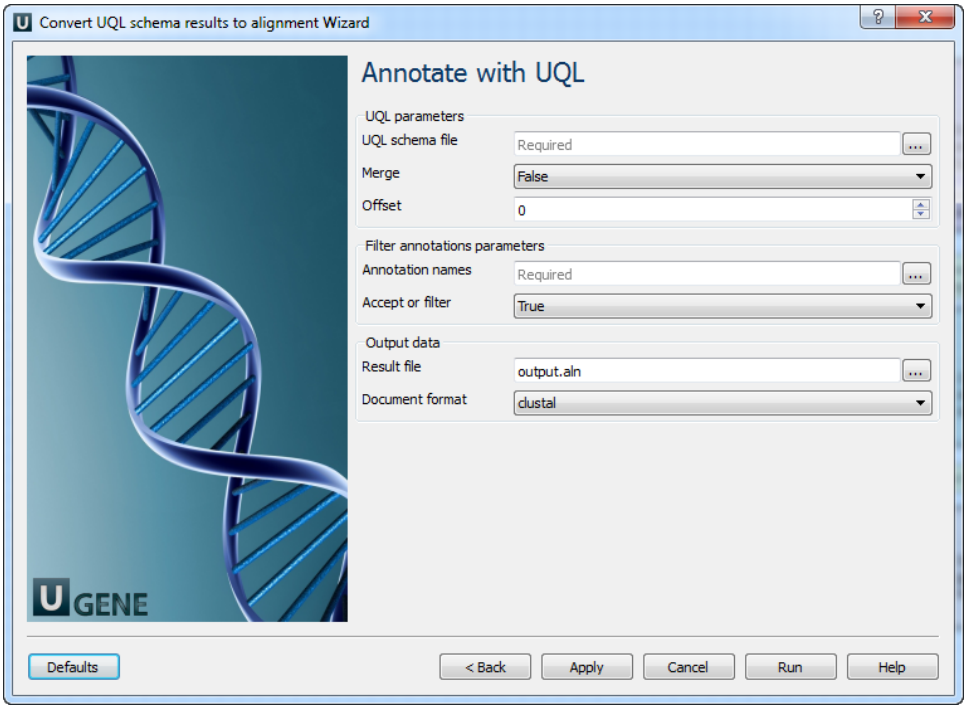
Workflow Wizard

The wizard has 2 pages.

1. Input sequence(s): On this page you must input sequence(s).



2. Annotate with UQL: On this page you can modify annotation and output settings.



The following parameters are available:

UQL schema file	Schema file.
Merge	Merges regions of each result into single annotation if true.
Offset	Specifies left and right offsets for merged annotation (if 'Merge' parameter is set to true).
Annotation names	File with annotation names, separated with whitespaces or list of annotation names which will be accepted or filtered. Use space as the separator.
Accept or filter	Selects the name filter: accept specified names or accept all except specified.
Result file	Location of output data file. If this attribute is set, slot "Location" in port will not be used.
Document format	Document format of output file.

Convert Sequence to Genbank

This workflow converts sequence file(s) of any format (including PDB, aligments etc) to Genbank document(s). If source format supports annotations, they are also saved as feature tables in target file. Sequence meta-information (accessions etc) is preserved as well.

How to Use This Sample

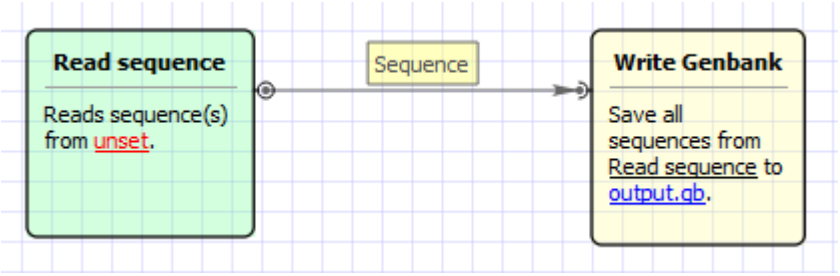
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Convert Sequence to Genbank" can be found in the "Conversions" section of the Workflow Designer samples.

Workflow Image

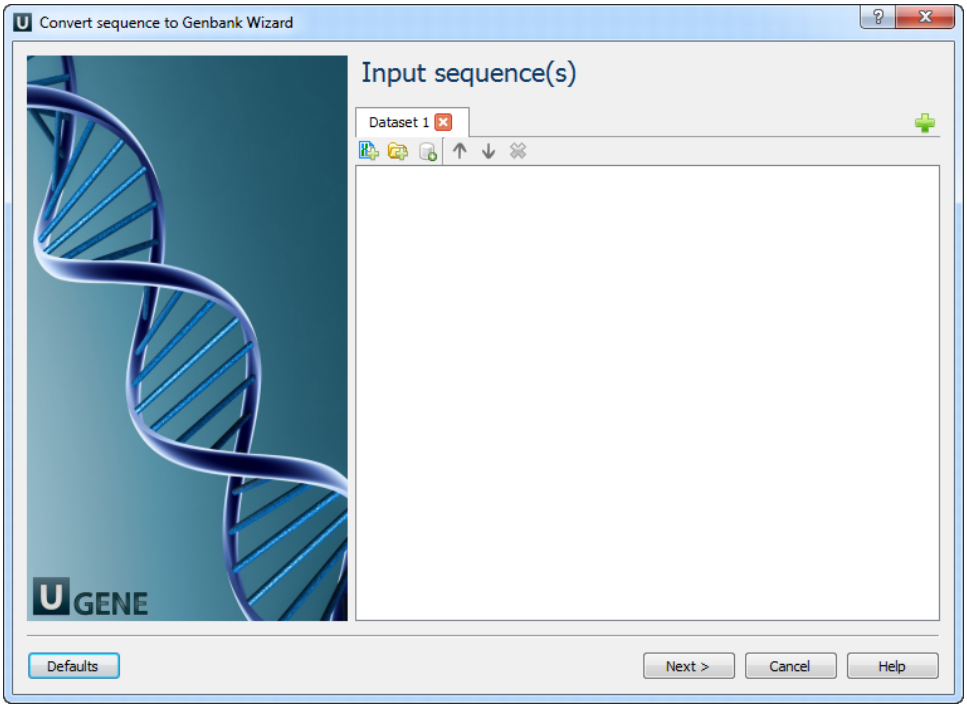
The workflow looks as follows:



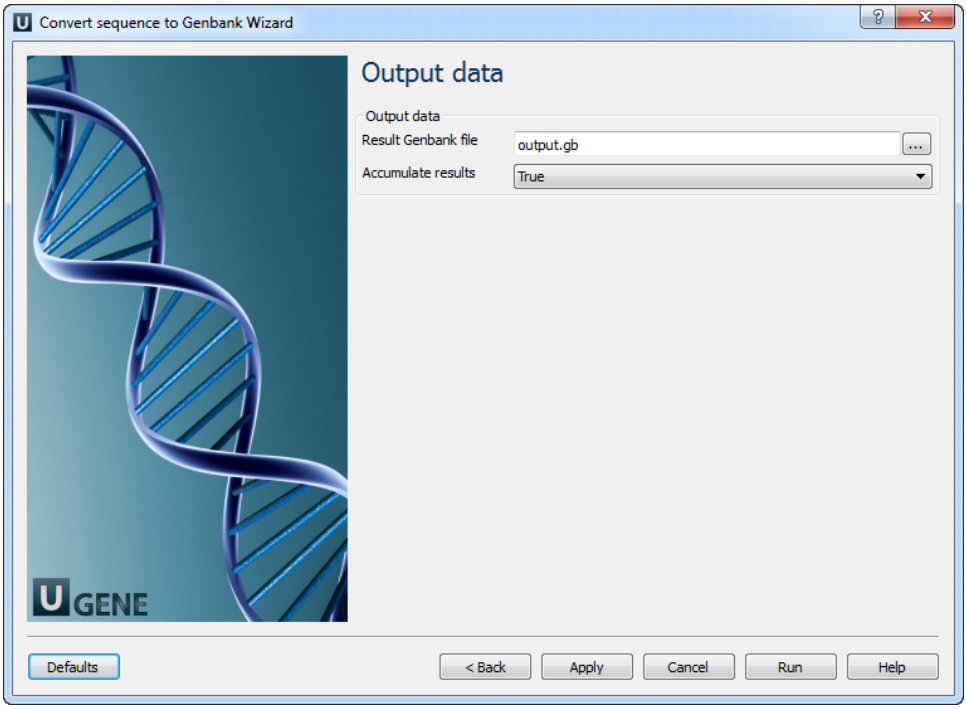
Workflow Wizard

The wizard has 2 pages.

1. Input sequence(s): On this page you must input sequence(s).



2. Output data: On this page you can modify output settings.



The following parameters are available:

Result Genbank file	Location of output data file. If this attribute is set, slot "Location" in port will not be used.
Accumulate results	Accumulate all incoming data in one file or create separate files for each input.In the latter case, an incremental numerical suffix is added to the file name.

Custom Elements

- CASAVA FASTQ Filter
- FASTQ Trimmer
- Dump Sequence Info
- LinkData Fetch
- Quality Filter

CASAVA FASTQ Filter

Reads in FASTQ file produced by CASAVA 1.8 contain 'N' or 'Y' as a part of an idetifier. 'Y' if a read if filtered, 'N' if the read if the read is not filtered. The workflow cleans up the filtered reads.

How to Use This Sample

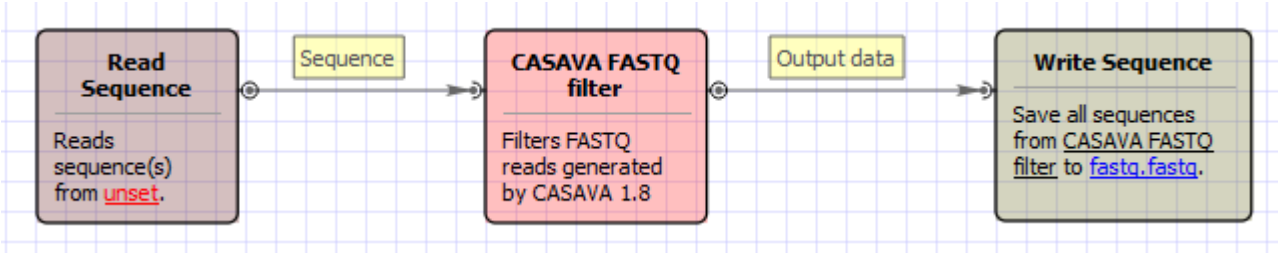
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "CASAVA FASTQ Filter" can be found in the "Custom Elements" section of the Workflow Designer samples.

Workflow Image

The workflow looks as follows:



FASTQ Trimmer

The workflow scans each input sequence from the end to find the first position where the quality is greater or equal to the minimum quality threshold. Then it trims the sequence to that position. If a the whole sequence has quality less than the threshold or the length of the output sequence less than the minimum length threshold then the sequence is skipped.

How to Use This Sample

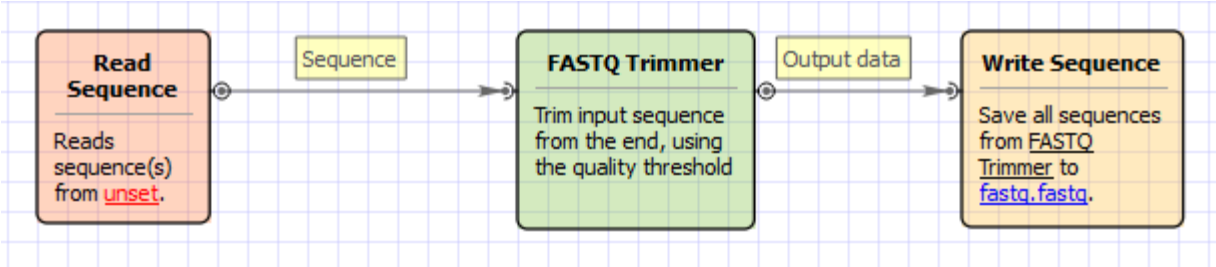
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "FASTQ Trimmer" can be found in the "Custom Elements" section of the Workflow Designer samples.

Workflow Image

The workflow looks as follows:



Dump Sequence Info

This workflow dump sequence name and sequence size to output for all incoming sequences.

How to Use This Sample

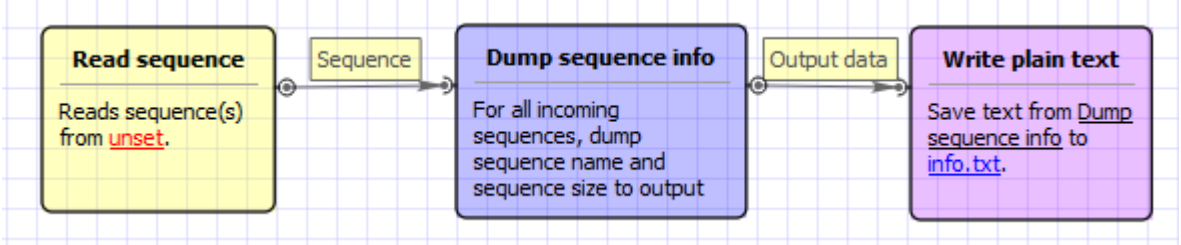
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Dump Sequence Info" can be found in the "Custom Elements" section of the Workflow Designer samples.

Workflow Image

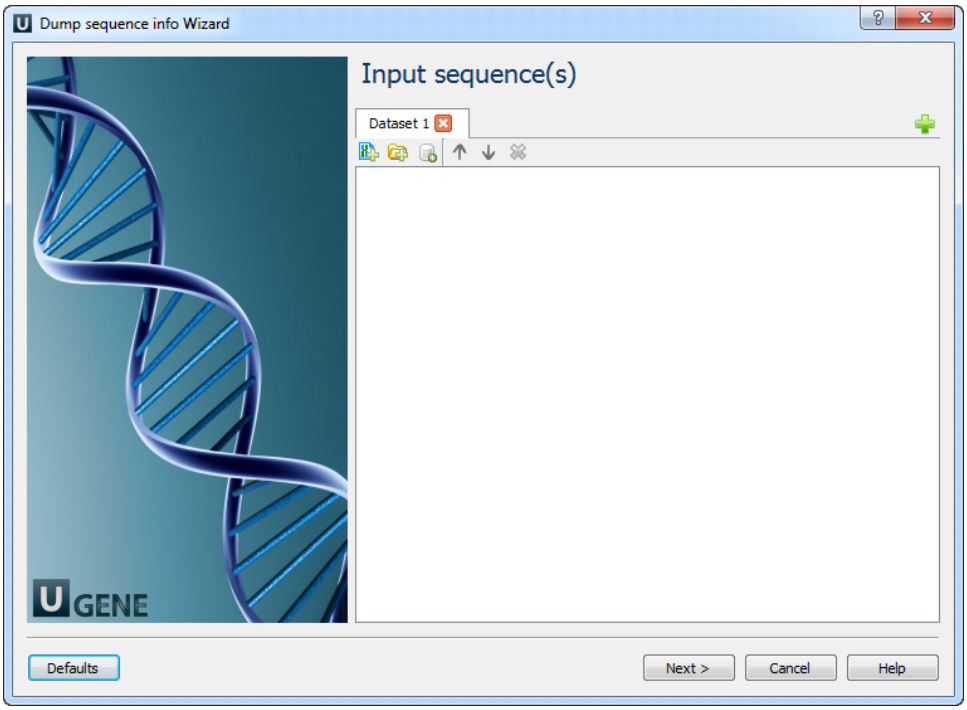
The workflow looks as follows:



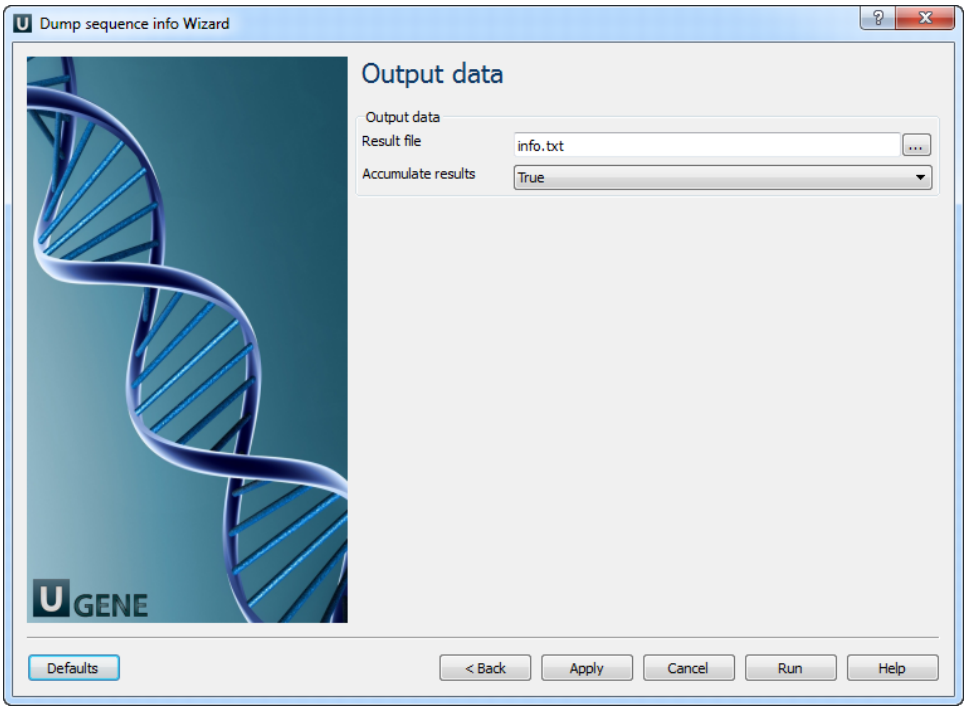
Workflow Wizard

The wizard has 2 pages.

1. Input sequence(s): On this page you must input sequence(s).



2. Output data: On this page you can modify output settings.



The following parameters are available:

Result file	Location of output data file. If this attribute is set, slot "Location" in port will not be used.
Accumulate results	Accumulate all incoming data in one file or create separate files for each input.In the latter case, an incremental numerical suffix is added to the file name.

LinkData Fetch

This workflow fetches sequence from LinkData by specified work ID, filename, subject ID, property ID and writes result in file in FASTA format.

How to Use This Sample

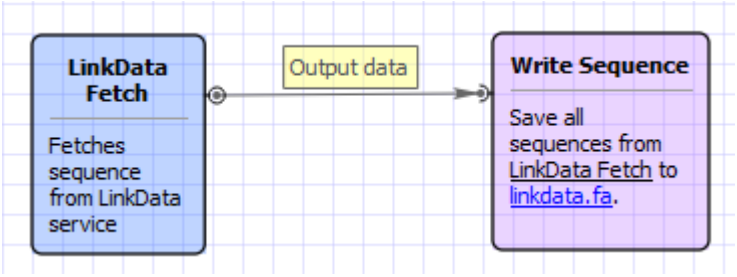
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "LinkData Fetch" can be found in the "Custom Elements" section of the Workflow Designer samples.

Workflow Image

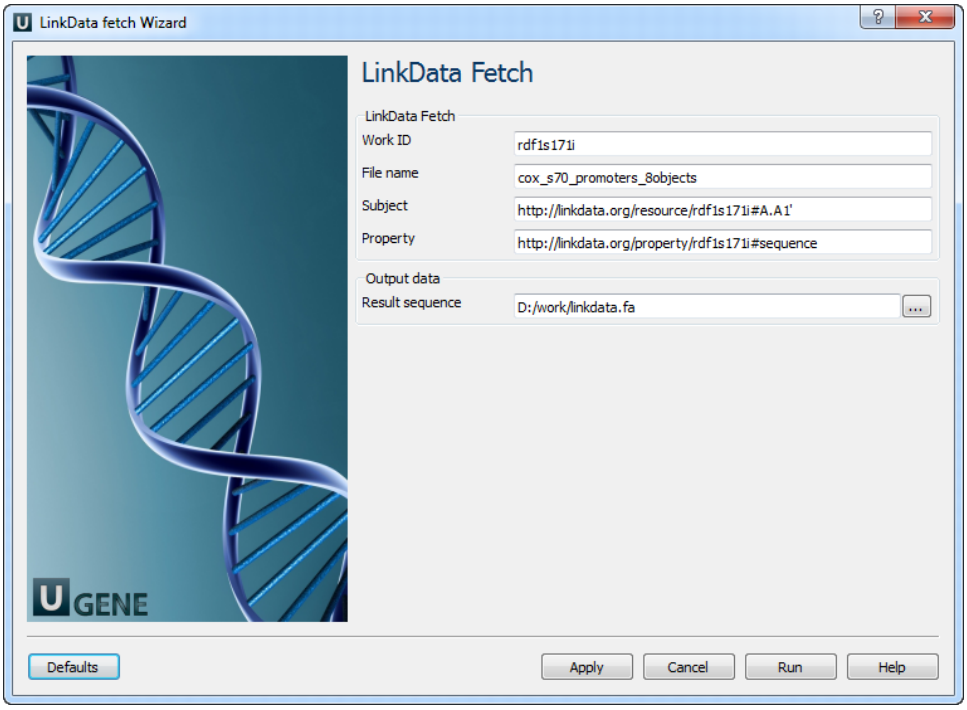
The workflow looks as follows:



Workflow Wizard

The wizard has 1 page.

1. LinkData Fetch: On this page you can modify LinkData and output settings.



The following parameters are available:

Work ID	Work ID
File name	File name
Subject	Subject

Property	Property
Result sequence	Location of output data file. If this attribute is set, slot "Location" in port will not be used.

Quality Filter

This workflow filters sequences with quality >= than parameter "quality" and writes result in file in FASTQ format.

How to Use This Sample

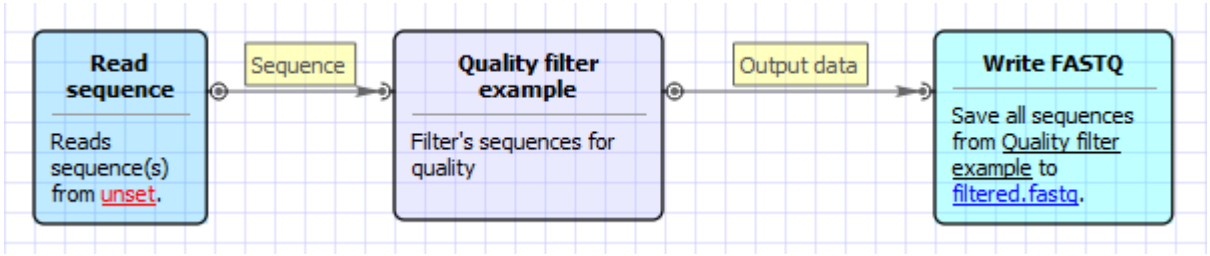
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Quality Filter" can be found in the "Custom Elements" section of the Workflow Designer samples.

Workflow Image

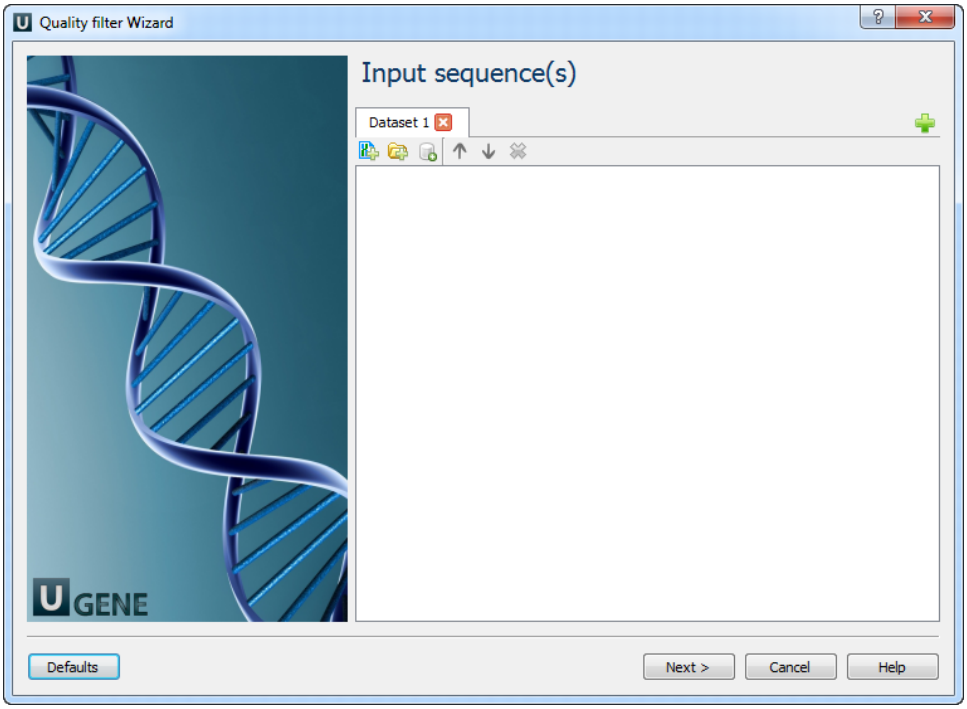
The workflow looks as follows:



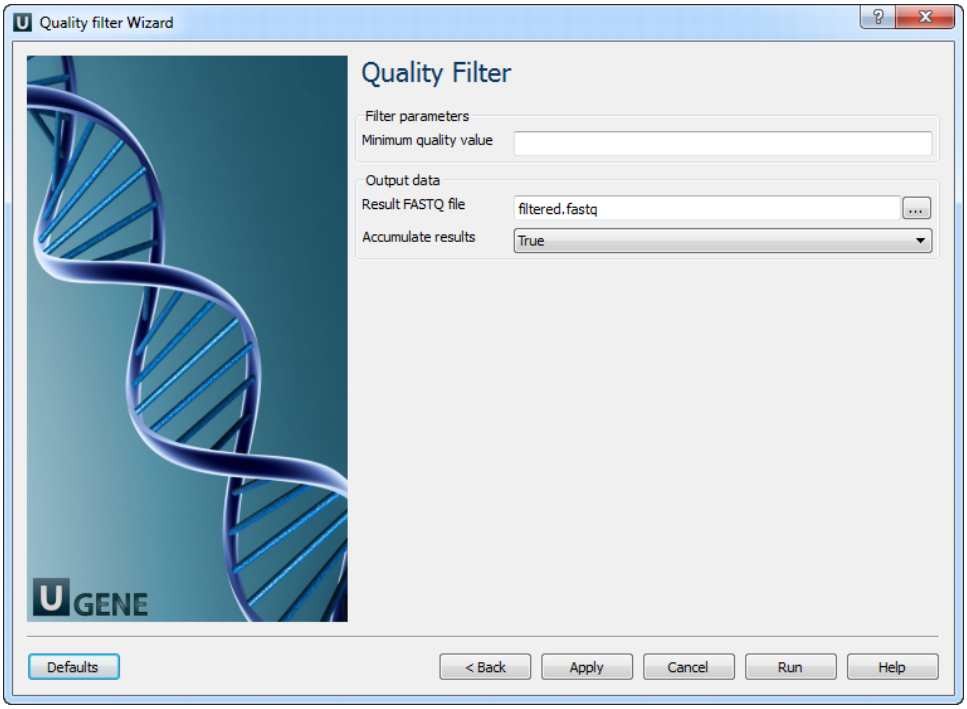
Workflow Wizard

The wizard has 2 pages.

1. Input sequence(s): On this page you must input sequence(s).



2. Quality Filter: On this page you can modify quality filter and output settings.



The following parameters are available:

Minimum quality value	Minimum quality value
Result FASTQ file	Location of output data file. If this attribute is set, slot "Location" in port will not be used.
Accumulate results	Accumulate all incoming data in one file or create separate files for each input.In the latter case, an incremental numerical suffix is added to the file name.

Data Marking

- Marking by Annotation Number
- Marking by Length

Marking by Annotation Number

This sample describes how to identify sequences with the specified number of annotations.

First, the schema reads sequences input by a user. Then, each sequence is marked either with the "Good" or with the "Rest" mark, depending on the number of the sequence annotations. After marking, the sequences are filtered by the marks. And finally, the filtered sequences are written into files, specified by a user.

By default, a sequence with 1 or more annotations is marked as "Good". You can configure this value in the *Sequence Marker* element parameters. Also, it is possible to set up the annotation names that should be taken into account.

How to Use This Sample

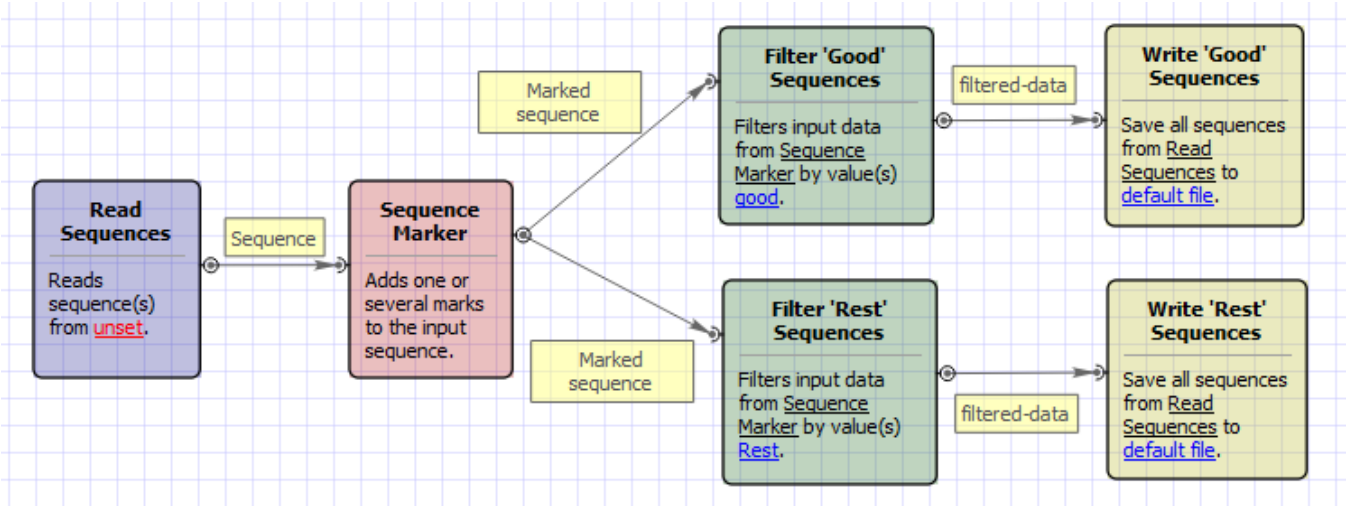
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Marking Sequences by Annotation Number" can be found in the "Data Marking" section of the Workflow Designer samples.

Workflow Image

The workflow looks as follows:



Marking by Length

This sample describes how to identify sequences with the specified length.

First, the workflow reads sequences input by a user. Then, each sequence is marked either with the “Short” or with the “Long” mark, depending on the sequence length. After marking, the sequences are filtered by the marks. And finally, the filtered sequences are written into files, specified by a user.

By default, a sequence with a length 200 or less bp is marked as “Short”. A sequence with a length of more than 200 bp is marked as “Long”. You can configure this value in the *Sequence Marker* element parameters.

How to Use This Sample

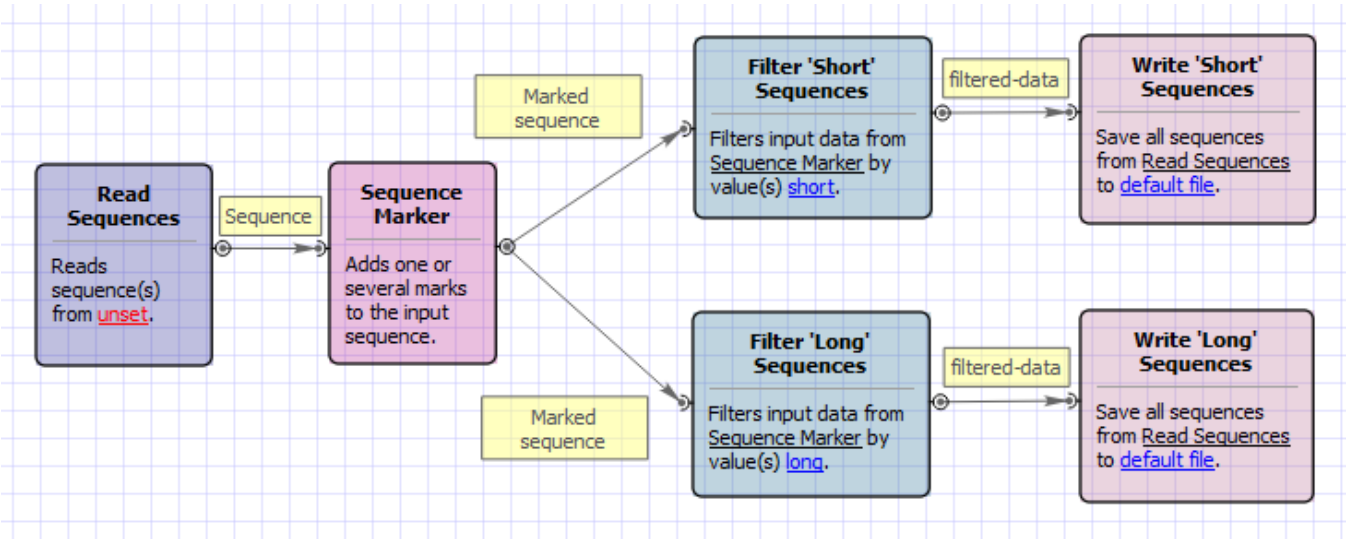
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Marking Sequences by Length" can be found in the "Data Marking" section of the Workflow Designer samples.

Workflow Image

The workflow looks as follows:



Data Merging

- Find Substrings in Sequences
- Merge Sequences and Shift Corresponding Annotations

- Search for TFBS

Find Substrings in Sequences

This sample workflow shows how to find substrings in input sequences, annotate them, and merge the found substring annotations with the original sequence annotations.

The steps of the workflow are these:

1. The workflow reads sequences from the input sequence files (e.g. GenBank). The input data may also contain the annotations, associated with the sequences.
2. The workflow reads text strings (patterns) from the input text files.
3. The data are multiplexed using the Multiplexer element. Multiplexing rule "1 to many" is used, so each input sequence is concatenated with each pattern. The concatenating results are sent to the *Find Substrings* element.
4. The *Find Substrings* element searches for the specified patterns in each sequence.
5. The next element Grouper merges annotations, read for the sequence in the *Read Sequence* element, with annotations, found for the sequence by the *Find Substrings* element. A sequence ID is used to group the appropriate sets of annotations.
6. And finally, the data are written to the output file ("substrings.gb" , by default).

How to Use This Sample

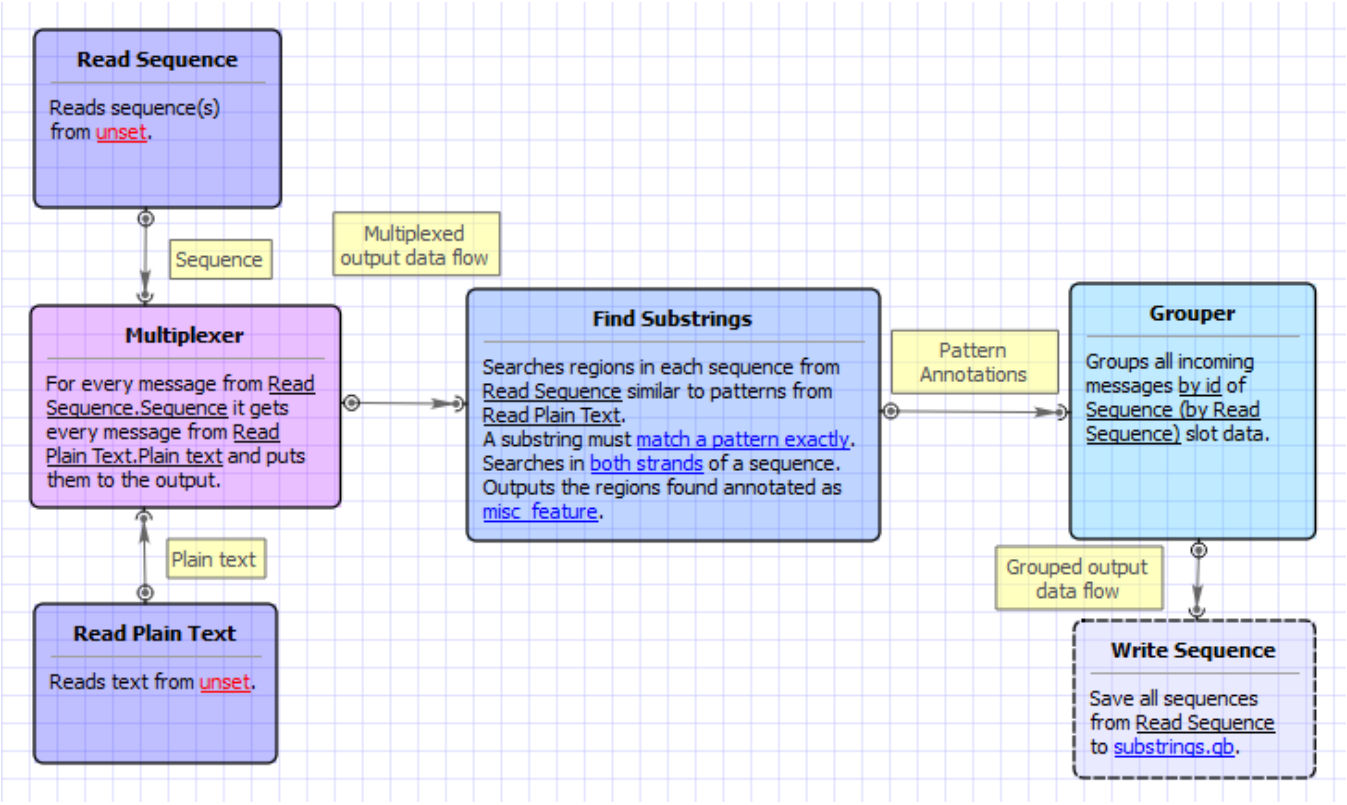
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Find Substrings at Sequences" can be found in the "Data Merging" section of the Workflow Designer samples.

Workflow Image

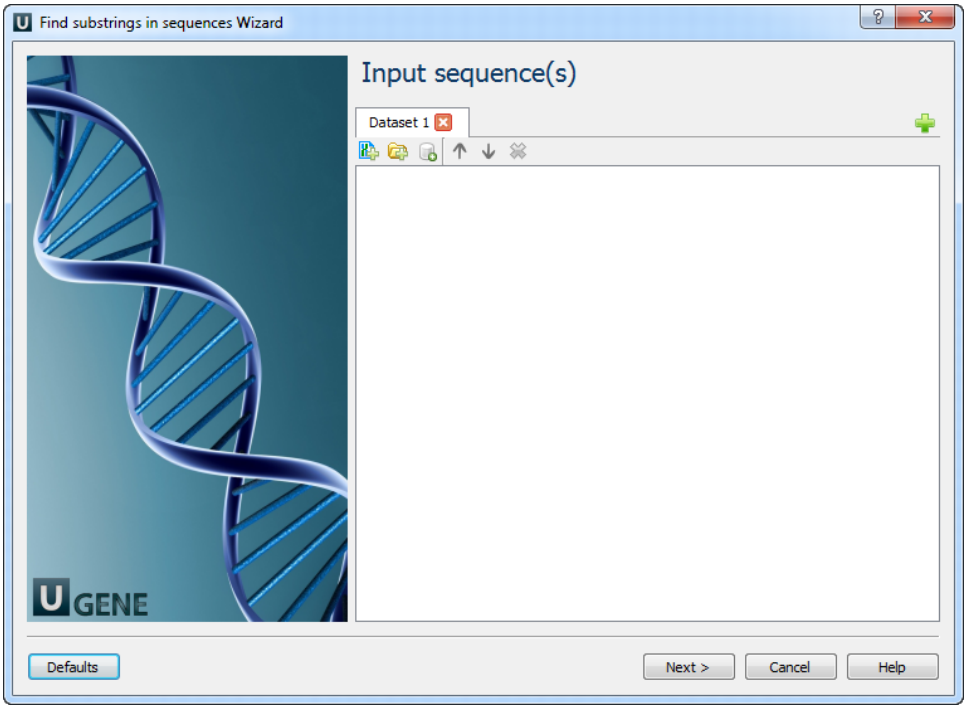
The workflow looks as follows:



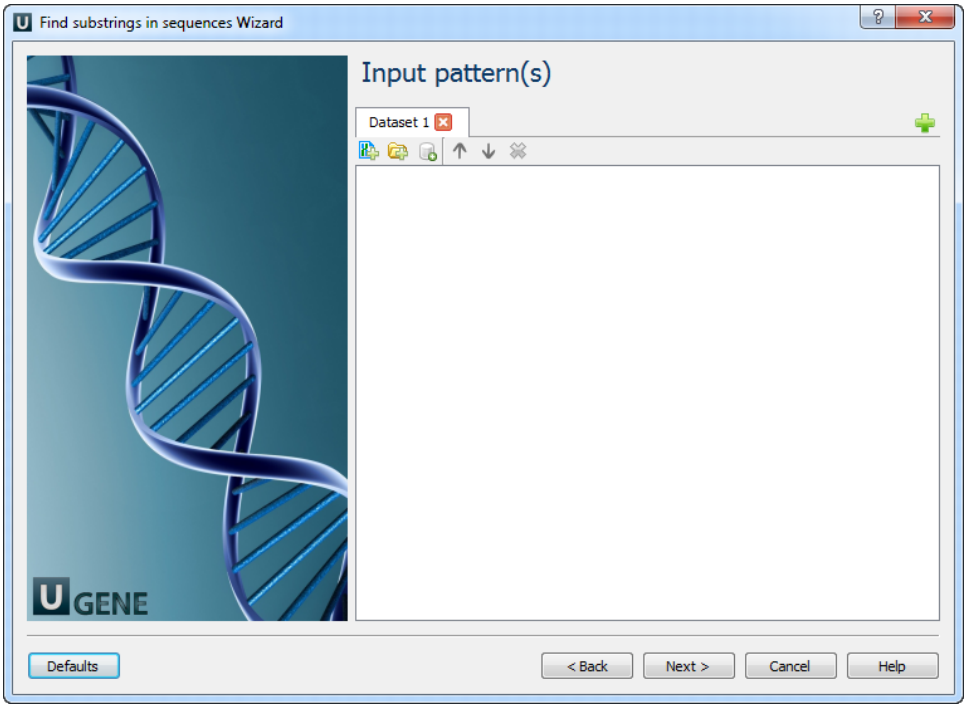
Workflow Wizard

The wizard has 3 pages.

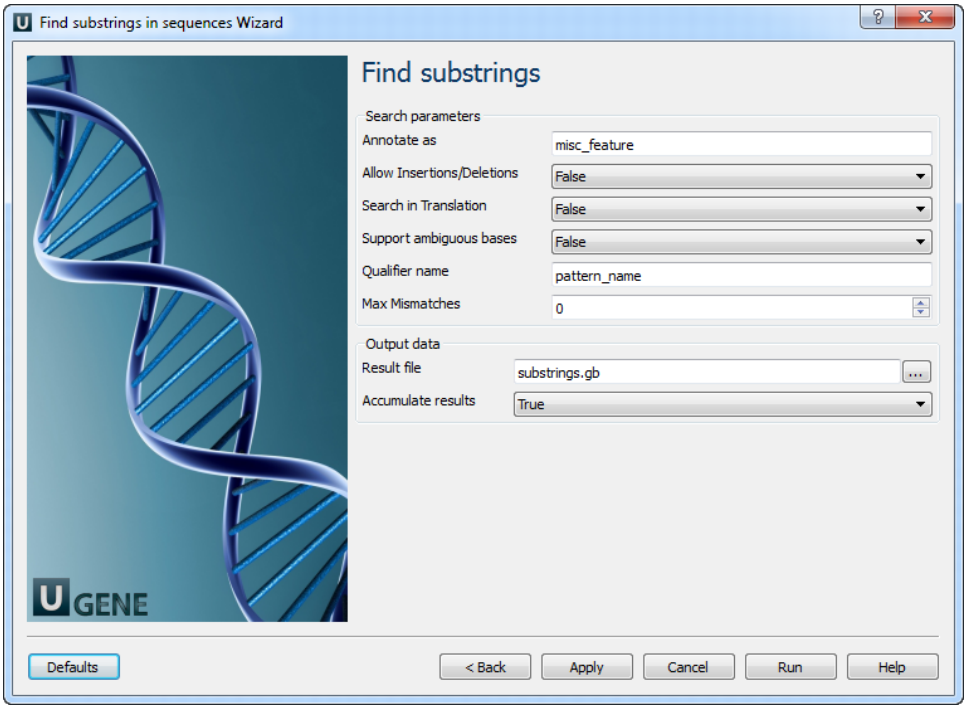
1. Input sequence(s): On this page you must input sequence(s).



2. Input pattern(s): On this page you must input pattern(s).



3. Find substrings: On this page you can modify search and output parameters.



The following parameters are available:

Annotate as	Name of the result annotations.
Allow Insertions /Deletions	Takes into account possibility of insertions/deletions when searching. By default substitutions are only considered.
Search in Translation	Translates a supplied nucleotide sequence to protein and searches in the translated sequence.
Support ambiguous bases	Performs correct handling of ambiguous bases. When this option is activated insertions and deletions are not considered.
Qualifier name	Name of qualifier in result annotations which is containing a pattern name.
Max Mismatches	Maximum number of mismatches between a substring and a pattern.
Result file	Location of output data file. If this attribute is set, slot "Location" in port will not be used.
Accumulate results	Accumulate all incoming data in one file or create separate files for each input.In the latter case, an incremental numerical suffix is added to the file name.

Merge Sequences and Shift Corresponding Annotations

This workflow describes how to merge sequences and manipulate with its annotations.

First, the workflow reads sequence(s) from file(s). Then, marks the input sequences with the sequence name marker. After marking the sequences are grouped by the marker. Sequences with equal markers are merged into one sequence. Annotations are shifted using the position of the corresponding sequence at the merged sequence. And finally, the grouped data are written into file, specified by a user.

By default, sequence is marked using the sequence name marker. You can configure this value in the *Marker* element parameters. Also, you can configure the *Grouper* element parameters.

How to Use This Sample

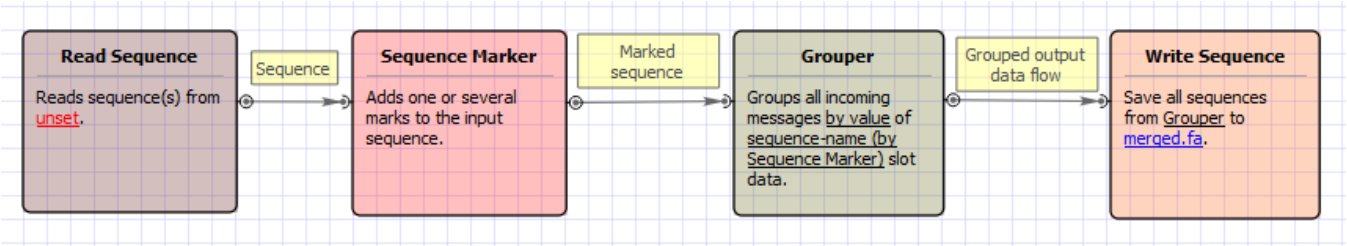
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Merge Sequences and Shift Corresponding Annotations" can be found in the "Data Merging" section of the Workflow Designer samples.

Workflow Image

The workflow looks as follows:



Search for TFBS

This sample shows how to search for transcription factor binding sites (TFBS) using two different approaches - weight matrices and SITECON models - and write the found TFBS annotations into one output file.

The workflow steps are these:

1. The workflow reads the input sequences.
2. Each sequence goes to the TFBS searching elements.
3. *Read Weight Matrix* reads the input weight matrices. *Read SITECON Model* reads the input SITECON models. The data are also transferred to the TFBS searching elements.
4. Each TFBS searching element produces the corresponding annotations.
5. After that the two annotation data flows are multiplexed into one data flow.
6. The multiplexed data and are written to the output file ("merged.gb", by default).

How to Use This Sample

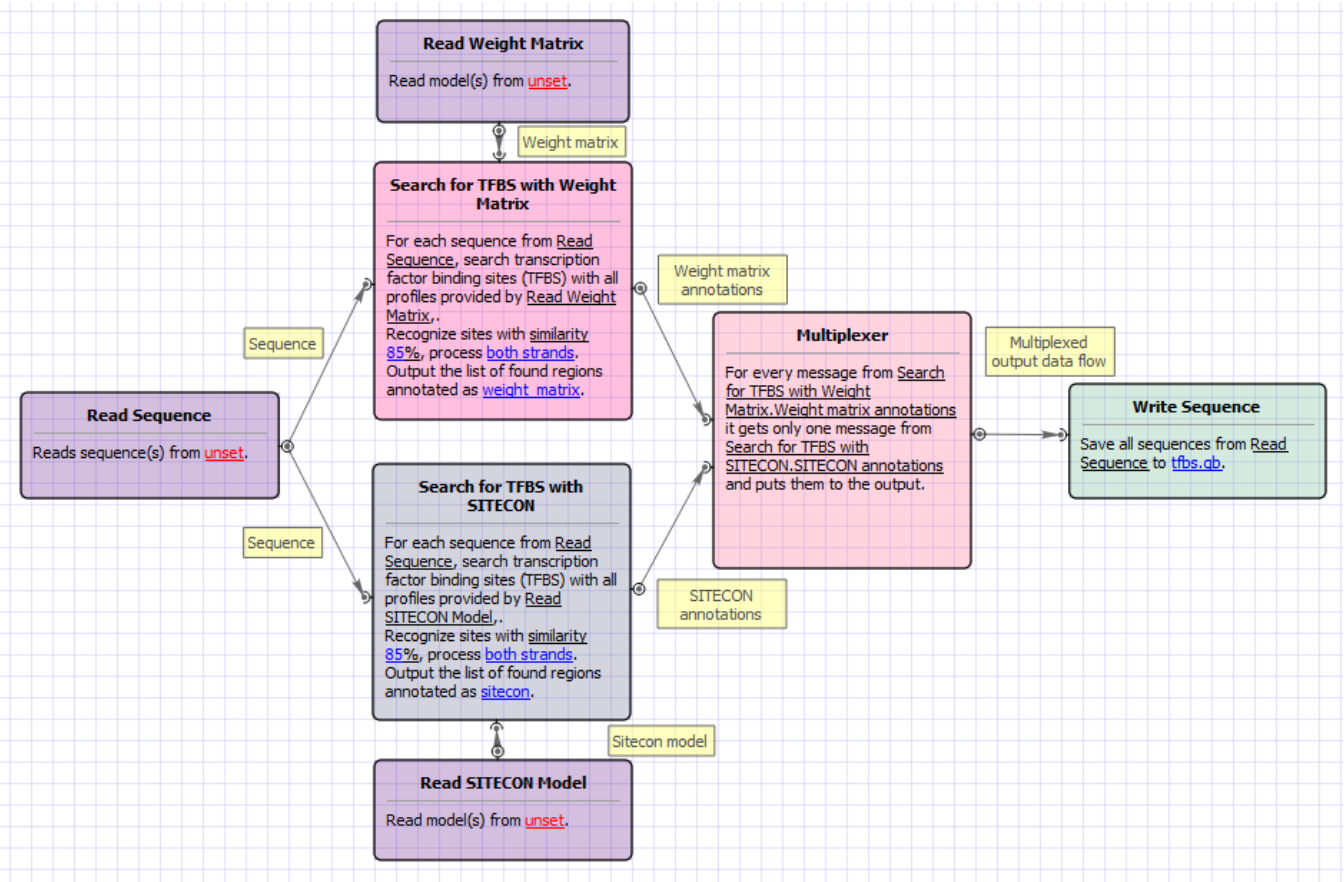
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Search for TFBS" can be found in the "Data Merging" section of the Workflow Designer samples.

Workflow Image

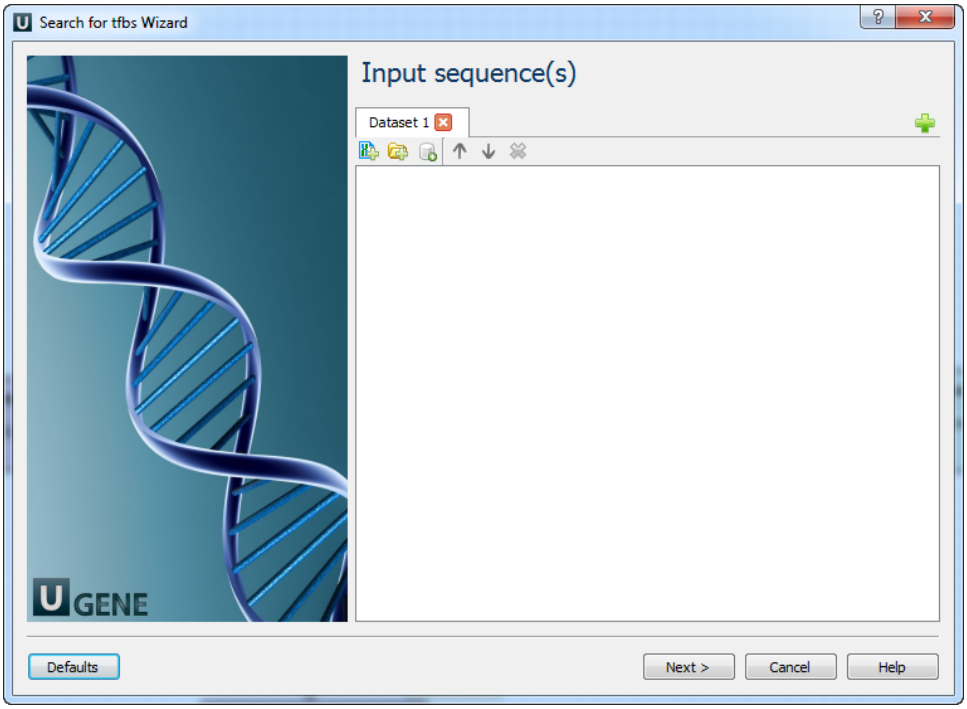
The workflow looks as follows:



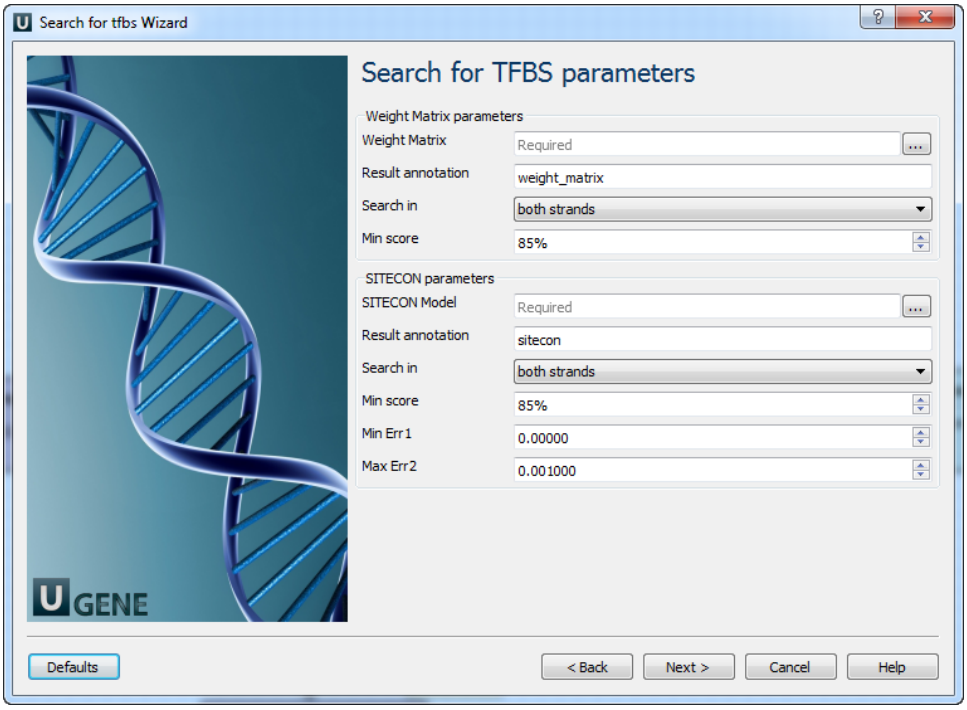
Workflow Wizard

The wizard has 3 pages.

1. Input sequence(s): On this page you must input sequence(s).



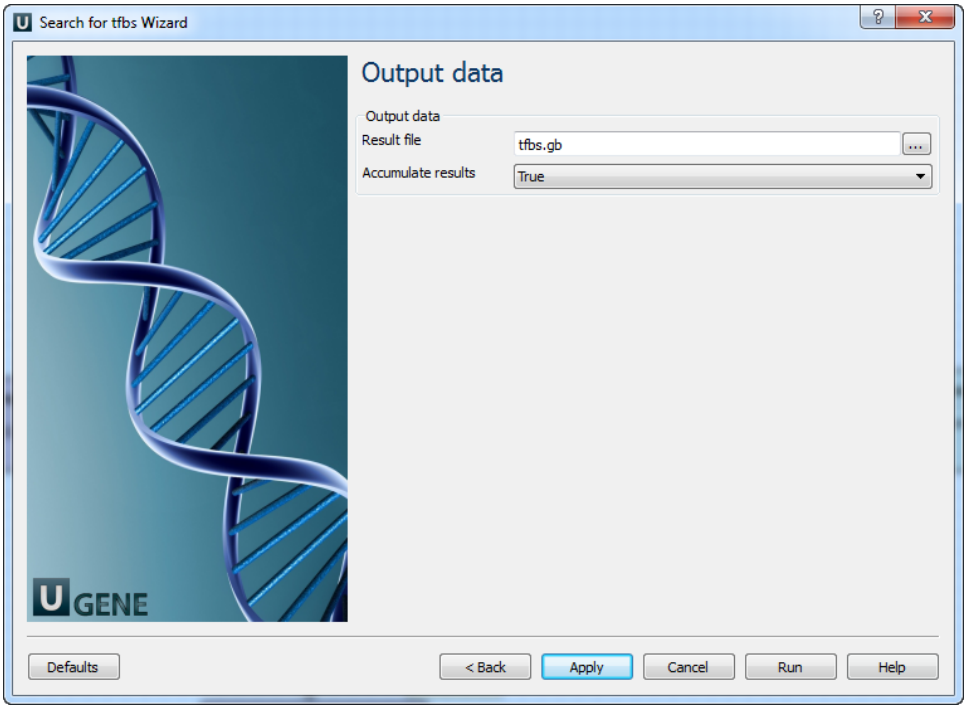
2. Search for TFBS parameters: On this page you can modify search for TFBS parameters.



The following parameters are available:

Weight Matrix	Semicolon-separated list of paths to the input files.
Result annotation	Annotation name for marking found regions.
Search in	Which strands should be searched: direct, complement or both.
Min score	Minimum score to detect transcription factor binding site
SITECON model	Semicolon-separated list of paths to the input files.
Result annotation	Annotation name for marking found regions.
Search in	Which strands should be searched: direct, complement or both.
Min score	Minimum score to detect transcription factor binding site
Min err1	Alternative setting for filtering results, minimal value of Error type I. Note that all thresholds (by score, by err1 and by err2) are applied when filtering results.
Max err2	Alternative setting for filtering results, max value of Error type II. Note that all thresholds (by score, by err1 and by err2) are applied when filtering results.

3. Output data: On this page you can modify output parameters.



The following parameters are available:

Result file	Location of output data file. If this attribute is set, slot "Location" in port will not be used.
Accumulate results	Accumulate all incoming data in one file or create separate files for each input.In the latter case, an incremental numerical suffix is added to the file name.

HMMER

- Build HMM from Alignment and test it
- Search Sequences with Profile HMM

Build HMM from Alignment and test it

This workflow builds a new profile HMM from input alignment, calibrates the HMM and saves to a file. Then runs a test HMM search over sample sequence and saves test results to Genbank file.

How to Use This Sample

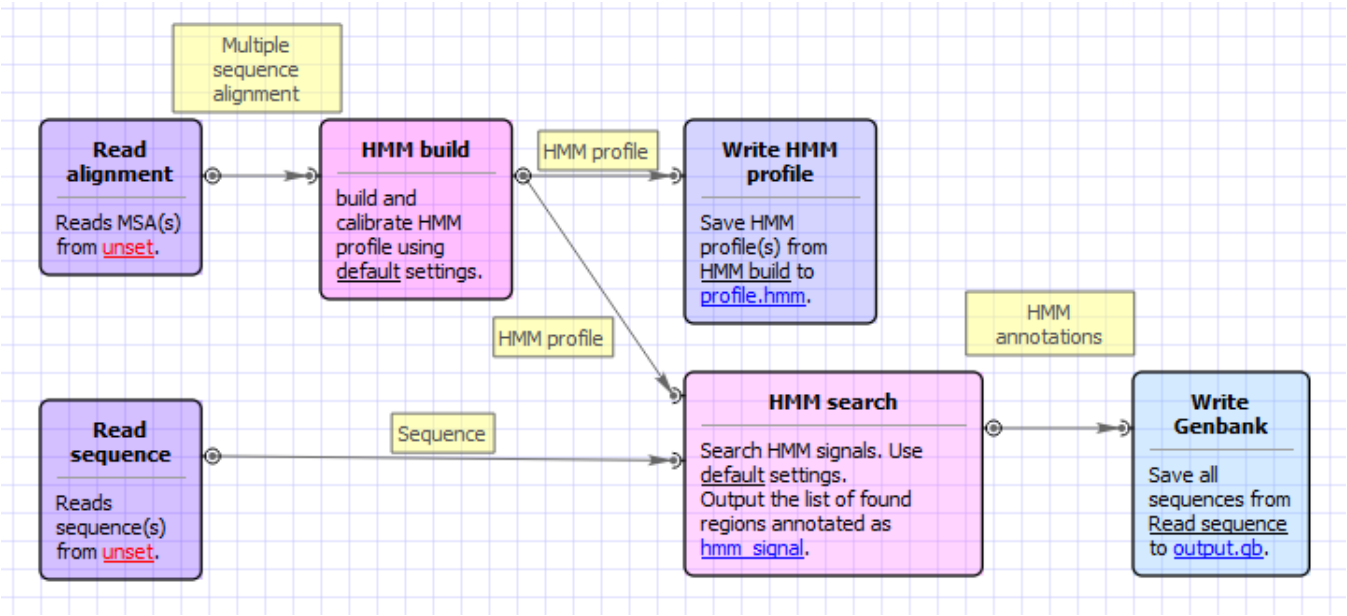
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Build HMM from Alignment and test it" can be found in the "HMMER" section of the Workflow Designer samples.

Workflow Image

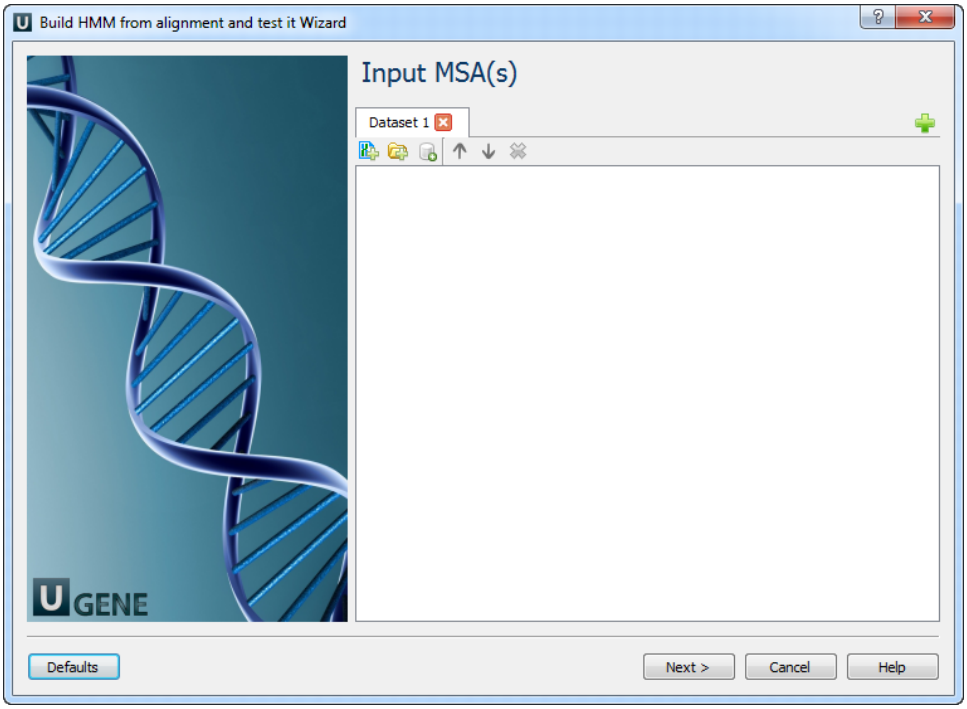
The workflow looks as follows:



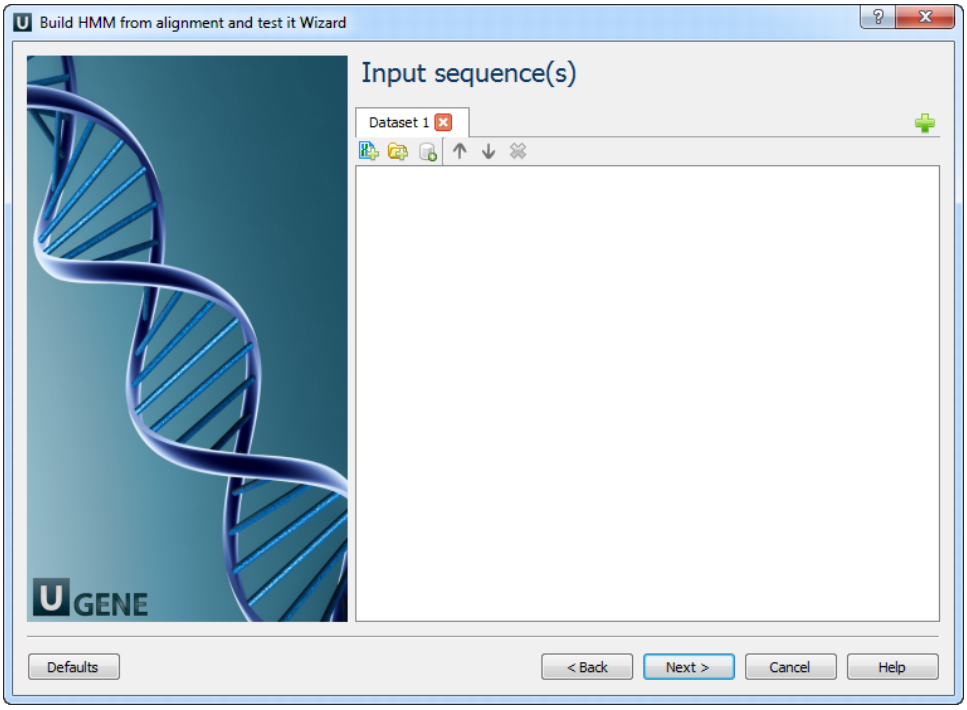
Workflow Wizard

The wizard has 4 pages.

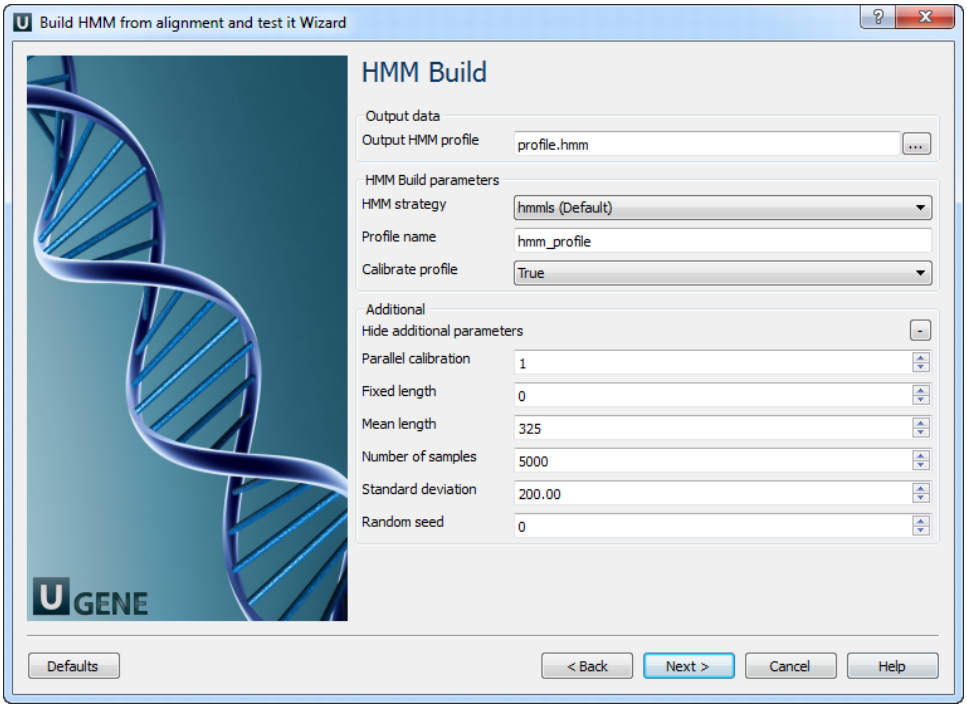
1. Input MSA(s): On this page you must input MSA(s).



2. Input sequence(s): On this page you must input sequence(s).



3. HMM build: On this page you can modify HMM build parameters.

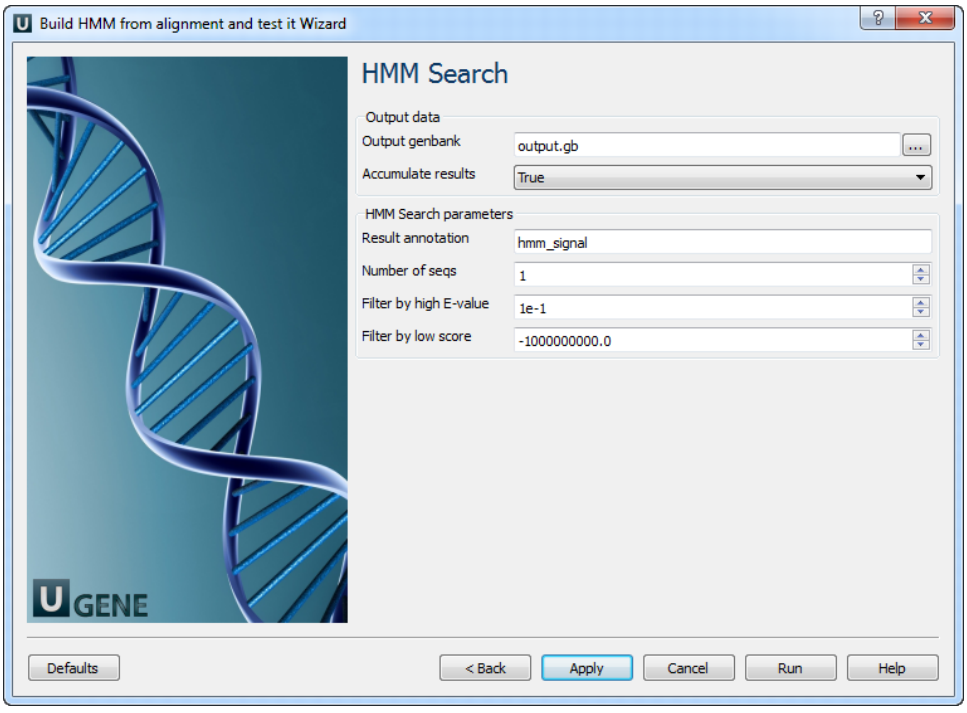


The following parameters are available:

Output HMM profile	Location of output data file. If this attribute is set, slot "Location" in port will not be used.
HMM strategy	Specifies kind of alignments you want to allow.
Profile name	Descriptive name of the HMM profile.

Calibrate profile	Enables/disables optional profile calibration. An empirical HMM calibration costs time but it only has to be done once per model, and can greatly increase the sensitivity of a database search.
Parallel calibration	Number of parallel threads that the calibration will run in.
Fixed length	Fix the length of the random sequences to , where is a positive (and reasonably sized) integer. The default is instead to generate sequences with a variety of different lengths, controlled by a Gaussian (normal) distribution.
Mean length	Mean length of the synthetic sequences, positive real number. The default value is 325.
Number of samples	Number of synthetic sequences. If is less than about 1000, the fit to the EVD may fail. Higher numbers of will give better determined EVD parameters. The default is 5000; it was empirically chosen as a tradeoff between accuracy and computation time.
Standard deviation	Standard deviation of the synthetic sequence length. A positive number. The default is 200. Note that the Gaussian is left-truncated so that no sequences have lengths
Random seed	The random seed, where is a positive integer. The default is to use time() to generate a different seed for each run, which means that two different runs of hmmcalibrate on the same HMM will give slightly different results. You can use this option to generate reproducible results for different hmmcalibrate runs on the same HMM.

4. HMM search: On this page you can modify HMM search and output parameters.



The following parameters are available:

Output genbank	Location of output data file. If this attribute is set, slot "Location" in port will not be used.
Accumulate results	Accumulate all incoming data in one file or create separate files for each input.In the latter case, an incremental numerical suffix is added to the file name.
Result annotation	A name of the result annotations.
Number of seqs	Calculate the E-value scores as if we had seen a sequence database of sequences.
Filter by high E-value	E-value filtering can be used to exclude low-probability hits from result.

Filter by low score	Score based filtering is an alternative to E-value filtering to exclude low-probability hits from result.

Search Sequences with Profile HMM

This workflow reads an HMM from a file and searches input sequences for significantly similar matches, saves found signals to a file. You can specify several input files for both HMM and sequences, the workflow will process Cartesian product of inputs. That is, each sequence will be searched with all specified HMMs in turn.

How to Use This Sample

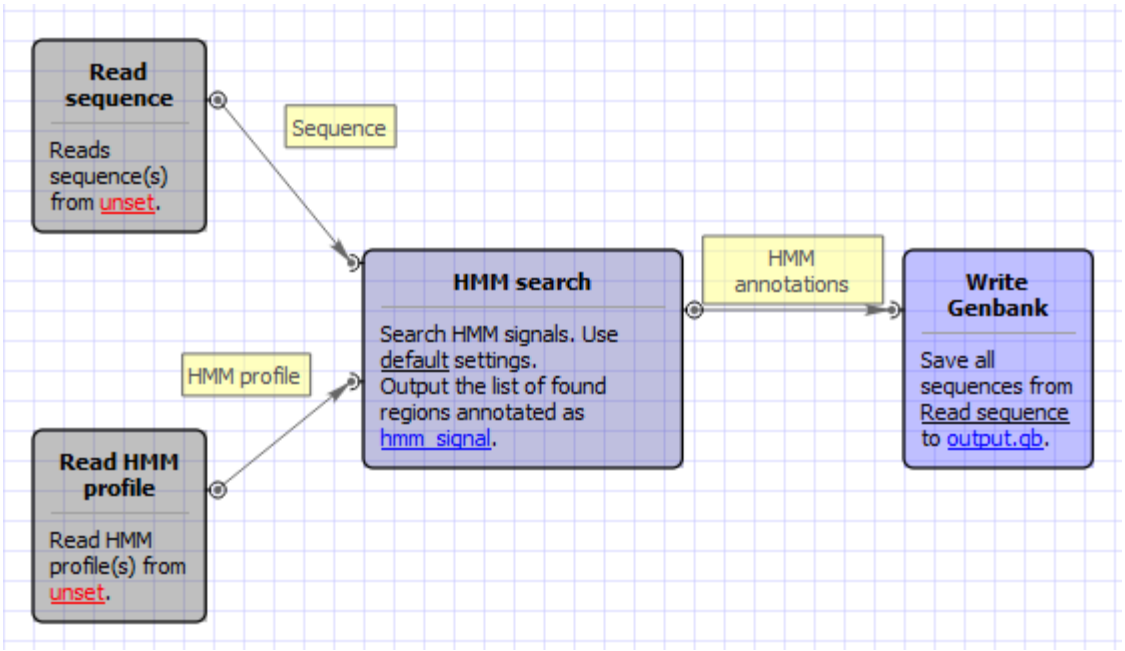
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Search Sequences with Profile HMM" can be found in the "HMMER" section of the Workflow Designer samples.

Workflow Image

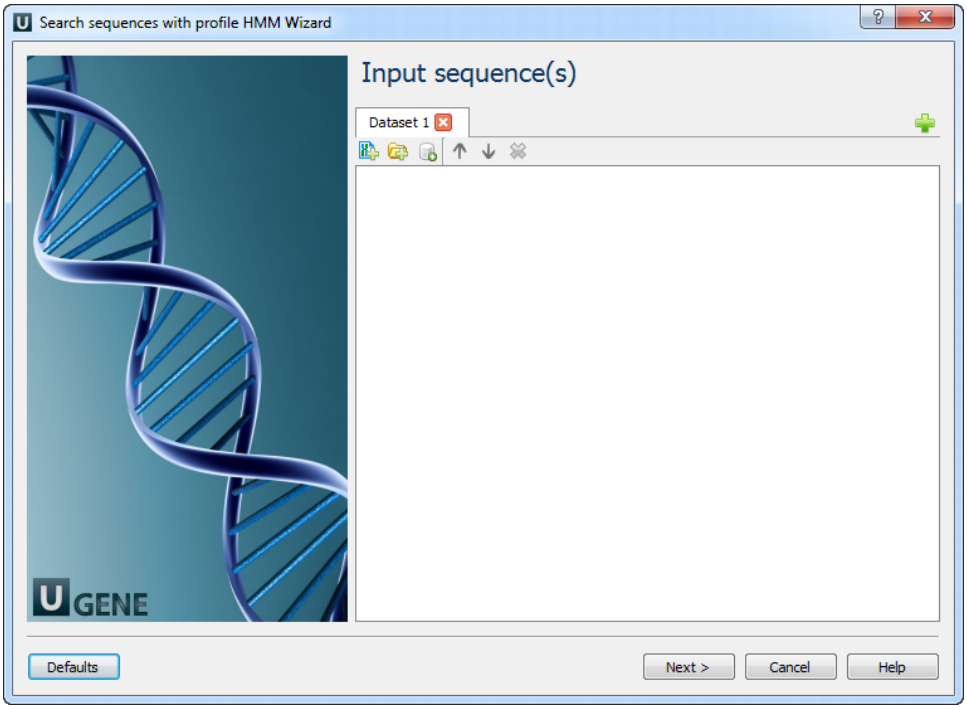
The workflow looks as follows:



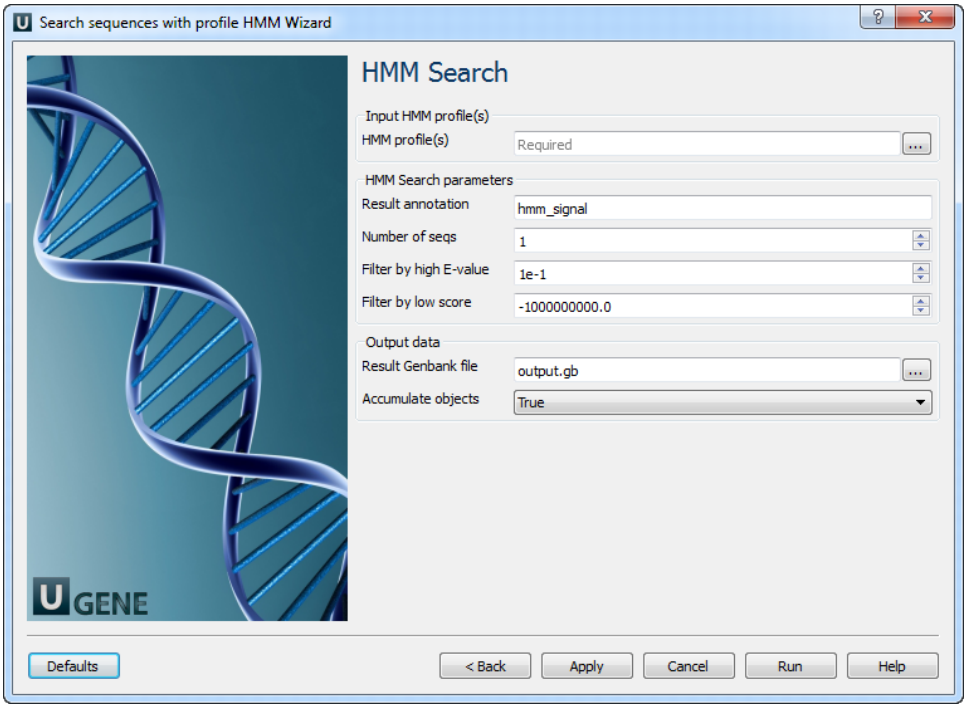
Workflow Wizard

The wizard has 2 pages.

1. Input sequence(s): On this page you must input sequence(s).



2. HMM search: On this page you can modify HMM search parameters.



The following parameters are available:

HMM profile(s)	Semicolon-separated list of paths to the input files.
Result annotation	A name of the result annotations.
Number of seqs	Calculate the E-value scores as if we had seen a sequence database of sequences.
Filter by high E-value	E-value filtering can be used to exclude low-probability hits from result.
	Score based filtering is an alternative to E-value filtering to exclude low-probability hits from result.

Filter by low score	
Result Genbank file	Location of output data file. If this attribute is set, slot "Location" in port will not be used.
Accumulate objects	Accumulate all incoming data in one file or create separate files for each input.In the latter case, an incremental numerical suffix is added to the file name.

NGS

- ChIP-Seq Coverage
- ChIP-seq Analysis with Cistrome Tools
- Extract Consensus from Assembly
- Extract Coverage from Assembly
- Extract Transcript Sequences
- Quality Control by FastQC
- De novo Assemble Illumina PE Reads
- De novo Assemble Illumina PE and Nanopore Reads
- De novo Assemble Illumina SE Reads
- De Novo Assembly and Contigs Classification
- Parallel NGS Reads Classification
- Serial NGS Reads Classification
- RNA-Seq Analysis with TopHat and StringTie
- RNA-seq Analysis with Tuxedo Tools
- Variation Annotation with SnpEff
- Call Variants with SAMtools
- Variant Calling and Effect Prediction
- Raw ChIP-Seq Data Processing
- Raw DNA-Seq Data Processing
- Raw RNA-Seq Data Processing
- Get Unmappet Reads

ChIP-Seq Coverage

The workflow sample, described below, prepare ChIP-Seq processed data (with BedTools and bedGraphToBigWig) for visualization in a genome browser. For input BED-file produces BigWig file.

How to Use This Sample

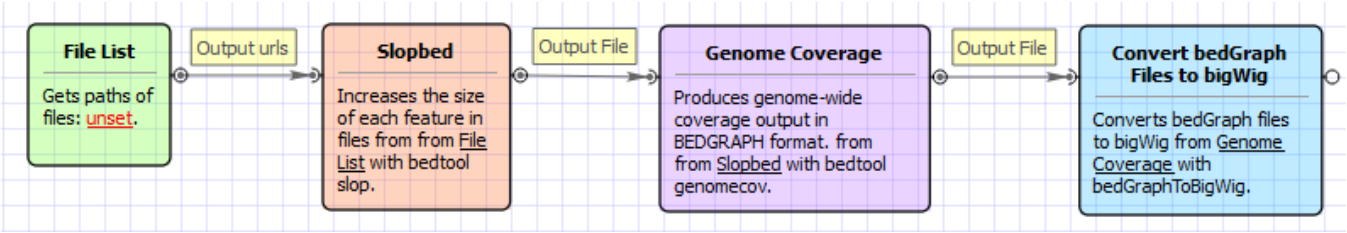
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "ChIP-Seq Coverage" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

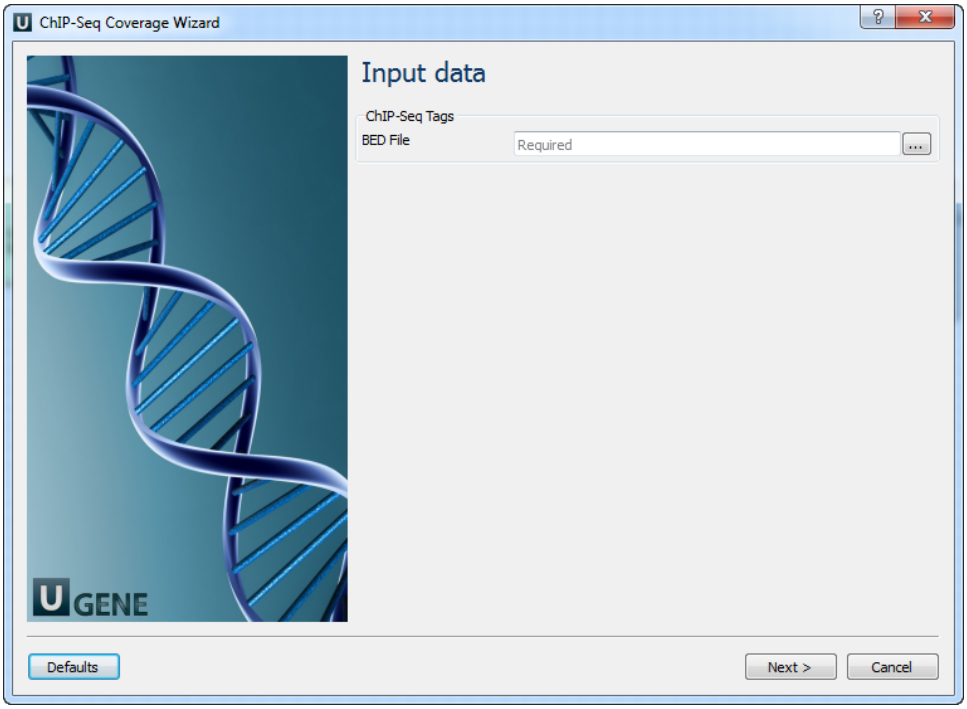
The opened workflow looks as follows:



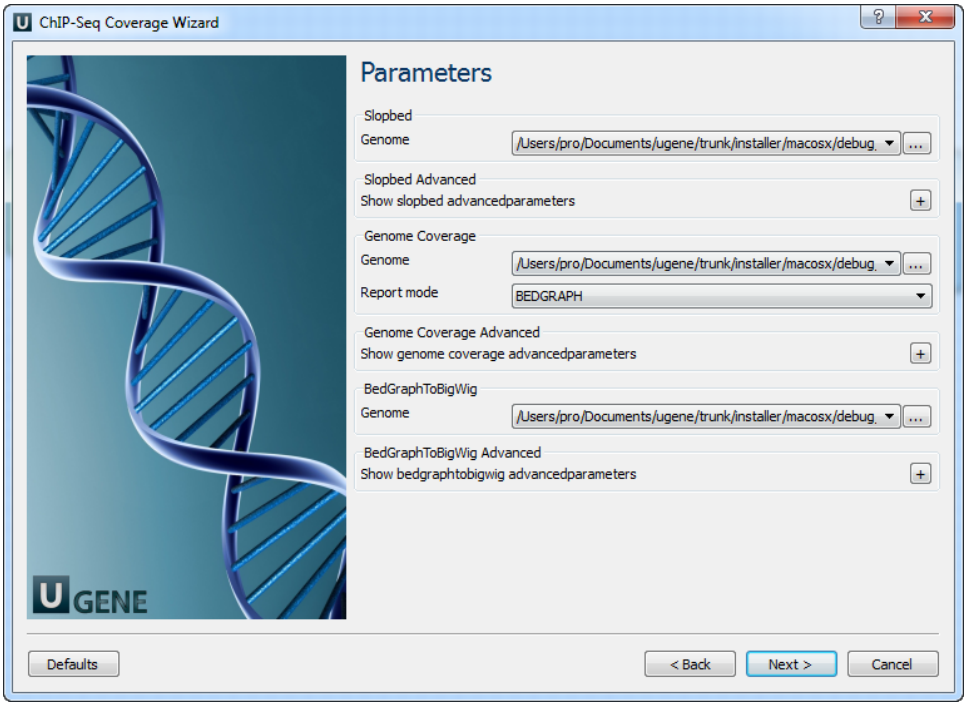
Workflow Wizard

The wizard has 3 pages.

1. Input data Page: On this page you must input BED file with ChIP-Seq tags.



2. Parameters Page: Here you can optionally modify parameters that should be used for the Slopbed, Genome Coverage and BedGraphToBigWig elements.

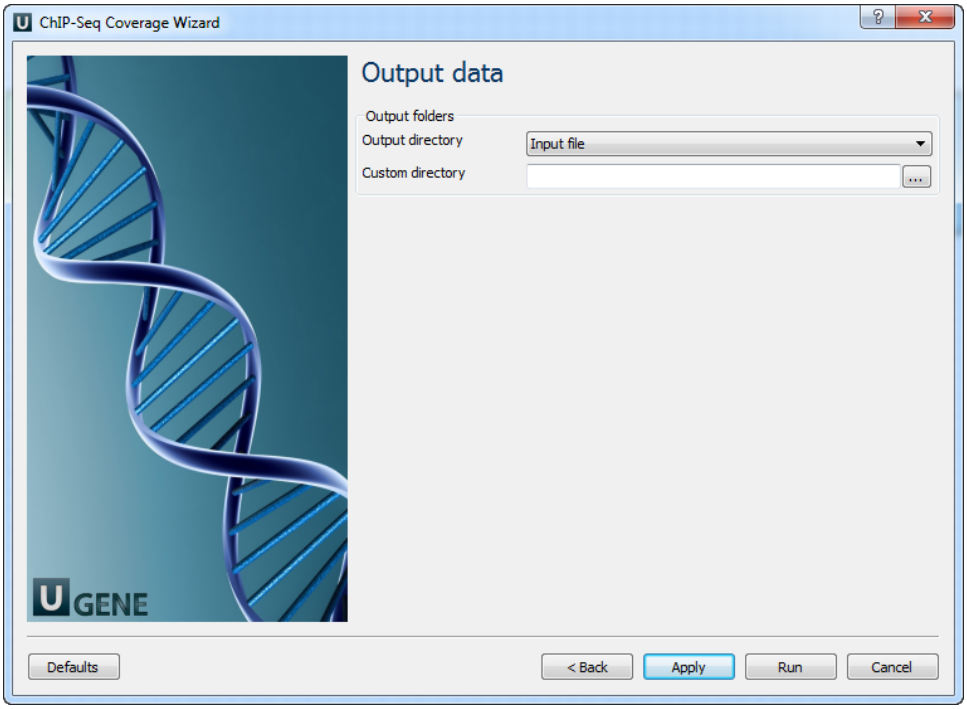


The following parameters are available:

Genome	In order to prevent the extension of intervals beyond chromosome boundaries, bedtools slop requires a genome file defining the length of each chromosome or contig. The format of the file is: (-g).
Each direction increase	Increase the BED/GFF/VCF entry by the same number base pairs in each direction. If this parameter is used -l and -l are ignored. Enter 0 to disable. (-b)

Subtract from start	The number of base pairs to subtract from the start coordinate. Enter 0 to disable. (-l)
Add to end	The number of base pairs to add to the end coordinate. Enter 0 to disable. (-r)
Strand-based	Define -l and -r based on strand. For example. if used, -l 500 for a negative-stranded feature, it will add 500 bp to the end coordinate. (-s)
As fraction	Define -l and -r as a fraction of the feature's length. E.g. if used on a 1000bp feature, -l 0.50, will add 500 bp upstream. (-pct)
Print header	Print the header from the input file prior to results. (-header)
Filter start>end fields	Remove lines with start postion greater than end position
Report mode	Histogram () - Compute a histogram of coverage. Per-base (0-based) (-dz) - Compute the depth of feature coverage for each base on each chromosome (0-based). Per-base (1-based) (-d) - Compute the depth of feature coverage for each base on each chromosome (1-based). BEDGRAPH (-bg) - Produces genome-wide coverage output in BEDGRAPH format. BEDGRAPH (including uncoveded) (-bga) - Produces genome-wide coverage output in BEDGRAPH format (including uncovered).
Split	Treat BAM or BED12 entries as distinct BED intervals when computing coverage. For BAM files, this uses the CIGAR and operations to infer the blocks for computing coverage. For BED12 files, this uses the BlockCount, BlockStarts, and BlockEnds fields (i.e., columns 10,11,12). (-split)
Strand	Calculate coverage of intervals from a specific strand. With BED files, requires at least 6 columns (strand is column 6). (-strand)
5 prime	Calculate coverage of 5' positions (instead of entire interval). (-5)
3 prime	Calculate coverage of 3' positions (instead of entire interval). (-3)
Max	Combine all positions with a depth >= max into a single bin in the histogram. (-max)
Scale	Scale the coverage by a constant factor.Each coverage value is multiplied by this factor before being reported. Useful for normalizing coverage by, e.g., reads per million (RPM). Default is 1.0; i.e., unscaled. (-scale)
Trackline	Adds a UCSC/Genome-Browser track line definition in the first line of the output. (-trackline)
Trackopts	Writes additional track line definition parameters in the first line. (-trackopts)
Block size	Number of items to bundle in r-tree (-blockSize).
Items per slot	Number of data points bundled at lowest level (-itemsPerSlot).
Uncompressed	If set, do not use compression (-unc).

3. Output Files Page: On this page you can select an output directory:



ChIP-seq Analysis with Cistrome Tools

The component for ChIP-seq data analysis is not installed by default. To use this sample, add the component via the UGENE Online Installer or, if you used an offline installer, manually configure the package, see "Configure ChIP-seq Analysis Data" chapter of the manual.

to check which genes are nearby so can be regarded as potential regulated genes, then perform GO analysis;The ChIP-seq pipeline “Cistrome” integrated into UGENE allows one to do the following analysis steps: peak calling and annotating, motif search and gene ontology. ChIP-seq analysis is started from MACS tool. CEAS then takes peak regions and signal wiggle file to check which chromosome is enriched with binding/modification sites, whether bindings events are significant at gene features like promoters, gene bodies, exons, introns or UTRs, and the signal aggregation at gene transcription start/end sites or meta-gene bodies (average all genes). Then peaks are investigated in these ways:

1. to check the conservation scores at the binding sites;
2. the DNA motifs at binding sites.

Note that it is originally based on the General ChIP-seq pipeline from the public Cistrome installation on the Galaxy workflow platform.

How to Use This Sample

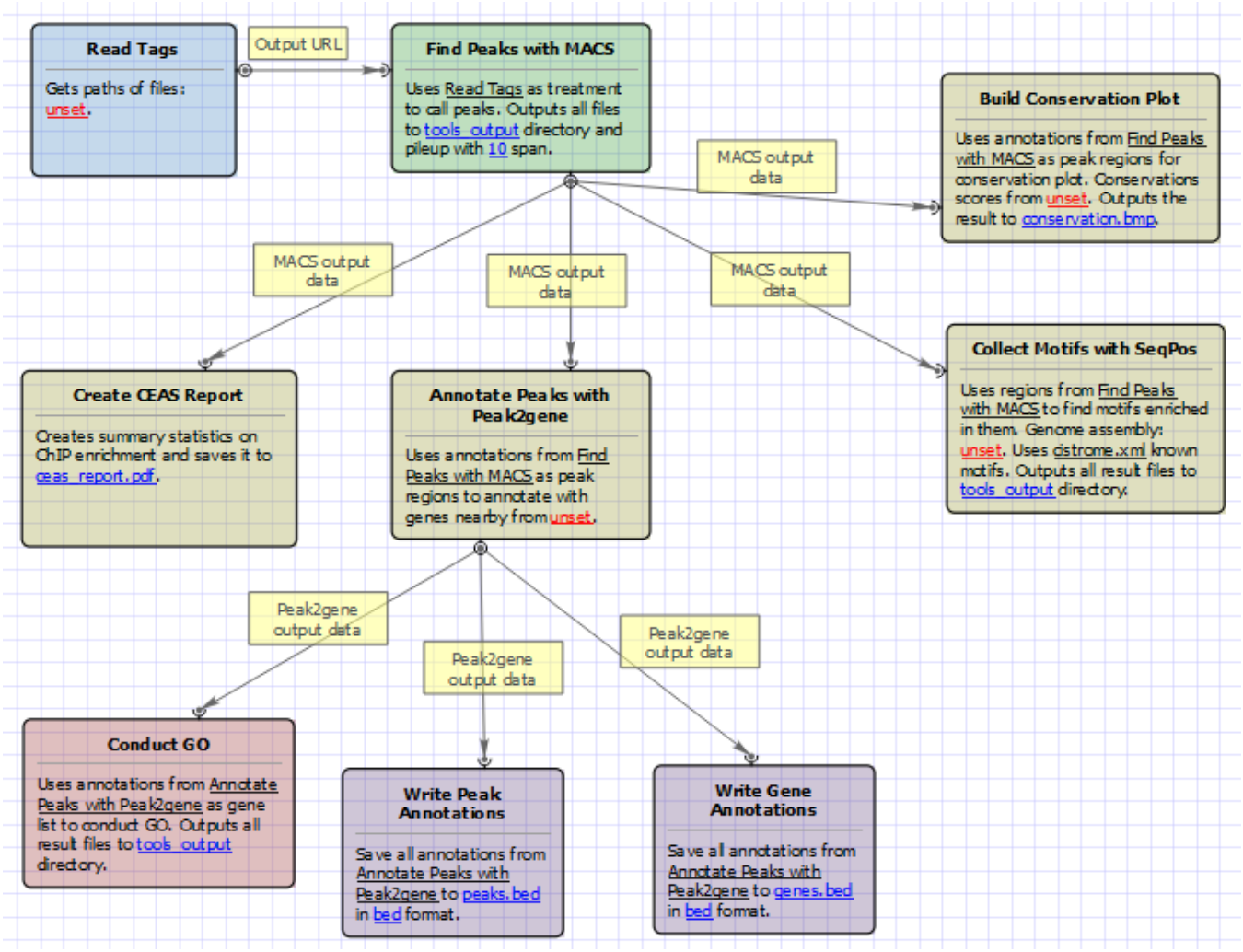
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

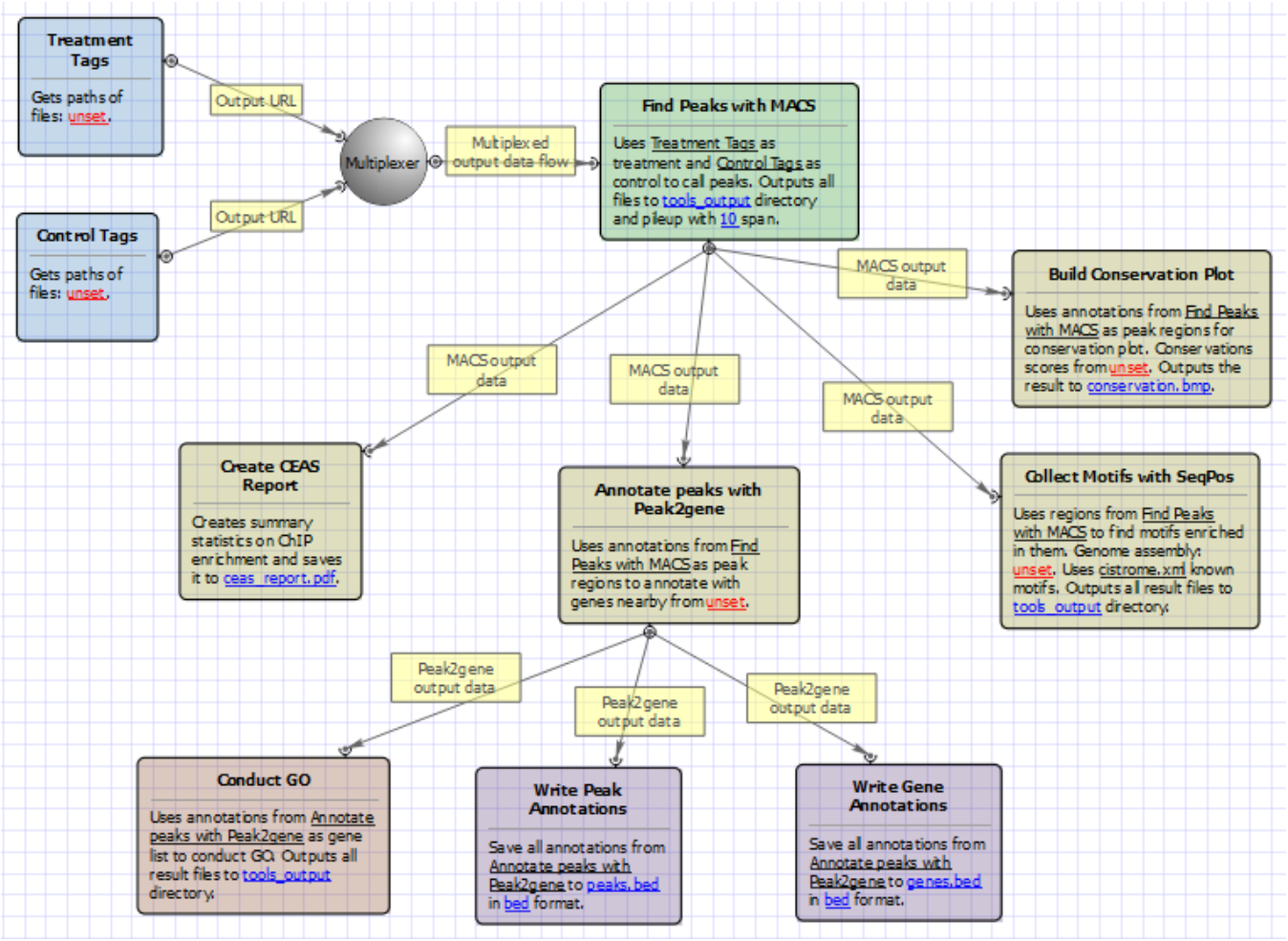
The workflow sample "ChIP-seq Analysis with Cistrome Tools" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

For treatment tags only analysis type the workflow looks as follows:



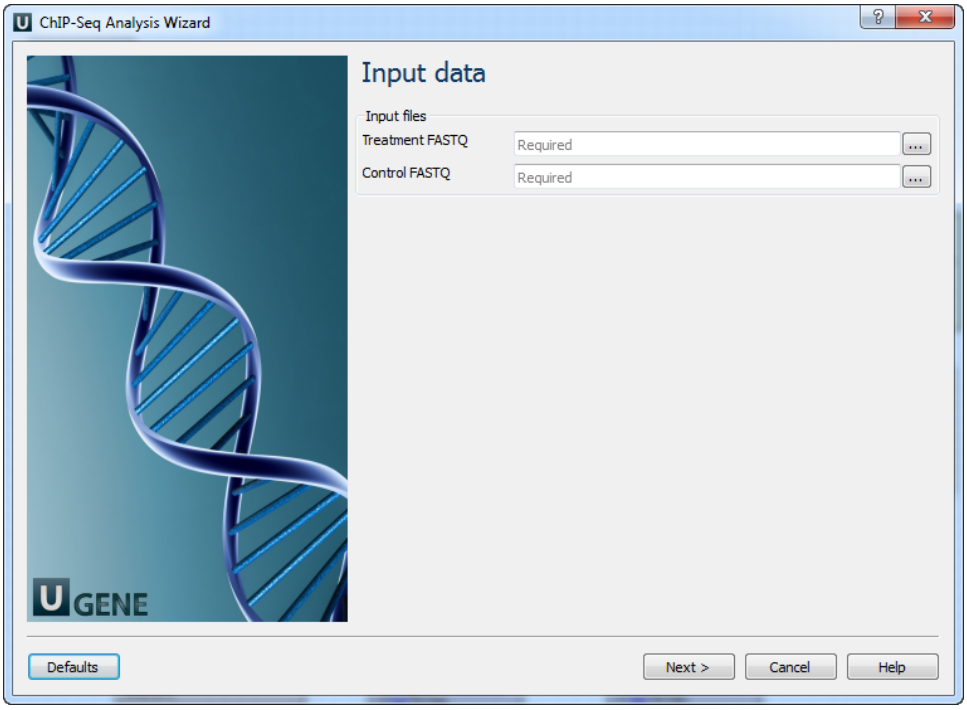
For treatment and control tags analysis type the workflow looks as follows:



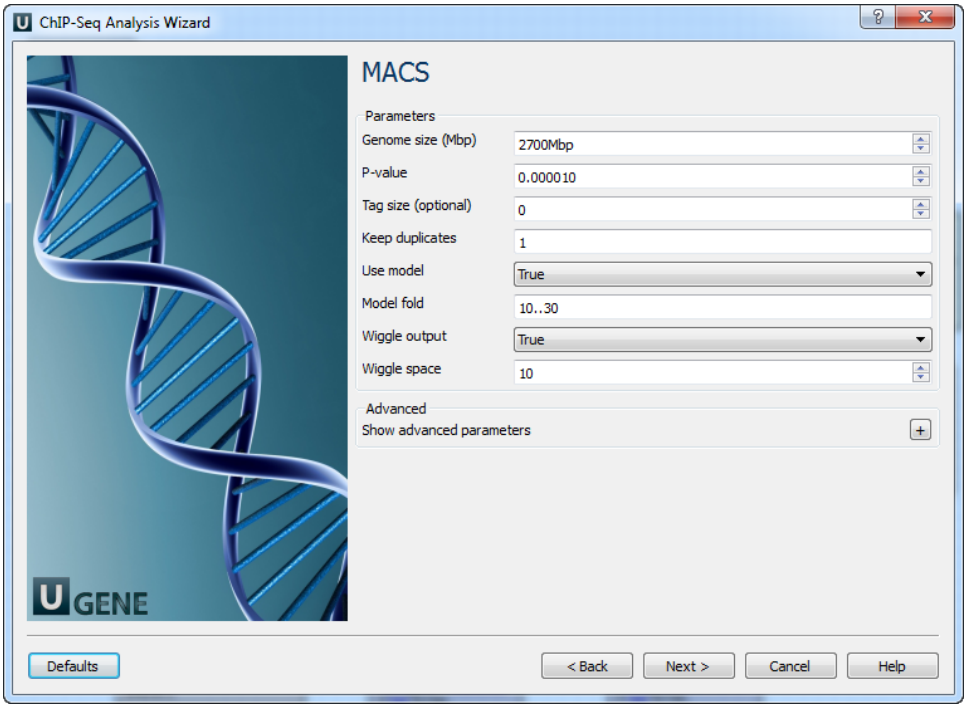
Workflow Wizard

The wizards are the same for both types of workflows. The wizard has 7 pages.

1. Input data: Here you need to input a file with treatment and control annotations for MACS.



2. MACS: Here you can change default MACS parameters.

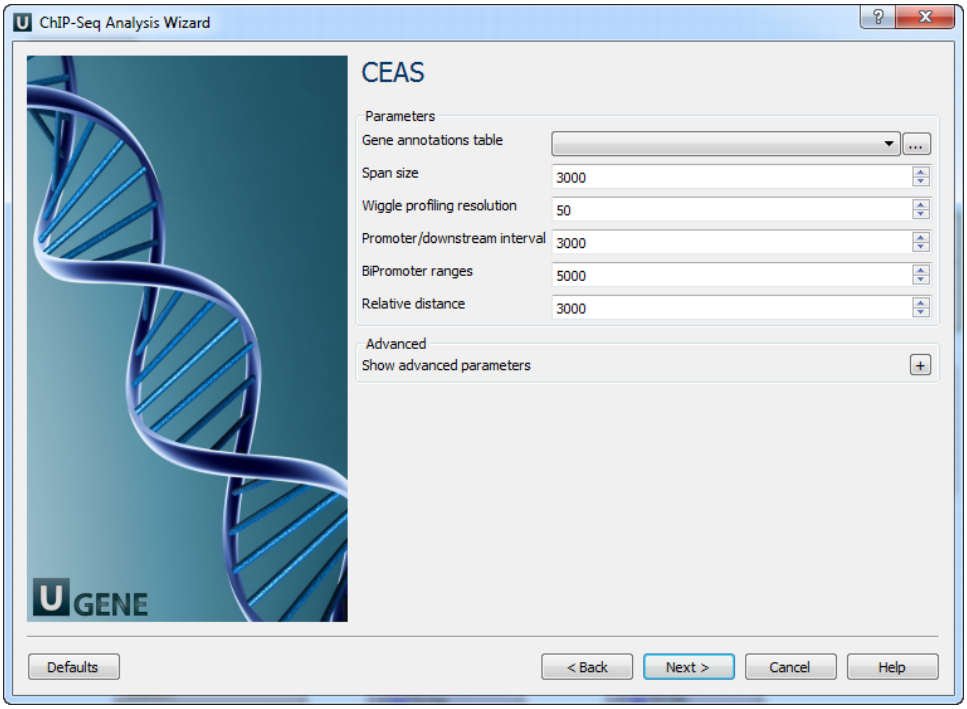


The following parameters are available:

Genome size (Mbp)	Homo sapience - 2700 Mbp Mus musculus - 1870 Mbp Caenorhabditis elegans - 90 Mbp Drosophila melanogaster - 120 Mbp

	It's the mappable genome size or effective genome size which is defined as the genome size which can be sequenced. Because of the repetitive features on the chromosomes, the actual mappable genome size will be smaller than the original size, about 90% or 70% of the genome size.
P-value	P-value cutoff. Default is 0.00001, for looser results, try 0.001 instead.
Tag size (optional)	Length of reads. Determined from first 10 reads if not specified (input 0).
Keep duplicates	It controls the MACS behavior towards duplicate tags at the exact same location -- the same coordination and the same strand. The default auto option makes MACS calculate the maximum tags at the exact same location based on binomal distribution using 1e-5 as pvalue cutoff; and the all option keeps every tags. If an integer is given, at most this number of tags will be kept at the same location.
Use model	Whether or not to use MACS paired peaks model.
Model fold	Select the regions within MFOLD range of high-confidence enrichment ratio against. Model fold is available when Use Model is true, which is the foldchange to chose paired peaks to build paired peaks model. Users need to set a lower(smaller) and upper(larger) number for fold change so that MACS will only use the peaks within these foldchange range to build model.
Wiggle output	If this flag is on, MACS will store the fragment pileup in wiggle format for the whole genome data instead of for every chromosomes.
Wiggle space	By default, the resolution for saving wiggle files is 10 bps, i.e., MACS will save the raw tag count every 10 bps. You can change it along with Wiggle output parameter.
Shift size	An arbitrary shift value used as a half of the fragment size when model is not built. Shift size is available when Use Model is false, which will represent the HALF of the fragment size of your sample. If your sonication and size selection size is 300 bps, after you trim out nearly 100 bps adapters, the fragment size is about 200 bps, so you can specify 100 here.
Band width	The band width which is used to scan the genome for model building. You can set this parameter as the sonication fragment size expected from wet experiment. Used only while building the shifting model.
Use lambda	Whether to use local lambda model which can use the local bias at peak regions to throw out false positives.
Small nearby region	The small nearby region in basepairs to calculate dynamic lambda. This is used to capture the bias near the peak summit region. Invalid if there is no control data.
Auto bimodal	Whether turn on the auto pair model process.If set, when MACS failed to build paired model, it will use the nomodelsettings, the Shift size parameter to shift and extend each tags.
Scale to large	When set, scale the small sample up to the bigger sample.By default, the bigger dataset will be scaled down towards the smaller dataset,which will lead to smaller p/qvalues and more specific results. Keep in mind that scaling down will bring down background noise more.

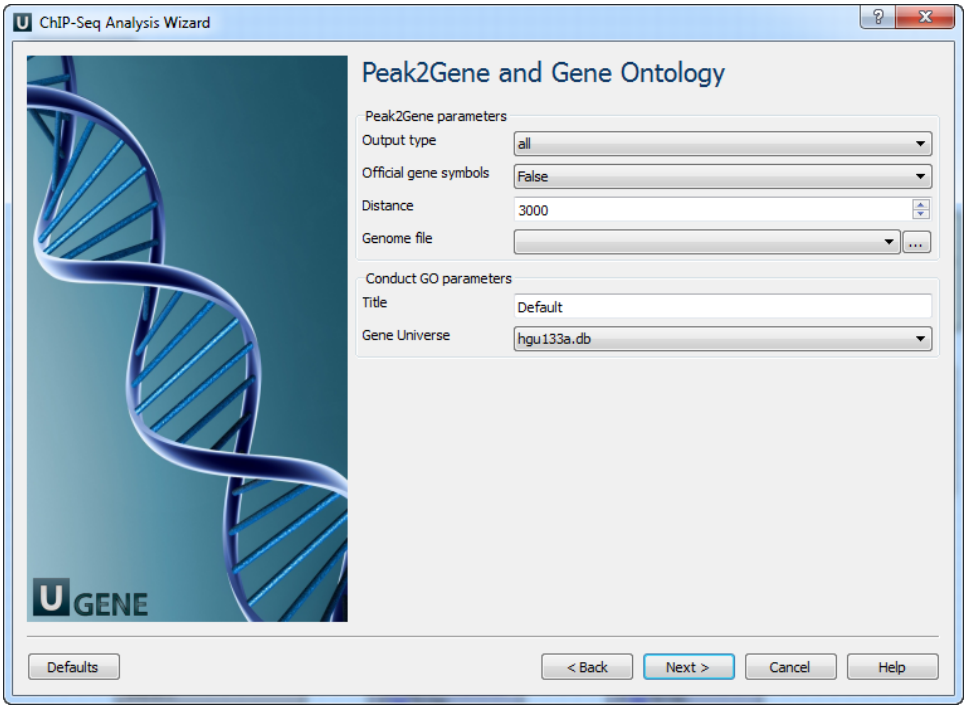
3. CEAS: The next page allows to configure CEAS parameters.



The following parameters are available:

Gene annotations table	Path to gene annotation table (e.g. a refGene table in sqlite3 db format.
Span size	Span from TSS and TTS in the gene-centered annotation (base pairs). ChIP regions within this range from TSS and TTS are considered when calculating the coverage rates in promoter and downstream.
Wiggle profiling resolution	Wiggle profiling resolution. WARNING: Value smaller than the wig interval (resolution) may cause aliasing error.
Promoter /downstream interval	Promoter/downstream intervals for ChIP region annotation are three values or a single value can be given. If a single value is given, it will be segmented into three equal fractions (e.g. 3000 is equivalent to 1000,2000,3000).
BiPromoter ranges	Bidirectional-promoter sizes for ChIP region annotation. It's two values or a single value can be given. If a single value is given, it will be segmented into two equal fractions (e.g. 5000 is equivalent to 2500,5000).
Relative distance	Relative distance to TSS/TTS in WIGGLE file profiling.
Gene group files	Gene groups of particular interest in wig profiling. Each gene group file must have gene names in the 1st column. The file names are separated by commas.
Gene group names	Set this parameter empty for using default values. The names of the gene groups from "Gene group files" parameter. These names appear in the legends of the wig profiling plots. Values range: comma-separated list of strings. Default value: 'Group 1, Group 2,...Group n'.

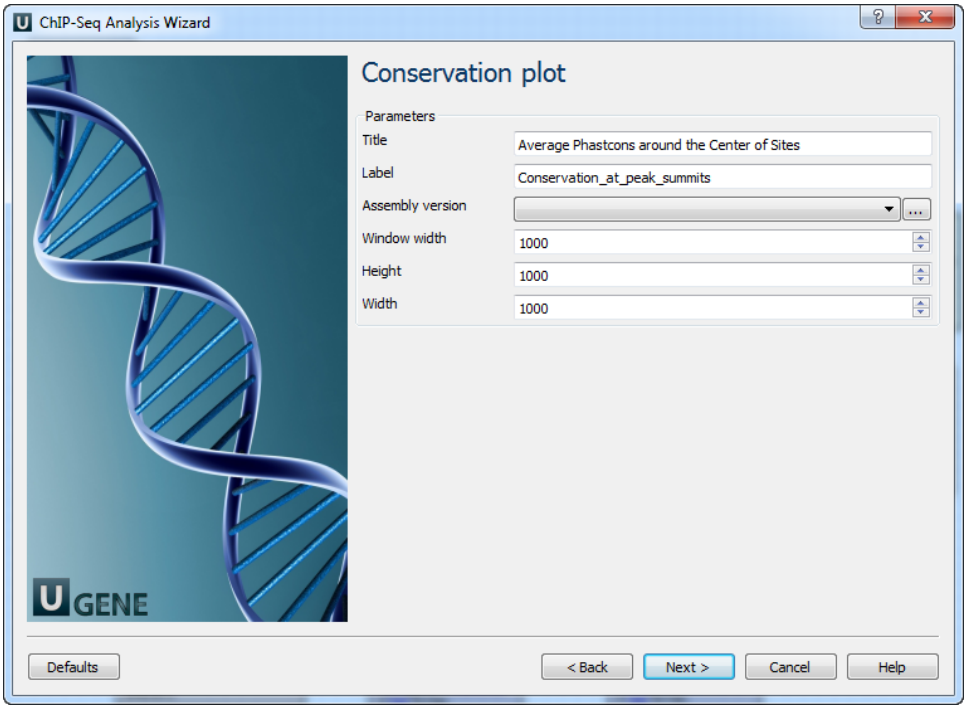
4. Peak2Gene and Gene Ontology: The next page allows to configure Peak2Gene and Gene Ontology parameters.



The following parameters are available:

Output type	The directory to store Conduct GO results.
Official gene symbols	Output official gene symbol instead of refseq name.
Distance	Set a number which unit is base. It will get the refGenes in n bases from peak center.
Genome file	Select a genome file (sqlite3 file) to search refGenes.
Title	Title is used to name the output files - so make it meaningful.
Gene Universe	Select a gene universe.

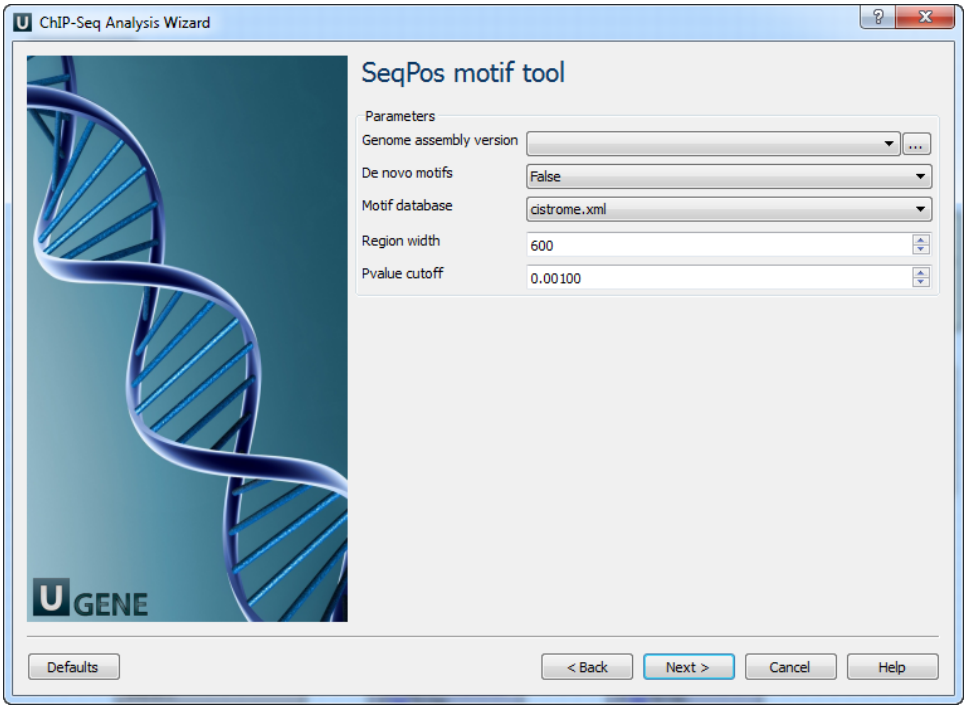
5. Conservation plot: On this page you can modify Conservation Plot parameters.



The following parameters are available:

Title	Title of the figure.
Label	Label of data in the figure.
Assembly version	The directory to store phastcons scores.
Window width	Window width centered at middle of regions.
Height	Height of plot.
Width	Width of plot.

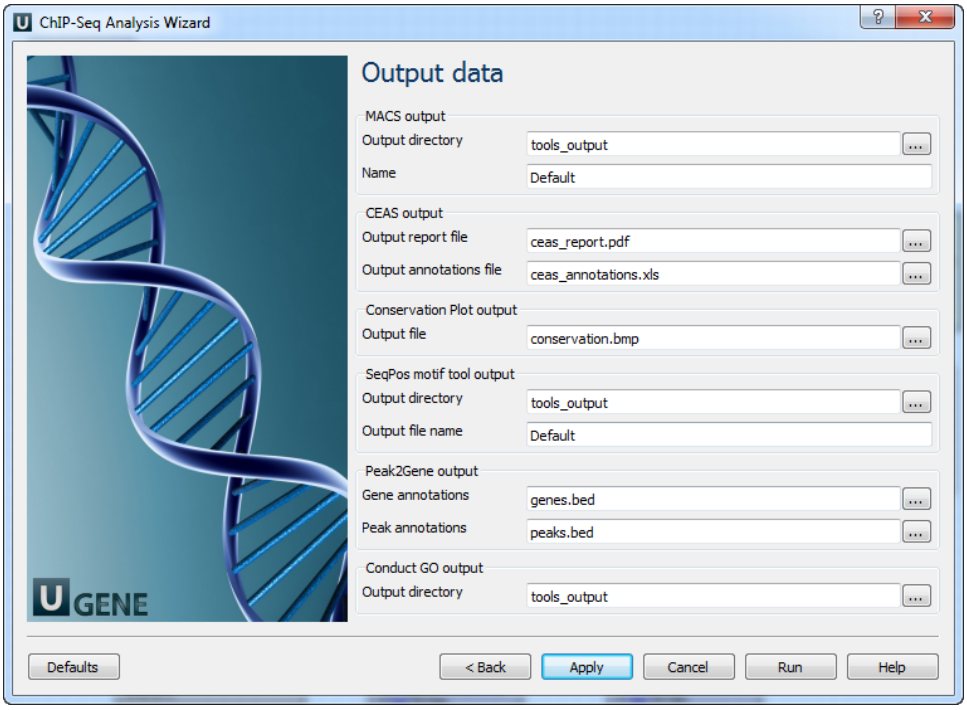
6. SeqPos motif tool: On this page you can modify SeqPos motif parameters.



The following parameters are available:

Genome assembly version	UCSC database version.
De novo motifs	Run de novo motif search.
Motif database	Known motif collections.
Region width	Width of the region to be scanned for motifs; depends on a resolution of assay.
Pvalue cutoff	Pvalue cutoff for the motif significance.

7. Output data: On this page you can modify output parameters.



The following parameters are available.

MACS output:

Output directory	Directory to save MACS output files.
Name	Name string of the experiment. MACS will use this string NAME to create output files like 'NAME_peaks.xls', 'NAME_negative_peaks.xls', 'NAME_peaks.bed', 'NAME_summits.bed', 'NAME_model.r' and so on. So please avoid any confliction between these filenames and your existing files.

CEAS output:

Output report file	Path to the report output file. Result for the CEAS analysis.
Output annotations file	Name of tab-delimited output text file, containing a row of annotations for every RefSeq gene. Note that the file is not generated if there is no peak regions input.

Conservation Plot output:

Output file	File to store phastcons results (BMP).

SeqPos motif tool output:

Output directory	Directory to store seqpos results.
Output file name	Name of the output file which stores new motifs found during a de novo search.

Peak2Gene output:

Gene annotations	Location of peak2gene gene annotations data file.
Peak annotations	Location of peak2gene peak annotations data file.

Conduct GO output:

Output directory	Directory to store Conduct GO results.

The work on this pipeline was supported by grant RUB1-31097-NO-12 from NIAID.

Extract Consensus from Assembly

The workflow sample, described below, uses input assemblies to extract the consensus and save them to a FASTA.

How to Use This Sample

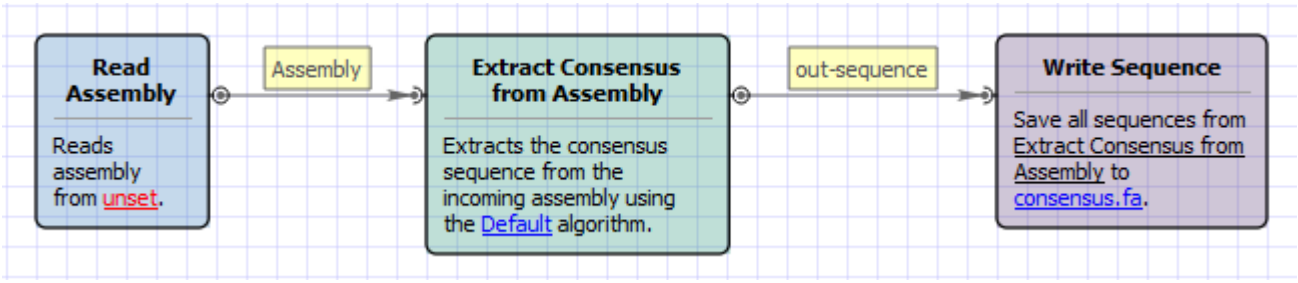
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Extract Consensus from Assembly" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

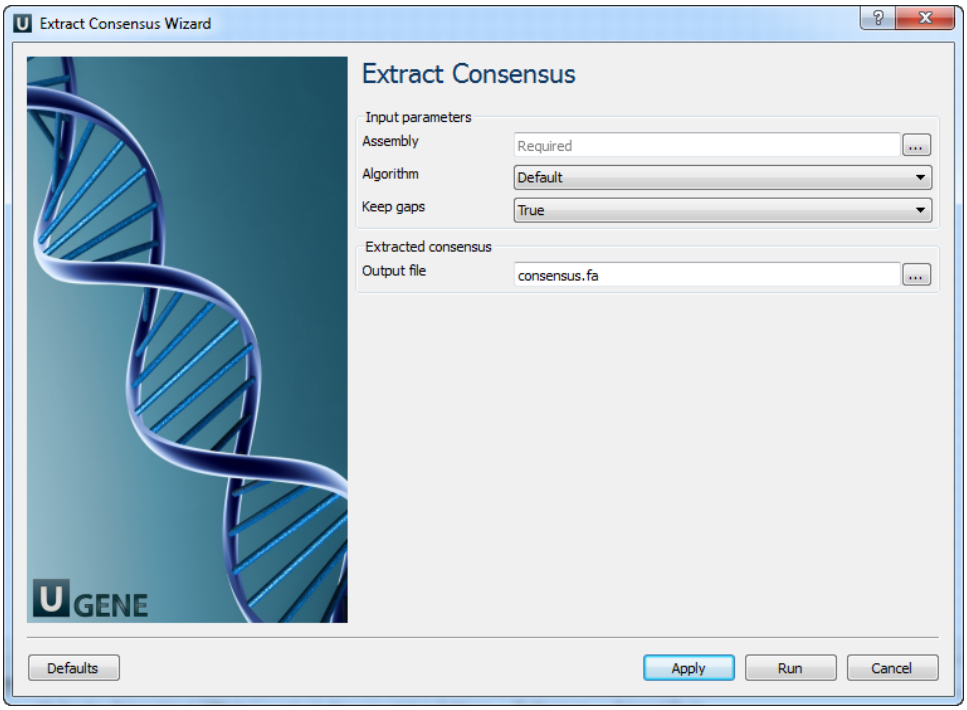
The opened workflow looks as follows:



Workflow Wizard

The wizard has 1 page.

1. Extract Consensus Page: On this page you must input assembly file and output file. Also you can modify other input parameters.



The following parameters are available:

Assembly	Semicolon-separated list of pathes to the input files.
Algorithm	The algorithm of consensus extracting.
Keep gaps	Set this parameter if the result consensus must keep the gaps.
Output files	Location of output data file. If this attribute is set, slot "Location" in port will not be used.

Extract Coverage from Assembly

The workflow sample, described below, allows one to extract a coverage and/or bases count from an assembly. It receives a number of assemblies and for each of them produces coverage as a tab delimited plain text file. The coverage is extracted considering a threshold value.

How to Use This Sample

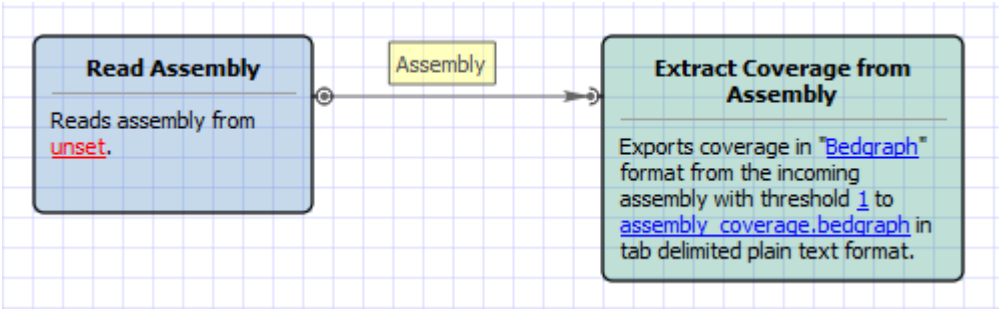
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Extract Coverage from Assembly" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

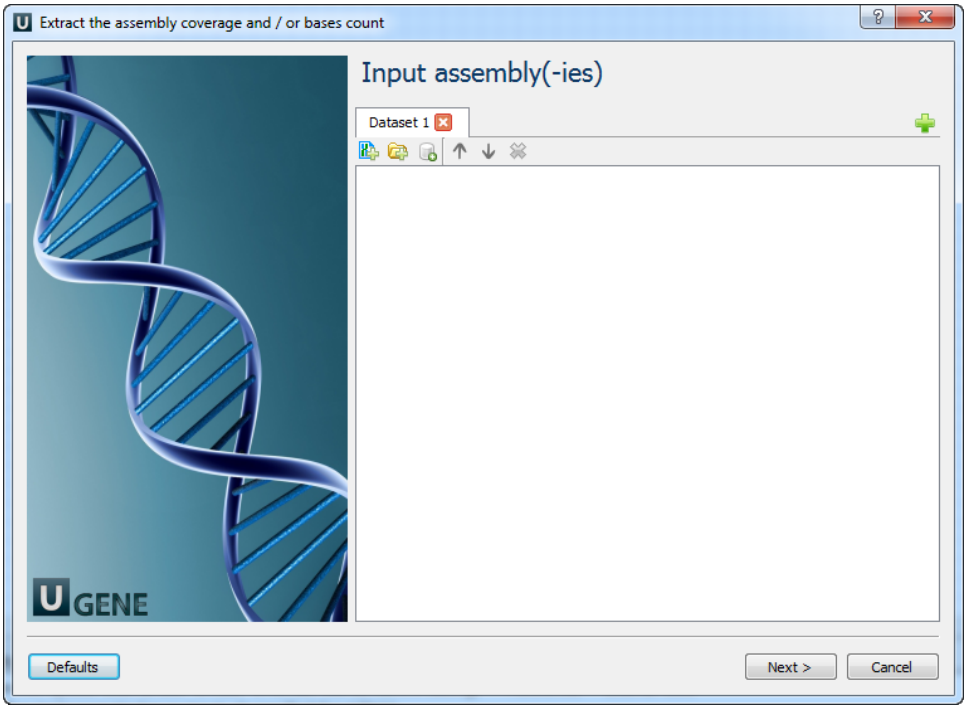
The opened workflow looks as follows:



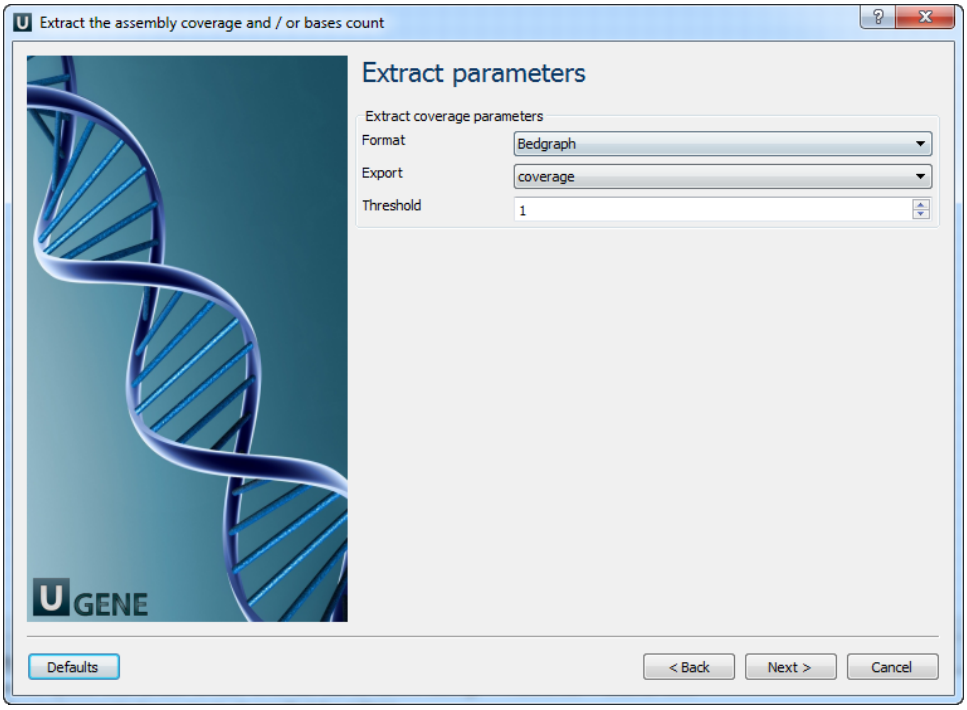
Workflow Wizard

The wizard has 3 pages.

1. Input assembly (-ies) Page: On this page you must input assembly(-ies).



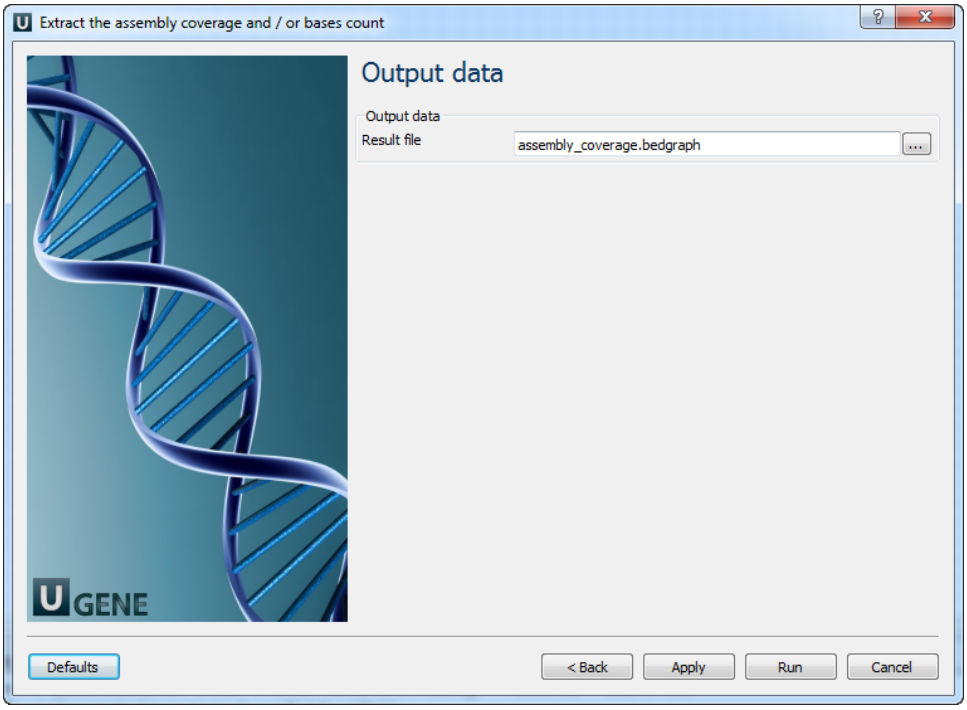
2. Extract parameters Page: Here you can optionally modify extract parameters.



The following parameters are available:

Format	Format to store the output.
Export	Data type to export.
Threshold	The minimum coverage value to export.

3. Output data Page: On this page you can select an output file:



Extract Transcript Sequences

This workflow uses input transcripts and genomic sequences to generate a FASTA file with the DNA sequences for the transcripts. Please make sure that contig or chromosome names in the transcript file(s) have corresponding entries in the input sequence(s).

How to Use This Sample

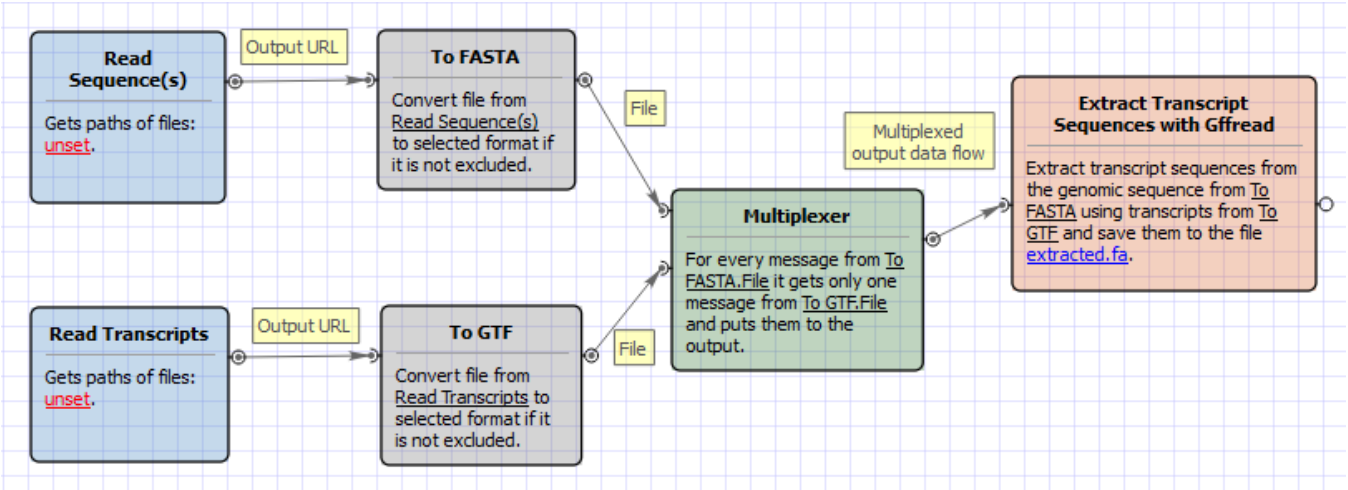
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Extract Transcript Sequences" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

The workflow looks as follows:



Quality Control by FastQC

FastQC aims to provide a simple way to do some quality control checks on raw sequence data coming from high througput sequencing pipelines. It provides a molecular set of analyses which you can use to give a quick impression of whether your data has any problems of which you should be aware before doing any further analysis.

How to Use This Sample

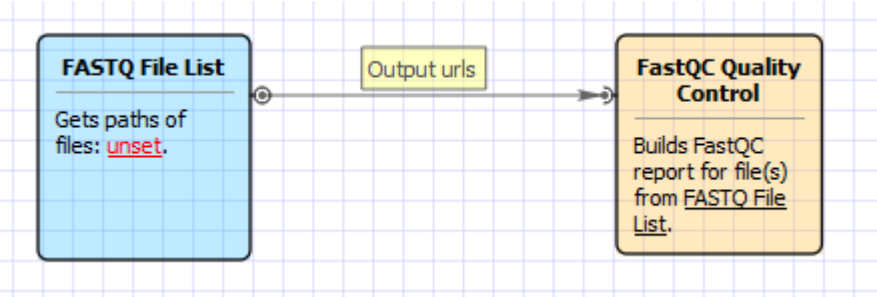
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Quality Control by FastQC" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

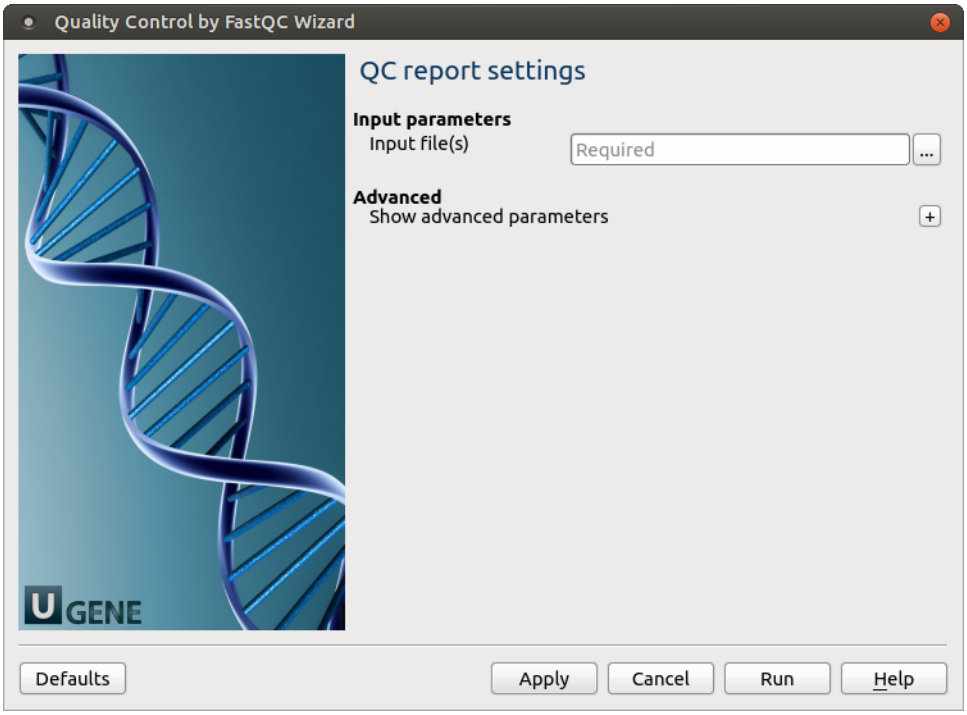
The workflow is the following:



Workflow Wizard

The wizard has 1 page.

1. High Throughput Sequence QC Report by FastQC: On this page you must input FASTQ file(s) and optionally modify advanced parameters.



The following parameters are available:

FASTQ URL(s)	Semicolon-separated list of pathes to the input files.
Output	Select an output directory. Custom - specify the output directory in the 'Custom directory' parameter. Workflow - internal workflow

directory	directory. Input file - the directory of the input file.
Custom directory	Select the custom output directory.
List of adapters	Specifies a non-default file which contains the list of adapter sequences which will be explicity searched against the library. The file must contain sets of named adapters in the form name[tab]sequence. Lines prefixed with a hash will be ignored.
List of contaminants	Specifies a non-default file which contains the list of contaminants to screen overrepresented sequences against. The file must contain sets of named contaminants in the form name[tab]sequence. Lines prefixed with a hash will be ignored.

De novo Assemble Illumina PE Reads

The workflow sample, described below, takes FASTQ files with paired-end Illumina reads as input and process them as follows:

- Improve reads quality with Trimmomatic
- Provide FastQC quality reports
- De novo assemble reads with SPAdes

How to Use This Sample

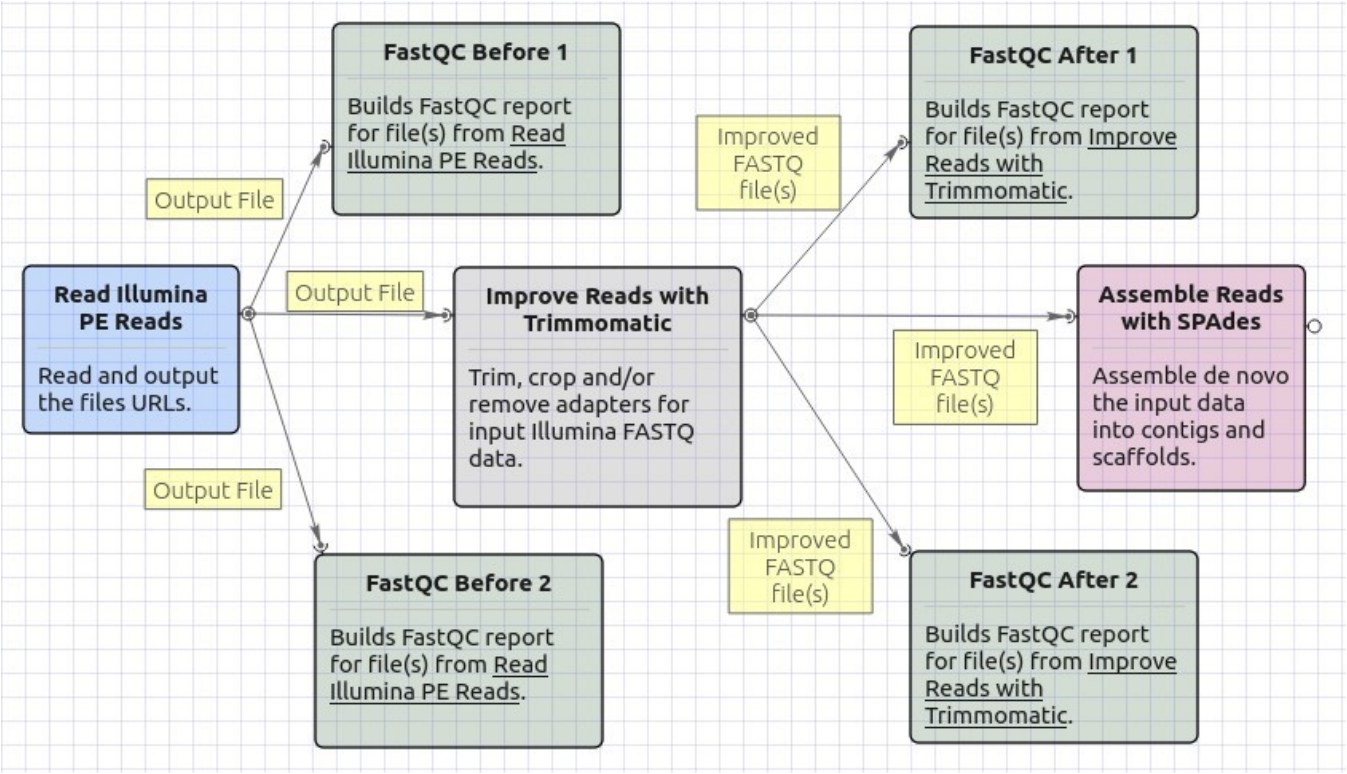
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "De novo Assemble Illumina PE Reads" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

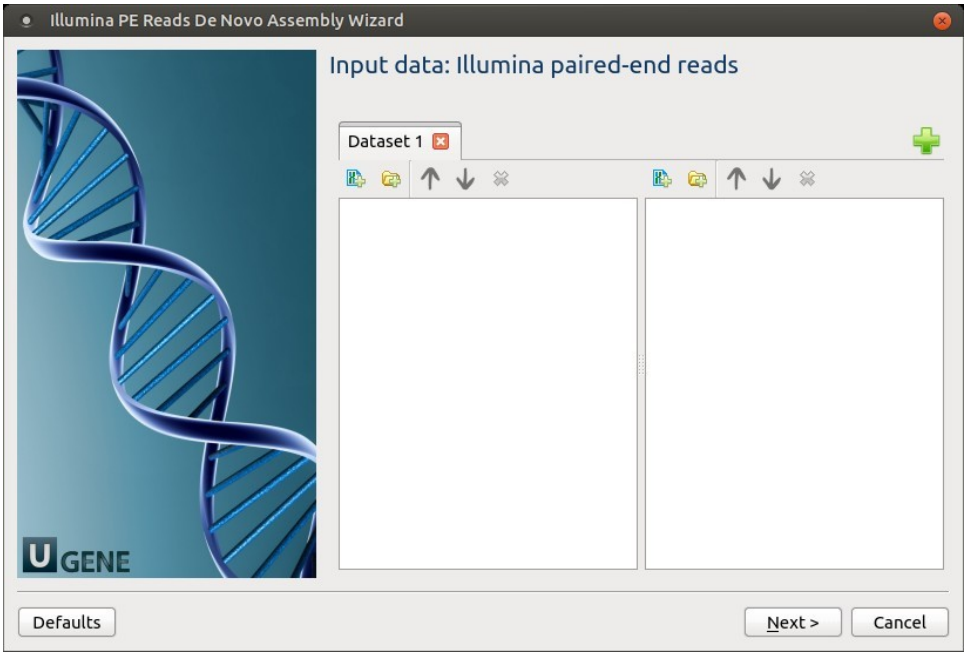
The opened workflow looks as follows:



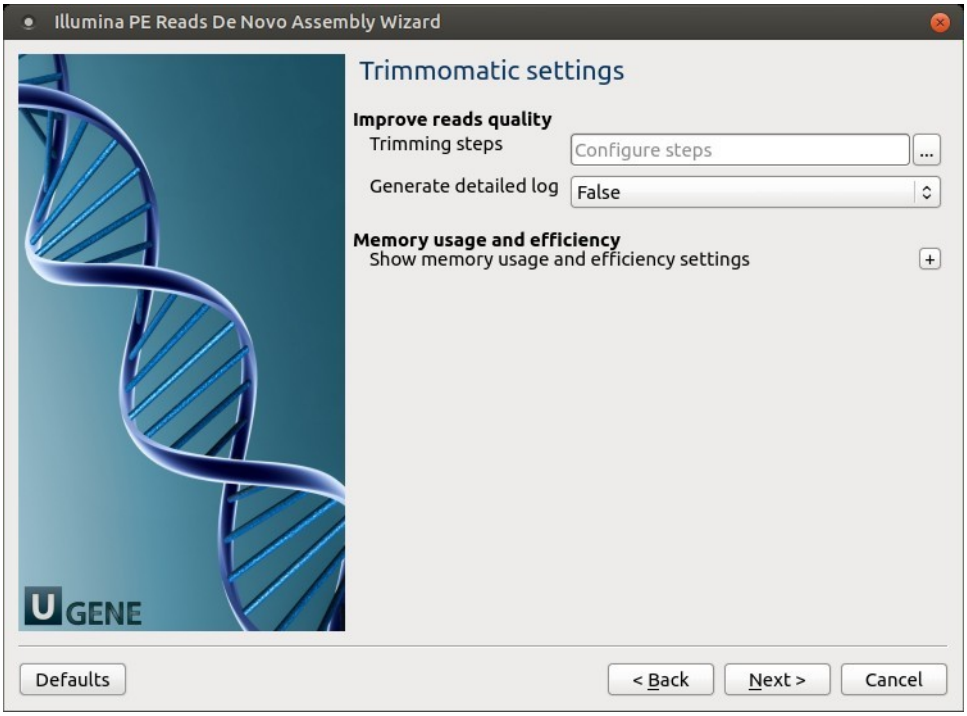
Workflow Wizard

The wizard has 4 pages.

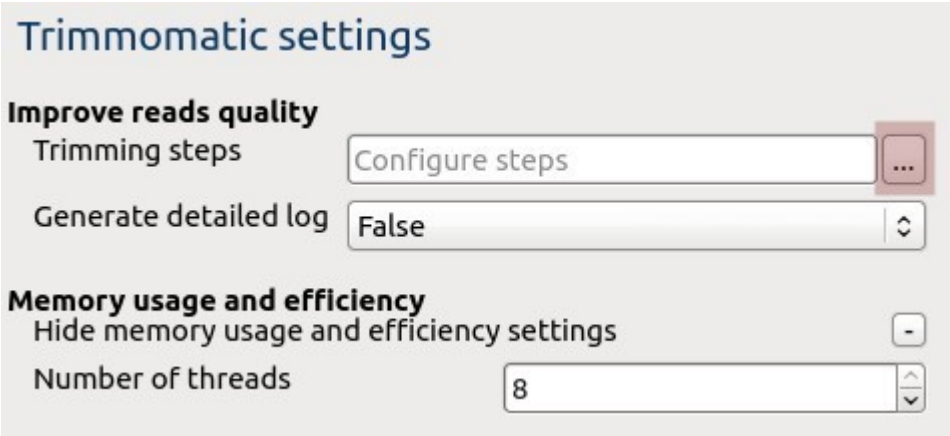
1. Input data: Illumina paired-end reads: On this page, files with Illumina paired-end reads must be set.



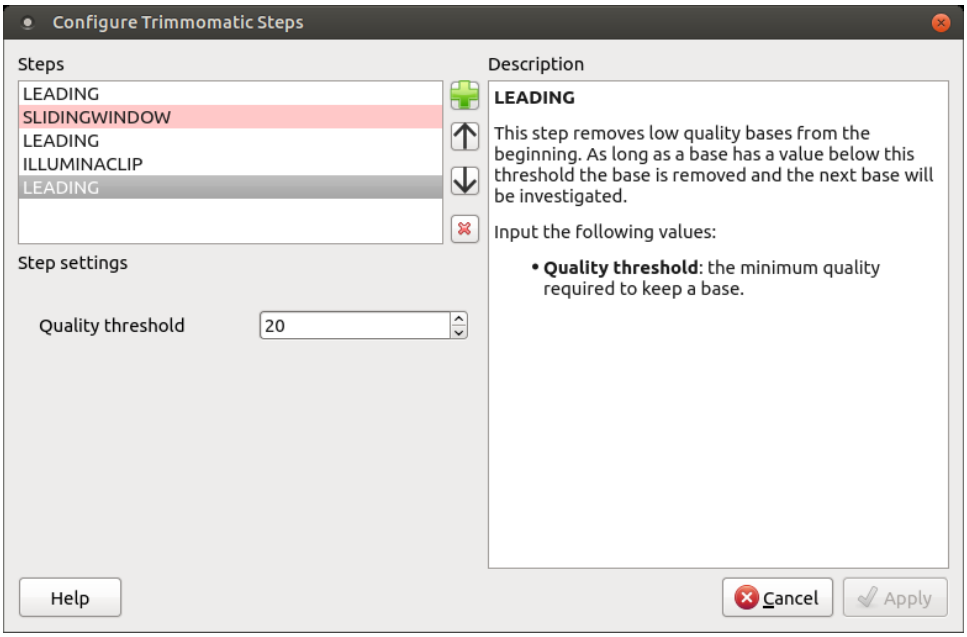
2. Trimmomatic settings: The Trimmomatic parameters can be changed here.



To configure trimming steps use the following button:



The following dialog will appear:



Click the *Add new step* button and select a step. The following options are available:

- ILLUMINACLIP: Cutadapterandotherillumina-specific sequences from the read.
- SLIDINGWINDOW: Perform a sliding window trimming, cutting once the average quality within the window falls below a threshold.
- LEADING: Cut bases off the start of a read, if below a threshold quality.
- TRAILING: Cut bases off the end of a read, if below a threshold quality.
- CROP: Cut the read to a specified length.
- HEADCROP: Cut the specified number of bases from the start of the read.
- MINLEN: Drop the read if it is below a specified length.
- AVGQUAL: Drop the read if the average quality is below the specified level.
- TOPHRED33: Convert quality scores to Phred-33.
- TOPHRED64: Convert quality scores to Phred-64.

Each step has the own parameters:

AVGQUAL

This step drops a read if the average quality is below the specified level.

Input the following values:

- Quality threshold: the minimum average quality required to keep a read.

CROP

This step removes bases regardless of quality from the end of thread, so that the readhas maximally the specified length after this step has been performed. Steps performed after CROP might of course further shorten the read.

Input the following values:

- Length: the number of bases to keep, from the start of the read.

HEADCROP

This step removes the specified number of bases, regardless of quality, from the beginning of the read.

Input the following values:

- Length: the number of bases to remove from the start of the read.

ILLUMINACLIP

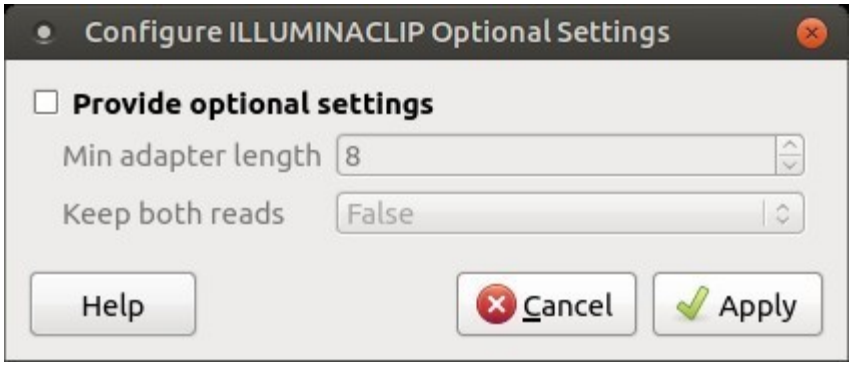
This step is used to find and remove Illumina adapters.

Trimmomatic first compares short sections of an adapter and a read. If they match enough, the entire alignment between the read and adapter is scored. For paired-end reads, the "palindrome" approach is also used to improve the result. See Trimmomatic manual for details.

Input the following values:

- Adapter sequences: a FASTA file with the adapter sequences. Files for TruSeq2 (GAII machines), TruSeq3 (HiSeq and MiSeq machines) and Nextera kits for SE and PE reads are now available by default. The naming of the various sequences within the specified file determines how they are used.
- Seed mismatches: the maximum mismatch count in short sections which will still allow a full match to be performed.
- Simple clip threshold: a threshold for simple alignment mode. Values between 7 and 15 are recommended. A perfect match of a 12 base sequence will score just over 7, while 25 bases are needed to score 15.
- Palindrome clip threshold: a threshold for palindrome alignment mode. For palindromic matches, a longer alignment is possible. Therefore the threshold can be in the range of 30. Even though this threshold is very high (requiring a match of almost 50 bases) Trimmomatic is still able to identify very, very short adapter fragments.

There are also two optional parameters for palindrome mode: Min adapter length and Keep both reads. Use the following dialog. To call the dialog press the *Optional* button.



LEADING

This step removes low-quality bases from the beginning. As long as a base has a value below this threshold the base is removed and the next base will be investigated.

Input the following values:

- Quality threshold: the minimum quality required to keep a base.

MAXINFO

This step performs an adaptive quality trim, balancing the benefits of retaining longer reads against the costs of retaining bases with errors. See Trimmomatic manual for details.

Input the following values:

- Target length: the read length which is likely to allow the location of the read within the target sequence. Extremely short reads, which can be placed into many different locations, provide little value. Typically, the length would be in the order of 40 bases, however, the value also depends on the size and complexity of the target sequence.
- Strictness: the balance between preserving as much read length as possible vs. removal of incorrect bases. A low value of this parameter (0.8) favours read correctness.

MINLEN

This step removes reads that fall below the specified minimum length. If required, it should normally be after all other processing steps. Reads removed by this step will be counted and included in the "dropped reads" count.

Input the following values:

- Length: the minimum length of reads to be kept.

SLIDINGWINDOW

This step performs a sliding window trimming, cutting once the average quality within the window falls below a threshold. By considering multiple bases, a single poor quality base will not cause the removal of high-quality data later in the read.

Input the following values:

- Window size: the number of bases to an average across.
- Quality threshold: the average quality required.

TOPHRED33

This step (re)encodes the quality part of the FASTQ file to base 33.

TOPHRED64

This step (re)encodes the quality part of the FASTQ file to base 64.

TRAILING

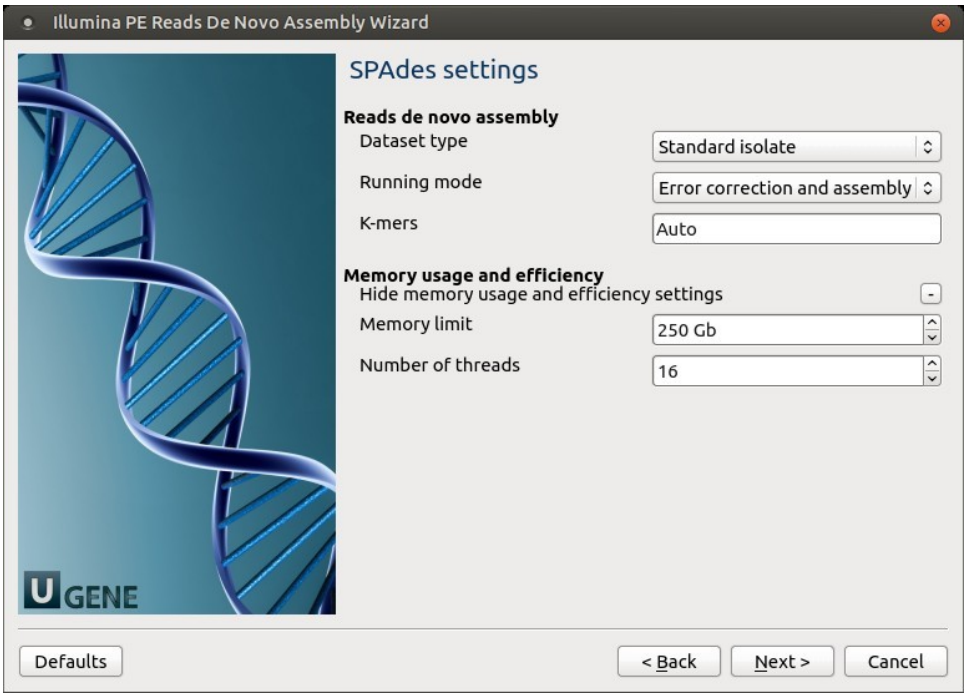
This step removes low-quality bases from the end. As long as a base has a value below this threshold the base is removed and the next base (i. e. the preceding one) will be investigated. This approach can be used removing the special Illumina " low-quality segment" regions (which are marked with a quality score of 2), but SLIDINGWINDOW or MAXINFO are recommended instead.

Input the following values:

- Quality threshold: the minimum quality required to keep a base.

To remove a step use the *Remove selected step* button. The pink highlighting means the required parameter has not been set.

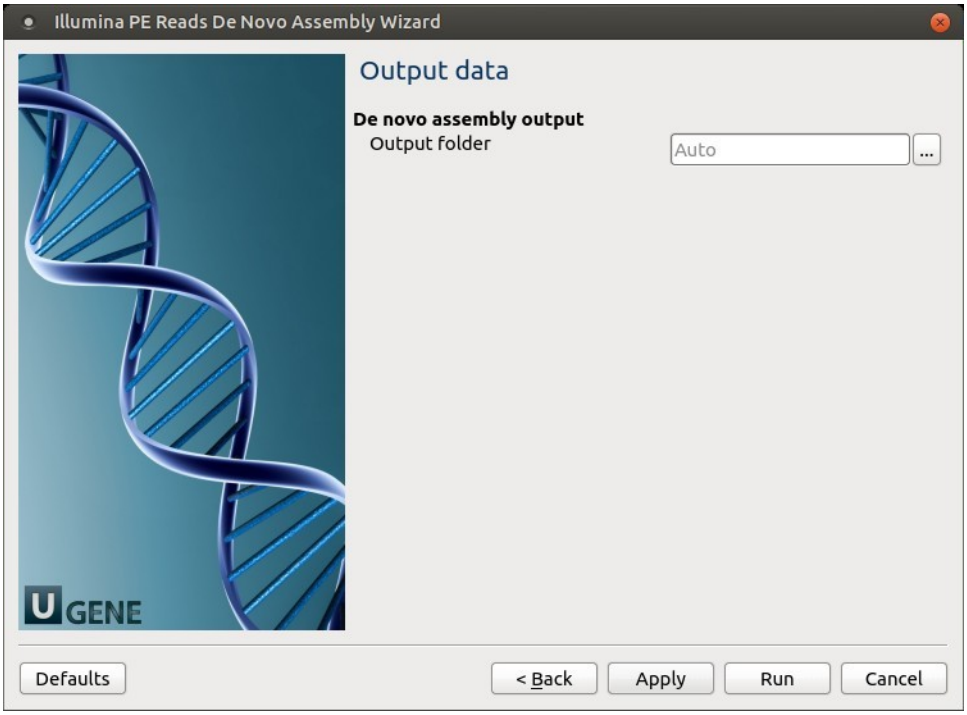
3. SPAdes settings: Default SPAdes parameters can be changed here.



The following parameters are available:

Dataset type	Select the input dataset type: standard isolate (the default value) or multiple displacement amplification (corresponds to --sc).
Running mode	By default, SPAdes performs both read error correction and assembly. You can select leave one of only (corresponds to --only-assembler, --only-error-correction). Error correction is performed using BayesHammer module in case of Illumina input reads andIonHammer in case of IonTorrent data. Note that you should not use error correction in case input reads do not have quality information(e.g. FASTA input files are provided).
K-mers	k-mer sizes (-k).

4. Output Files Page: On this page, you can select an output directory:



De novo Assemble Illumina PE and Nanopore Reads

The workflow sample, described below, takes FASTQ files with paired-end Illumina reads and FASTQ file(s) with Oxford Nanopore reads and assembles these data de novo with SPAdes.

How to Use This Sample

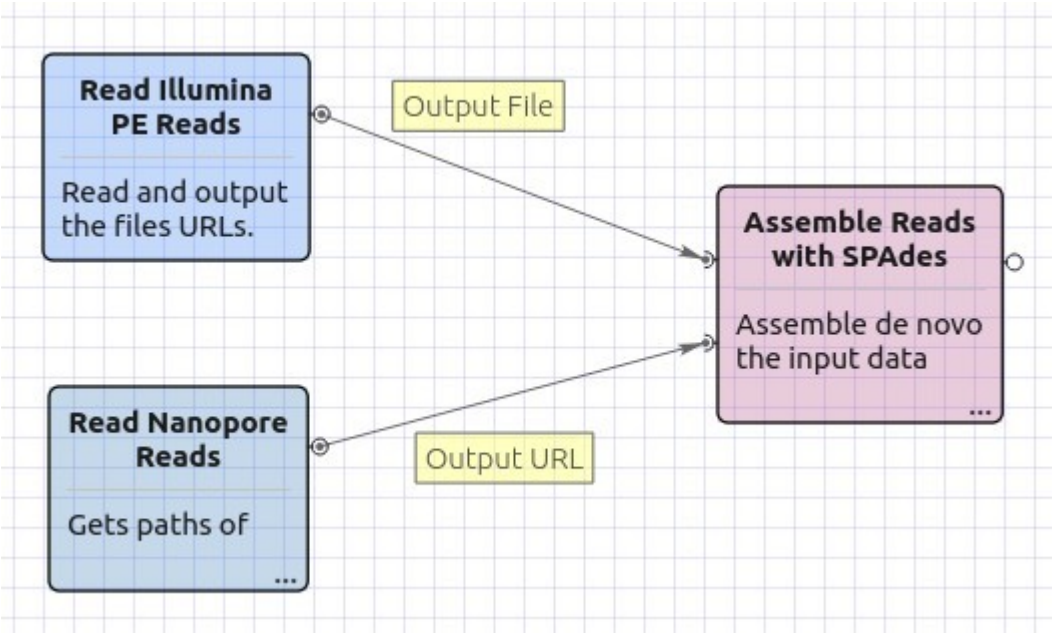
If you haven't used the workflow samples in UGENEbefore, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "De novo Assemble Illumina PE and Nanopore Reads" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

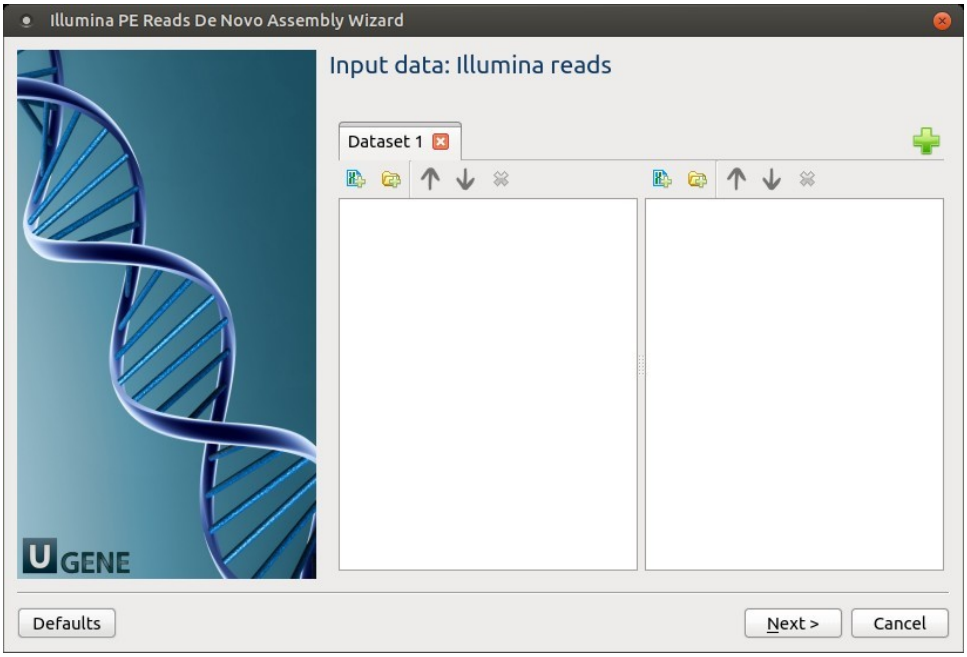
The opened workflow looks as follows:



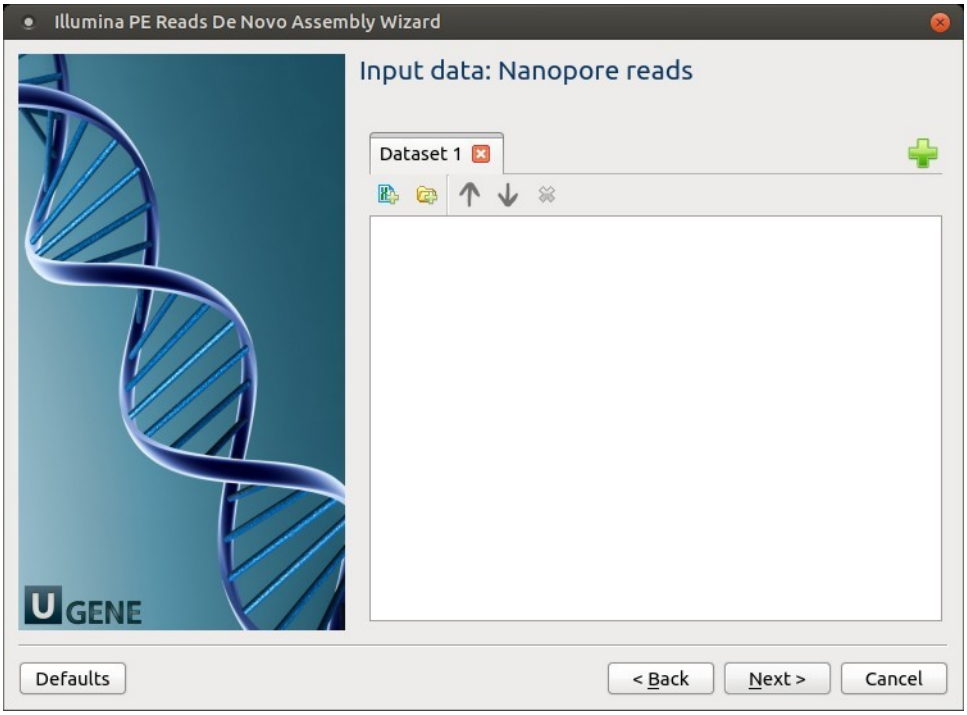
Workflow Wizard

The wizard has 4 pages.

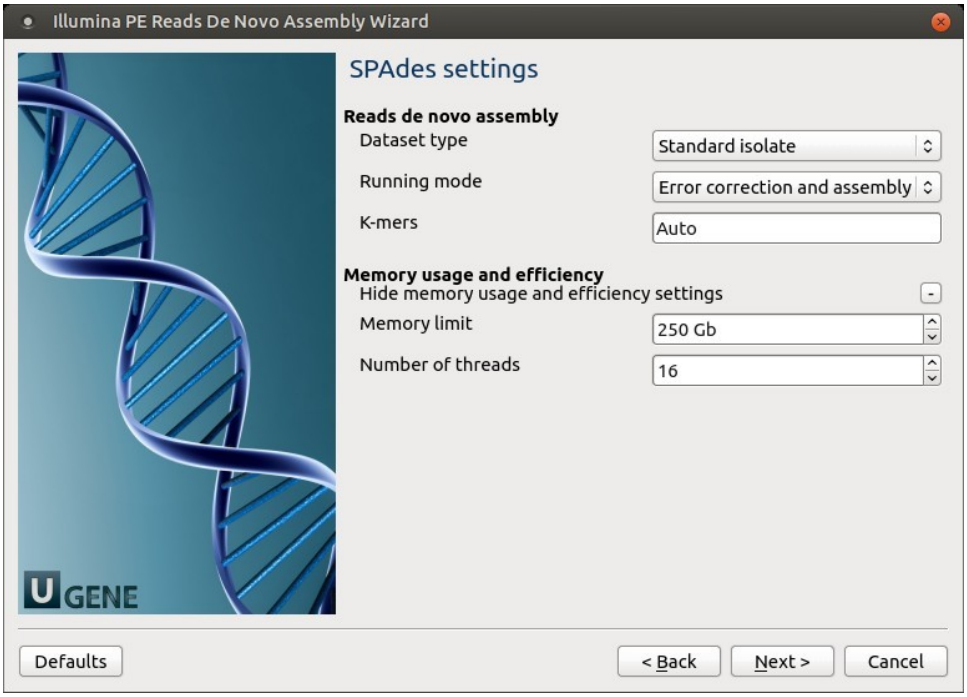
1. Input data: Illumina reads: On this page, files with Illumina reads must be set.



2. Input data: Nanopore reads: The Nanopore reads must be set on this page.



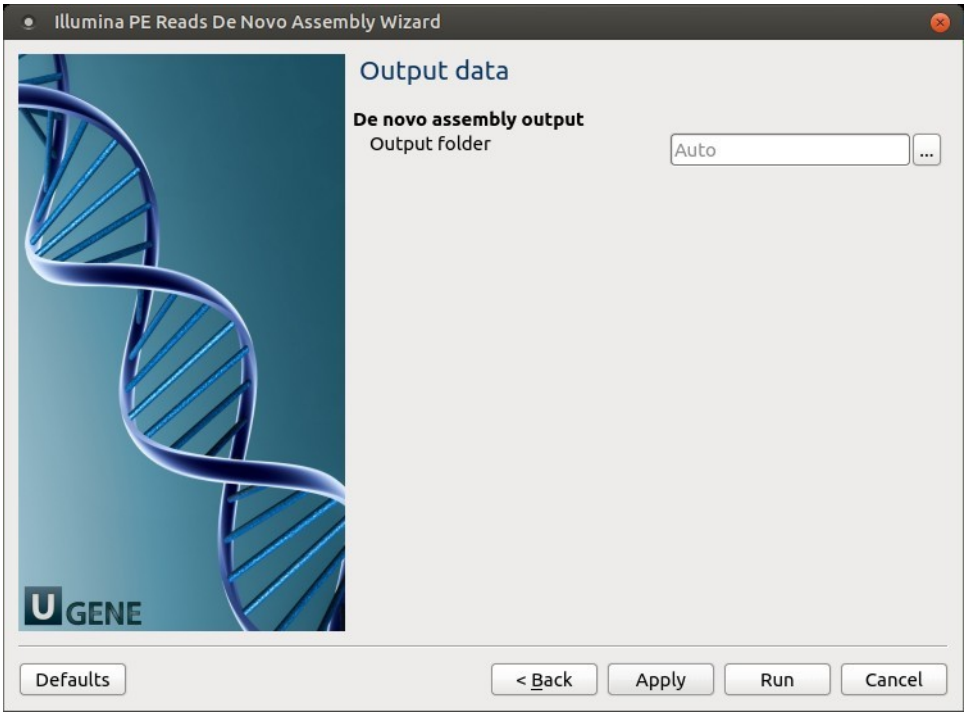
3. SPAdes settings: Default SPAdes parameters can be changed here.



The following parameters are available:

Dataset type	Select the input dataset type: standard isolate (the default value) or multiple displacement amplification (corresponds to --sc).
Running mode	By default, SPAdes performs both read error correction and assembly. You can select leave one of only (corresponds to --only-assembler, --only-error-correction). Error correction is performed using BayesHammer module in case of Illumina input reads andIonHammer in case of IonTorrent data. Note that you should not use error correction in case input reads do not have quality information(e.g. FASTA input files are provided).
K-mers	k-mer sizes (-k).

4. Output Files Page: On this page, you can select an output directory:



De novo Assemble Illumina SE Reads

The workflow sample, described below, takes FASTQ files with single-end Illumina reads as input and process them as follows:

- Improve reads quality with Trimmomatic
- Provide FastQC quality reports
- De novo assemble reads with SPAdes

How to Use This Sample

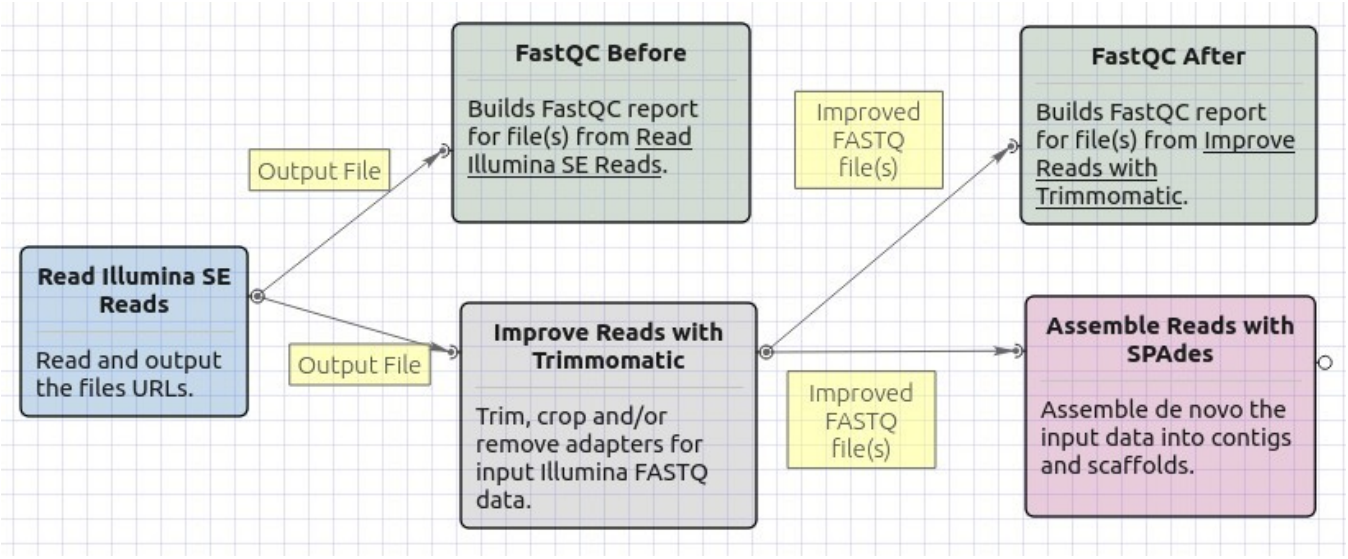
If you haven't used the workflow samples in UGENEbefore, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "De novo Assemble Illumina PE Reads" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

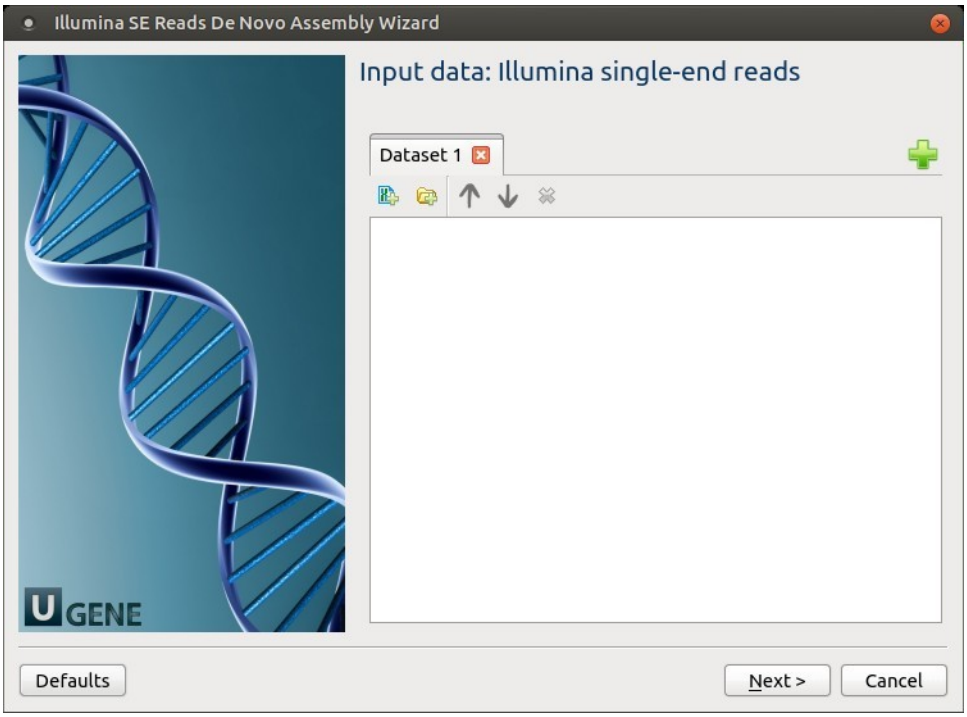
The opened workflow looks as follows:



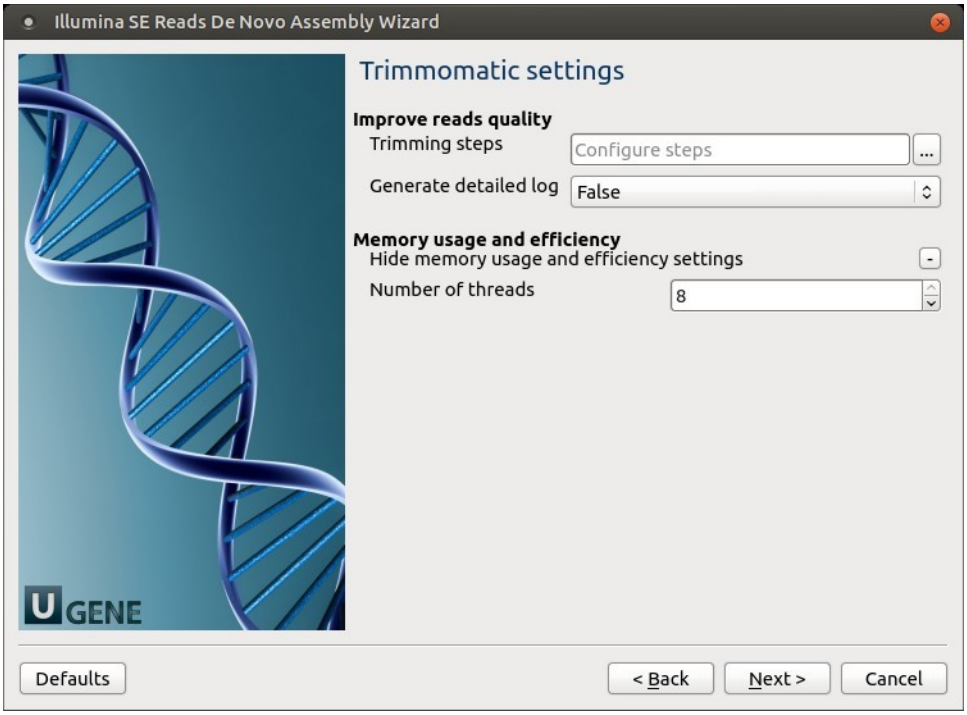
Workflow Wizard

The wizard has 4 pages.

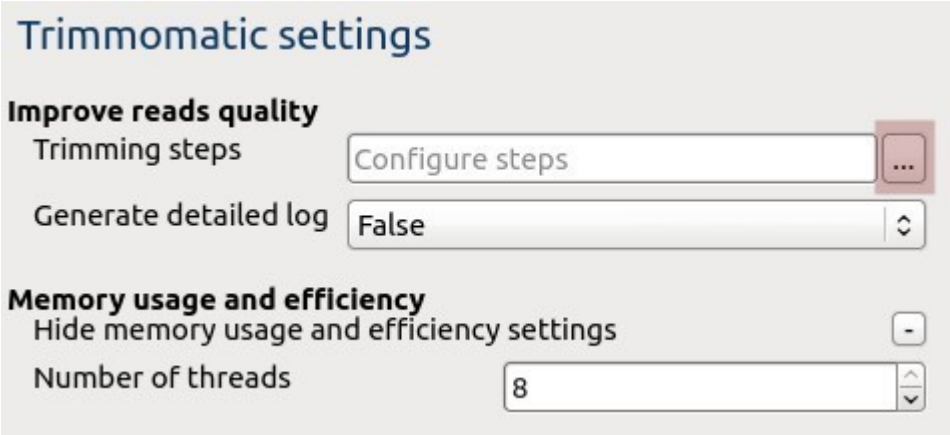
1. Input data: Illumina single-end reads: On this page, files with Illumina single-end reads must be set.



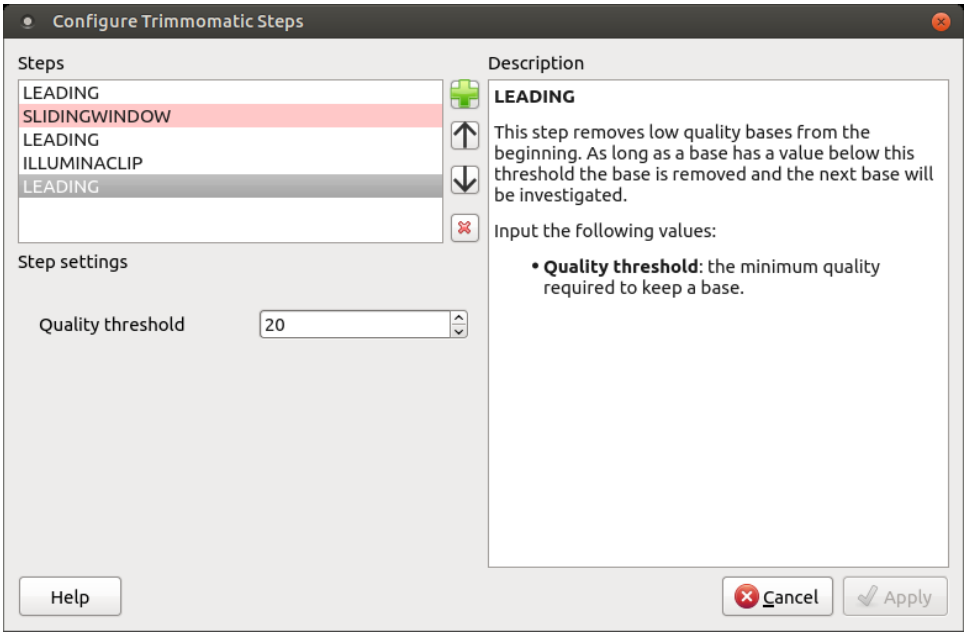
2. Trimmomatic settings: The Trimmomatic parameters can be changed here.



To configure trimming steps use the following button:



The following dialog will appear:



Click the *Add new step* button and select a step. The following options are available:

- ILLUMINACLIP: Cutadapterandotherillumina-specific sequences from the read.
- SLIDINGWINDOW: Perform a sliding window trimming, cutting once the average quality within the window falls below a threshold.
- LEADING: Cut bases off the start of a read, if below a threshold quality.
- TRAILING: Cut bases off the end of a read, if below a threshold quality.
- CROP: Cut the read to a specified length.
- HEADCROP: Cut the specified number of bases from the start of the read.
- MINLEN: Drop the read if it is below a specified length.
- AVGQUAL: Drop the read if the average quality is below the specified level.
- TOPHRED33: Convert quality scores to Phred-33.
- TOPHRED64: Convert quality scores to Phred-64.

Each step has the own parameters:

AVGQUAL

This step drops a read if the average quality is below the specified level.

Input the following values:

- Quality threshold: the minimum average quality required to keep a read.

CROP

This step removes bases regardless of quality from the end of thread, so that the readhas maximally the specified length after this step has been performed. Steps performed after CROP might of course further shorten the read.

Input the following values:

- Length: the number of bases to keep, from the start of the read.

HEADCROP

This step removes the specified number of bases, regardless of quality, from the beginning of the read.

Input the following values:

- Length: the number of bases to remove from the start of the read.

ILLUMINACLIP

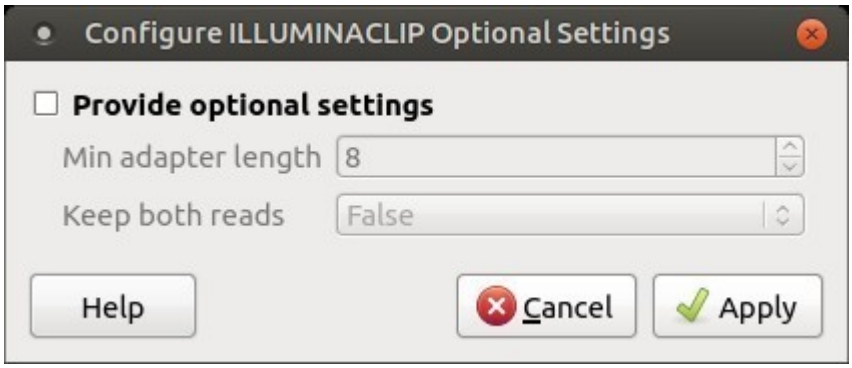
This step is used to find and remove Illumina adapters.

Trimmomatic first compares short sections of an adapter and a read. If they match enough, the entire alignment between the read and adapter is scored. For paired-end reads, the "palindrome" approach is also used to improve the result. See Trimmomatic manual for details.

Input the following values:

- Adapter sequences: a FASTA file with the adapter sequences. Files for TruSeq2 (GAII machines), TruSeq3 (HiSeq and MiSeq machines) and Nextera kits for SE and PE reads are now available by default. The naming of the various sequences within the specified file determines how they are used.
- Seed mismatches: the maximum mismatch count in short sections which will still allow a full match to be performed.
- Simple clip threshold: a threshold for simple alignment mode. Values between 7 and 15 are recommended. A perfect match of a 12 base sequence will score just over 7, while 25 bases are needed to score 15.
- Palindrome clip threshold: a threshold for palindrome alignment mode. For palindromic matches, a longer alignment is possible. Therefore the threshold can be in the range of 30. Even though this threshold is very high (requiring a match of almost 50 bases) Trimmomatic is still able to identify very, very short adapter fragments.

There are also two optional parameters for palindrome mode: Min adapter length and Keep both reads. Use the following dialog. To call the dialog press the *Optional* button.



LEADING

This step removes low-quality bases from the beginning. As long as a base has a value below this threshold the base is removed and the next base will be investigated.

Input the following values:

- Quality threshold: the minimum quality required to keep a base.

MAXINFO

This step performs an adaptive quality trim, balancing the benefits of retaining longer reads against the costs of retaining bases with errors. See Trimmomatic manual for details.

Input the following values:

- Target length: the read length which is likely to allow the location of the read within the target sequence. Extremely short reads, which can be placed into many different locations, provide little value. Typically, the length would be in the order of 40 bases, however, the value also depends on the size and complexity of the target sequence.
- Strictness: the balance between preserving as much read length as possible vs. removal of incorrect bases. A low value of this parameter (0.8) favours read correctness.

MINLEN

This step removes reads that fall below the specified minimum length. If required, it should normally be after all other processing steps. Reads removed by this step will be counted and included in the "dropped reads" count.

Input the following values:

- Length: the minimum length of reads to be kept.

SLIDINGWINDOW

This step performs a sliding window trimming, cutting once the average quality within the window falls below a threshold. By considering multiple bases, a single poor quality base will not cause the removal of high-quality data later in the read.

Input the following values:

- Window size: the number of bases to an average across.
- Quality threshold: the average quality required.

TOPHRED33

This step (re)encodes the quality part of the FASTQ file to base 33.

TOPHRED64

This step (re)encodes the quality part of the FASTQ file to base 64.

TRAILING

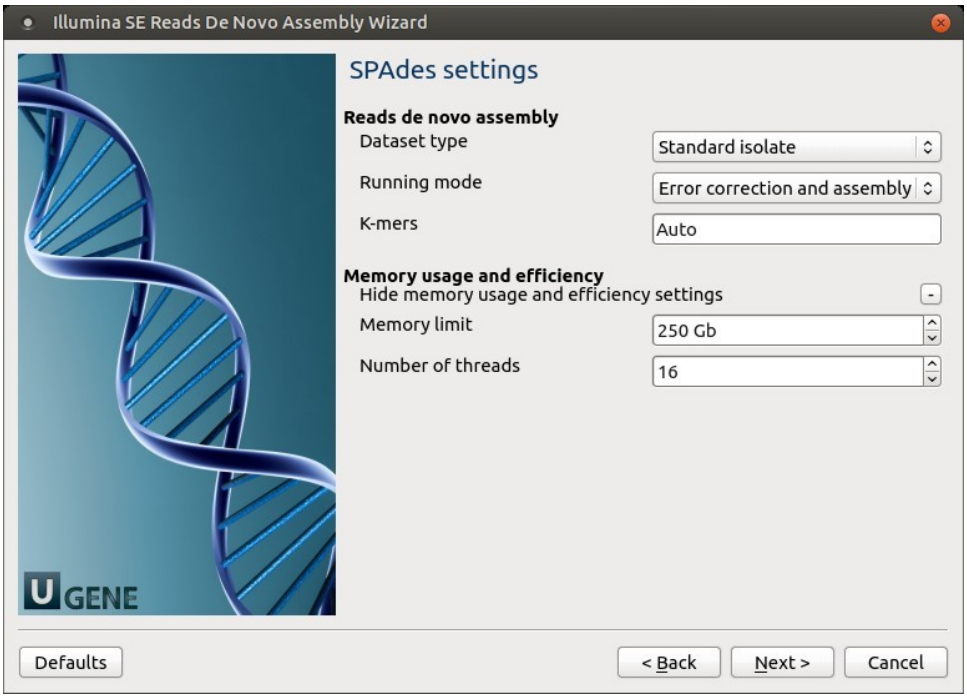
This step removes low-quality bases from the end. As long as a base has a value below this threshold the base is removed and the next base (i. e. the preceding one) will be investigated. This approach can be used removing the special Illumina " low-quality segment" regions (which are marked with a quality score of 2), but SLIDINGWINDOW or MAXINFO are recommended instead.

Input the following values:

- Quality threshold: the minimum quality required to keep a base.

To remove a step use the *Remove selected step* button. The pink highlighting means the required parameter has not been set.

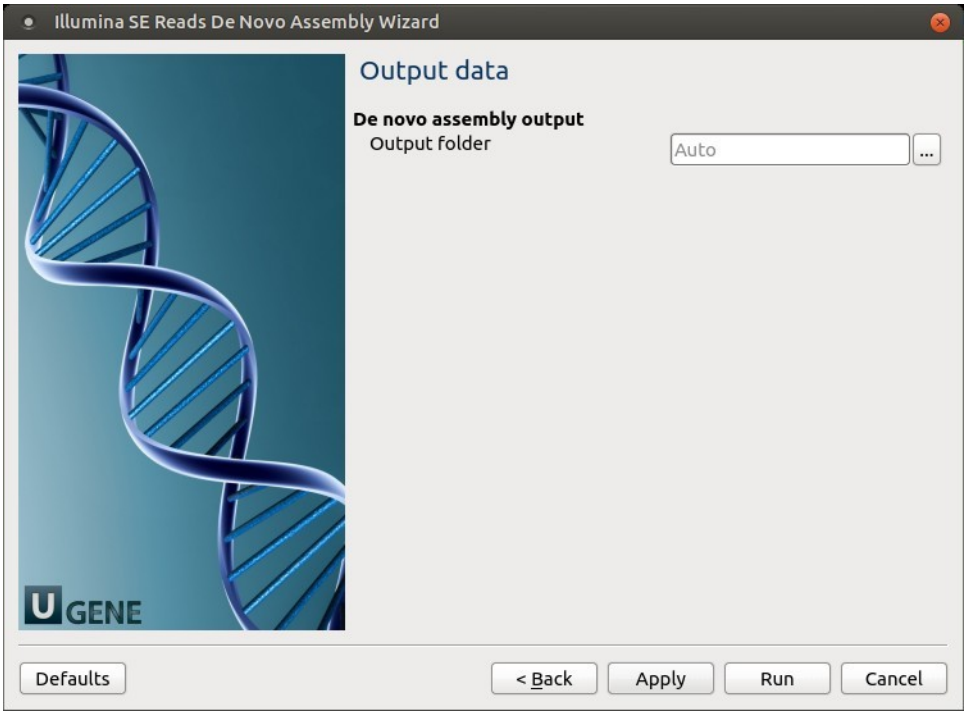
3. SPAdes settings: Default SPAdes parameters can be changed here.



The following parameters are available:

Dataset type	Select the input dataset type: standard isolate (the default value) or multiple displacement amplification (corresponds to --sc).
Running mode	By default, SPAdes performs both read error correction and assembly. You can select leave one of only (corresponds to --only-assembler, --only-error-correction). Error correction is performed using BayesHammer module in case of Illumina input reads andIonHammer in case of IonTorrent data. Note that you should not use error correction in case input reads do not have quality information(e.g. FASTA input files are provided).
K-mers	k-mer sizes (-k).

4. Output Files Page: On this page, you can select an output directory:



De Novo Assembly and Contigs Classification

The workflow sample, described below, takes FASTQ files with metagenomic NGS reads as input and process them as follows:

- Improve reads quality with Trimmomatic
- Provide FastQC reads quality reports
- De novo assembly:
  Assemble the reads into contigs with SPAdes
- Classification:
  Classify the assembled contigs with Kraken
- Provide general classification report

How to Use This Sample

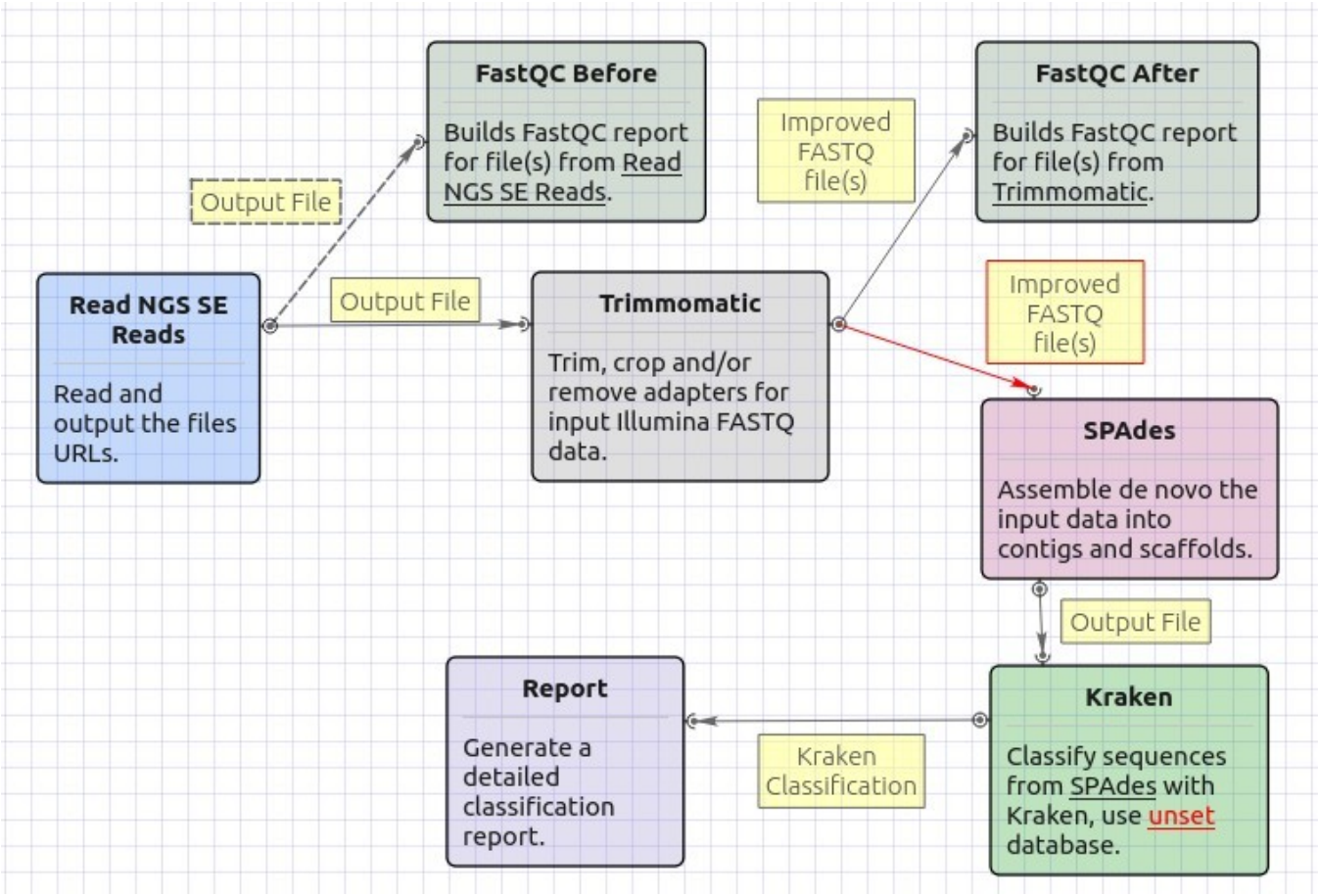
If you haven't used the workflow samples in UGENEbefore, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

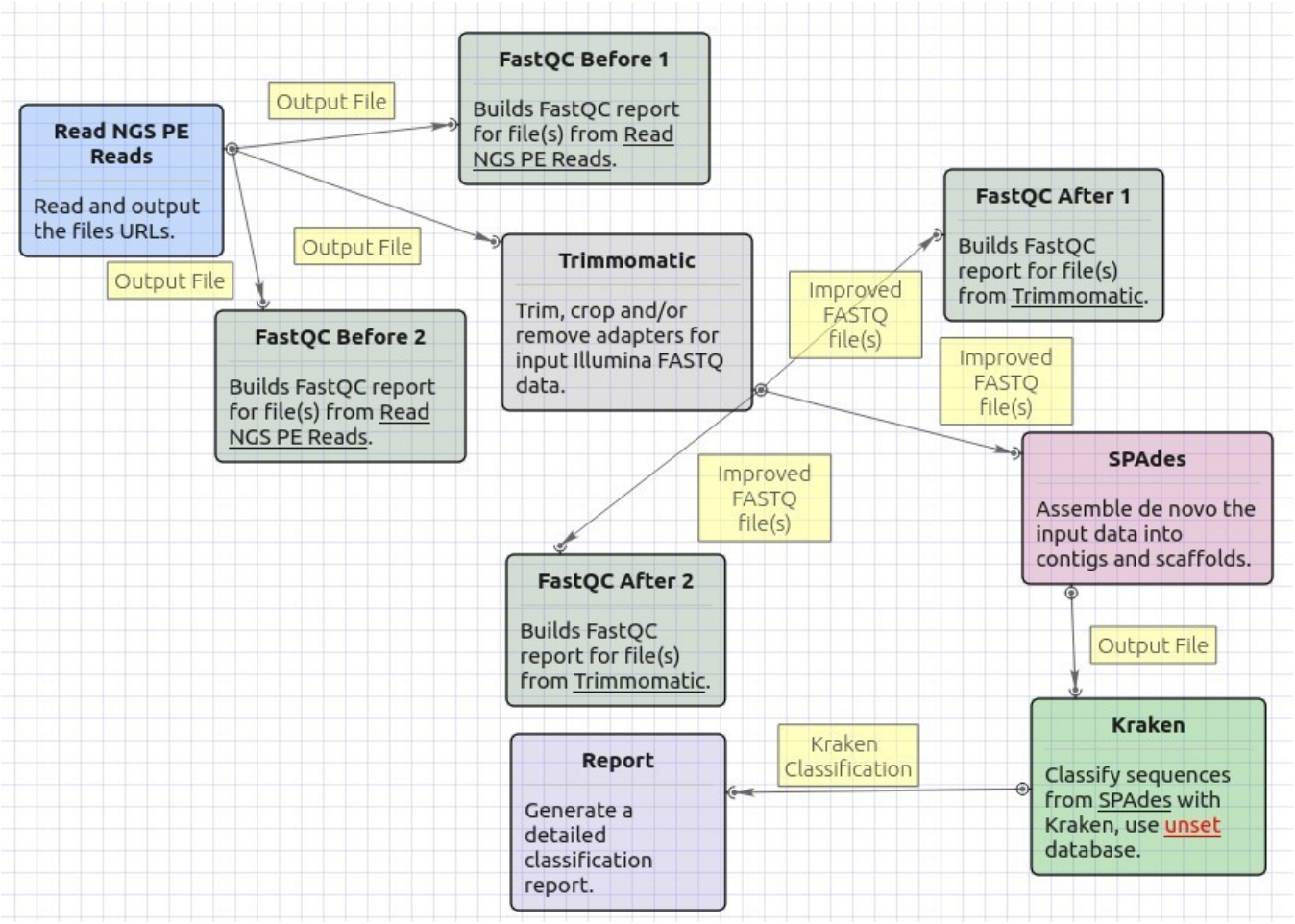
The workflow sample "De Novo Assembly and Contigs Classification" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

The opened workflow for single-end reads looks as follows:



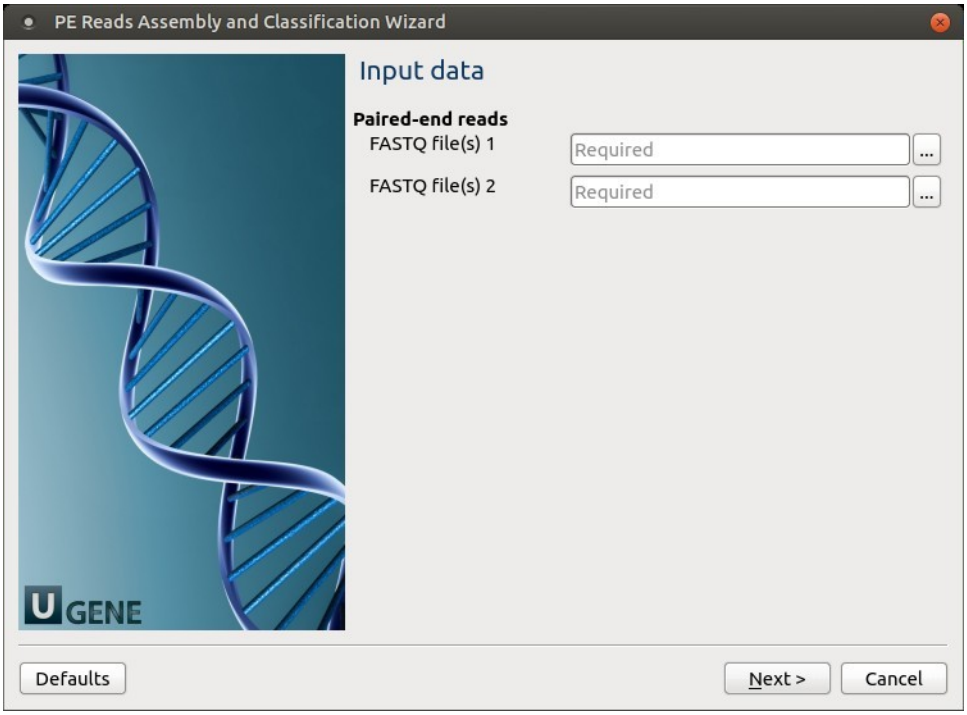
The opened workflow for paired-end reads looks as follows:



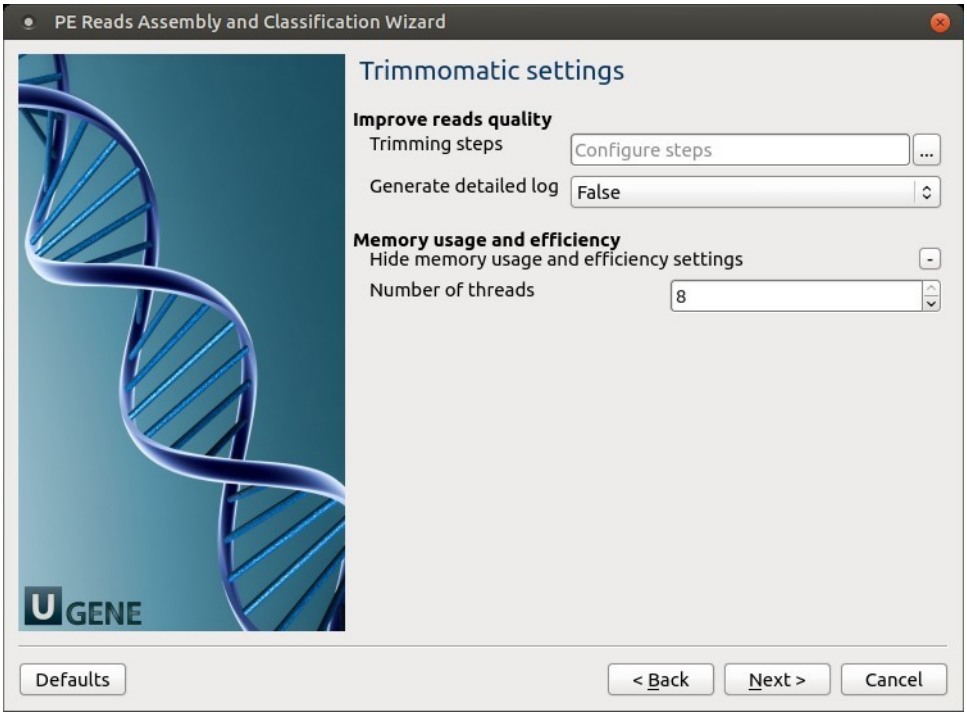
Workflow Wizard

The wizard has 5 pages.

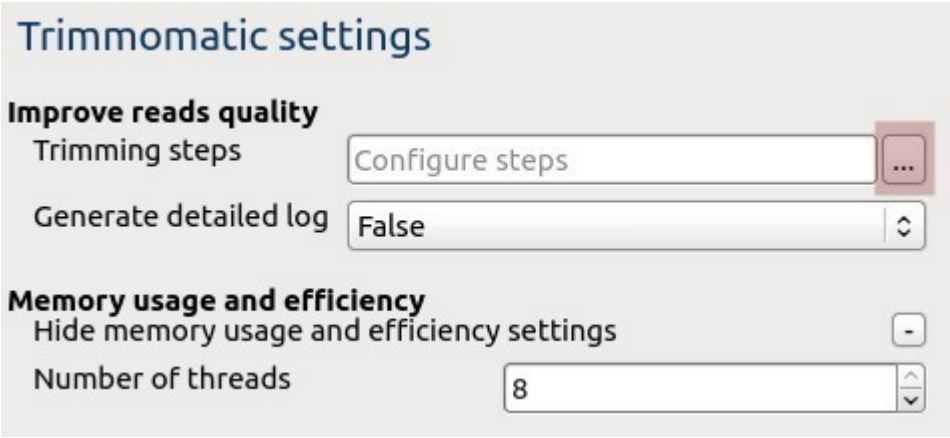
1. Input data: On this page, input files must be set.



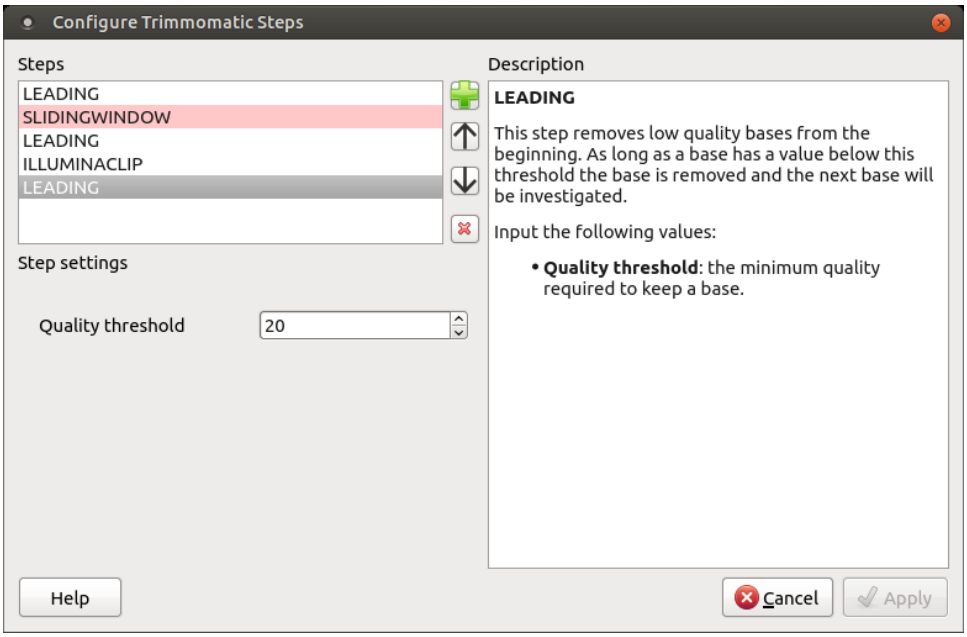
2. Trimmomatic settings: The Trimmomatic parameters can be changed here.



To configure trimming steps use the following button:



The following dialog will appear:



Click the *Add new step* button and select a step. The following options are available:

- ILLUMINACLIP: Cutadapterandotherillumina-specific sequences from the read.
- SLIDINGWINDOW: Perform a sliding window trimming, cutting once the average quality within the window falls below a threshold.
- LEADING: Cut bases off the start of a read, if below a threshold quality.
- TRAILING: Cut bases off the end of a read, if below a threshold quality.
- CROP: Cut the read to a specified length.
- HEADCROP: Cut the specified number of bases from the start of the read.
- MINLEN: Drop the read if it is below a specified length.
- AVGQUAL: Drop the read if the average quality is below the specified level.
- TOPHRED33: Convert quality scores to Phred-33.
- TOPHRED64: Convert quality scores to Phred-64.

Each step has the own parameters:

AVGQUAL

This step drops a read if the average quality is below the specified level.

Input the following values:

- Quality threshold: the minimum average quality required to keep a read.

CROP

This step removes bases regardless of quality from the end of thread, so that the readhas maximally the specified length after this step has been performed. Steps performed after CROP might of course further shorten the read.

Input the following values:

- Length: the number of bases to keep, from the start of the read.

HEADCROP

This step removes the specified number of bases, regardless of quality, from the beginning of the read.

Input the following values:

- Length: the number of bases to remove from the start of the read.

ILLUMINACLIP

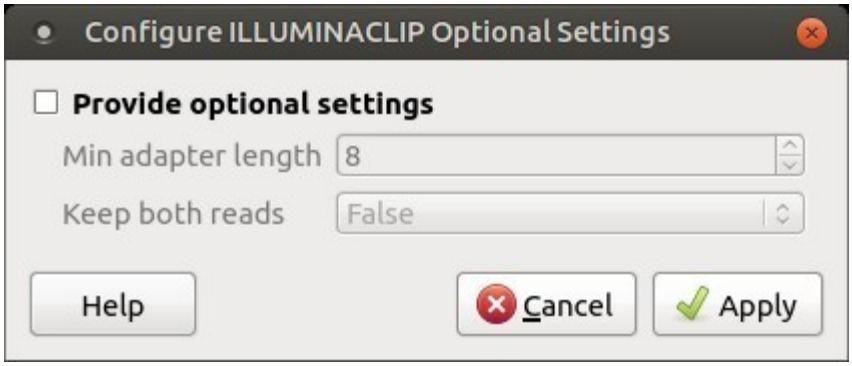
This step is used to find and remove Illumina adapters.

Trimmomatic first compares short sections of an adapter and a read. If they match enough, the entire alignment between the read and adapter is scored. For paired-end reads, the "palindrome" approach is also used to improve the result. See Trimmomatic manual for details.

Input the following values:

- Adapter sequences: a FASTA file with the adapter sequences. Files for TruSeq2 (GAII machines), TruSeq3 (HiSeq and MiSeq machines) and Nextera kits for SE and PE reads are now available by default. The naming of the various sequences within the specified file determines how they are used.
- Seed mismatches: the maximum mismatch count in short sections which will still allow a full match to be performed.
- Simple clip threshold: a threshold for simple alignment mode. Values between 7 and 15 are recommended. A perfect match of a 12 base sequence will score just over 7, while 25 bases are needed to score 15.
- Palindrome clip threshold: a threshold for palindrome alignment mode. For palindromic matches, a longer alignment is possible. Therefore the threshold can be in the range of 30. Even though this threshold is very high (requiring a match of almost 50 bases) Trimmomatic is still able to identify very, very short adapter fragments.

There are also two optional parameters for palindrome mode: Min adapter length and Keep both reads. Use the following dialog. To call the dialog press the *Optional* button.



LEADING

This step removes low-quality bases from the beginning. As long as a base has a value below this threshold the base is removed and the next base will be investigated.

Input the following values:

- Quality threshold: the minimum quality required to keep a base.

MAXINFO

This step performs an adaptive quality trim, balancing the benefits of retaining longer reads against the costs of retaining bases with errors. See Trimmomatic manual for details.

Input the following values:

- Target length: the read length which is likely to allow the location of the read within the target sequence. Extremely short reads, which can be placed into many different locations, provide little value. Typically, the length would be in the order of 40 bases, however, the value also depends on the size and complexity of the target sequence.
- Strictness: the balance between preserving as much read length as possible vs. removal of incorrect bases. A low value of this parameter (0.8) favours read correctness.

MINLEN

This step removes reads that fall below the specified minimum length. If required, it should normally be after all other processing steps. Reads removed by this step will be counted and included in the "dropped reads" count.

Input the following values:

- Length: the minimum length of reads to be kept.

SLIDINGWINDOW

This step performs a sliding window trimming, cutting once the average quality within the window falls below a threshold. By considering multiple bases, a single poor quality base will not cause the removal of high-quality data later in the read.

Input the following values:

- Window size: the number of bases to an average across.
- Quality threshold: the average quality required.

TOPHRED33

This step (re)encodes the quality part of the FASTQ file to base 33.

TOPHRED64

This step (re)encodes the quality part of the FASTQ file to base 64.

TRAILING

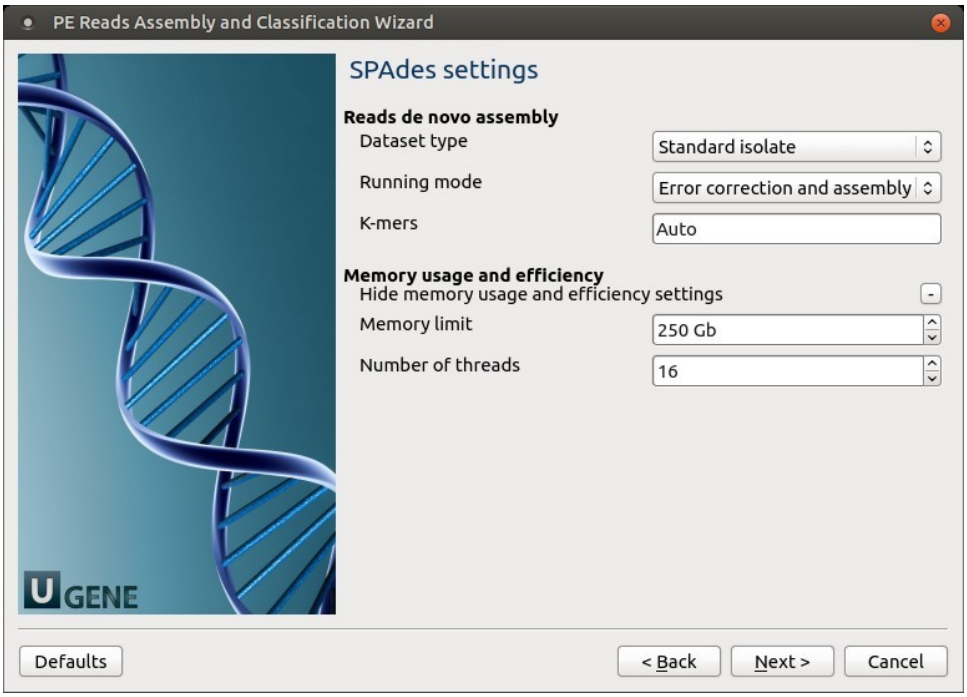
This step removes low-quality bases from the end. As long as a base has a value below this threshold the base is removed and the next base (i. e. the preceding one) will be investigated. This approach can be used removing the special Illumina " low-quality segment" regions (which are marked with a quality score of 2), but SLIDINGWINDOW or MAXINFO are recommended instead.

Input the following values:

- Quality threshold: the minimum quality required to keep a base.

To remove a step use the *Remove selected step* button. The pink highlighting means the required parameter has not been set.

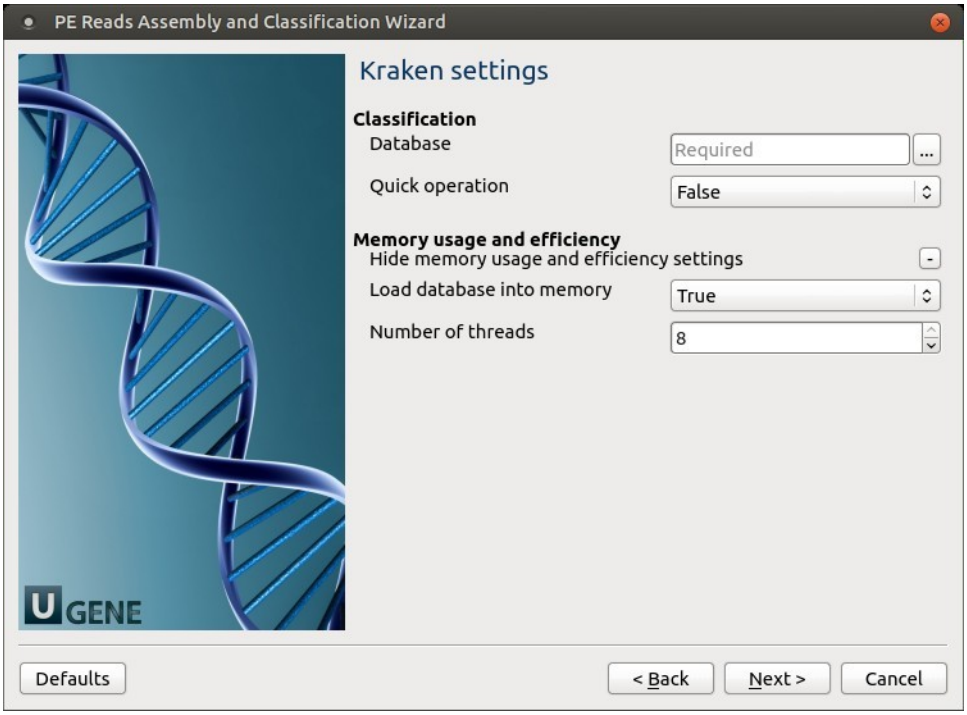
3. SPAdes settings: Default SPAdes parameters can be changed here.



The following parameters are available:

Dataset type	Select the input dataset type: standard isolate (the default value) or multiple displacement amplification (corresponds to --sc).
Running mode	By default, SPAdes performs both read error correction and assembly. You can select leave one of only (corresponds to --only-assembler, --only-error-correction). Error correction is performed using BayesHammer module in case of Illumina input reads andIonHammer in case of IonTorrent data. Note that you should not use error correction in case input reads do not have quality information(e.g. FASTA input files are provided).
K-mers	k-mer sizes (-k).

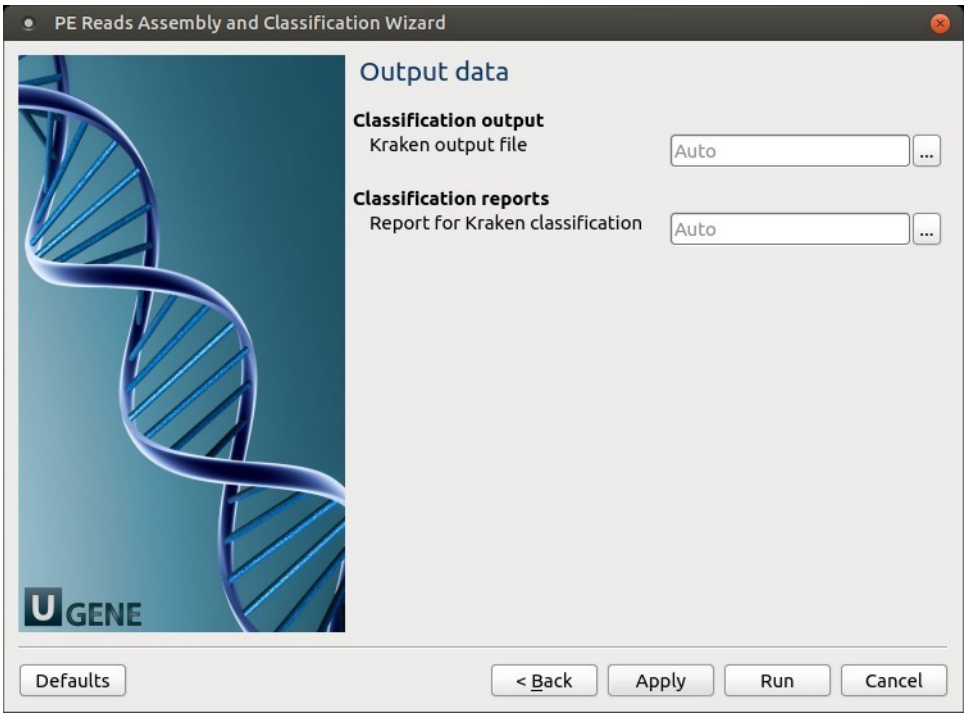
4. Kraken settings: Default Kraken parameters can be changed here.



The following parameters are available:

Database	A path to the folder with the Kraken database files.
Quick operation	Stop classification of an input read after the certain number of hits. The value can be specified in the "Minimum number of hits" parameter.

5. Output Files Page: On this page, you can select an output directory:



Parallel NGS Reads Classification

The workflow sample, described below, takes FASTQ files with metagenomic NGS reads as input and process them as follows:

- Improve reads quality with Trimmomatic
- Provide FastQC reads quality reports
- Classification:
  1. Classify the pre-processed reads with Kraken
  2. Classify the assembled contigs with CLARK
  3. Classify these reads with DIAMOND (in case of SE reads)
  4. Classify these reads with MetaPhlAn2
  5. Ensemble the classification output from Kraken, CLARK, and DIAMOND into a CSV file.
  6. Improve classification from these tools with WEVOTE.
  7. Provide general classification reports

How to Use This Sample

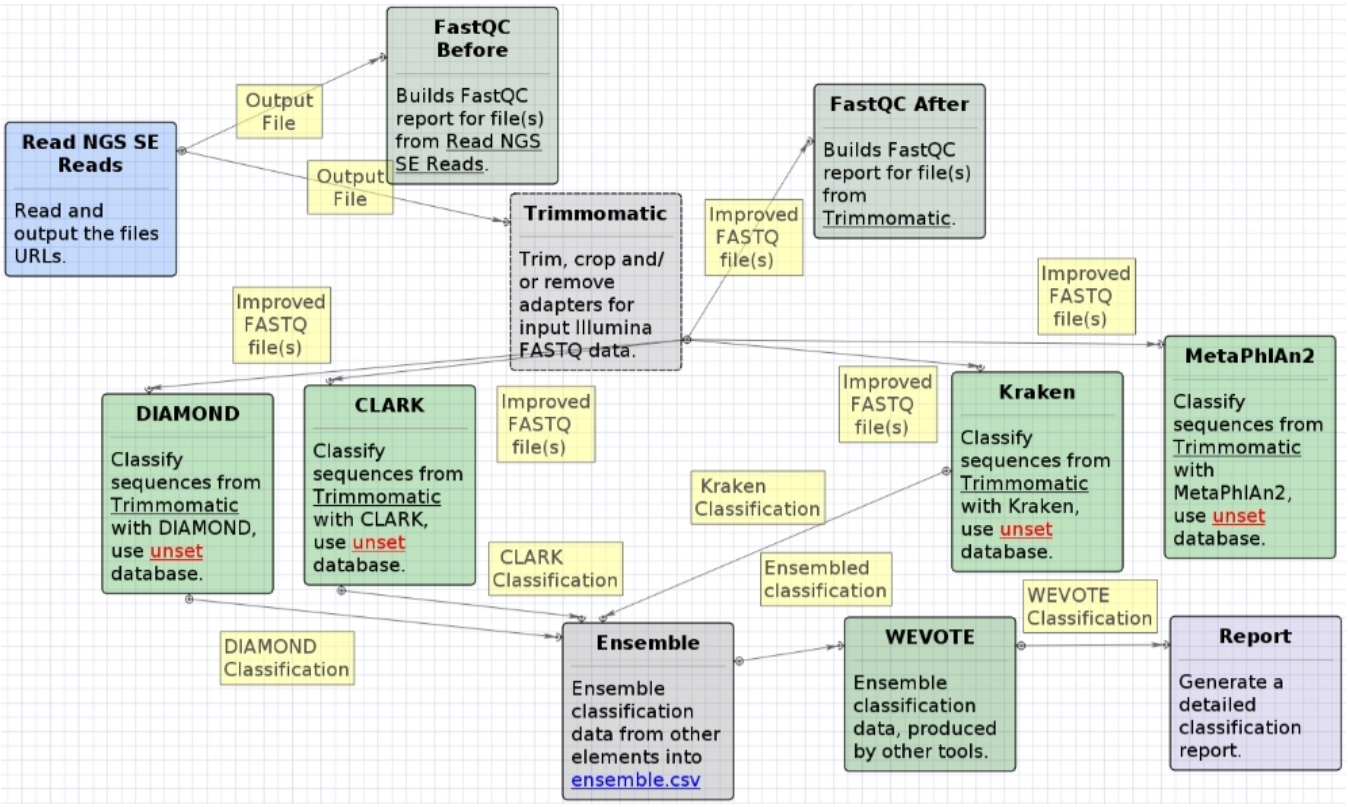
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

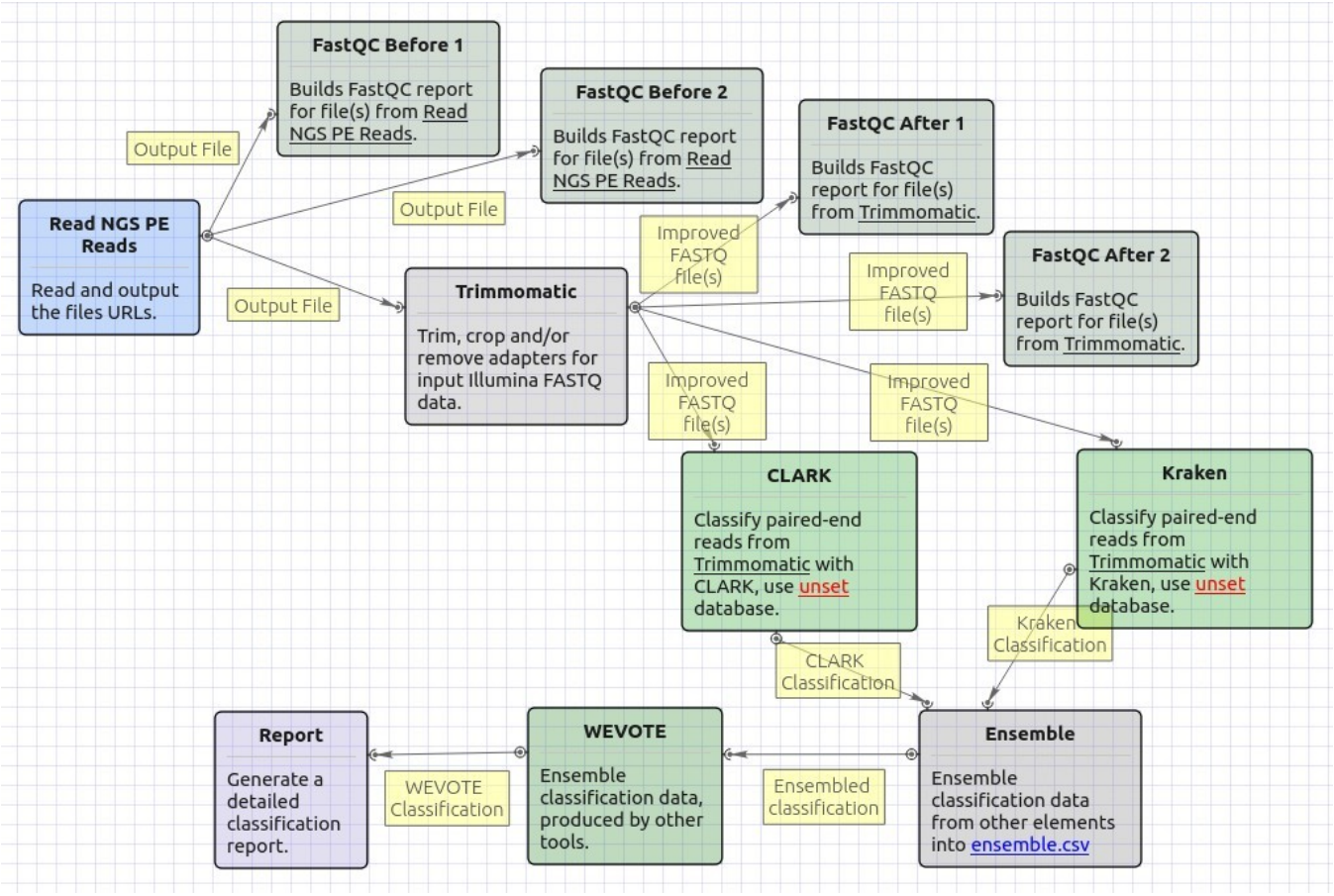
The workflow sample "Parallel NGS Reads Classification" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

The opened workflow for single-end reads looks as follows:



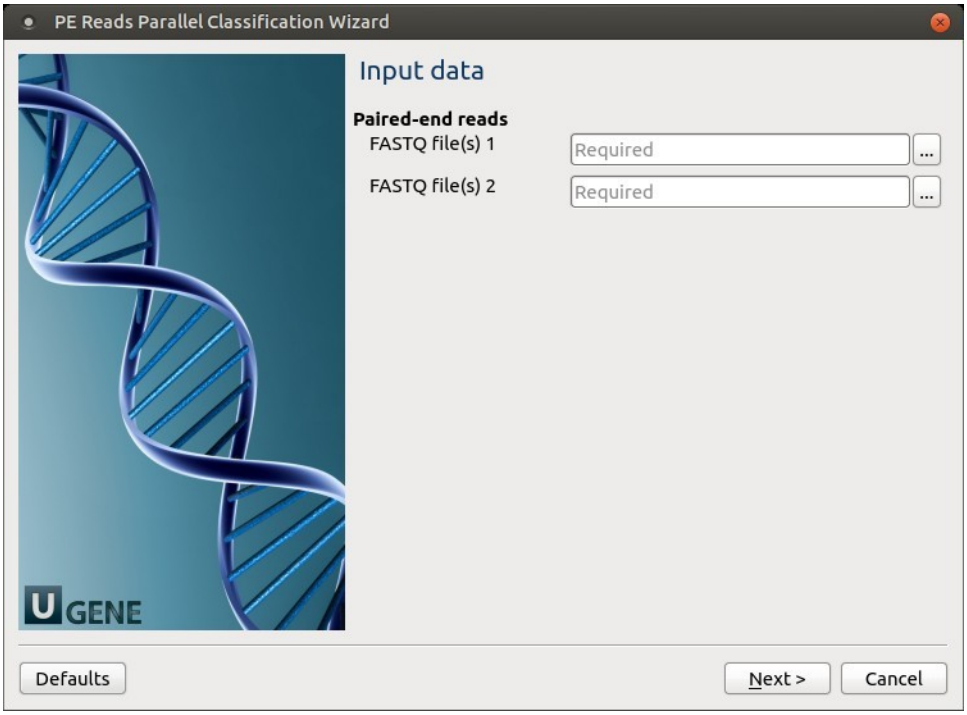
The opened workflow for paired-end reads looks as follows:



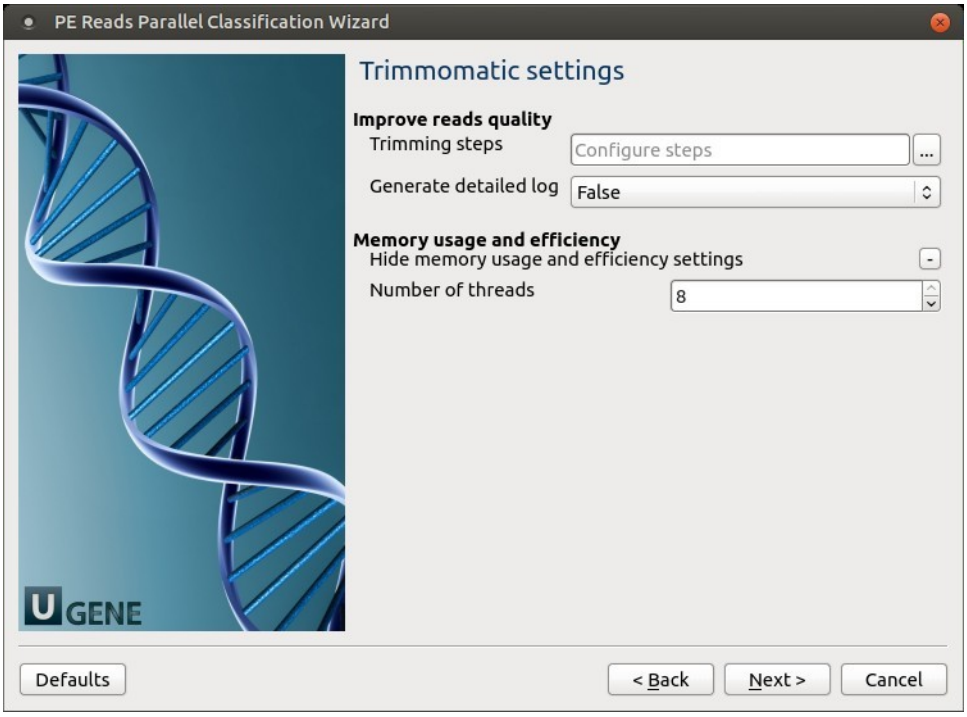
Workflow Wizard

The wizard has 6 pages.

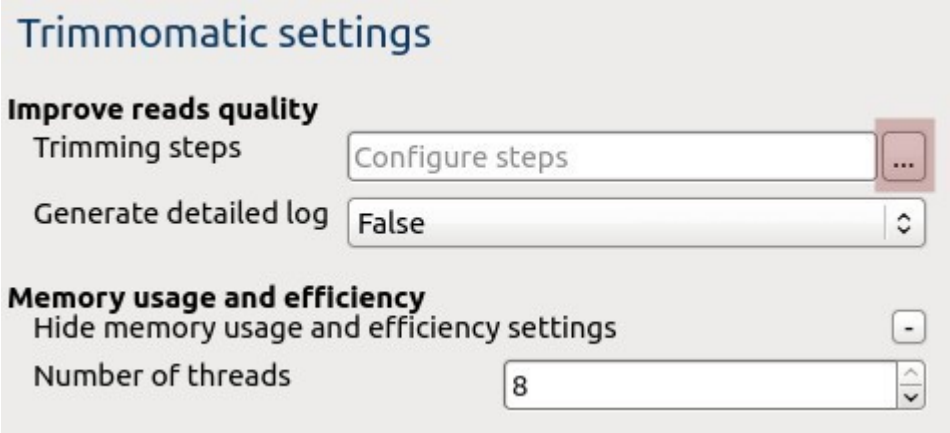
1. Input data: On this page, input files must be set.



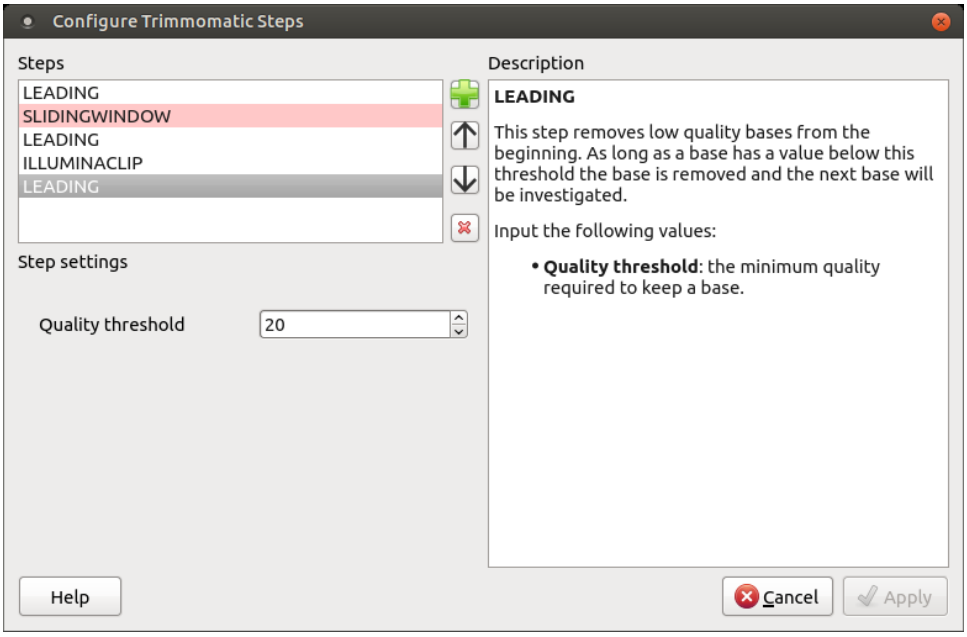
2. Trimmomatic settings: The Trimmomatic parameters can be changed here.



To configure trimming steps use the following button:



The following dialog will appear:



Click the *Add new step* button and select a step. The following options are available:

- ILLUMINACLIP: Cutadapterandotherillumina-specific sequences from the read.
- SLIDINGWINDOW: Perform a sliding window trimming, cutting once the average quality within the window falls below a threshold.
- LEADING: Cut bases off the start of a read, if below a threshold quality.
- TRAILING: Cut bases off the end of a read, if below a threshold quality.
- CROP: Cut the read to a specified length.
- HEADCROP: Cut the specified number of bases from the start of the read.
- MINLEN: Drop the read if it is below a specified length.
- AVGQUAL: Drop the read if the average quality is below the specified level.
- TOPHRED33: Convert quality scores to Phred-33.
- TOPHRED64: Convert quality scores to Phred-64.

Each step has the own parameters:

AVGQUAL

This step drops a read if the average quality is below the specified level.

Input the following values:

- Quality threshold: the minimum average quality required to keep a read.

CROP

This step removes bases regardless of quality from the end of thread, so that the readhas maximally the specified length after this step has been performed. Steps performed after CROP might of course further shorten the read.

Input the following values:

- Length: the number of bases to keep, from the start of the read.

HEADCROP

This step removes the specified number of bases, regardless of quality, from the beginning of the read.

Input the following values:

- Length: the number of bases to remove from the start of the read.

ILLUMINACLIP

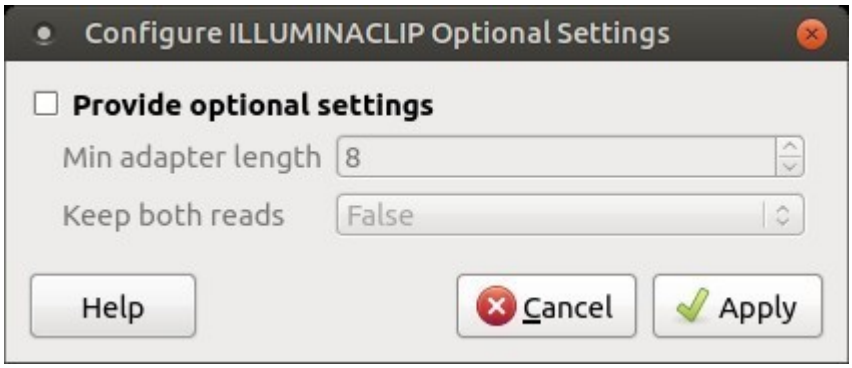
This step is used to find and remove Illumina adapters.

Trimmomatic first compares short sections of an adapter and a read. If they match enough, the entire alignment between the read and adapter is scored. For paired-end reads, the "palindrome" approach is also used to improve the result. See Trimmomatic manual for details.

Input the following values:

- Adapter sequences: a FASTA file with the adapter sequences. Files for TruSeq2 (GAII machines), TruSeq3 (HiSeq and MiSeq machines) and Nextera kits for SE and PE reads are now available by default. The naming of the various sequences within the specified file determines how they are used.
- Seed mismatches: the maximum mismatch count in short sections which will still allow a full match to be performed.
- Simple clip threshold: a threshold for simple alignment mode. Values between 7 and 15 are recommended. A perfect match of a 12 base sequence will score just over 7, while 25 bases are needed to score 15.
- Palindrome clip threshold: a threshold for palindrome alignment mode. For palindromic matches, a longer alignment is possible. Therefore the threshold can be in the range of 30. Even though this threshold is very high (requiring a match of almost 50 bases) Trimmomatic is still able to identify very, very short adapter fragments.

There are also two optional parameters for palindrome mode: Min adapter length and Keep both reads. Use the following dialog. To call the dialog press the *Optional* button.



LEADING

This step removes low-quality bases from the beginning. As long as a base has a value below this threshold the base is removed and the next base will be investigated.

Input the following values:

- Quality threshold: the minimum quality required to keep a base.

MAXINFO

This step performs an adaptive quality trim, balancing the benefits of retaining longer reads against the costs of retaining bases with errors. See Trimmomatic manual for details.

Input the following values:

- Target length: the read length which is likely to allow the location of the read within the target sequence. Extremely short reads, which can be placed into many different locations, provide little value. Typically, the length would be in the order of 40 bases, however, the value also depends on the size and complexity of the target sequence.
- Strictness: the balance between preserving as much read length as possible vs. removal of incorrect bases. A low value of this parameter (0.8) favours read correctness.

MINLEN

This step removes reads that fall below the specified minimum length. If required, it should normally be after all other processing steps. Reads removed by this step will be counted and included in the "dropped reads" count.

Input the following values:

- Length: the minimum length of reads to be kept.

SLIDINGWINDOW

This step performs a sliding window trimming, cutting once the average quality within the window falls below a threshold. By considering multiple bases, a single poor quality base will not cause the removal of high-quality data later in the read.

Input the following values:

- Window size: the number of bases to an average across.
- Quality threshold: the average quality required.

TOPHRED33

This step (re)encodes the quality part of the FASTQ file to base 33.

TOPHRED64

This step (re)encodes the quality part of the FASTQ file to base 64.

TRAILING

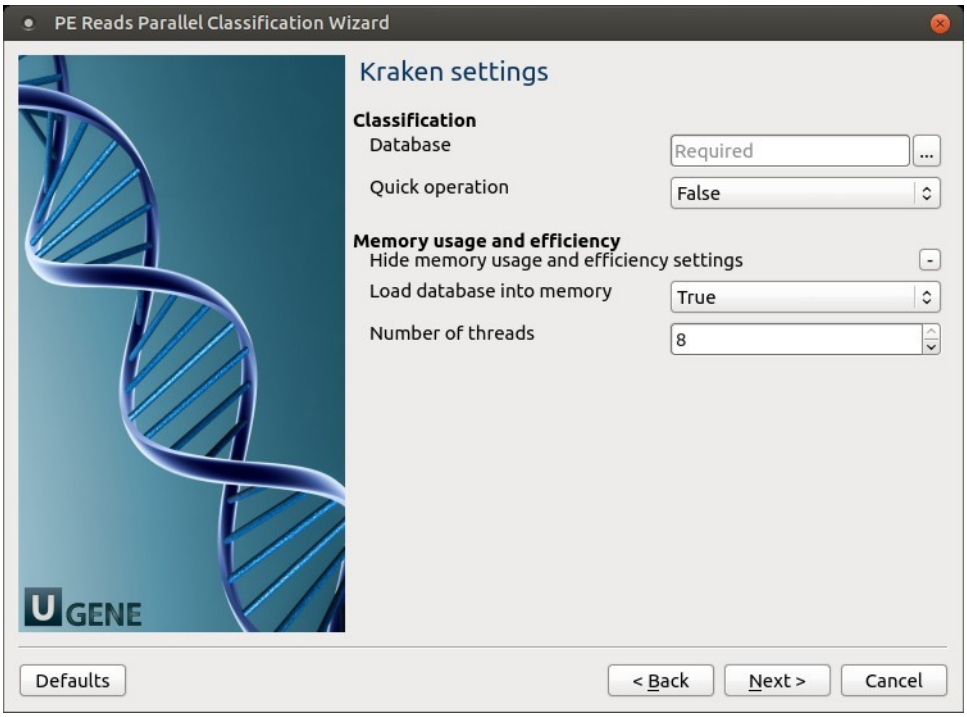
This step removes low-quality bases from the end. As long as a base has a value below this threshold the base is removed and the next base (i. e. the preceding one) will be investigated. This approach can be used removing the special Illumina " low-quality segment" regions (which are marked with a quality score of 2), but SLIDINGWINDOW or MAXINFO are recommended instead.

Input the following values:

- Quality threshold: the minimum quality required to keep a base.

To remove a step use the *Remove selected step* button. The pink highlighting means the required parameter has not been set.

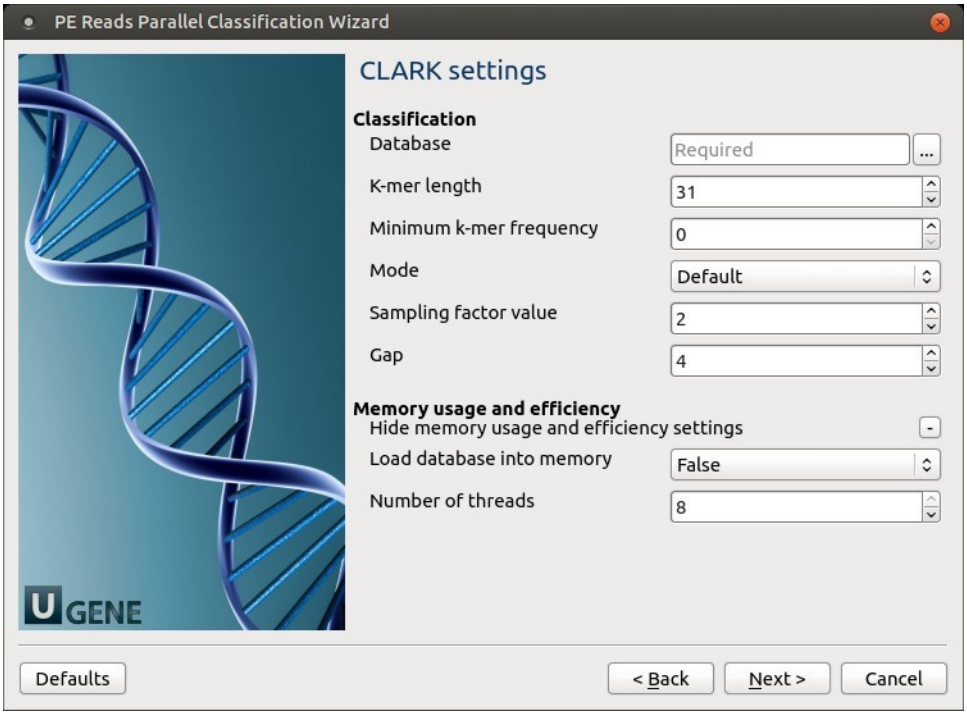
3. Kraken settings: Default Kraken parameters can be changed here.



The following parameters are available:

Database	A path to the folder with the Kraken database files.
Quick operation	Stop classification of an input read after the certain number of hits. The value can be specified in the "Minimum number of hits" parameter.

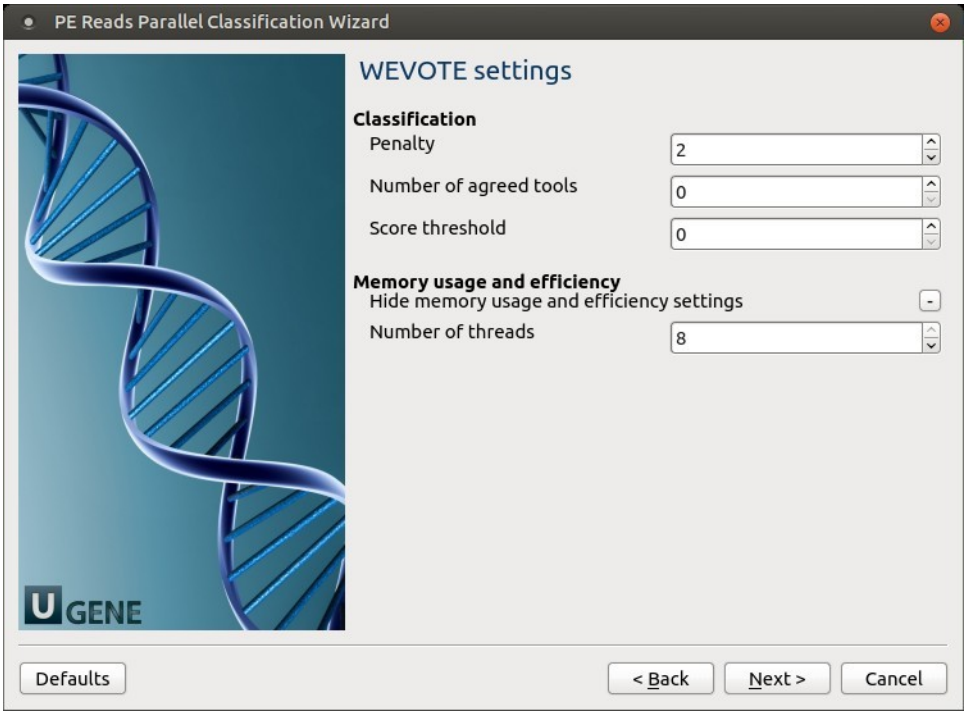
4. CLARK settings: Default CLARK parameters can be changed here.



The following parameters are available:

Database	A folder that should be used to store the database files.
K-mer length	This value is critical for the classification accuracy and speed. For high sensitivity, it is recommended to set this value to 20 or 21 (along with the "Full" mode). However, if the precision and the speed are the main concern, use any value between 26 and 32. Note that the higher the value, the higher is the RAM usage. So, as a good tradeoff between speed, precision, and RAM usage , it is recommended to set this value to 31 (along with the "Default" or "Express" mode).
Minimumk-merfrequency	Minimum of k-mer frequency/occurrence for the discriminative k-mers (-t). For example, for 1 (or, 2), the program will discard any discriminative k-mer that appear only once (or, less than twice).
Mode	Set the mode of the execution (-m): • "Full" to get detailed results, confidence scores, and other statistics. • "Default" to get results summary and perform the best trade-off between classification speed, accuracy and RAM usage. • "Express" to get results summary with the highest speed possible.
Sampling factor value	
Gap	"Gap" or number of non-overlapping k-mers to pass when creating the database (-). Increase the value if it is required to reduce the RAM usage. Note that this will degrade the sensitivity.

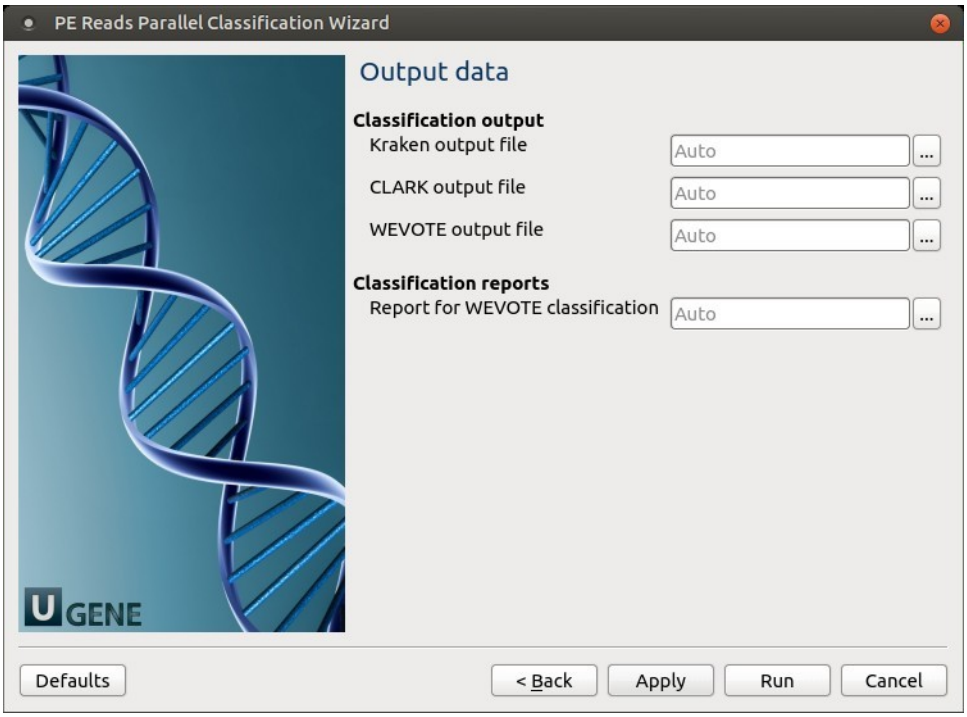
5. WEVOTE settings: DefaultWEVOTE parameters can be changed here.



The following parameters are available:

Penalty	Score penalty for disagreements (-k)
Numberof agreed tools	Specify the minimum number of tools agreed on WEVOTE decision (-a).
Score threshold	Score threshold (-s)

6. Output Files Page: On this page, you can select an output directory:



Serial NGS Reads Classification

The workflow sample, described below, takes FASTQ files with metagenomic NGS reads as input and process them as follows:

- Improve reads quality with Trimmomatic
- Provide FastQC reads quality reports
- Classification:
  Classify the pre-processed reads with Kraken
  Get reads that were not classified by Kraken
  Classify these reads with CLARK
  Get reads that were not classified (in case of SE reads)
  Classify these reads with DIAMOND (in case of SE reads)
  Provide general classification reports

How to Use This Sample

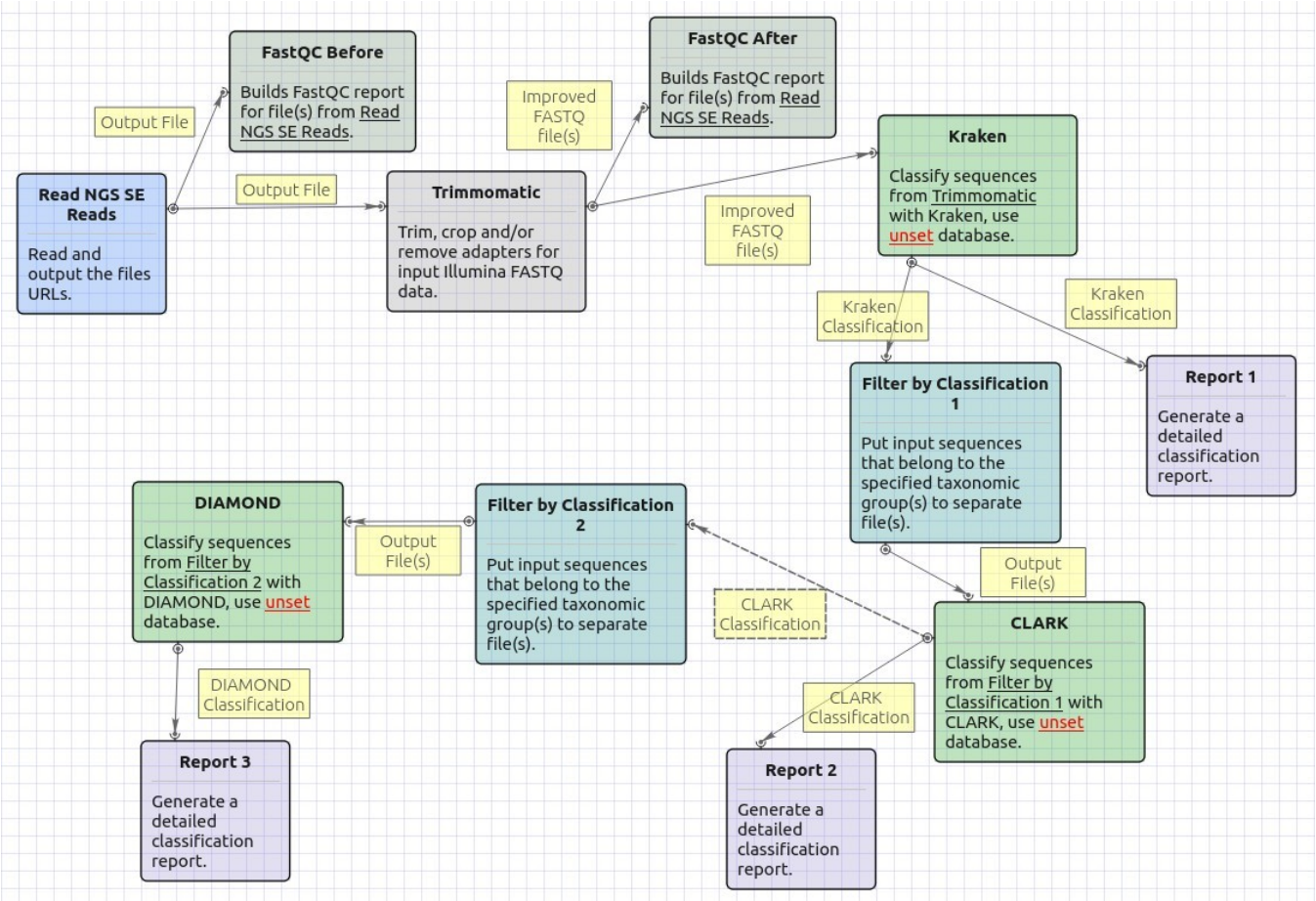
If you haven't used the workflow samples in UGENEbefore, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

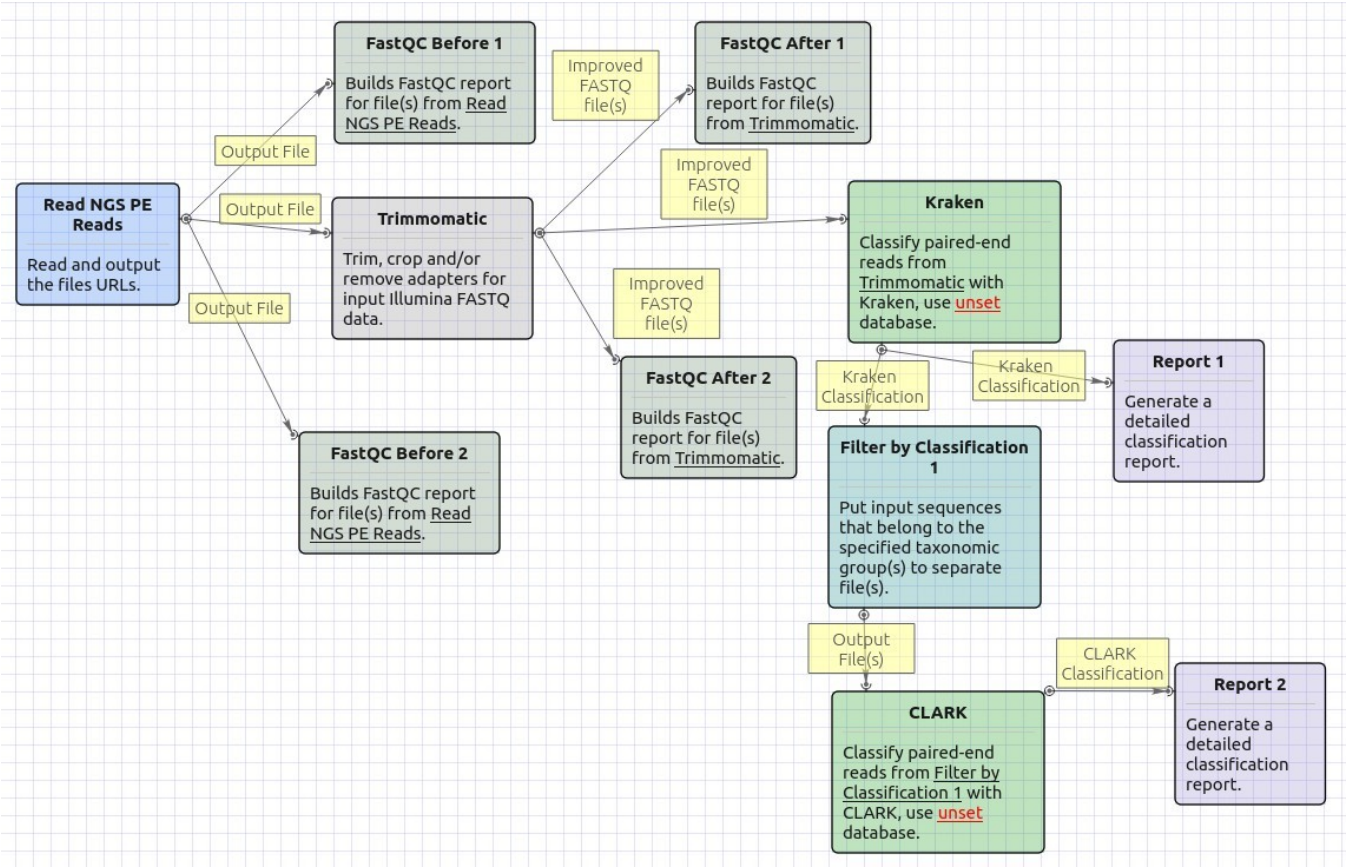
The workflow sample "Serial NGS Reads Classification" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

The opened workflow for single-end reads looks as follows:



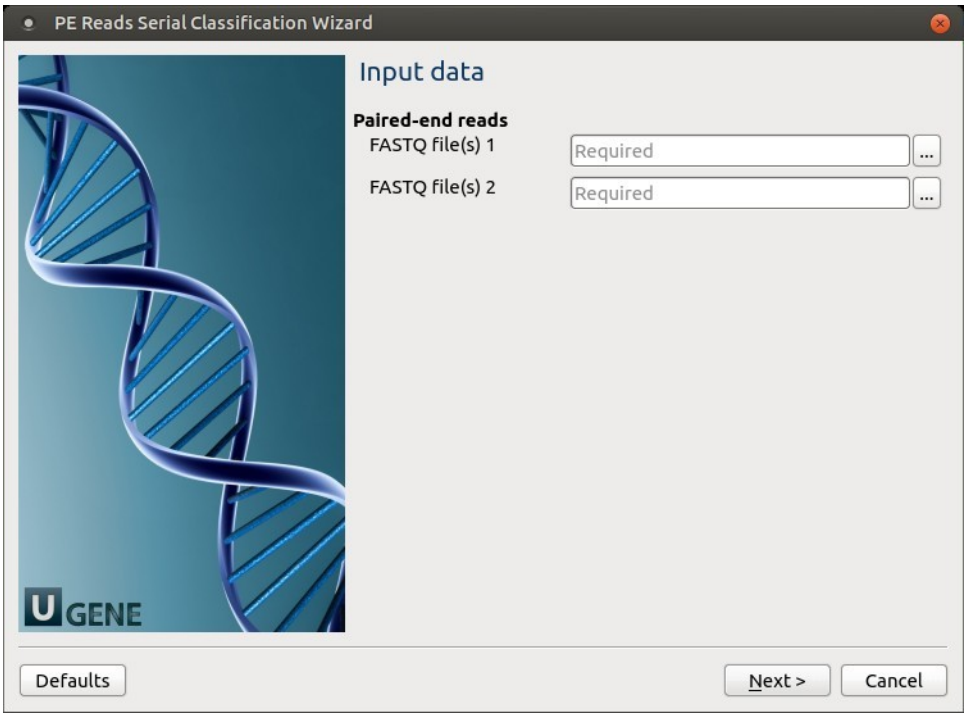
The opened workflow for paired-end reads looks as follows:



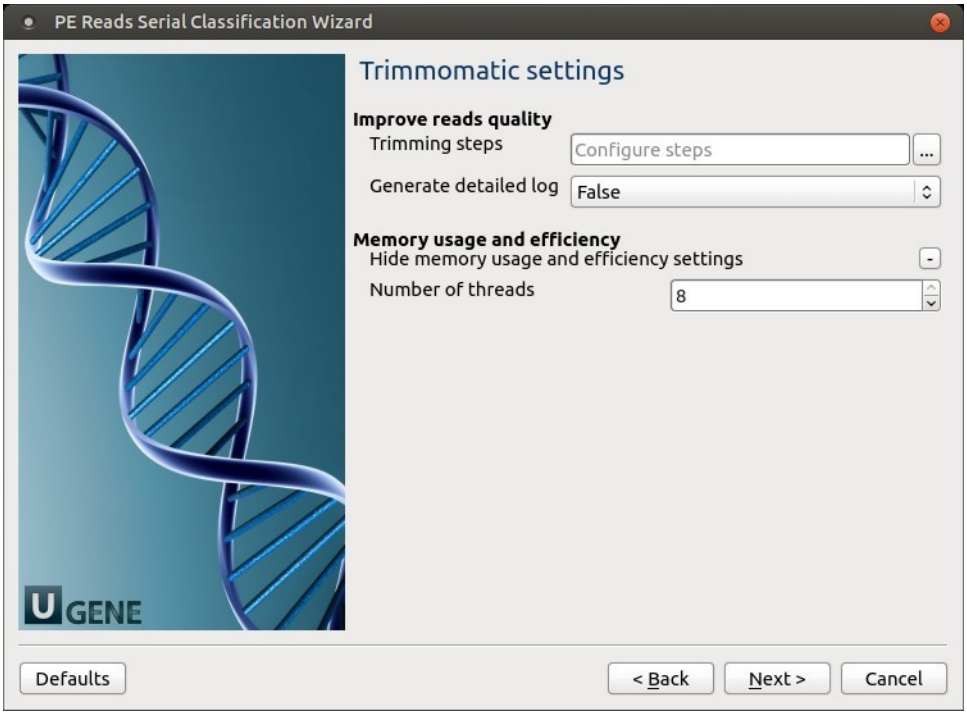
Workflow Wizard

The wizard has 5 pages.

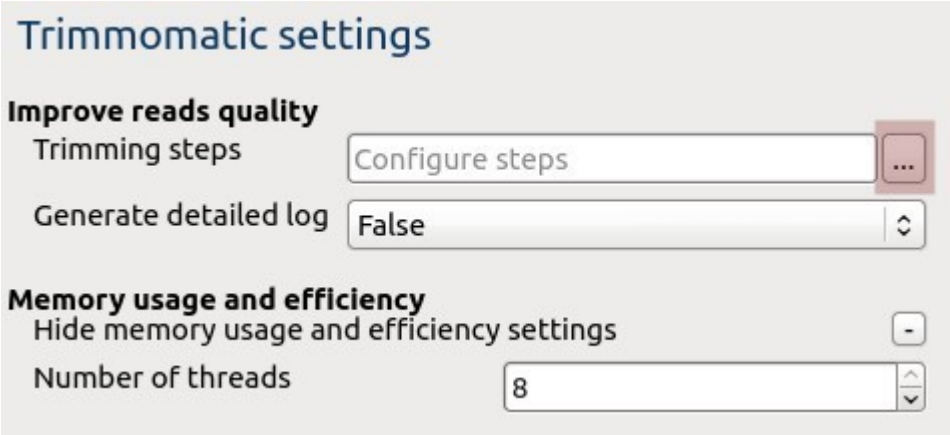
1. Input data: On this page, input files must be set.



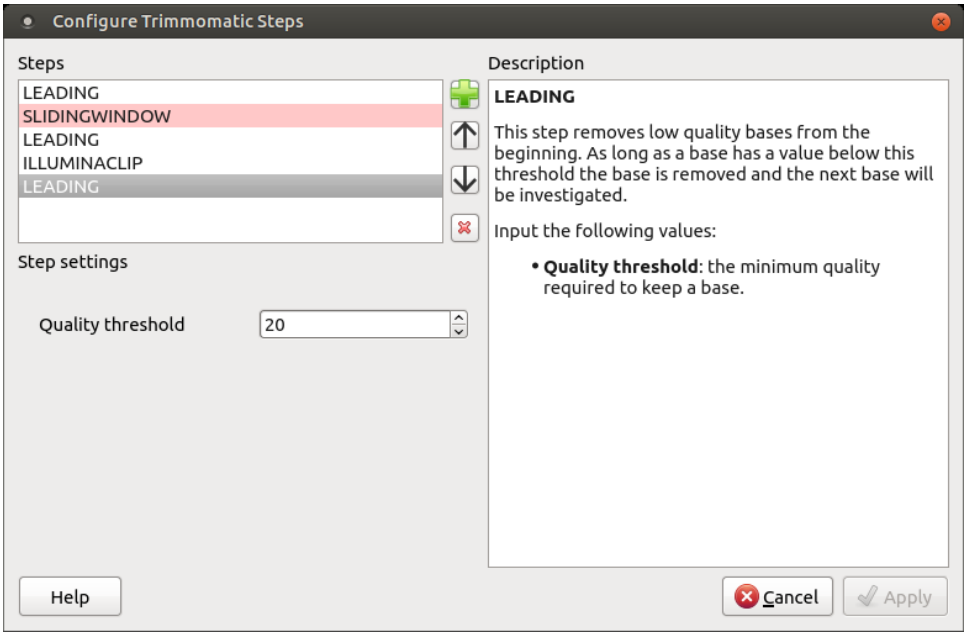
2. Trimmomatic settings: The Trimmomatic parameters can be changed here.



To configure trimming steps use the following button:



The following dialog will appear:



Click the *Add new step* button and select a step. The following options are available:

- ILLUMINACLIP: Cutadapterandotherillumina-specific sequences from the read.
- SLIDINGWINDOW: Perform a sliding window trimming, cutting once the average quality within the window falls below a threshold.
- LEADING: Cut bases off the start of a read, if below a threshold quality.
- TRAILING: Cut bases off the end of a read, if below a threshold quality.
- CROP: Cut the read to a specified length.
- HEADCROP: Cut the specified number of bases from the start of the read.
- MINLEN: Drop the read if it is below a specified length.
- AVGQUAL: Drop the read if the average quality is below the specified level.
- TOPHRED33: Convert quality scores to Phred-33.
- TOPHRED64: Convert quality scores to Phred-64.

Each step has the own parameters:

AVGQUAL

This step drops a read if the average quality is below the specified level.

Input the following values:

- Quality threshold: the minimum average quality required to keep a read.

CROP

This step removes bases regardless of quality from the end of thread, so that the readhas maximally the specified length after this step has been performed. Steps performed after CROP might of course further shorten the read.

Input the following values:

- Length: the number of bases to keep, from the start of the read.

HEADCROP

This step removes the specified number of bases, regardless of quality, from the beginning of the read.

Input the following values:

- Length: the number of bases to remove from the start of the read.

ILLUMINACLIP

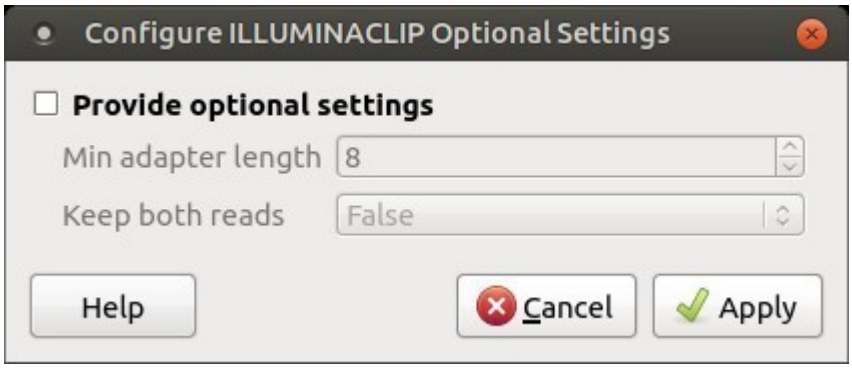
This step is used to find and remove Illumina adapters.

Trimmomatic first compares short sections of an adapter and a read. If they match enough, the entire alignment between the read and adapter is scored. For paired-end reads, the "palindrome" approach is also used to improve the result. See Trimmomatic manual for details.

Input the following values:

- Adapter sequences: a FASTA file with the adapter sequences. Files for TruSeq2 (GAII machines), TruSeq3 (HiSeq and MiSeq machines) and Nextera kits for SE and PE reads are now available by default. The naming of the various sequences within the specified file determines how they are used.
- Seed mismatches: the maximum mismatch count in short sections which will still allow a full match to be performed.
- Simple clip threshold: a threshold for simple alignment mode. Values between 7 and 15 are recommended. A perfect match of a 12 base sequence will score just over 7, while 25 bases are needed to score 15.
- Palindrome clip threshold: a threshold for palindrome alignment mode. For palindromic matches, a longer alignment is possible. Therefore the threshold can be in the range of 30. Even though this threshold is very high (requiring a match of almost 50 bases) Trimmomatic is still able to identify very, very short adapter fragments.

There are also two optional parameters for palindrome mode: Min adapter length and Keep both reads. Use the following dialog. To call the dialog press the *Optional* button.



LEADING

This step removes low-quality bases from the beginning. As long as a base has a value below this threshold the base is removed and the next base will be investigated.

Input the following values:

- Quality threshold: the minimum quality required to keep a base.

MAXINFO

This step performs an adaptive quality trim, balancing the benefits of retaining longer reads against the costs of retaining bases with errors. See Trimmomatic manual for details.

Input the following values:

- Target length: the read length which is likely to allow the location of the read within the target sequence. Extremely short reads, which can be placed into many different locations, provide little value. Typically, the length would be in the order of 40 bases, however, the value also depends on the size and complexity of the target sequence.
- Strictness: the balance between preserving as much read length as possible vs. removal of incorrect bases. A low value of this parameter (0.8) favours read correctness.

MINLEN

This step removes reads that fall below the specified minimum length. If required, it should normally be after all other processing steps. Reads removed by this step will be counted and included in the "dropped reads" count.

Input the following values:

- Length: the minimum length of reads to be kept.

SLIDINGWINDOW

This step performs a sliding window trimming, cutting once the average quality within the window falls below a threshold. By considering multiple bases, a single poor quality base will not cause the removal of high-quality data later in the read.

Input the following values:

- Window size: the number of bases to an average across.
- Quality threshold: the average quality required.

TOPHRED33

This step (re)encodes the quality part of the FASTQ file to base 33.

TOPHRED64

This step (re)encodes the quality part of the FASTQ file to base 64.

TRAILING

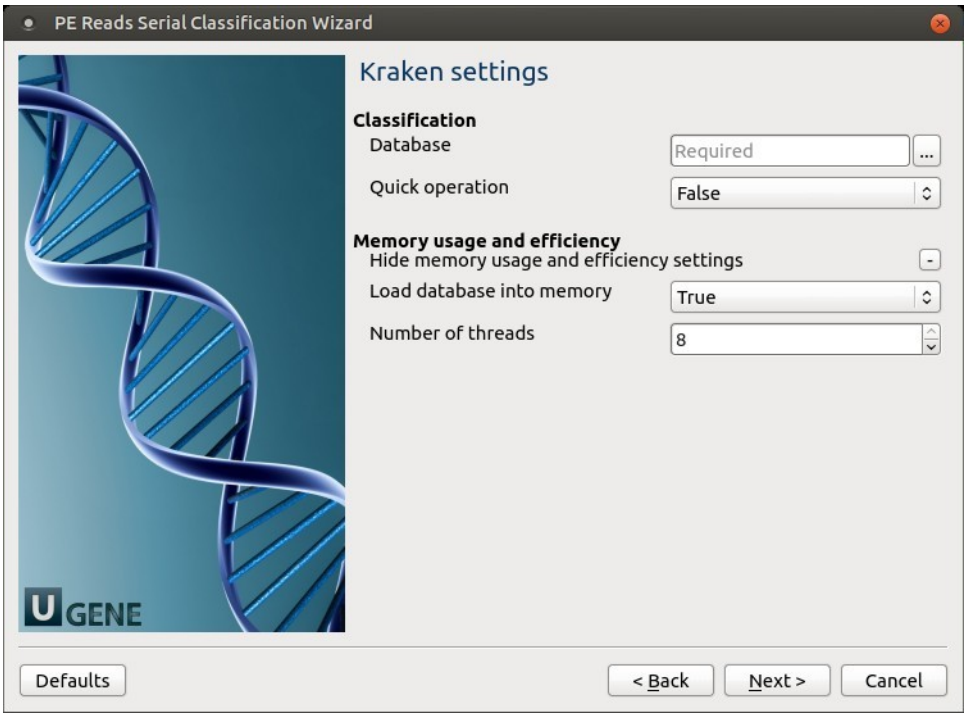
This step removes low-quality bases from the end. As long as a base has a value below this threshold the base is removed and the next base (i. e. the preceding one) will be investigated. This approach can be used removing the special Illumina " low-quality segment" regions (which are marked with a quality score of 2), but SLIDINGWINDOW or MAXINFO are recommended instead.

Input the following values:

- Quality threshold: the minimum quality required to keep a base.

To remove a step use the *Remove selected step* button. The pink highlighting means the required parameter has not been set.

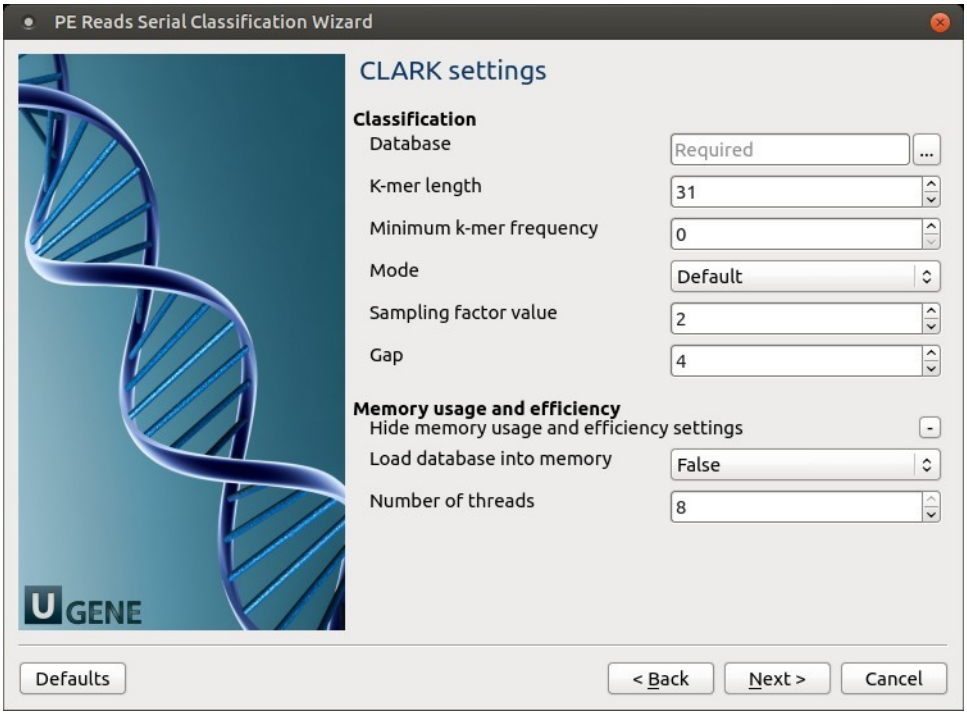
3. Kraken settings: Default Kraken parameters can be changed here.



The following parameters are available:

Database	A path to the folder with the Kraken database files.
Quick operation	Stop classification of an input read after the certain number of hits. The value can be specified in the "Minimum number of hits" parameter.

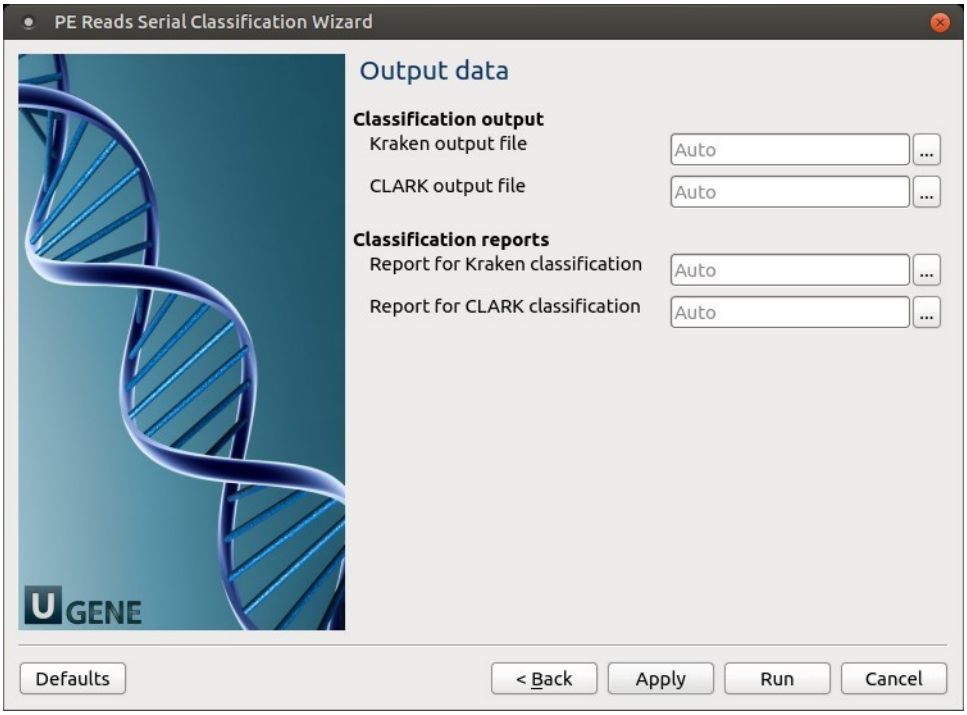
4. CLARK settings: Default CLARK parameters can be changed here.



The following parameters are available:

Database	A folder that should be used to store the database files.
K-mer length	This value is critical for the classification accuracy and speed. For high sensitivity, it is recommended to set this value to 20 or 21 (along with the "Full" mode). However, if the precision and the speed are the main concern, use any value between 26 and 32. Note that the higher the value, the higher is the RAM usage. So, as a good tradeoff between speed, precision, and RAM usag e, it is recommended to set this value to 31 (along with the "Default" or "Express" mode).
Minimum k-mer frequency	Minimum of k-mer frequency/occurrence for the discriminative k-mers(-t). For example, for 1 (or, 2), the program will discard any discriminative k-mer that appear only once (or, less than twice).
Mode	Set the mode of the execution (-m): • "Full" to get detailed results, confidence scores, and other statistics. • "Default" to get results summary and perform the best trade-off between classification speed, accuracy and RAM usage. • "Express" to get results summary with the highest speed possible.
Sampling factor value	
Gap	"Gap" or number of non-overlapping k-mers to pass when creating the database (-). Increase the value if it is required to reduce the RAM usage. Note that this will degrade the sensitivity.

5. Output Files Page: On this page, you can select an output directory:



RNA-Seq Analysis with TopHat and StringTie

The workflow sample, described below, takes FASTQ files with paired-end RNA-Seq reads and process them as follows:

- Improve reads quality with Trimmomatic
- Provide FastQC quality reports
- Map improved reads to a reference sequence with TopHat
- Assemble transcripts and generate gene abundance output with StringTie
- Produce a common gene abundance report (one for several input samples)

How to Use This Sample

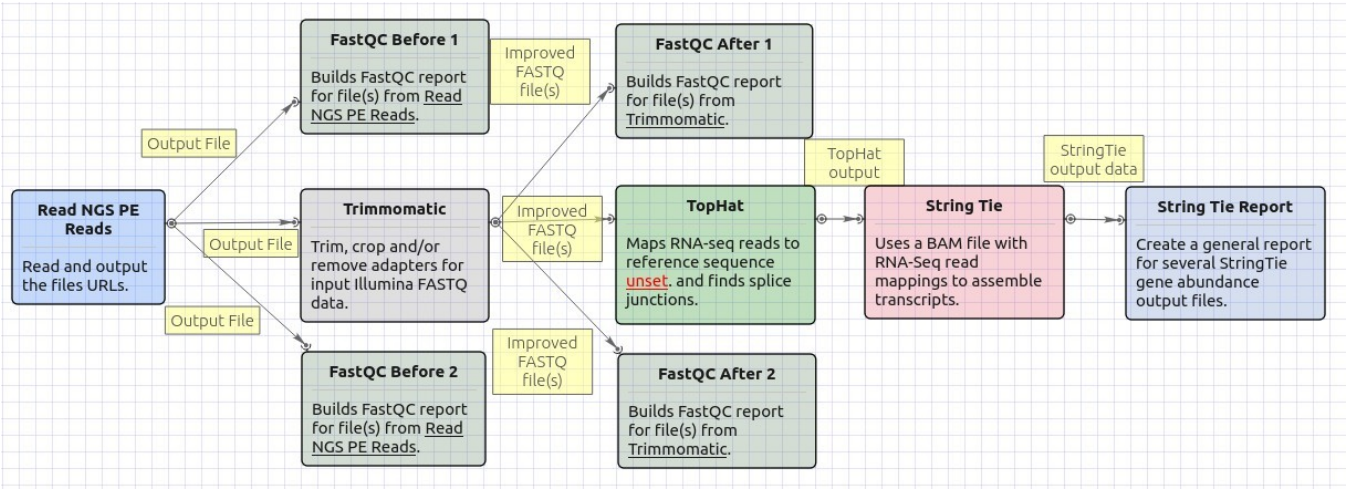
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "RNA-Seq Analysis with TopHat and StringTie" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

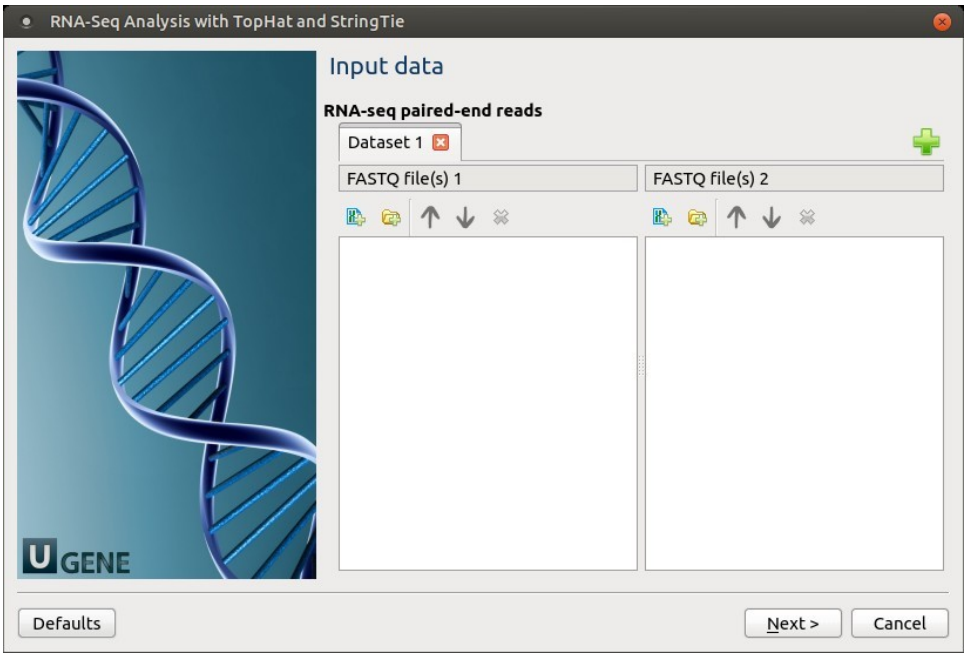
The opened workflow looks as follows:



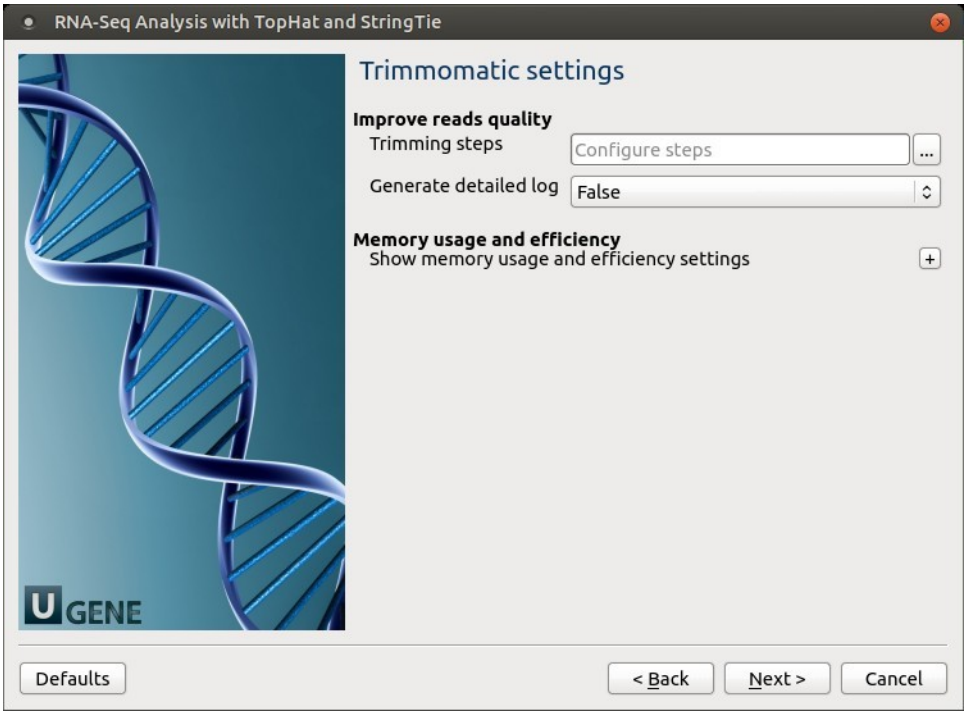
Workflow Wizard

The wizard has 5 pages.

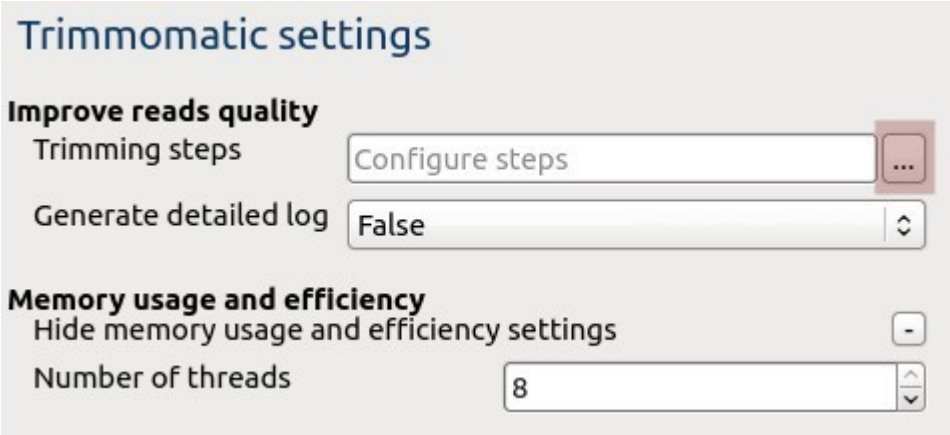
1. Input data: RNA-seq paired-end reads: On this page, files with RNA-seq paired-end reads must be set.



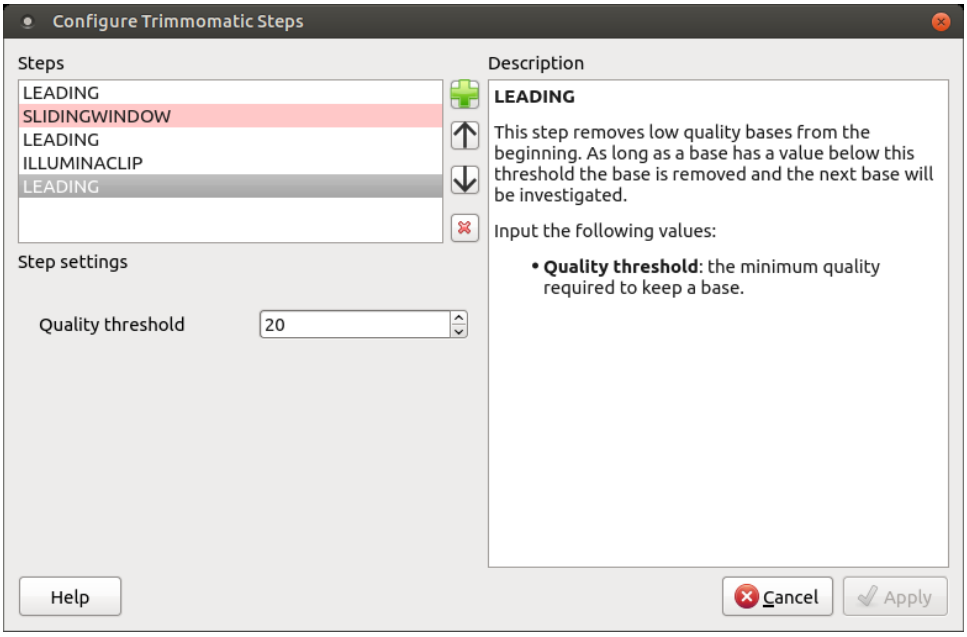
2. Trimmomatic settings: The Trimmomatic parameters can be changed here.



To configure trimming steps use the following button:



The following dialog will appear:



Click the *Add new step* button and select a step. The following options are available:

- ILLUMINACLIP: Cutadapterandotherillumina-specific sequences from the read.
- SLIDINGWINDOW: Perform a sliding window trimming, cutting once the average quality within the window falls below a threshold.
- LEADING: Cut bases off the start of a read, if below a threshold quality.
- TRAILING: Cut bases off the end of a read, if below a threshold quality.
- CROP: Cut the read to a specified length.
- HEADCROP: Cut the specified number of bases from the start of the read.
- MINLEN: Drop the read if it is below a specified length.
- AVGQUAL: Drop the read if the average quality is below the specified level.
- TOPHRED33: Convert quality scores to Phred-33.
- TOPHRED64: Convert quality scores to Phred-64.

Each step has the own parameters:

AVGQUAL

This step drops a read if the average quality is below the specified level.

Input the following values:

- Quality threshold: the minimum average quality required to keep a read.

CROP

This step removes bases regardless of quality from the end of thread, so that the readhas maximally the specified length after this step has been performed. Steps performed after CROP might of course further shorten the read.

Input the following values:

- Length: the number of bases to keep, from the start of the read.

HEADCROP

This step removes the specified number of bases, regardless of quality, from the beginning of the read.

Input the following values:

- Length: the number of bases to remove from the start of the read.

ILLUMINACLIP

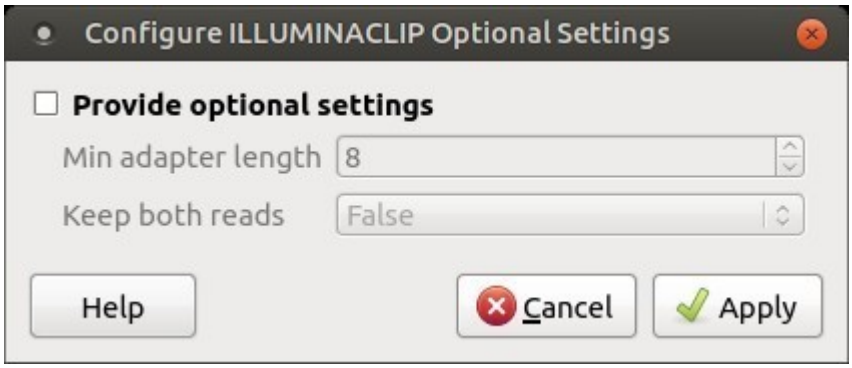
This step is used to find and remove Illumina adapters.

Trimmomatic first compares short sections of an adapter and a read. If they match enough, the entire alignment between the read and adapter is scored. For paired-end reads, the "palindrome" approach is also used to improve the result. See Trimmomatic manual for details.

Input the following values:

- Adapter sequences: a FASTA file with the adapter sequences. Files for TruSeq2 (GAII machines), TruSeq3 (HiSeq and MiSeq machines) and Nextera kits for SE and PE reads are now available by default. The naming of the various sequences within the specified file determines how they are used.
- Seed mismatches: the maximum mismatch count in short sections which will still allow a full match to be performed.
- Simple clip threshold: a threshold for simple alignment mode. Values between 7 and 15 are recommended. A perfect match of a 12 base sequence will score just over 7, while 25 bases are needed to score 15.
- Palindrome clip threshold: a threshold for palindrome alignment mode. For palindromic matches, a longer alignment is possible. Therefore the threshold can be in the range of 30. Even though this threshold is very high (requiring a match of almost 50 bases) Trimmomatic is still able to identify very, very short adapter fragments.

There are also two optional parameters for palindrome mode: Min adapter length and Keep both reads. Use the following dialog. To call the dialog press the *Optional* button.



LEADING

This step removes low-quality bases from the beginning. As long as a base has a value below this threshold the base is removed and the next base will be investigated.

Input the following values:

- Quality threshold: the minimum quality required to keep a base.

MAXINFO

This step performs an adaptive quality trim, balancing the benefits of retaining longer reads against the costs of retaining bases with errors. See Trimmomatic manual for details.

Input the following values:

- Target length: the read length which is likely to allow the location of the read within the target sequence. Extremely short reads, which can be placed into many different locations, provide little value. Typically, the length would be in the order of 40 bases, however, the value also depends on the size and complexity of the target sequence.
- Strictness: the balance between preserving as much read length as possible vs. removal of incorrect bases. A low value of this parameter (0.8) favours read correctness.

MINLEN

This step removes reads that fall below the specified minimum length. If required, it should normally be after all other processing steps. Reads removed by this step will be counted and included in the "dropped reads" count.

Input the following values:

- Length: the minimum length of reads to be kept.

SLIDINGWINDOW

This step performs a sliding window trimming, cutting once the average quality within the window falls below a threshold. By considering multiple bases, a single poor quality base will not cause the removal of high-quality data later in the read.

Input the following values:

- Window size: the number of bases to an average across.
- Quality threshold: the average quality required.

TOPHRED33

This step (re)encodes the quality part of the FASTQ file to base 33.

TOPHRED64

This step (re)encodes the quality part of the FASTQ file to base 64.

TRAILING

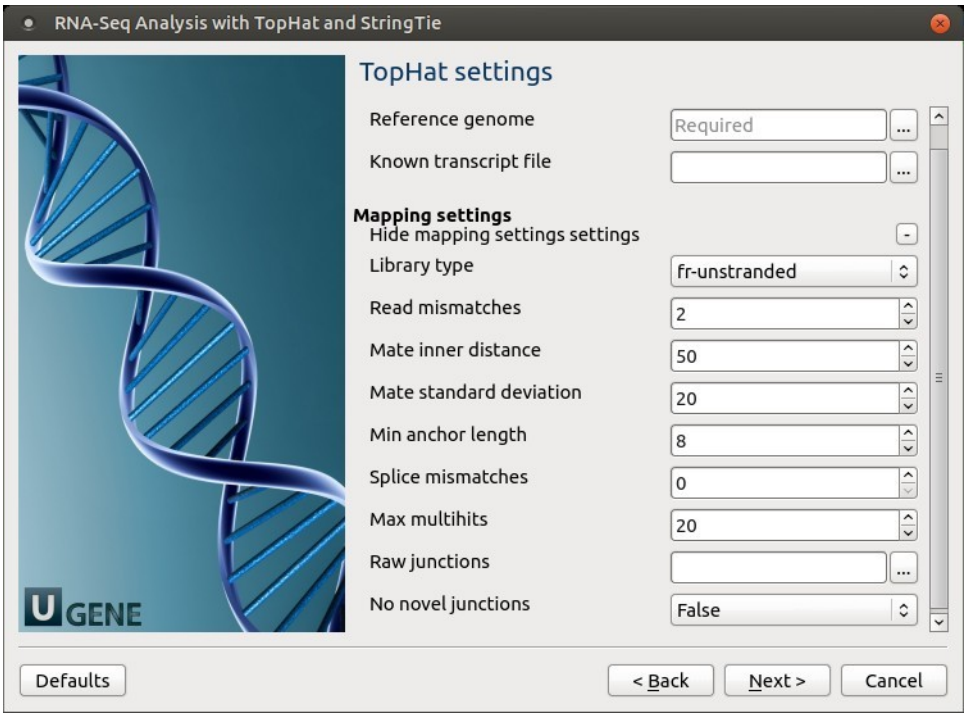
This step removes low-quality bases from the end. As long as a base has a value below this threshold the base is removed and the next base (i. e. the preceding one) will be investigated. This approach can be used removing the special Illumina " low-quality segment" regions (which are marked with a quality score of 2), but SLIDINGWINDOW or MAXINFO are recommended instead.

Input the following values:

- Quality threshold: the minimum quality required to keep a base.

To remove a step use the *Remove selected step* button. The pink highlighting means the required parameter has not been set.

3. TopHat settings: TopHat parameters can be set here.

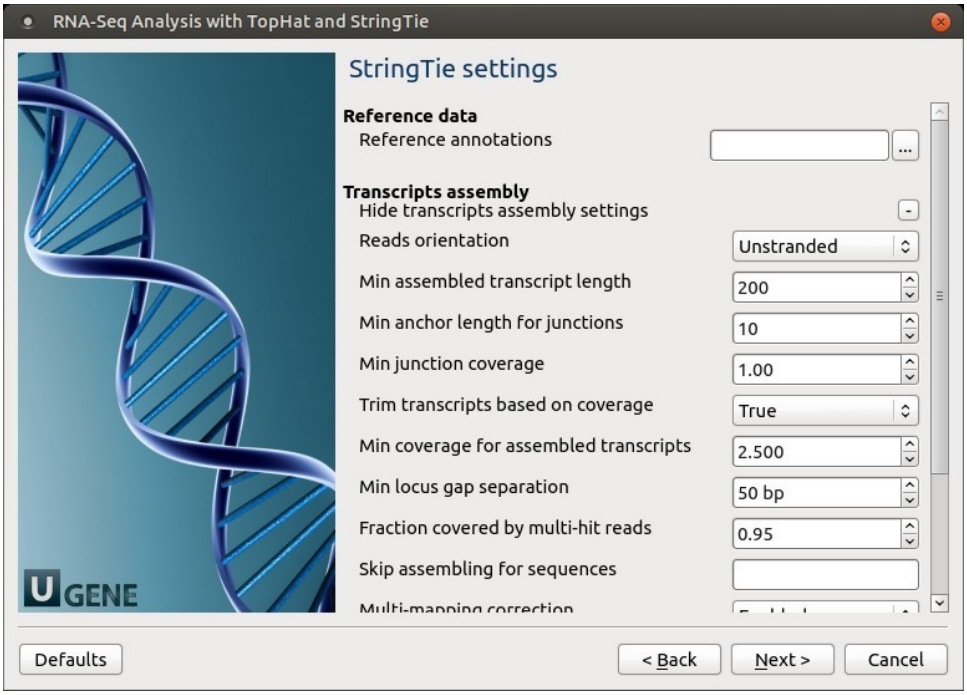


The following parameters are available:

Reference genome	Path to the indexed reference genome.
Known transcript file	A set of gene model annotations and/or known transcripts.
Library type	Specifies RNA-Seq protocol.
Read mismatches	Final read alignments having more than these many mismatches are discarded.
Mate inner distance	The expected (mean) inner distance between mate pairs.
Mate standard deviation	The standard deviation for the distribution on inner distances between mate pairs.
Min anchor length	The anchor length. TopHat will report junctions spanned by reads with at least this many bases on each side of the junction. Note that individual spliced alignments may span a junction with fewer than this many bases on one side. However, every junction involved in spliced alignments is supported by at least one read with this many bases on each side.
Splice mismatches	The maximum number of mismatches that may appear in the anchor region of a spliced alignment.

Max multihits	Instruct TopHat to allow up to this many alignments to the reference for a given read and suppresses all alignments for reads with more than this many alignments.
Raw junctions	The list of raw junctions.
No novel junctions	Only look for reads across junctions indicated in the supplied GFF or junctions file. This parameter is ignored if Raw junctions or Known transcript file is not set.

4. StringTie settings: StringTie parameters can be set here.

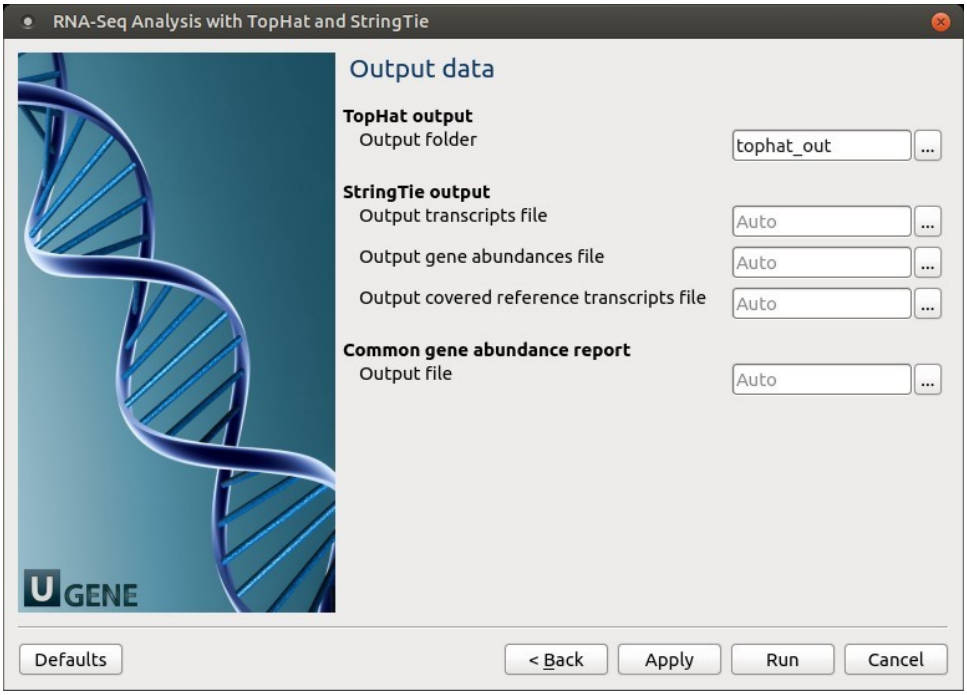


The following parameters are available:

Reference annotations	Use the reference annotation file (in GTF or GFF3 format) to guide the assembly process (-G). The output will include expressed reference transcripts as well as any novel transcripts that are assembled.
Reads orientation	Select the NGS libraries type: unstranded, stranded fr-secondstrand (--fr), or stranded fr-firststand (--rf).
Min assembled transcript length	Specify the minimum length for the predicted transcripts (-m).
Min anchor length for junctions	Junctions that don't have spliced reads that align them with at least this amount of bases on both sides is filtered out (-a).
Min junction coverage	There should be at least this many spliced reads that align across a junction (-j). This number can be fractional since some reads align in more than one place. A read that aligns in n places will contribute 1/n to the junction coverage.

Trim transcripts based on coverage	By default StringTie adjusts the predicted transcript's start and/or stop coordinates based on sudden drops in coverage of the assembled transcript. Set this parameter to "False" to disable the trimming at the ends of the assembled transcripts (-t).
Min coverage for assembled transcripts	Specifies the minimum read coverage allowed for the predicted transcripts (-c). A transcript with a lower coverage than this value is not shown in the output. This number can be fractional since some reads align in more than one place. A read that aligns in n places will contribute 1/n to the coverage.
Min locus gap separation	Reads that are mapped closer than this distance are merged together in the same processing bundle (-g).
Fraction covered by multi-hit reads	Specify the maximum fraction of muliple-location-mapped reads that are allowed to be present at a given locus (-M). A read that aligns in n places will contribute 1/n to the coverage.
Skip assembling for sequences	Ignore all read alignments (and thus do not attempt to perform transcript assembly) on the specified reference sequences (-x). The value can be a single reference sequence name (e.g. "chrM") or a comma-delimited list of sequence names (e.g. "chrM,chrX,chrY"). This can speed up StringTie especially in the case of excluding the mitochondrial genome, whose genes may have very high coverage in some cases, even though they may be of no interest for a particular RNA-Seq analysis. The reference sequence names are case sensitive, they must match identically the names of chromosomes/contigs of the target genome against which the RNA-Seq reads were aligned in the first place.
Multi-mapping correction	Enables or disables (-u) multi-mapping correction.
Verbose log	Enable detailed logging, if required (-v). The messages will be written to the UGENE log (enabling of "DETAILS" and "TRACE" logging may be required) and to the dashboard.
Label	Use the specified string as the prefix for the name of the output transcripts (-l).

5. Output Files Page: On this page, output directories can be selected:



RNA-seq Analysis with Tuxedo Tools

The RNA-seq pipeline “Tuxedo” consists of the TopHat spliced read mapper, that internally uses Bowtie or Bowtie 2 short read aligners, and several Cufflinks tools that allows one to assemble transcripts, estimate their abundances, and tests for differential expression and regulation in RNA-seq samples.

Environment Requirements

The pipeline is currently available on Linux and macOS systems only.

How to Use This Sample

If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

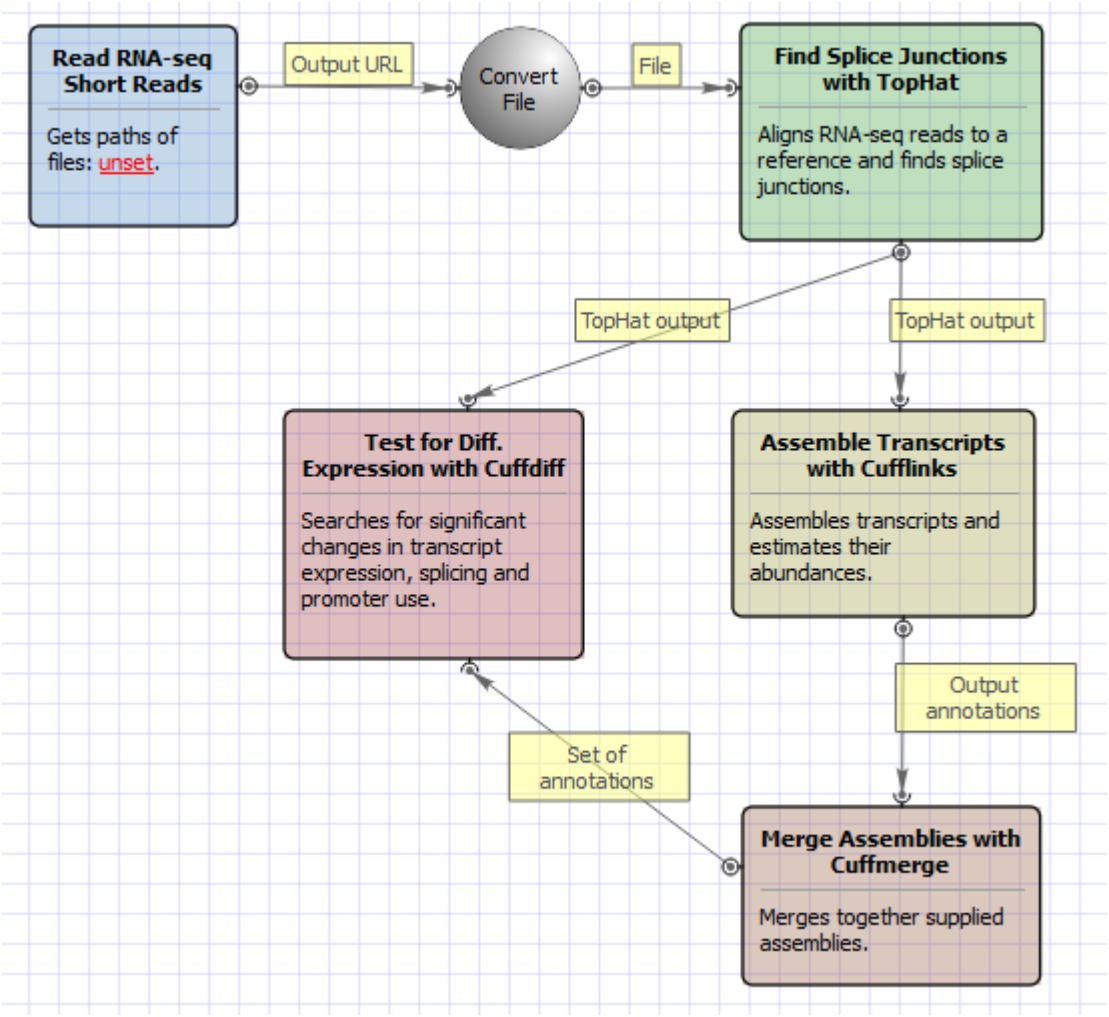
The workflow sample "RNA-seq Analysis with Tuxedo Tools" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

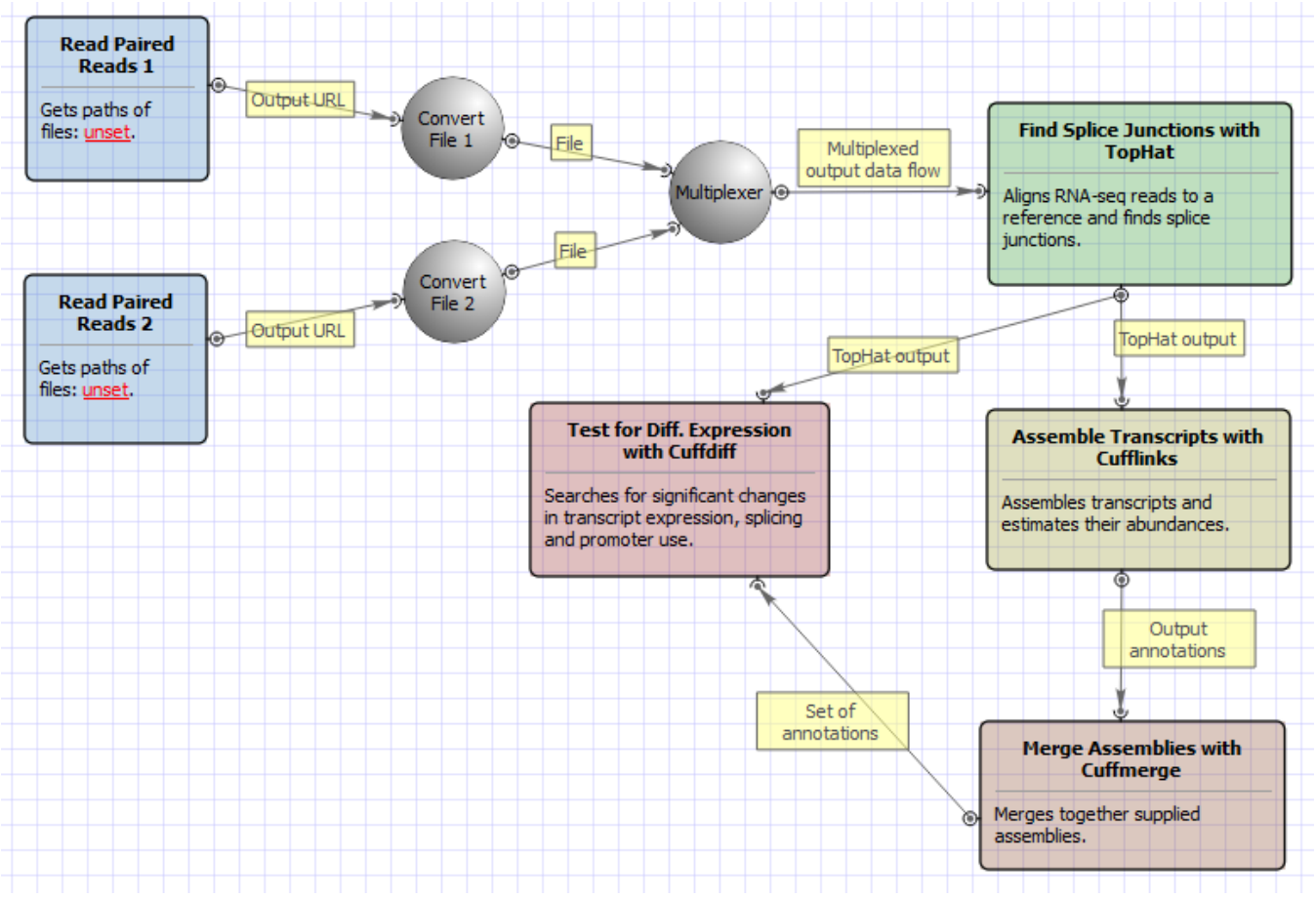
There are two short reads types of workflow: single-end and paired-end reads. For both of them there are three analysis types:

1. Full Tuxedo Pipeline - use this pipeline to analyze multiple samples with TopHat, Cufflinks, Cuffmerge and Cuffdiff tools.
2. Single-sample Tuxedo Pipeline - use this pipeline to analyze a single sample with TopHat and Cufflinks tools.
3. No-new-transcripts Tuxedo Pipeline - use this pipeline to analyze multiple samples with TopHat and Cuffdiff tools only, i.e. without producing new transcripts.

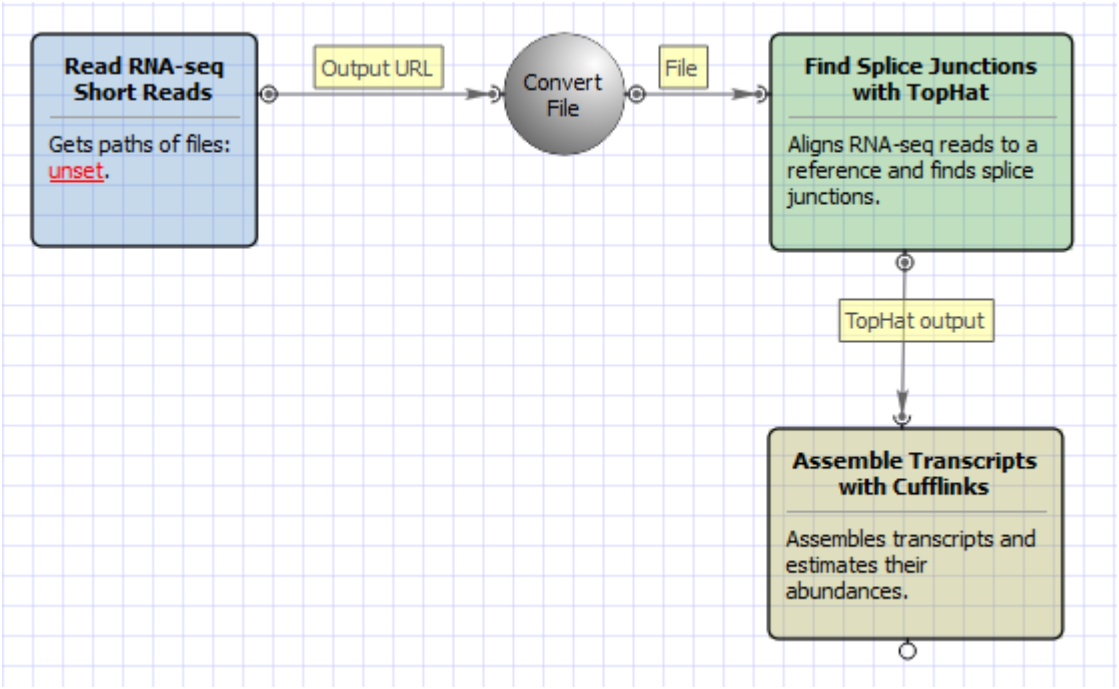
For **Full Tuxedo Pipeline** analysis type and **single-end reads** type the following workflow appears:



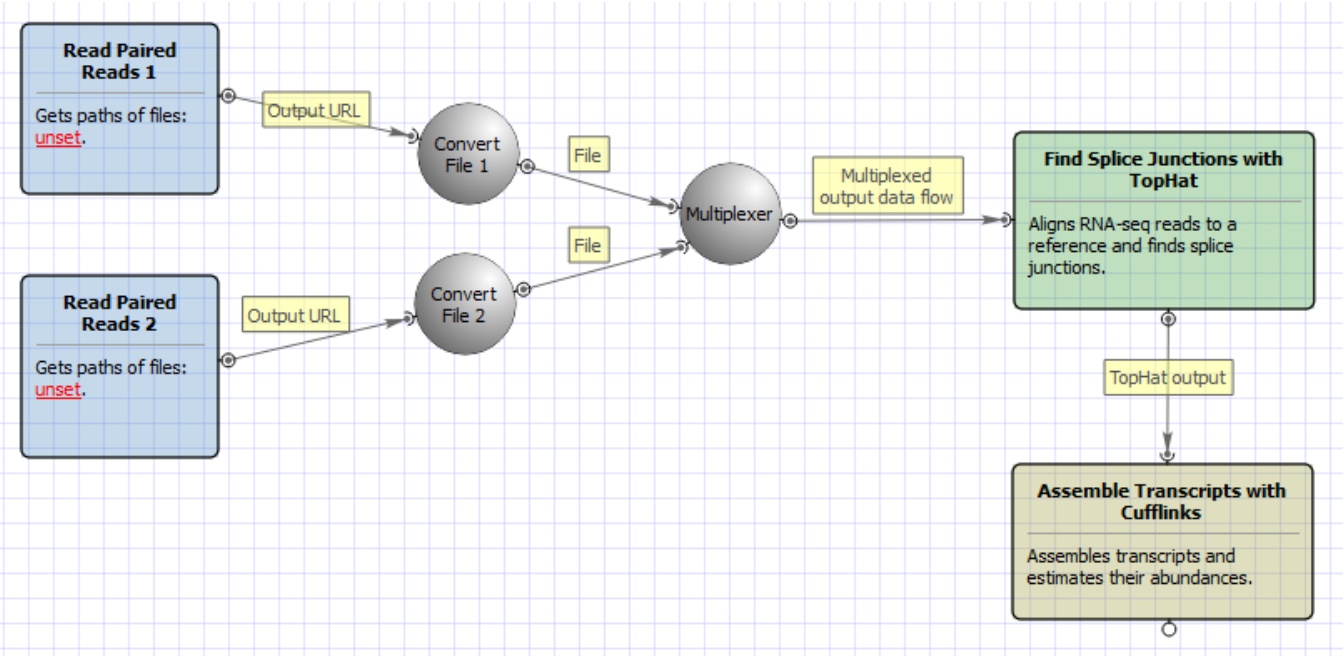
For **Full Tuxedo Pipeline** analysis type and **paired-end reads** type the following workflow appears:



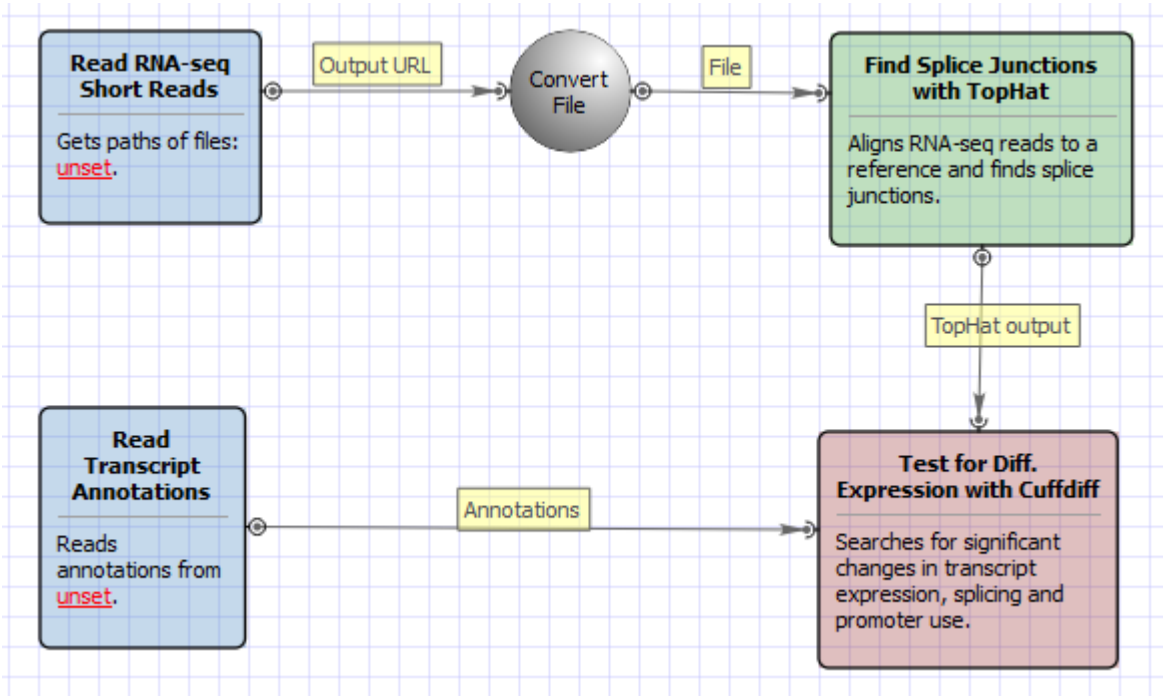
For **Single-sample Tuxedo Pipeline** analysis type and **single-end reads** type the following workflow appears:



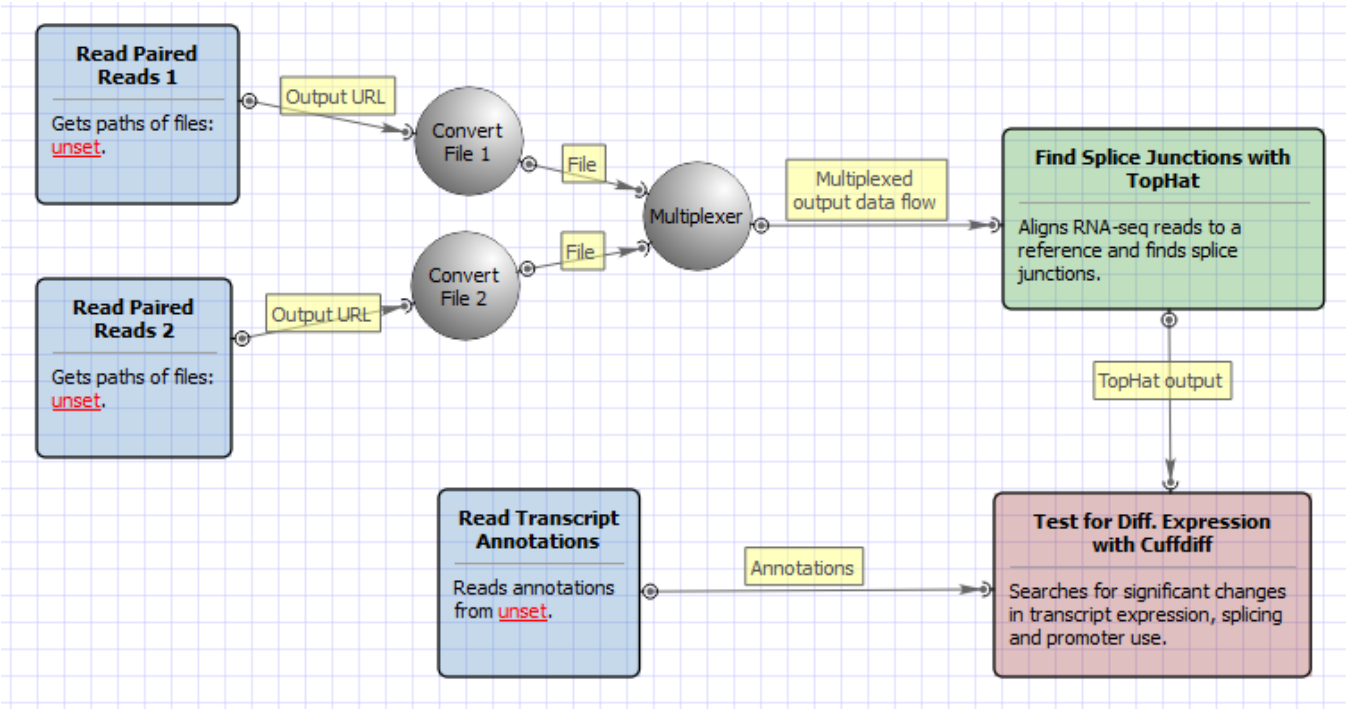
For **Single-sample Tuxedo Pipeline** analysis type and **paired-end reads** type the following workflow appears:



For **No-new-transcripts Tuxedo Pipeline** analysis type and **single-end reads** type the following workflow appears:



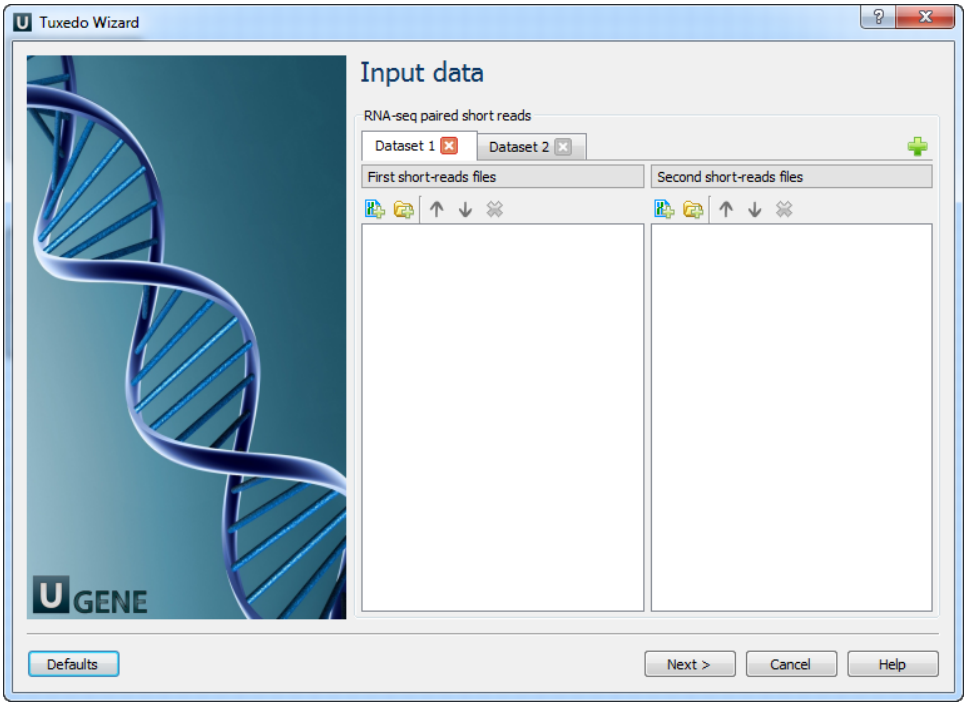
For **No-new-transcripts Tuxedo Pipeline** analysis type and **paired-end reads** type the following workflow appears:



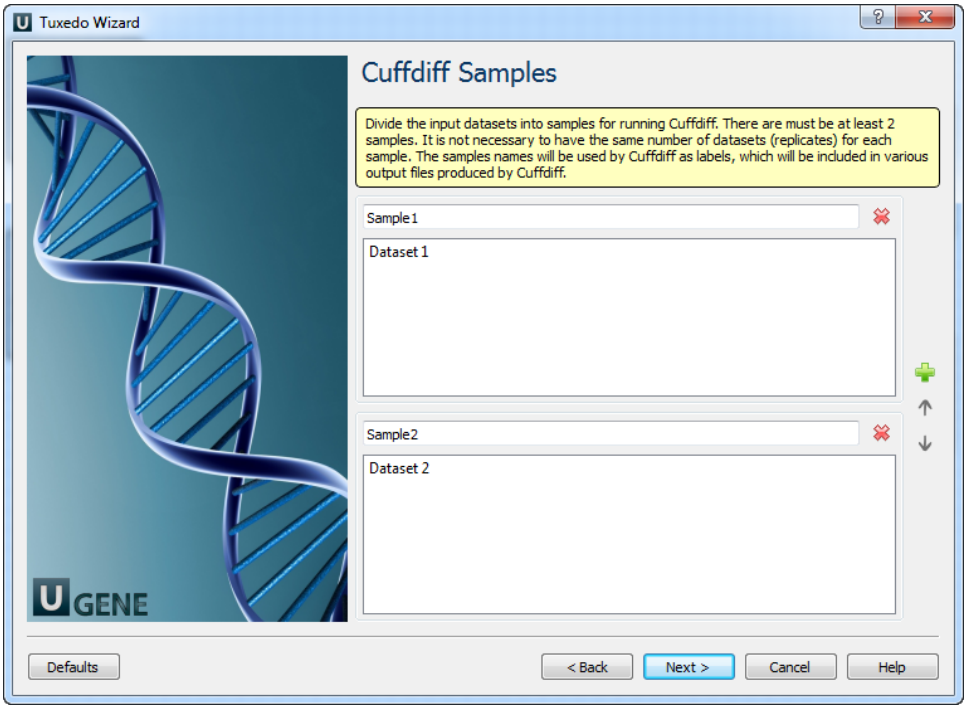
Workflow Wizard

All of these workflows have the similar wizards. For **Full Tuxedo Pipeline** analysis type and **paired-end reads** type wizard has 7 pages.

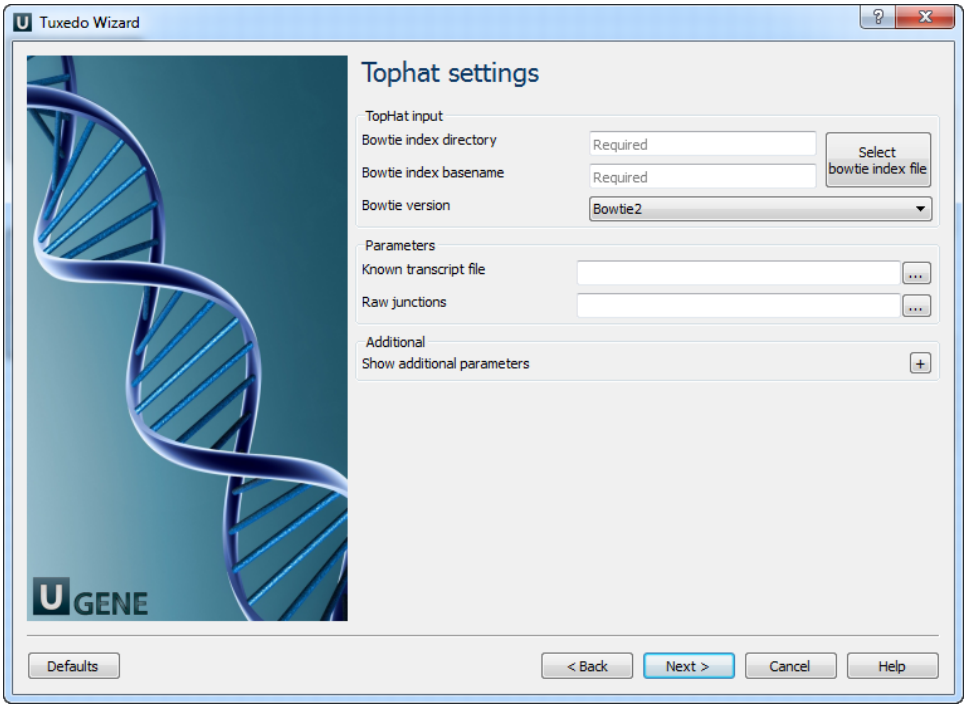
1. Input data: Here you need to input RNA-seq short reads in FASTA or FASTQ formats. Many datasets with different reads can be added.



2. Cuffdiff Samples: Here you need to divide the input datasets into samples for running Cuffdiff. There are must be at least 2 samples. It is not necessary to have the same number of datasets (replicates) for each sample. The samples names will be used by Cuffdiff as labels, which will be included in various output files produced by Cuffdiff.



3. Tophat Settings: Here you can configure TopHat settings. To show additional parameters click on the + button.



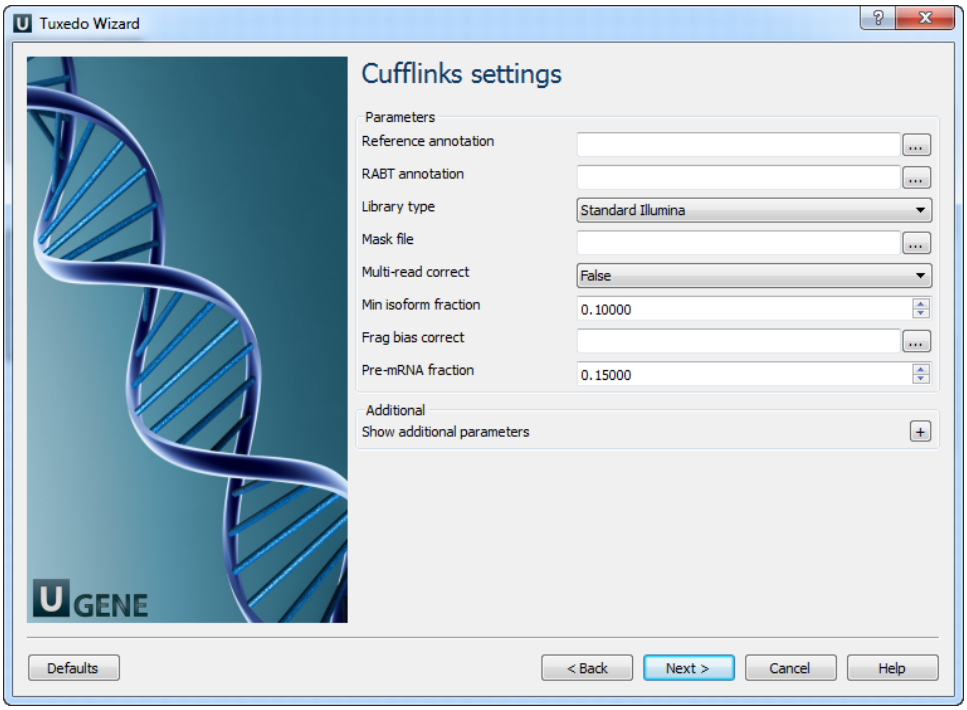
The following parameters are available:

Bowtie index directory	The directory with the Bowtie index for the reference sequence.
Bowtie index basename	The basename of the Bowtie index for the reference sequence.

Bowtie version	Specifies which Bowtie version should be used.
Known transcript file	A set of gene model annotations and/or known transcripts.
Raw junctions	The list of raw junctions.
Mate inner distance	Expected (mean) inner distance between mate pairs.
Mate standard deviation	Standard deviation for the distribution on inner distances between mate pairs.
Library type	Specifies RNA-seq protocol.
No novel junctions	Only look for reads across junctions indicated in the supplied GFF or junctions file. This parameter is ignored if Raw junctions or Known transcript file is not set.
Max multihits	Instructs TopHat to allow up to this many alignments to the reference for a given read, and suppresses all alignments for reads with more than this many alignments.
Segment length	Each read is cut up into segments, each at least this long. These segments are mapped independently.
Fusion search	Turn on fusion mapping.
Transcritome max hits	Only align the reads to the transcriptome and report only those mappings as genomic mappings.
Prefilter multihits	When mapping reads on the transcriptome, some repetitive or low complexity reads that would be discarded in the context of the genome may appear to align to the transcript sequences and thus may end up reported as mapped to those genes only. This option directs TopHat to first align the reads to the whole genome in order to determine and exclude such multi-mapped reads (according to the value of the Max multihits option).
Min anchor length	The anchor length. TopHat will report junctions spanned by reads with at least this many bases on each side of the junction. Note that individual spliced alignments may span a junction with fewer than this many bases on one side. However, every junction involved in spliced alignments is supported by at least one read with this many bases on each side.
Splice mismatches	The maximum number of mismatches that may appear in the anchor region of a spliced alignment.
Read mismatches	Final read alignments having more than these many mismatches are discarded.
Segment mismatches	Read segments are mapped independently, allowing up to this many mismatches in each segment alignment.
Solexa 1.3 quals	As of the Illumina GA pipeline version 1.3, quality scores are encoded in Phred-scaled base-64. Use this option for FASTQ files from pipeline 1.3 or later.

Bowtie version	specifies which Bowtie version should be used.
Bowtie -n mode	TopHat uses -v in Bowtie for initial read mapping (the default), but with this option, -n is used instead. Read segments are always mapped using -v option.
Bowtie tool path	The path to the Bowtie external tool.
SAMtools tool path	The path to the SAMtools tool. Note that the tool is available in the UGENE External Tool Package.
TopHat tool path	The path to the TopHat external tool in UGENE.
Temporary directory	The directory for temporary files.

4. Cufflinks Settings: The following page allows one to configure Cufflinks settings:

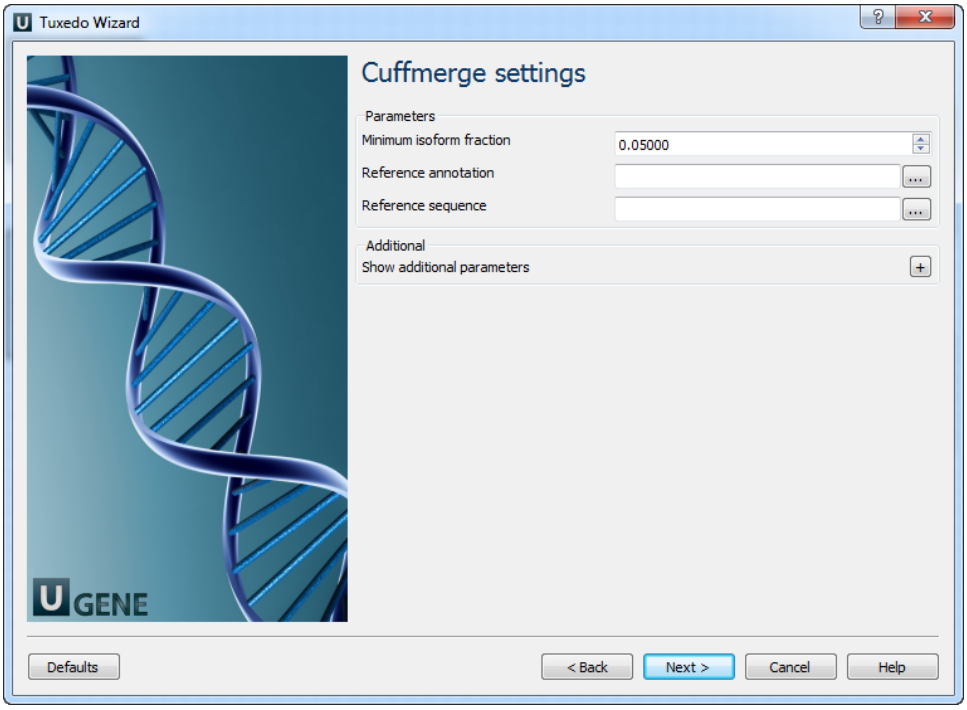


The following parameters are available:

Reference annotation	Tells Cufflinks to use the supplied reference annotation to estimate isoform expression. Cufflinks will not assemble novel transcripts and the program will ignore alignments not structurally compatible with any reference transcript.
RABT annotation	Tells Cufflinks to use the supplied reference annotation to guide Reference Annotation Based Transcript (RABT) assembly. Reference transcripts will be tiled with faux-reads to provide additional information in an assembly. The output will include all reference transcripts as well as any novel genes and isoforms that are assembled.
	Specifies RNA-seq protocol.

Library type	
Mask file	Ignore all reads that could have come from transcripts in this file. It is recommended to include any annotated rRNA, mitochondrial transcripts other abundant transcripts you wish to ignore in your analysis in this file. Due to variable efficiency of mRNA enrichment methods and rRNA depletion kits, masking these transcripts often improves the overall robustness of transcript abundance estimates.
Multi-read correct	Tells Cufflinks to do an initial estimation procedure to more accurately weight reads mapping to multiple locations in the genome.
Minisoform fraction	After calculating isoform abundance for a gene, Cufflinks filters out transcripts that it believes are very low abundance, because isoforms expressed at extremely low levels often cannot reliably be assembled, and may even be artifacts of incompletely spliced precursors of processed transcripts. This parameter is also used to filter out introns that have far fewer spliced alignments supporting them.
Frag bias correct	Providing Cufflinks with a multifasta file via this option instructs it to run the bias detection and correction algorithm which can significantly improve accuracy of transcript abundance estimates.
Pre-mRNA fraction	Some RNA-Seq protocols produce a significant amount of reads that originate from incompletely spliced transcripts, and these reads can confound the assembly of fully spliced mRNAs. Cufflinks uses this parameter to filter out alignments that lie within the intronic intervals implied by the spliced alignments. The minimum depth of coverage in the intronic region covered by the alignment is divided by the number of spliced reads, and if the result is lower than this parameter value, the intronic alignments are ignored.
Cufflinks tool path	The path to the Cufflinks external tool in UGENE.
Temporary directory	The directory for temporary files.

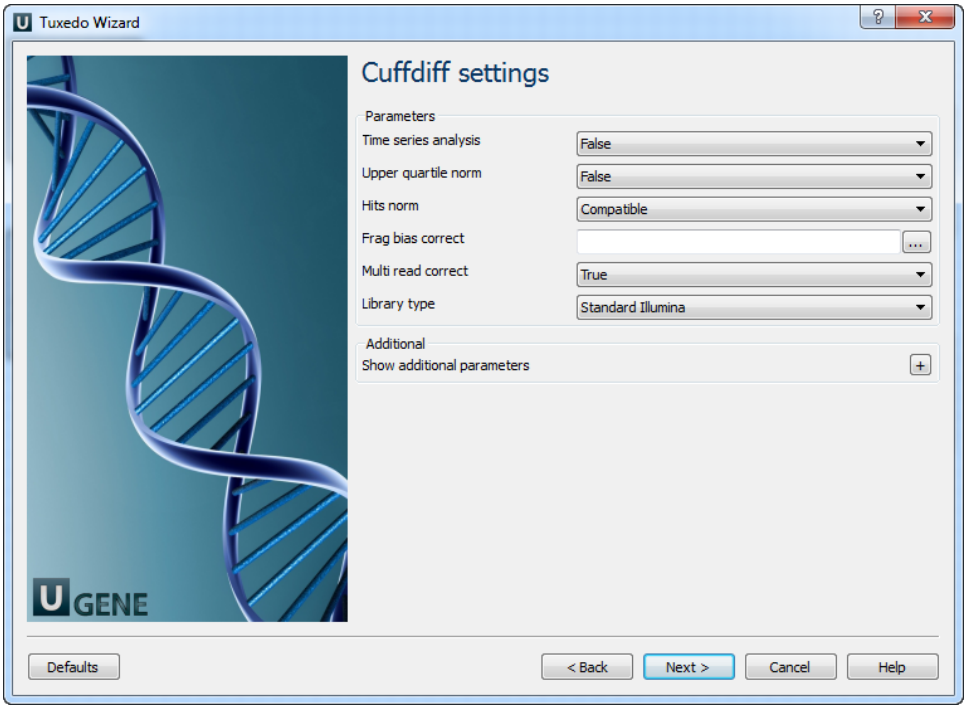
5. Cuffmerge settings: On this page, you can modify Cuffmerge parameters.



The following parameters are available:

Minimum isoform fraction	Discard isoforms with abundance below this.
Reference annotation	Merge the input assemblies together with this reference annotation.
Reference sequence	The genomic DNA sequences for the reference. It is used to assist in classifying transfrags and excluding artifacts (e.g. repeats). For example, transcripts consisting mostly of lower-case bases are classified as repeats.
Cuffcompare tool path	The path to the Cuffcompare external tool in UGENE.
Cuffmerge tool path	The path to the Cuffmerge external tool in UGENE.
Temporary directory	The directory for temporary files.

6. Cuffdiff settings: On the following page you may configure Cuffdiff settings:

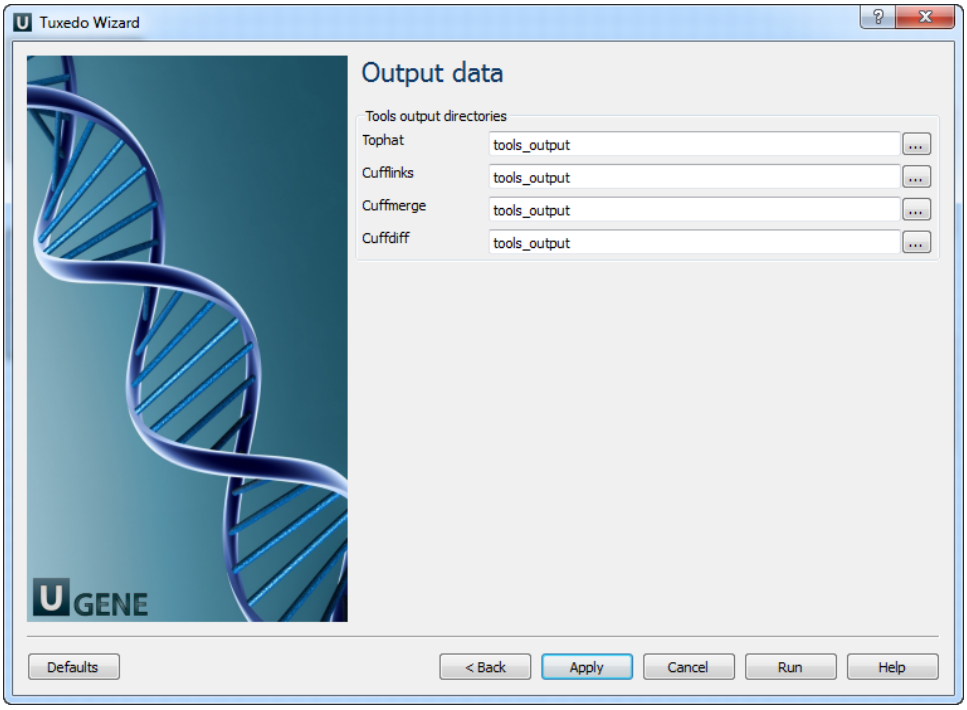


The following parameters are available:

Time series analysis	If set to True, instructs Cuffdiff to analyze the provided samples as a time series, rather than testing for differences between all pairs of samples. Samples should be provided in increasing time order.
Upper quartile norm	If set to True, normalizes by the upper quartile of the number of fragments mapping to individual loci instead of the total number of sequenced fragments. This can improve the robustness of differential expression calls for less abundant genes and transcripts.
Hits norm	Instructs how to count all fragments. Total specifies to count all fragments, including those not compatible with any reference transcript, towards the number of mapped fragments used in the FPKM denominator. Compatible specifies to use only compatible fragments. Selecting Compatible is generally recommended in Cuffdiff to reduce certain types of bias caused by differential amounts of ribosomal reads which can create the impression of falsely differentially expressed genes.
Frag bias correct	Providing the sequences your reads were mapped to instructs Cuffdiff to run bias detection and correction algorithm which can significantly improve the accuracy of transcript abundance estimates.
Multi read correct	Do an initial estimation procedure to more accurately weight reads mapping to multiple locations in the genome.
Library type	Specifies RNA-Seq protocol.
Mask file	Ignore all reads that could have come from transcripts in this file. It is recommended to include any annotated rRNA, mitochondrial transcripts other abundant transcripts you wish to ignore in your analysis in this file. Due to variable efficiency of mRNA enrichment methods and rRNA depletion kits, masking these transcripts often improves the overall robustness of transcript abundance estimates.
Min alignm	The minimum number of alignments in a locus for needed to conduct significance testing on changes in that locus observed between samples. If no testing is performed, changes in the locus are deemed not significant, and the locus' observed changes don't contribute to correction for multiple testing.

ent cou nt	
FDR	Allowed false discovery rate used in testing.
Ma x ML E iter atio ns	Sets the number of iterations allowed during maximum likelihood estimation of abundances.
Emi t cou nt tabl es	Include information about the fragment counts, fragment count variances, and fitted variance model into the report.
Cuf fdiff tool path	The path to the Cuffdiff external tool in UGENE.
Te mp orar y dire ctory	The directory for temporary files.

7. Output data: On this page, you can modify output parameters.



The work on this pipeline was supported by grant RUB1-31097-NO-12 from NIAID.

Variation Annotation with SnpEff

SnpEff is a variant annotation and effect prediction tool. It annotates and predicts the effects of genetic variants (such as amino acid changes).

A typical SnpEff use case would be:
-Input: The inputs are predicted variants (SNPs, insertions, deletions and MNPs). The input file is usually obtained as a result of a sequencing experiment, and it is usually in variant call format (VCF).
-Output: SnpEff analyzes the input variants. It annotates the variants and calculates the effects they produce on known genes (e.g. amino acid changes).

How to Use This Sample

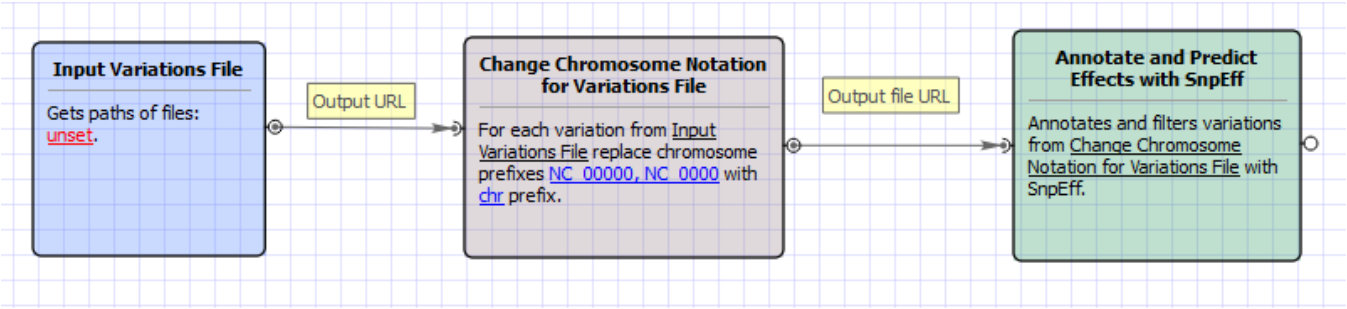
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Variation Annotation with SnpEff" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

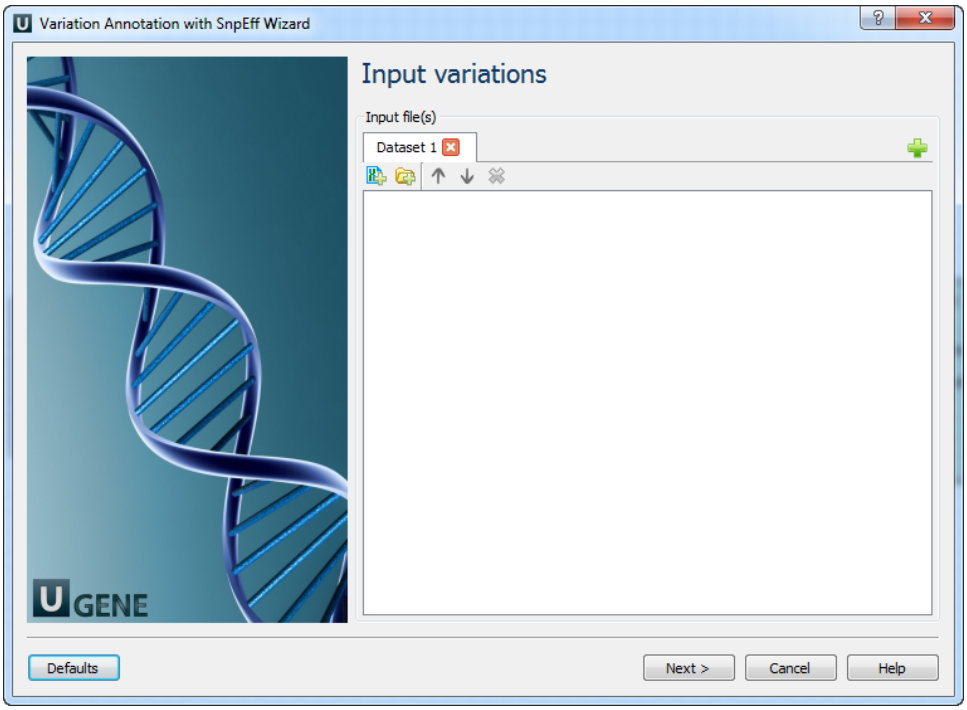
The opened workflow looks as follows:



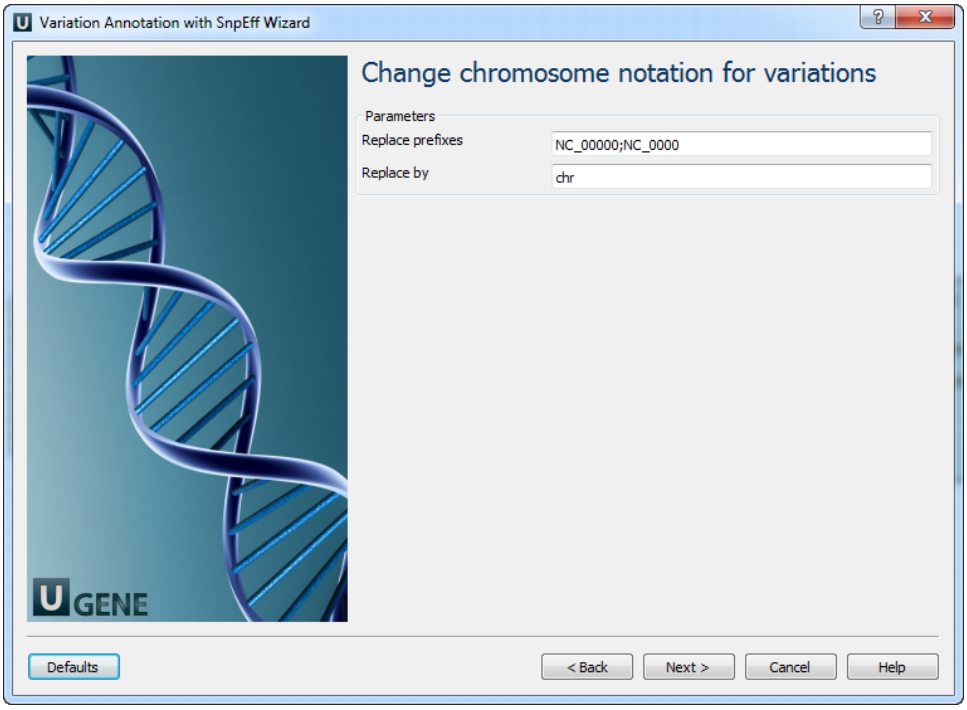
Workflow Wizard

The wizard has 3 pages.

1. Input Variations: On this page you must input variations file(s).



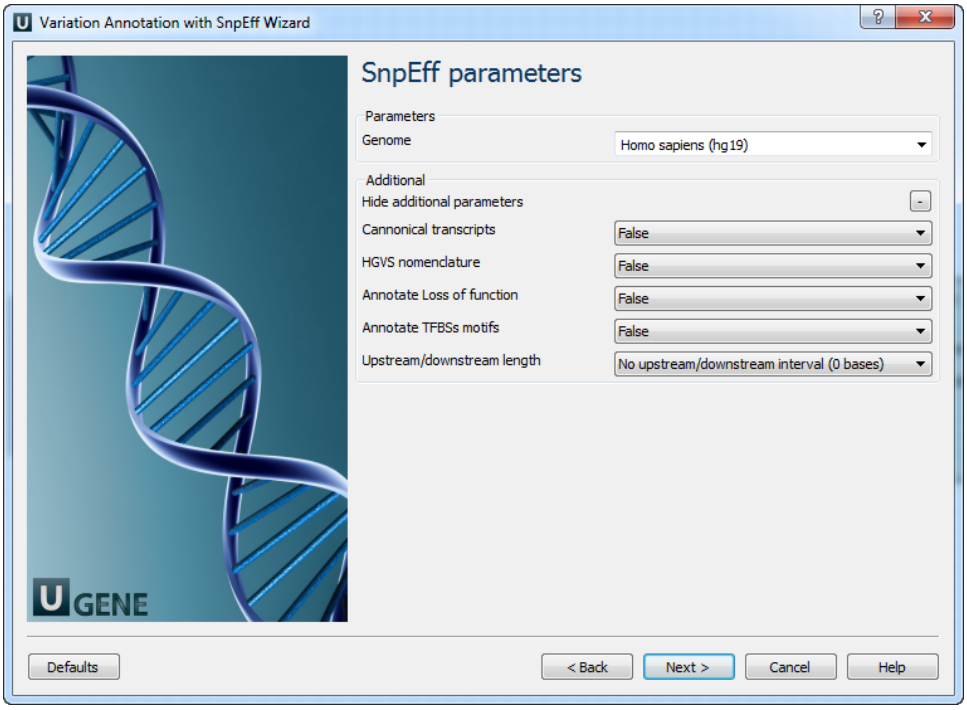
2. Change chromosome notation for variations: On this page you can change the chromosome notation for variations.



The following parameters are available:

Replace prefixes	Input the list of chromosome prefixes that you would like to replace. For example "NC_000". Separate different prefixes by semicolons.
Replace by	Input the prefix that should be set instead, for example "chr".

3. SnpEff Parameters: On this page you can modify SnpEff parameters.

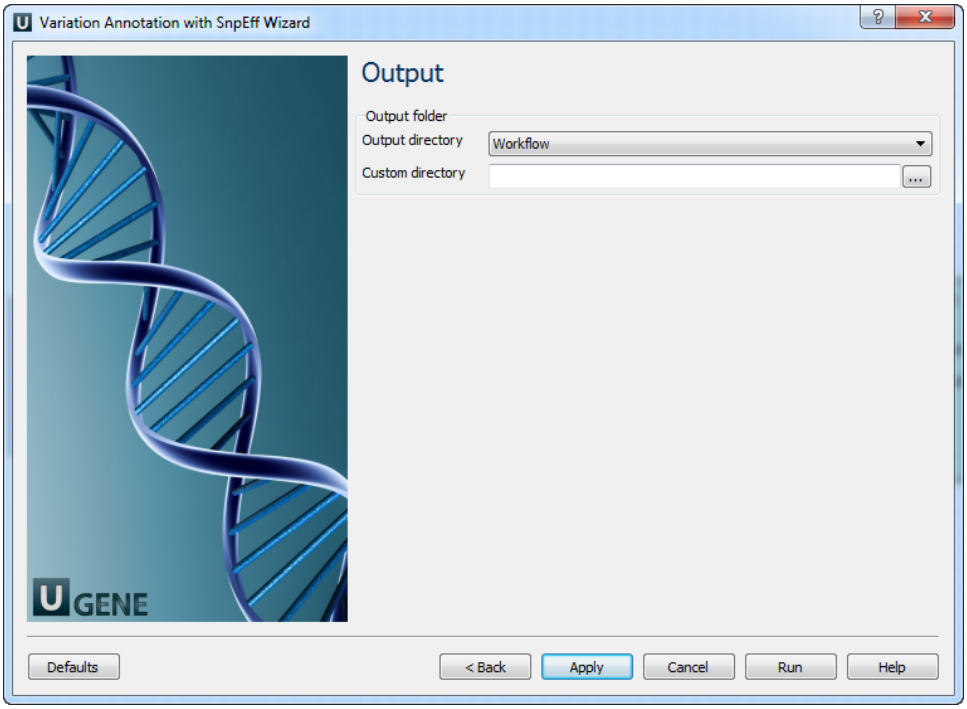


The following parameters are available:

Genome	Select the target genome. Genome data will be downloaded if it is not found.
Cannonical transcripts	Use only cannonical transcripts
HGVS nomenclature	Annotate using HGVS nomenclature

Annotate Loss of function	Annotate Loss of function (LOF) and Nonsense mediated decay (NMD)
Annotate TFBSs motifs	Annotate transcription factor binding site motifs (only available for latest GRCh37)
Upstream/downstream length	Upstream and downstream interval size. Eliminate any upstream and downstream effect by using 0 length

4. Output: On this page you need input output parameters.



Call Variants with SAMtools

Call variants in UGENE can be done using SAMtools mpileup and bcftools view utilities. To read additional information about SAMtools and its utilities visit SAMTools homepage. Both utilities are embedded into UGENE and there is no need in additional configuration.

How to Use This Sample

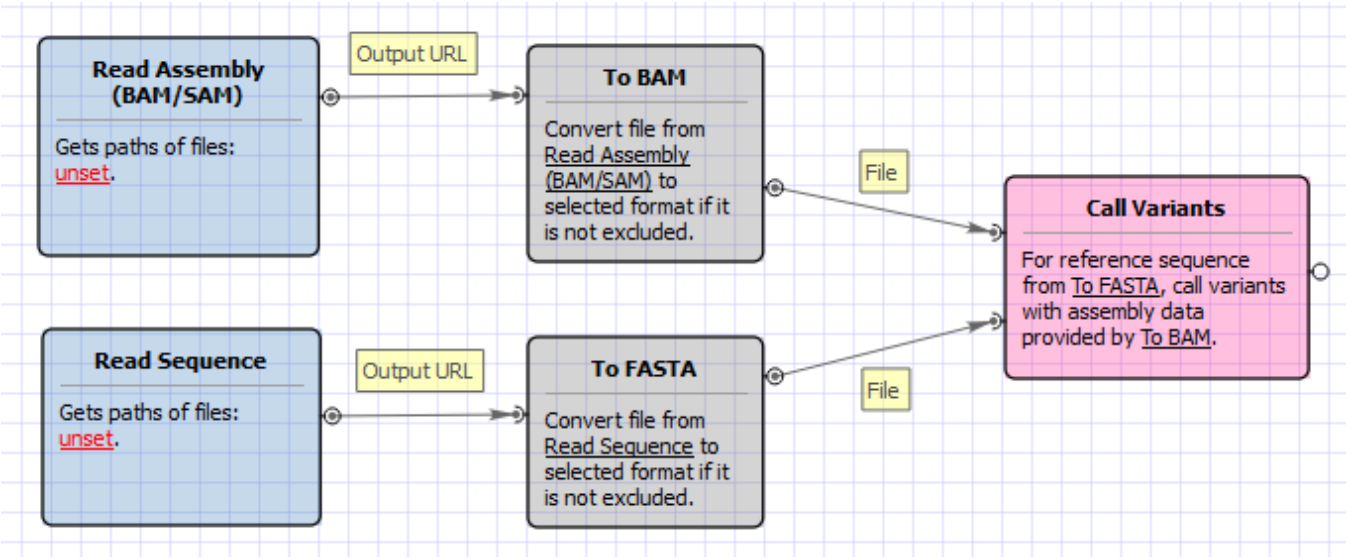
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Call Variants with SAMtools" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

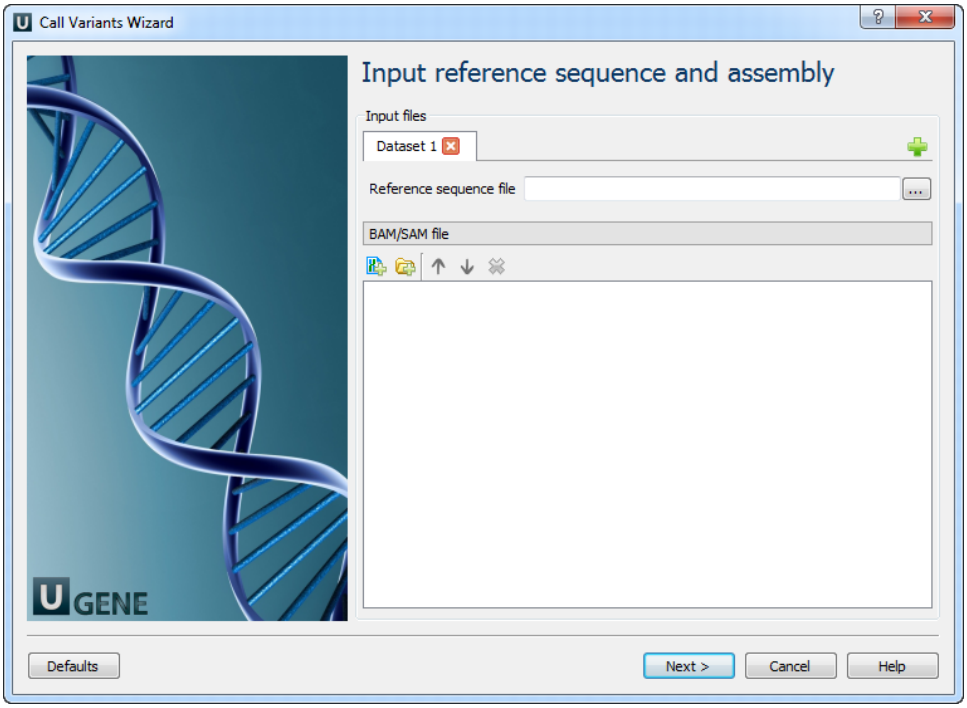
The workflow looks as follows:



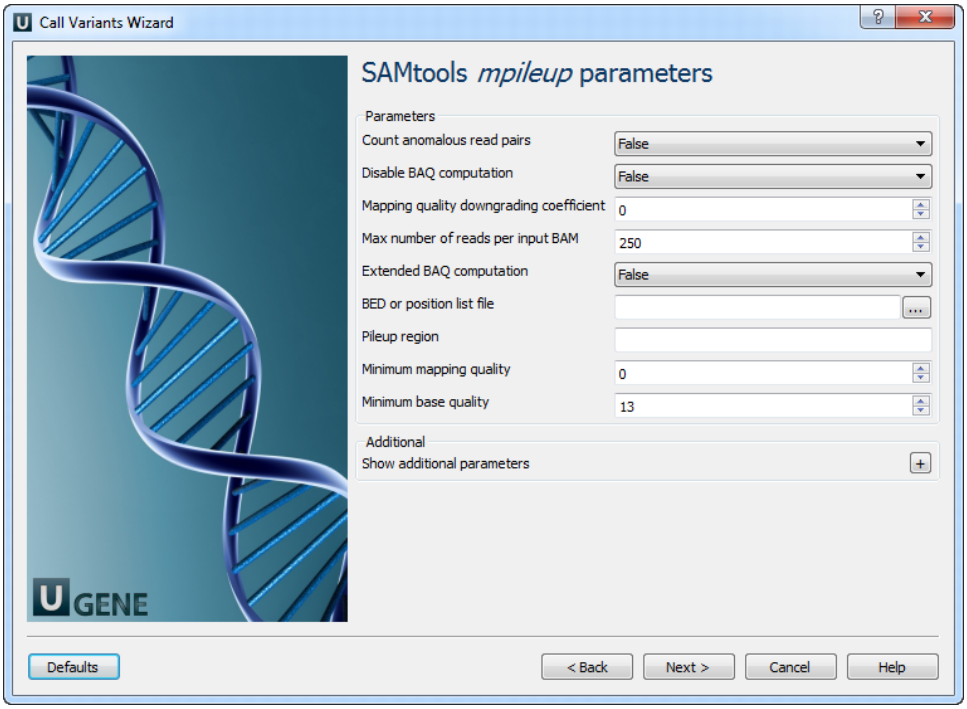
Workflow Wizard

The wizard has 5 pages.

1. Input reference sequence and assembly: Here you need to input a file with a reference sequence and a sorted BAM or SAM file. Note that the input BAM or SAM file may be unsorted.



2. SAMTolls *mpileup* parameters: Here you can change default parameters of the SAMtools mpileup utility. To show additional parameters click the + button.

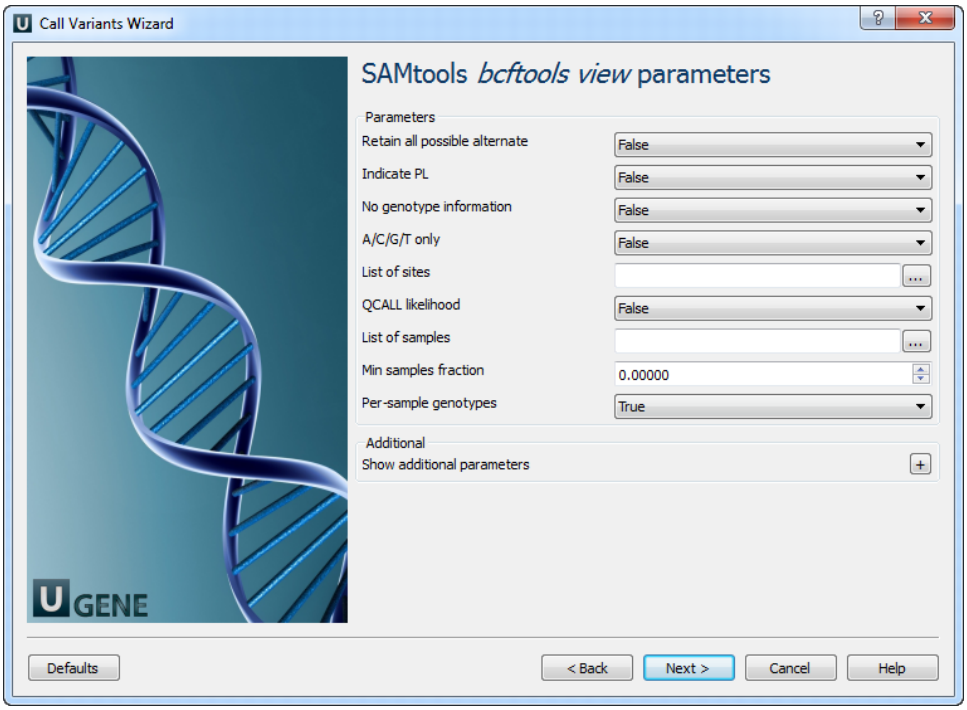


The following parameters are available:

Count anomalous read pairs	Do not skip anomalous read pairs in variant calling.
Disable BAQ computation	Disable probabilistic realignment for the computation of base alignment quality (BAQ). BAQ is the Phred-scaled probability of a read base being misaligned. Applying this option greatly helps to reduce false SNPs caused by misalignments.
Mapping quality downgroading coefficient	Coefficient for downgrading mapping quality for reads containing excessive mismatches. Given a read with a phred-scaled probability q of being generated from the mapped position, the new mapping quality is about sqrt((INT-q)/INT)*INT. A zero value disables this functionality; if enabled, the recommended value for BWA is 50.
Max number of reads per input BAM	At a position, read maximally INT reads per input BAM.
Extended BAQ computation	Extended BAQ computation. This option helps sensitivity especially for MNPs, but may hurt specificity a little bit.
BED or position list file	BED or position list file containing a list of regions or sites where pileup or BCF should be generated.
Pileup region	Only generate pileup in region STR.
Minimum mapping quality	Minimum mapping quality for an alignment to be used.
Minimum base quality	Minimum base quality for a base to be considered.
Illumina-1.3 +encoding	Assume the quality is in the Illumina 1.3+ encoding.
Gap extension error	Phred-scaled gap extension sequencing error probability. Reducing INT leads to longer indels.
Homopolymer errors coefficient	Coefficient for modeling homopolymer errors. Given an l-long homopolymer run, the sequencing error of an indel of size s is modeled as INT*s/l.
No INDELs	Do not perform INDEL calling.

Max INDEL depth	Skip INDEL calling if the average per-sample depth is above INT.
Gap open error	Phred-scaled gap open sequencing error probability. Reducing INT leads to more indel calls.
List of platforms for indels	Comma dilimited list of platforms (determined by @RG-PL) from which indel candidates are obtained. It is recommended to collect indel candidates from sequencing technologies that have low indel error rate such as ILLUMINA.

3. SAMTolls *bcftools* view parameters: The next page allows one to configure SAMtools bcftools view utility parameters.

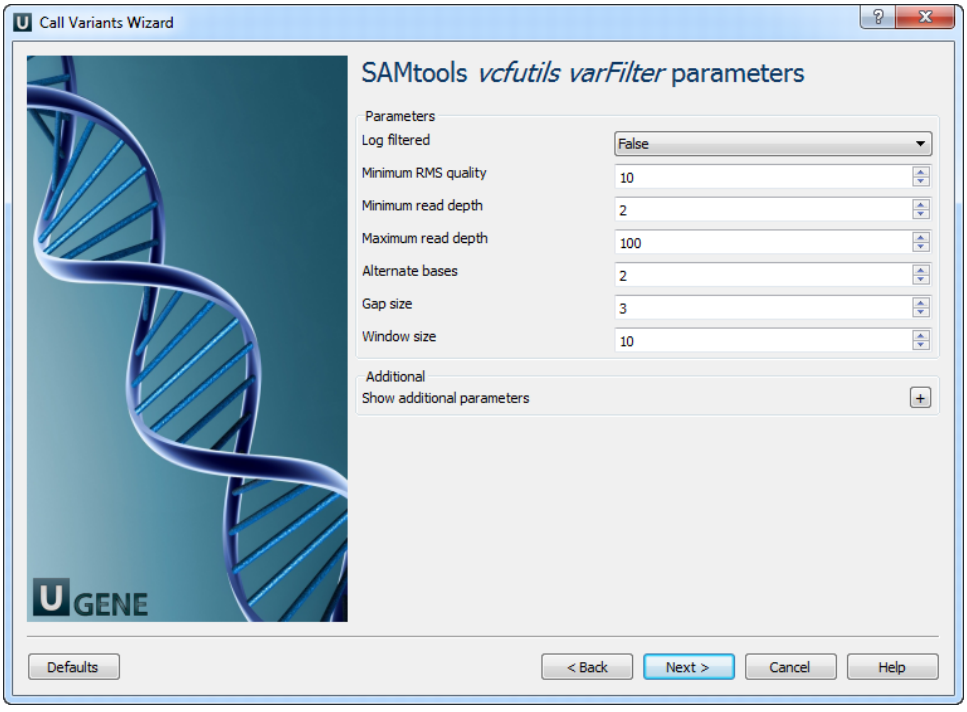


The following parameters are available:

Retain all possible alternative	Retain all possible alternate alleles at variant sites. By default, the view command discards unlikely alleles.
Indicate PL	Indicate PL is generated by r921 or before (ordering is different).
No genotype information	Suppress all individual genotype information.
A/C/G/T only	Skip sites where the REF field is not A/C/G/T.
List of sites	List of sites at which information are outputted.
QCALL likelihood	Output the QCALL likelihood format.
List of samples	List of samples to use. The first column in the input gives the sample names and the second gives the ploidy, which can only be 1 or 2. When the 2nd column is absent, the sample ploidy is assumed to be 2. In the output, the ordering of samples will be identical to the one in FILE.
	Skip loci where the fraction of samples covered by reads is below FLOAT.

Min samples fraction	
Per-sample genotypes	Call per-sample genotypes at variant sites.
INDEL-to-SNP Ratio	Ratio of INDEL-to-SNP mutation rate.
Gap open error	Phred-scaled gap open sequencing error probability. Reducing INT leads to more indel calls.
Max P (ref|D)	A site is considered to be a variant if P(ref|D).
Pair /trio calling	Enable pair/trio calling. For trio calling, option -s is usually needed to be applied to configure the trio members and their ordering. In the file supplied to the option -s, the first sample must be the child, the second the father and the third the mother. The valid values of STR are “pair”, “trioauto”, “trioxd” and “trioxs”, where “pair” calls differences between two input samples, and “trioxd” (“trioxs”) specifies that the input is from the X chromosome non-PAR regions and the child is a female (male).
N group-1 samples	Number of group-1 samples. This option is used for dividing the samples into two groups for contrast SNP calling or association test. When this option is in use, the following VCF INFO will be outputted: PC2, PCHI2 and QCHI2.
N permutations	Number of permutations for association test (effective only with -1).
Max P (chi^2)	Only perform permutations for P(chi^2).

4. SAMTolls *vcfutils varFilter* parameters: The next page allows one to configure SAMtools vcfutils parameters.

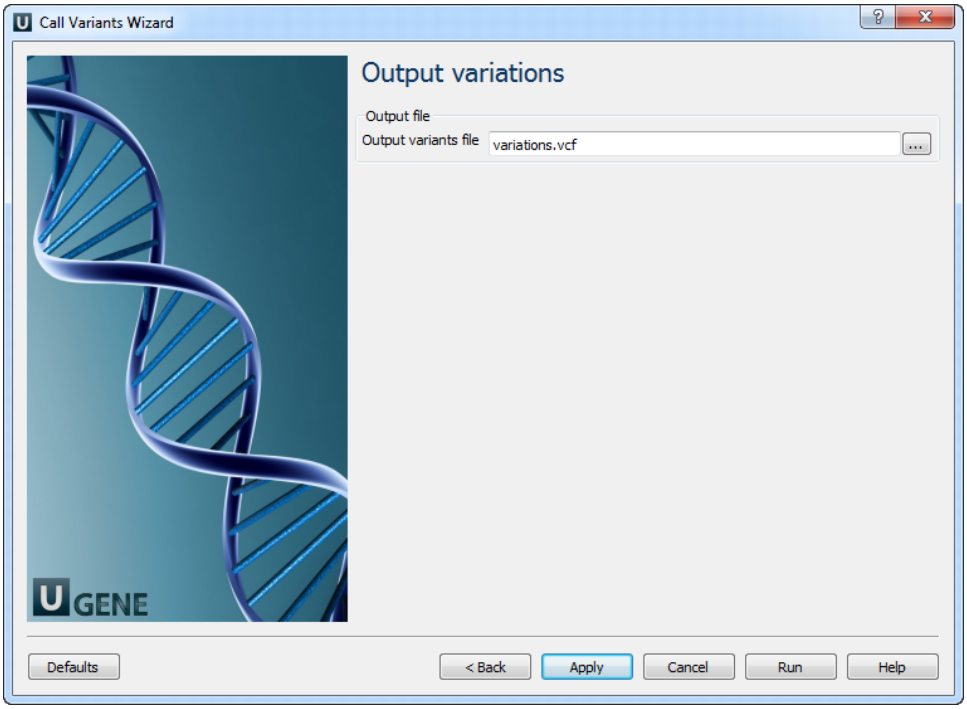


The following parameters are available:

Log filtered	Print filtered variants into the log (varFilter) (-p).
Minimum RMS quality	Minimum RMS mapping quality for SNPs (varFilter) (-Q).

Minimum read depth	Minimum read depth (varFilter) (-d).
Maximum read depth	Maximum read depth (varFilter) (-D).
Alternate bases	Minimum number of alternate bases (varFilter) (-a).
Gap size	SNP within INT bp around a gap to be filtered (varFilter) (-w).
Window size	Window size for filtering adjacent gaps (varFilter) (-W).
Strand bias	Minimum P-value for strand bias (given PV4) (varFilter) (-1).
BaseQ bias	Minimum P-value for baseQ bias (varFilter) (-2).
MapQ bias	Minimum P-value for mapQ bias (varFilter) (-3).
End distance bias	Minimum P-value for end distance bias (varFilter) (-4).
HWE	Minimum P-value for HWE (plus F<0) (varFilter) (-e).

5. Output variations: On this page you can modify output parameters.



The work on this pipeline was supported by grant RUB1-31097-NO-12 from NIAID.

Variant Calling and Effect Prediction

The workflow sample, described below, call variants for an input assembly and a reference sequence using SAMtools mpileup and bcftool. Predict effects of the variants using SnpEff.

How to Use This Sample

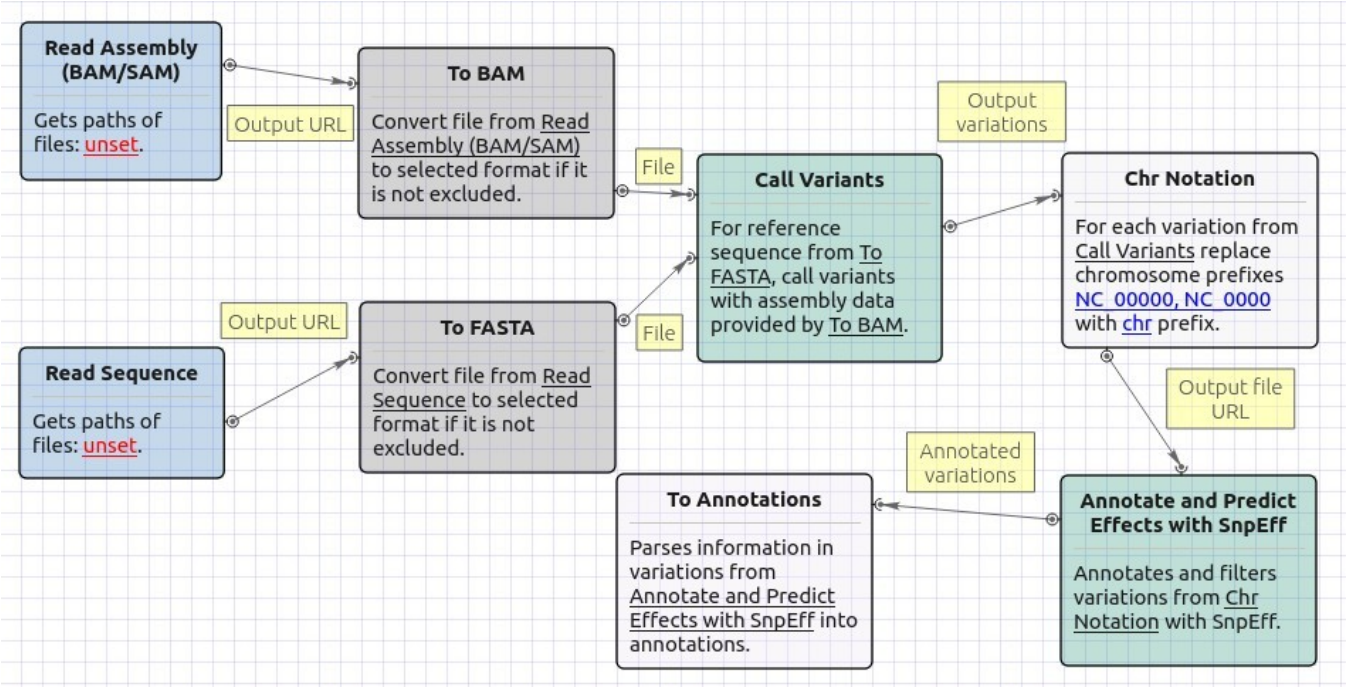
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Variant Calling and Effect Prediction" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

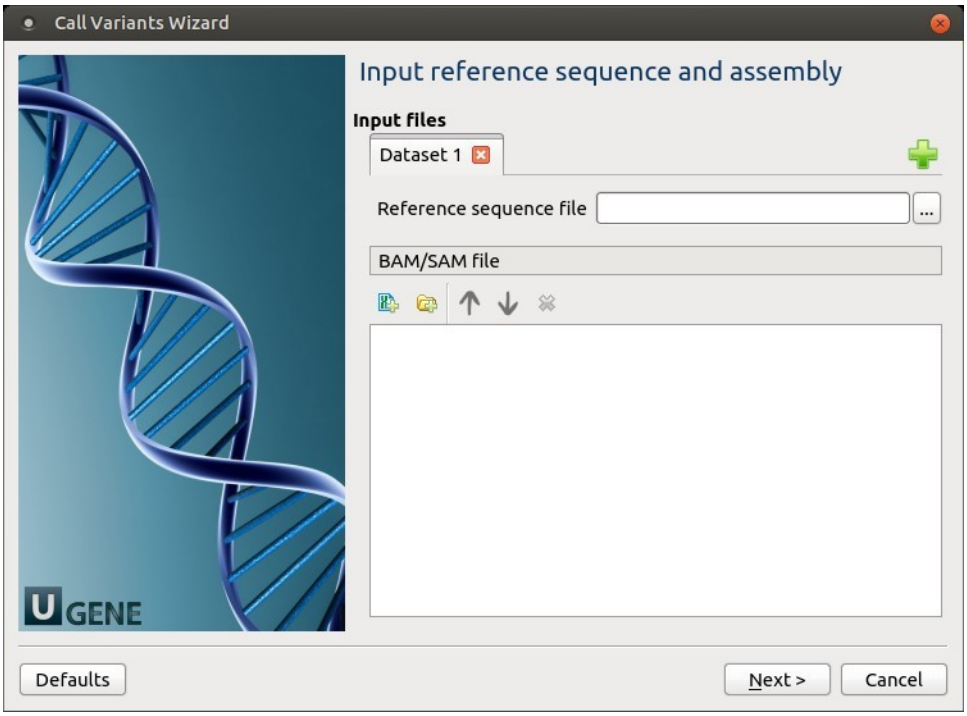
The opened workflow looks as follows:



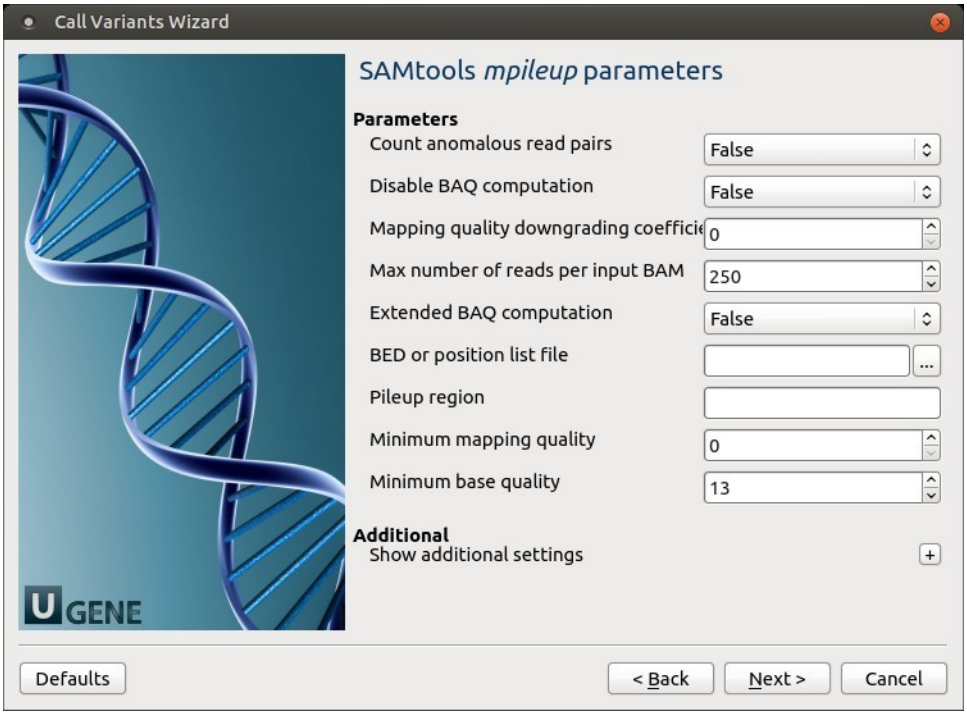
Workflow Wizard

The wizard has 7 pages.

1. Input reference sequence and assembly On this page, input files must be set.



2. SAMtools mpileup parameters: The SAMtoolsmpileup parameters can be changed here.

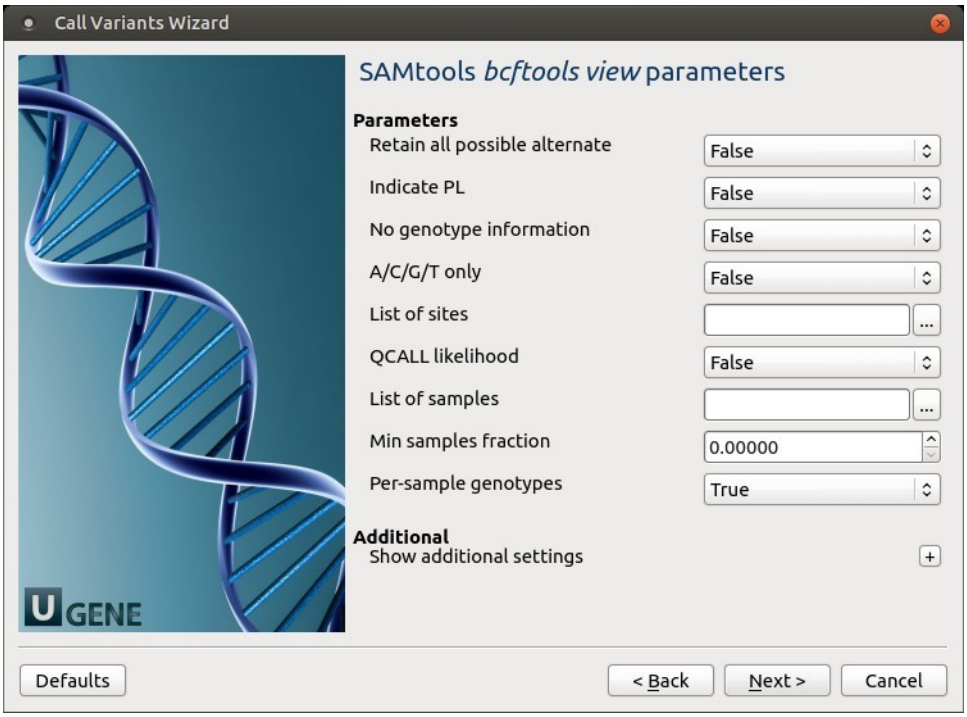


The following parameters are available:

Count anomalous read pairs	Do not skip anomalous read pairs in variant calling(mpileup)(-A).
Disable BAQ computation	Disable probabilistic realignment for the computation of base alignment quality (BAQ). BAQ is the Phred-scaled probability of a read base being misaligned. Applying this option greatly helps to reduce false SNPs caused by misalignments. (mpileup)(-B).
Mapping quality downgrading coefficient	Coefficient for downgrading mapping quality for reads containing excessive mismatches. Given a read with a phred-scaled mapping quality q of being generated from the mapped position, the new mapping quality is about sqrt((INT-q)/INT)*INT. A zero value disables this functionality; if enabled, the recommended value for BWA is 50 (mpileup)(-C).
Max number of reads per input BAM	At a position, read maximally the number of reads per input BAM (mpileup)(-d).
Extended BAQ computation	Extended BAQ computation. This option helps sensitivity especially for MNPs, but may hurt specificity a little bit (mpileup)(-E).
BED or position list file	BED or position list file containing a list of regions or sites where pileup or BCF should be generated (mpileup)(-l).
Pileup region	Only generate pileup in region STR (mpileup)(-r).
Minimum mapping quality	Minimum mapping quality for an alignment to be used (mpileup)(-q).
Minimum base quality	Minimum base quality for a base to be considered (mpileup)(-Q).
Illumina-1.3 + encoding	Assume the quality is in the Illumina 1.3+ encoding (mpileup)(-6).
Gap extension error	Phred-scaled gap extension sequencing error probability. Reducing INT leads to longer indels (mpileup)(-e).
Homopolymer errors coefficient	Coefficient for modeling homopolymer errors. Given an l-long homopolymer run, the sequencing error of an indel of size s is modeled as INT*s/l (mpileup)(-h).

No INDELs	Do not perform INDEL calling (mpileup)(-I).
Max INDEL depth	Skip INDEL calling if the average per-sample depth is above INT (mpileup)(-L).
Gap open error	Phred-scaled gap open sequencing error probability. Reducing INT leads to more indel calls (mpileup)(-o).
List of platforms for indels	Comma delimited list of platforms (determined by @RG-PL) from which indel candidates are obtained.It is recommended to collect indel candidates from sequencing technologies that have low indel error rate such as ILLUMINA (mpileup)(-P).

3. SAMtools bcftools view parameters: The SAMtoolsbcftools parameters can be changed here.

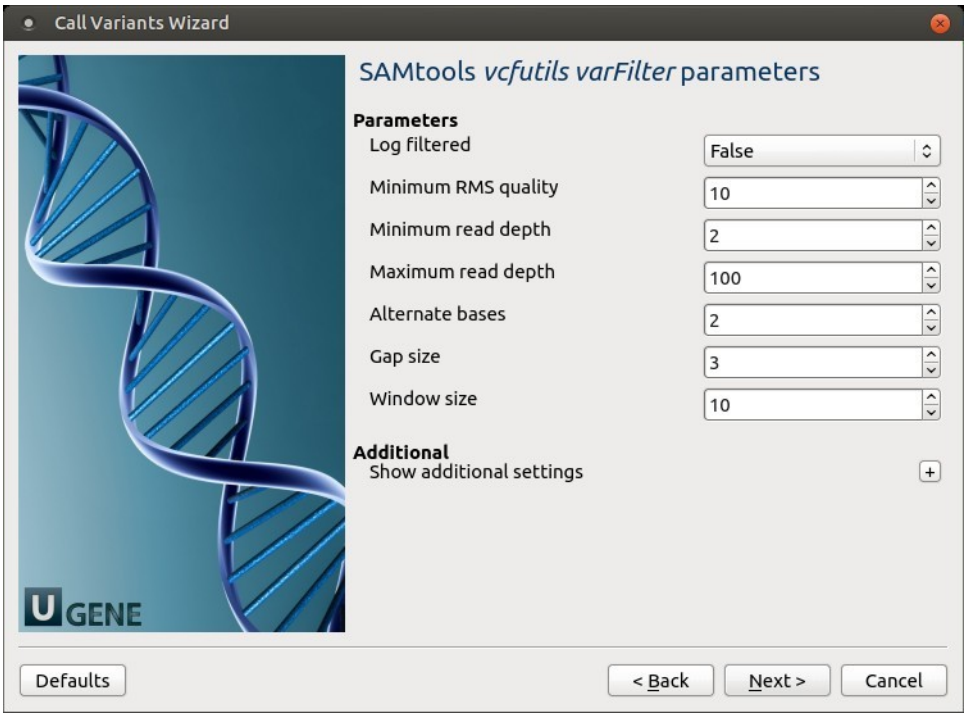


The following parameters are available:

Retain all possible alternate	Retain all possible alternate alleles at variant sites. By default, the view command discards unlikely alleles.
Indicate PL	Indicate PL is generated by r921 or before (ordering is different).
No genotype information	Suppress all individual genotype information.
A/C/G/T only	Skip sites where the REF field is not A/C/G/T.
List of sites	List of sites at which information are outputted.
QCALL likelihood	Output the QCALL likelihood format.
List of samples	List of samples to use. The first column in the input gives the sample names and the second gives the ploidy, which can only be 1 or 2. When the 2nd column is absent, the sample ploidy is assumed to be 2. In the output, the ordering of samples will be identical to the one in FILE.
Min	Skip loci where the fraction of samples covered by reads is below FLOAT.

samples fraction	
Per-sample genotypes	Call per-sample genotypes at variant sites.
INDEL-to-SNP Ratio	Ratio of INDEL-to-SNP mutation rate.
Max p(ref|D)	A site is considered to be a variant if P(ref|D).
Prior allele frequency spectrum	If STR can be full, cond2, flat or the file consisting of error output from a previous variant calling run (bcf view)(-P).
Mutation rate	Scaled mutation rate for variant calling (bcf view)(-t).
Pair/trio calling	Enable pair/trio calling. For trio calling, option -s is usually needed to be applied to configure the trio members and their ordering. In the file supplied to the option -s, the first sample must be the child, the second the father and the third the mother. The valid values of STR are “pair”, “trioauto”, “trioxd” and “trioxs”, where “pair” calls differences between two input samples, and “trioxd” (“trioxs”) specifies that the input is from the X chromosome non-PAR regions and the child is a female (male).
N group-1 samples	Number of group-1 samples. This option is used for dividing the samples into two groups for contrast SNP calling or association test. When this option is in use, the following VCF INFO will be outputted: PC2, PCHI2 and QCHI2.
N permutations	Number of permutations for association test (effective only with -1).
Max P(chi^2)	Only perform permutations for P(chi^2).

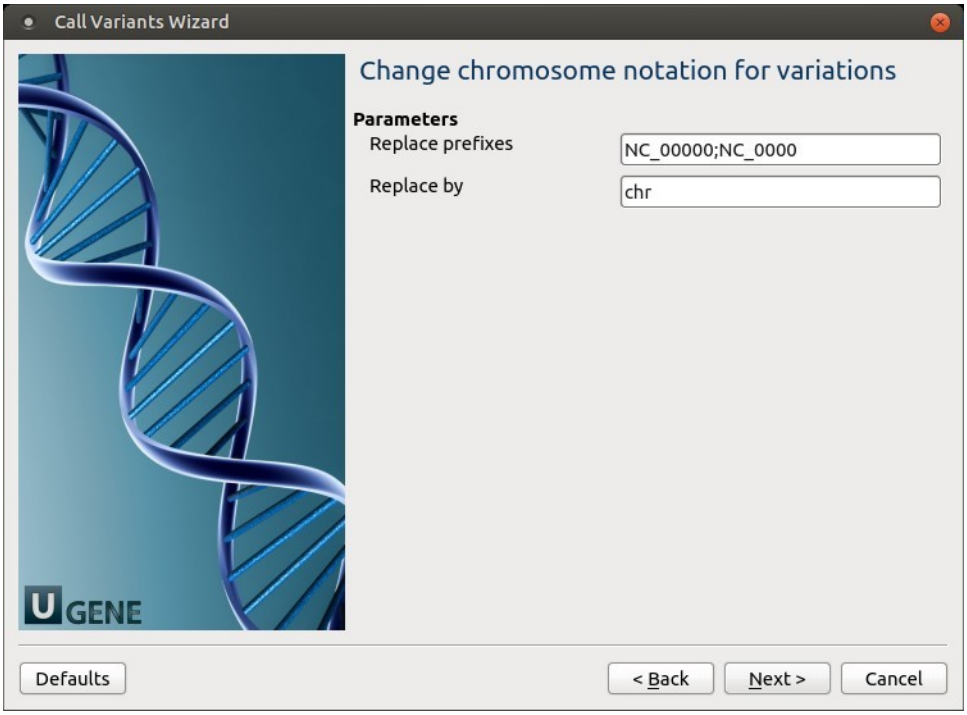
4. SAMTolls *vcfutils varFilter* parameters: The next page allows one to configure SAMtools vcfutils parameters.



The following parameters are available:

Log filtered	Print filtered variants into the log (varFilter) (-p).
Minimum RMS quality	Minimum RMS mapping quality for SNPs (varFilter) (-Q).
Minimum read depth	Minimum read depth (varFilter) (-d).
Maximum read depth	Maximum read depth (varFilter) (-D).
Alternate bases	Minimum number of alternate bases (varFilter) (-a).
Gap size	SNP within INT bp around a gap to be filtered (varFilter) (-w).
Window size	Window size for filtering adjacent gaps (varFilter) (-W).
Strand bias	Minimum P-value for strand bias (given PV4) (varFilter) (-1).
BaseQ bias	Minimum P-value for baseQ bias (varFilter) (-2).
MapQ bias	Minimum P-value for mapQ bias (varFilter) (-3).
End distance bias	Minimum P-value for end distance bias (varFilter) (-4).
HWE	Minimum P-value for HWE (plus F<0) (varFilter) (-e).

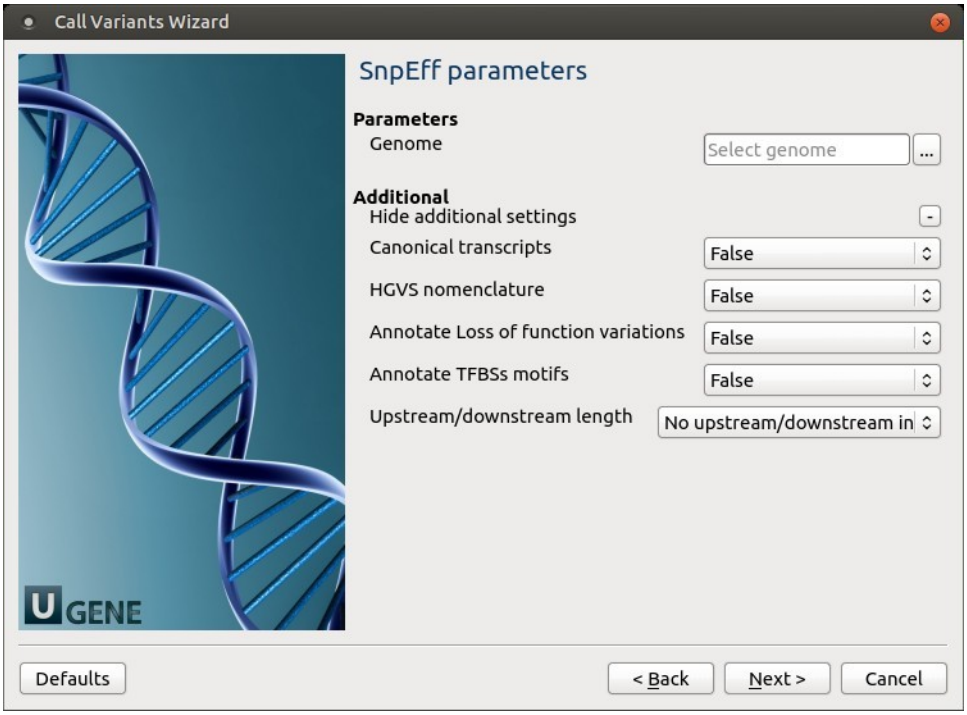
5. Change chromosome notation for variations: The next page allows change chromosome notation for variations.



The following parameters are available:

Replace prefixes	Input the list of chromosome prefixes that you would like to replace, for example, "NC_000". Separate different prefixes by semicolons.
Replace by	Input the prefix that should be set instead, for example, "chr".

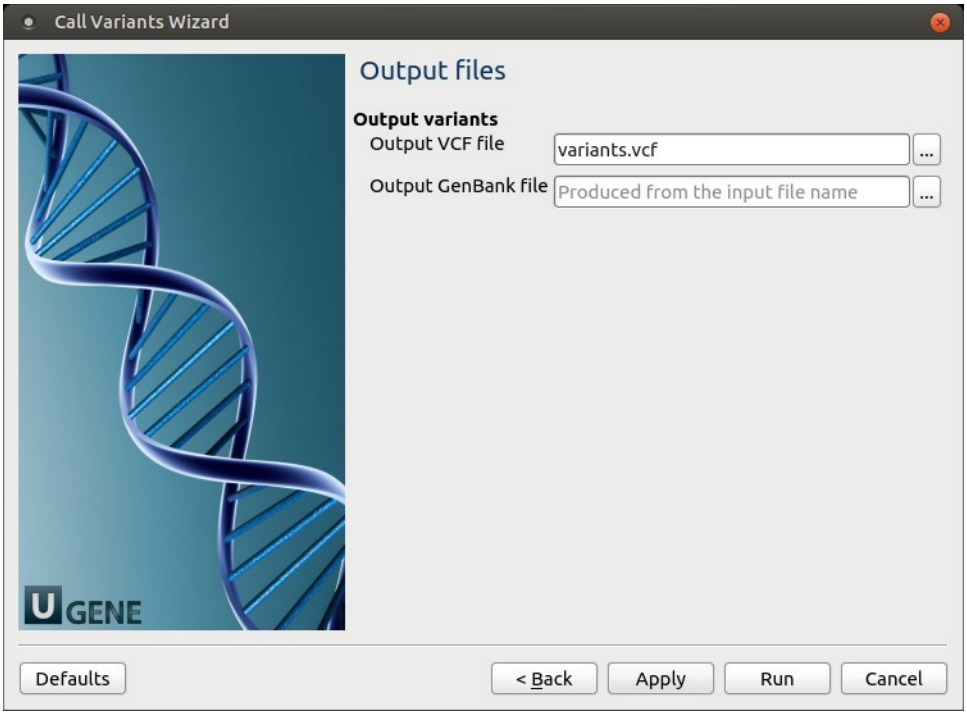
6. SnpEff parameters: The next page allows one to configure SnpEff parameters.



The following parameters are available:

Genome	Select the target genome. Genome data will be downloaded if it is not found.
Canonical transcripts	Use only canonical transcripts
HGVS nomenclature	Annotate using HGVS nomenclature
Annotate Loss of function variations	Annotate Loss of function variations (LOF) and Nonsense mediated decay (NMD)
Annotate TFBSs motifs	Annotate transcription factor binding site motifs (only available for latest GRCh37)
Upstream/downstream length	Upstream and downstream interval size. Eliminate any upstream and downstream effect by using 0 length

7. Output files Page: On this page, output files can be selected:



Raw ChIP-Seq Data Processing

The component for ChIP-seq data analysis is not installed by default. To use this sample, add the component via the UGENE Online Installer or, if you used an offline installer, manually configure the package, see "Configure ChIP-seq Analysis Data" chapter of the manual.

Use this workflow sample to process raw ChIP-seq next-generation sequencing (NGS) data from the Illumina platform. The processing includes:

- *Filtration:*
  - Filtering of the NGS short reads by the CASAVA 1.8 header;
  - Trimming of the short reads by quality;
- *Mapping:*
  - Mapping of the short reads to the specified reference sequence (the BWA-MEM tool is used in the sample);
- *Post-filtration:*
  - Filtering of the aligned short reads by SAMtools to remove reads with low mapping quality, unpaired/unaligned reads;
  - Removing of duplicated short reads.

The result of the data processing is provided in the BED format. Intermediate data files from the filtration and mapping steps are also available in the output.

How to Use This Sample

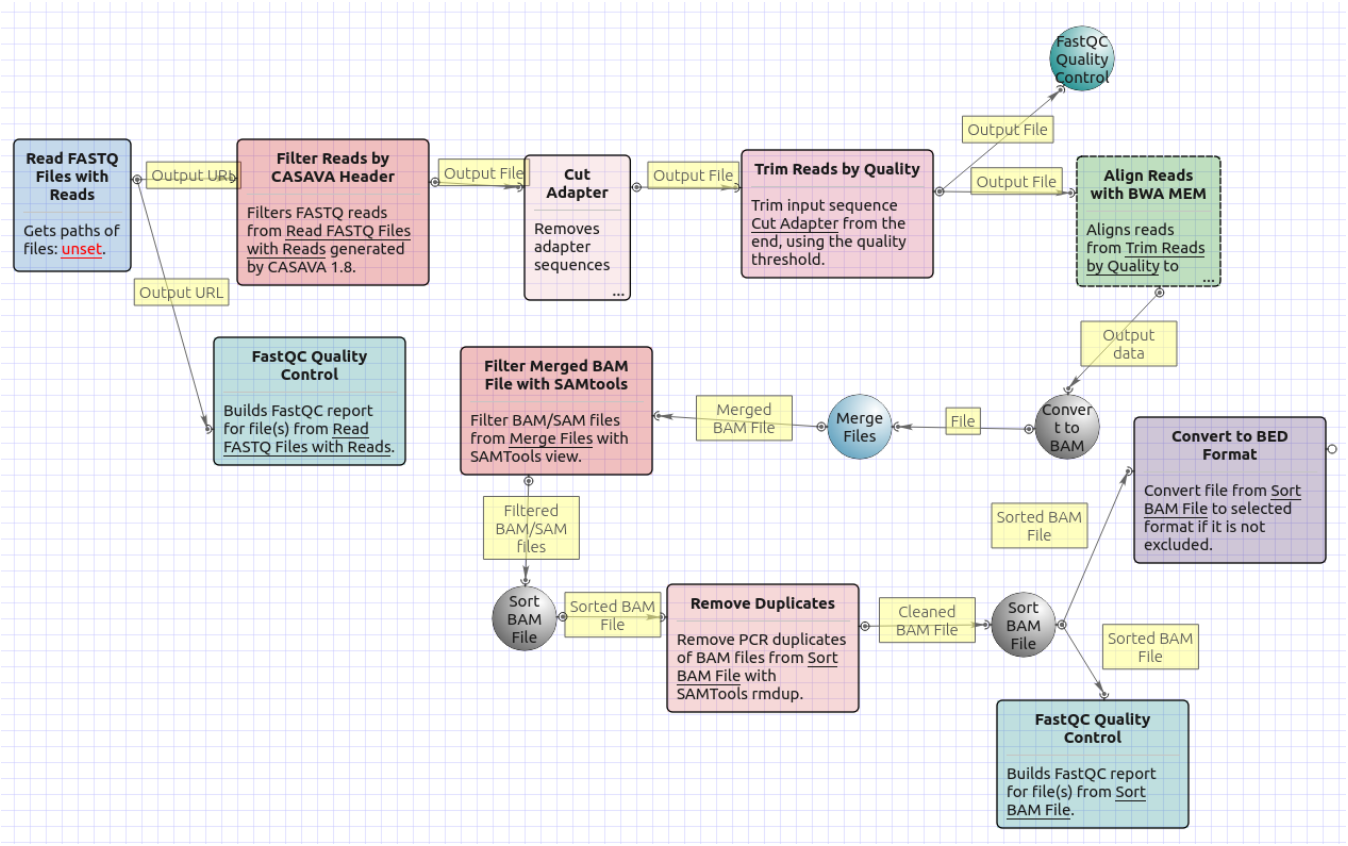
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

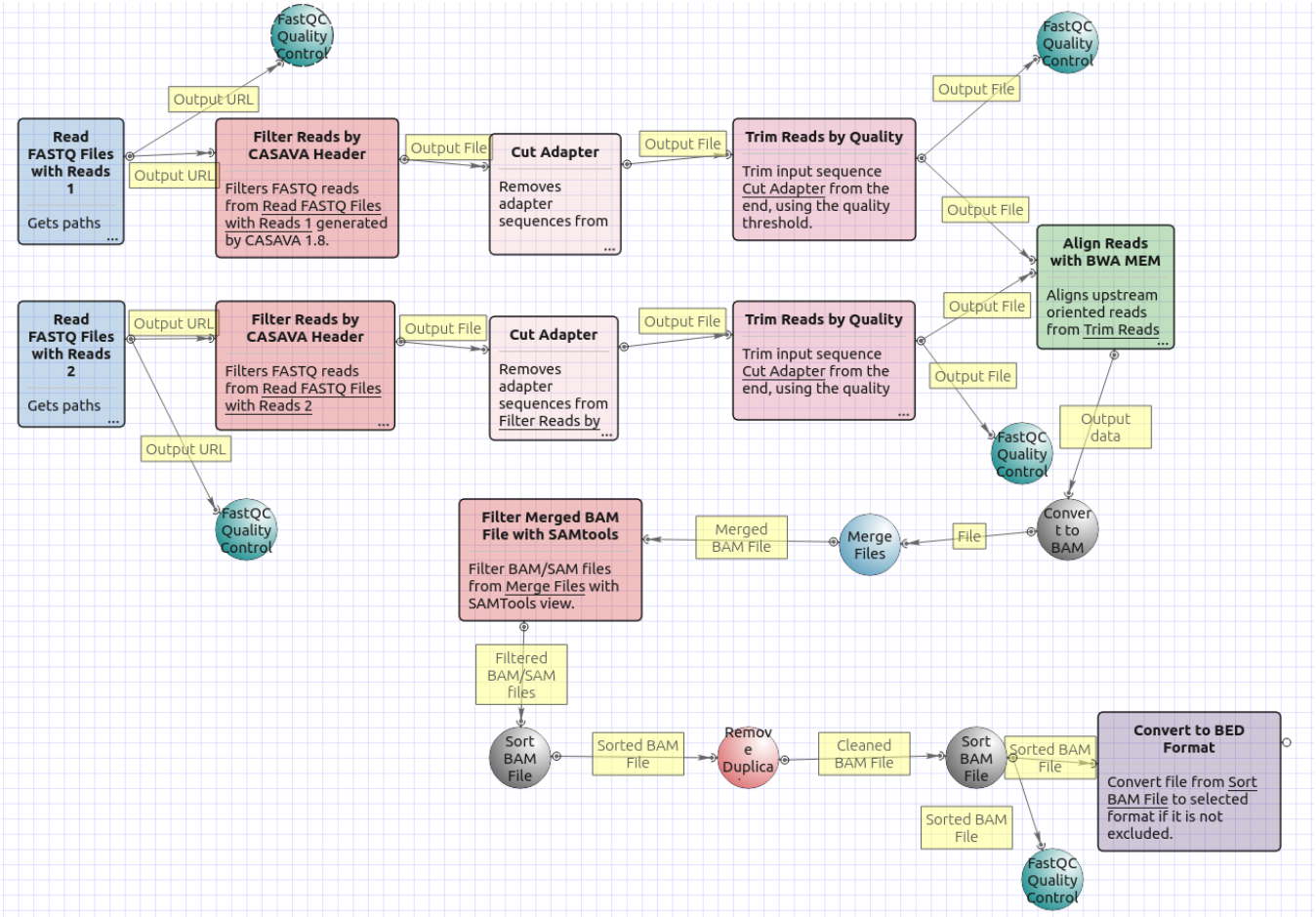
The workflow sample "Raw ChIP-Seq processing" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

There are two versions of the workflow available. The workflow for single-end reads looks as follows:



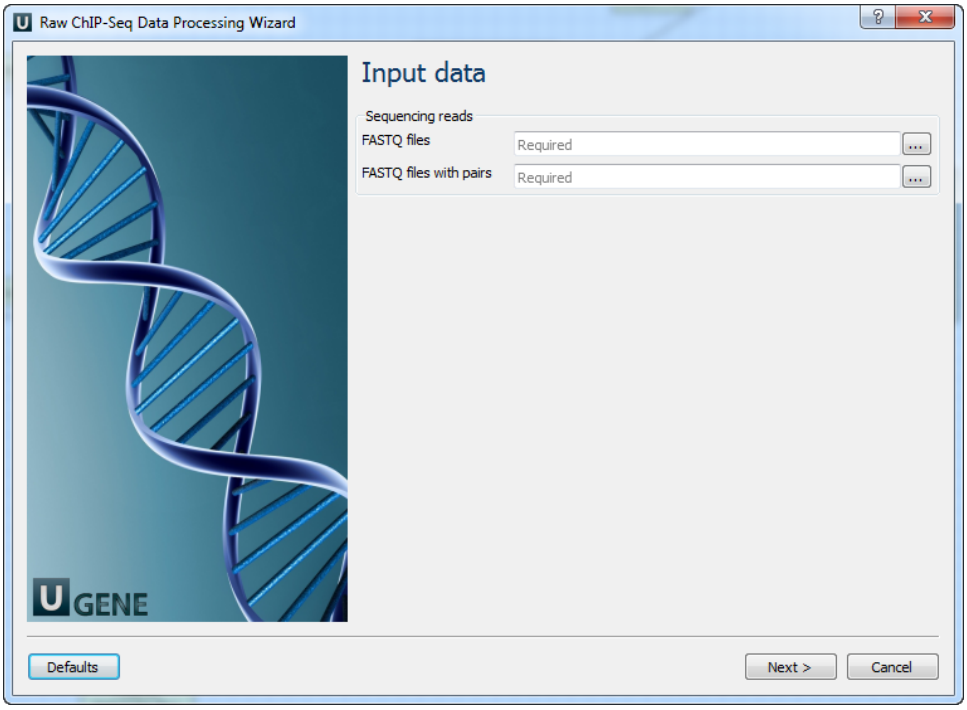
The workflow for paired-end short appearance is the following:



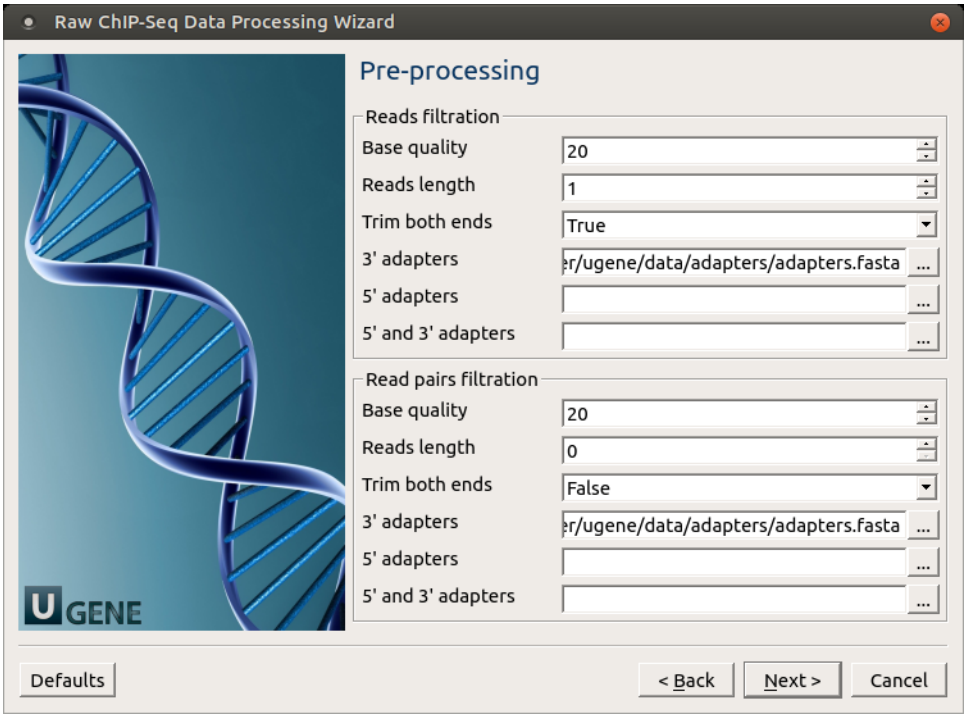
Workflow Wizard

The workflows have the similar wizards. The wizard for paired-end reads has 5 pages.

1. Input data: On this page you must input FASTQ file(s).



2. Pre-processing: On this page you can modify filtration parameters.

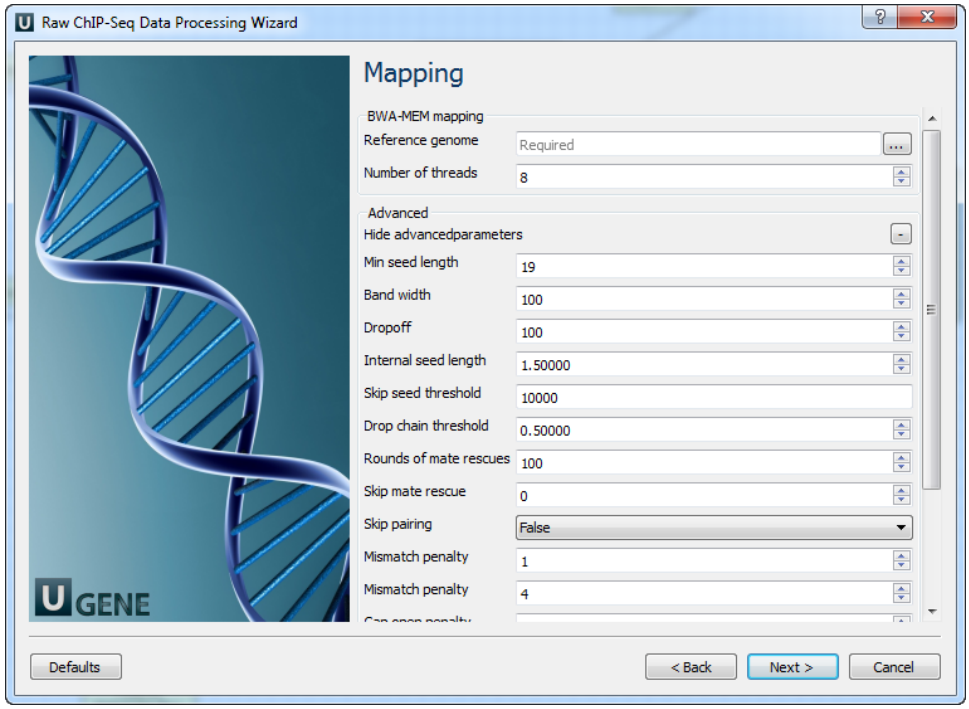


The following parameters are available for reads and reads pairs filtration:

Base quality	Quality threshold for trimming.
Reads length	Too short reads are discarded by the filter.
Trim both ends	Trim the both ends of a read or not. Usually, you need to set True for Sanger sequencing and False for NGS
3'	

adapters	A FASTA file with one or multiple sequences of adapter that were ligated to the 3' end. The adapter itself and anything that follows is trimmed. If the adapter sequence ends with the '$ character, the adapter is anchored to the end of the read and only found if it is a suffix of the read.
5' adapters	A FASTA file with one or multiple sequences of adapters that were ligated to the 5' end. If the adapter sequence starts with the character ^, the adapter is 'anchored'. An anchored adapter must appear in its entirety at the 5' end of the read (it is a prefix of the read). A non-anchored adapter may appear partially at the 5' end, or it may occur within the read. If it is found within a read, the sequence preceding the adapter is also trimmed. In all cases, the adapter itself is trimmed.
5' and 3' adapters	A FASTA file with one or multiple sequences of adapter that were ligated to the 5' end or 3' end.

3. Mapping: On this page you must input reference and optionally modify advanced parameters.

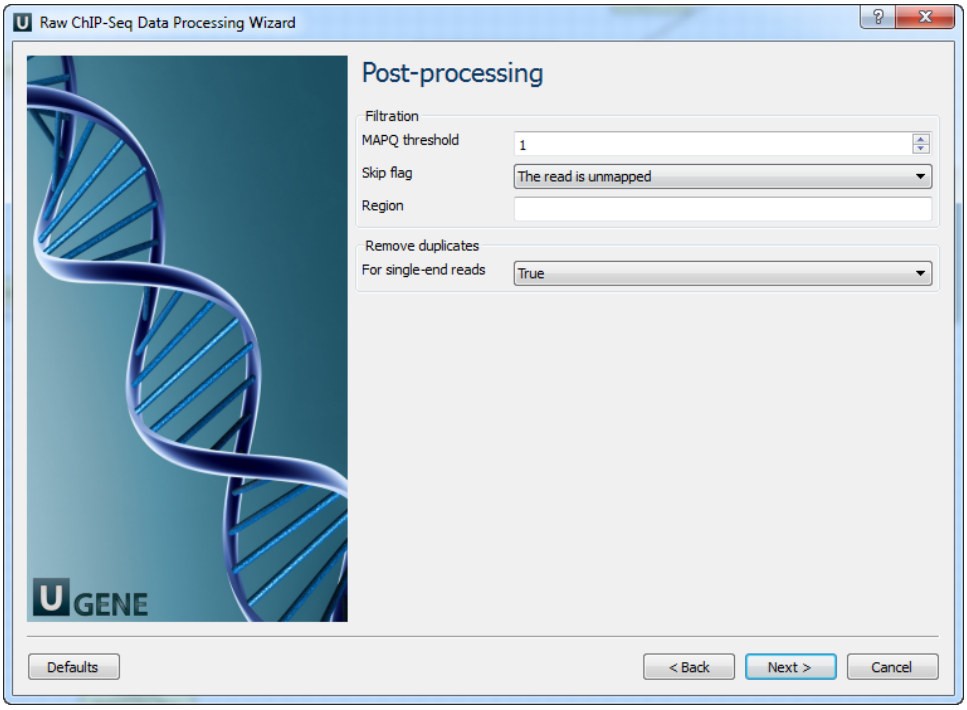


The following parameters are available:

Reference genome	Path to indexed reference genome.
Number of threads	Number of threads (-t).
Min seed length	Path to indexed reference genome (-k).
Band width	Band width for banded alignment (-w).
Dropoff	Off-diagonal X-dropoff (-d).
Internal seed length	Look for internal seeds inside a seed longer than {-k} (-r).
Skip seed threshold	Skip seeds with more than INT occurrences (-c).
Drop chain threshold	Drop chains shorter than FLOAT fraction of the longest overlapping chain (-D).
Rounds of mate rescues	Perform at most INT rounds of mate rescues for each read (-m).
Skip mate rescue	Skip mate rescue (-S).
Skip pairing	Skip pairing; mate rescue performed unless -S also in use (-P).
Mismatch penalty	Score for a sequence match (-A).
Mismatch penalty	Penalty for a mismatch (-B).

Gap open penalty	Gap open penalty (-O).
Gap extention penalty	Gap extension penalty; a gap of size k cost {-O} (-E).
Penalty for clipping	Penalty for clipping (-L).
Penalty unpaired	Penalty for an unpaired read pair (-U).
Score threshold	Minimum score to output (-T).

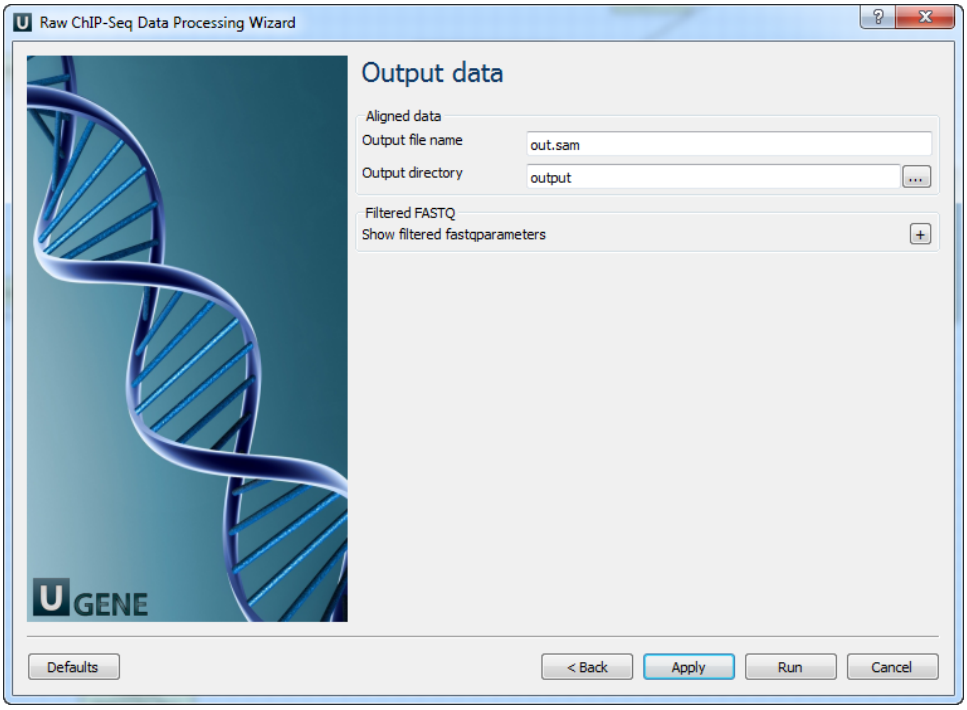
4. Post-processing: On this page you can modify post-processing parameters.



The following parameters are available:

MAPQ threshold	Minimum MAPQ quality score.
Skip flag	Skip alignment with the selected items. Select the items in the combobox to configure bit flag. Do not select the items to avoid filtration by this parameter.
Region	Regions to filter. For BAM output only. chr2 to output the whole chr2. chr2:1000 to output regions of chr 2 starting from 1000. chr2:1000-2000 to ouput regions of chr2 between 1000 and 2000 including the end point. To input multiple regions use the space seprator (e.g. chr1 chr2 chr3:1000-2000).
For single-end reads	Remove duplicates for single-end reads.

5. Output data: On this page you must input output parameters.



Raw DNA-Seq Data Processing

Use this workflow sample to process raw DNA-seq next-generation sequencing (NGS) data from the Illumina platform. The processing includes:

- *Filtration:*
  - Filtering of the NGS short reads by the CASAVA 1.8 header;
  - Trimming of the short reads by quality;
- *Mapping:*
  - Mapping of the short reads to the specified reference sequence (the BWA-MEM tool is used in the sample);
- *Post-filtration:*
  - Filtering of the aligned short reads by SAMtools to remove reads with low mapping quality, unpaired/unaligned reads;
  - Removing of duplicated short reads.

The result filtered short reads assembly is provided in the SAM format. Intermediate data files are also available in the output.

How to Use This Sample

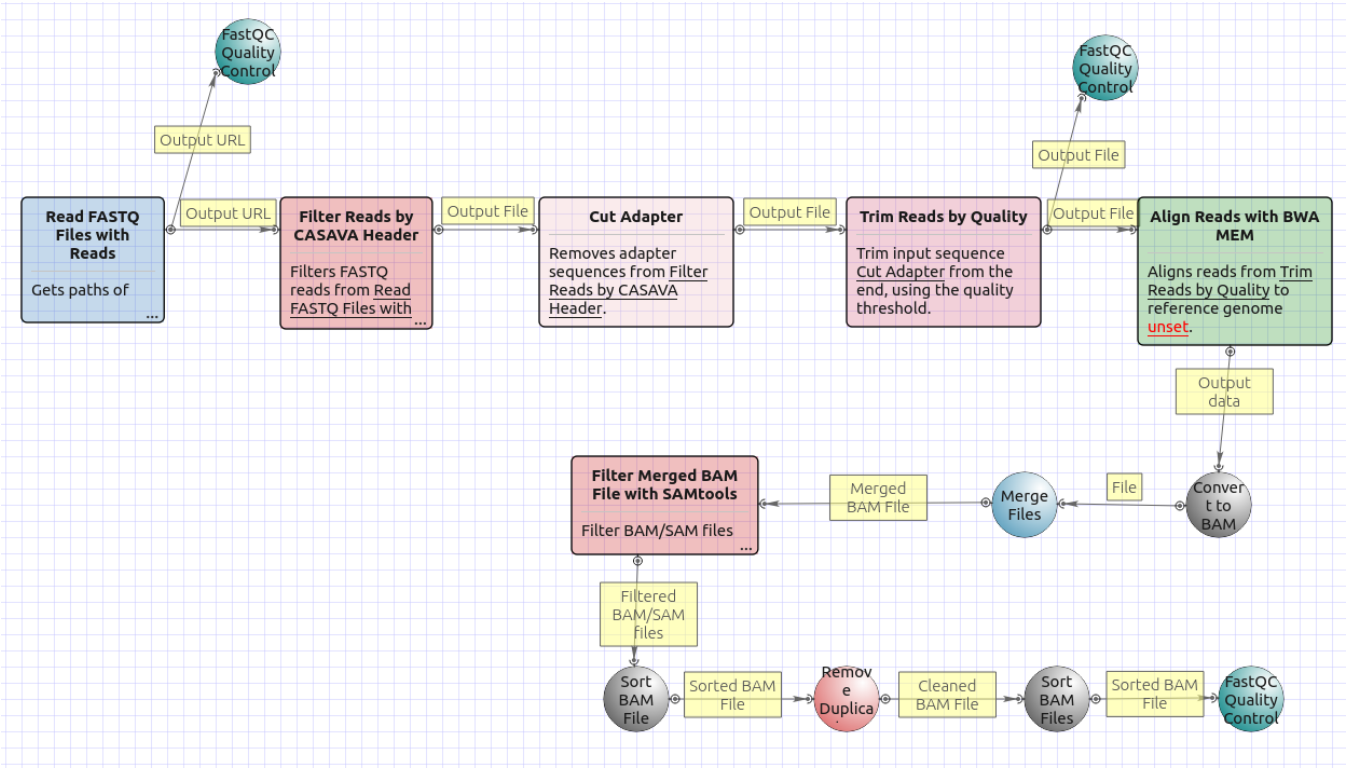
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

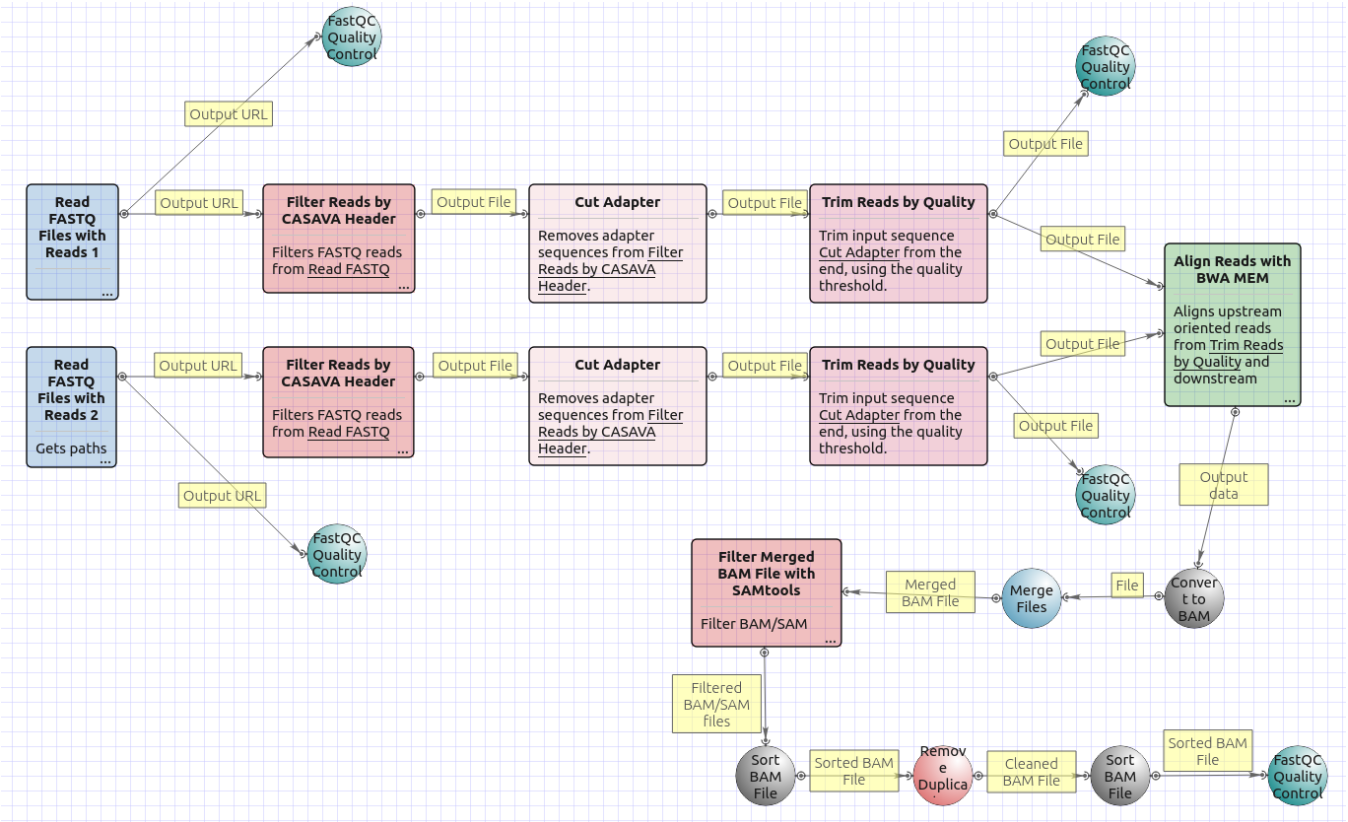
The workflow sample "Raw DNA-Seq processing" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

There are two versions of the workflow available. The workflow for single-end reads looks as follows:



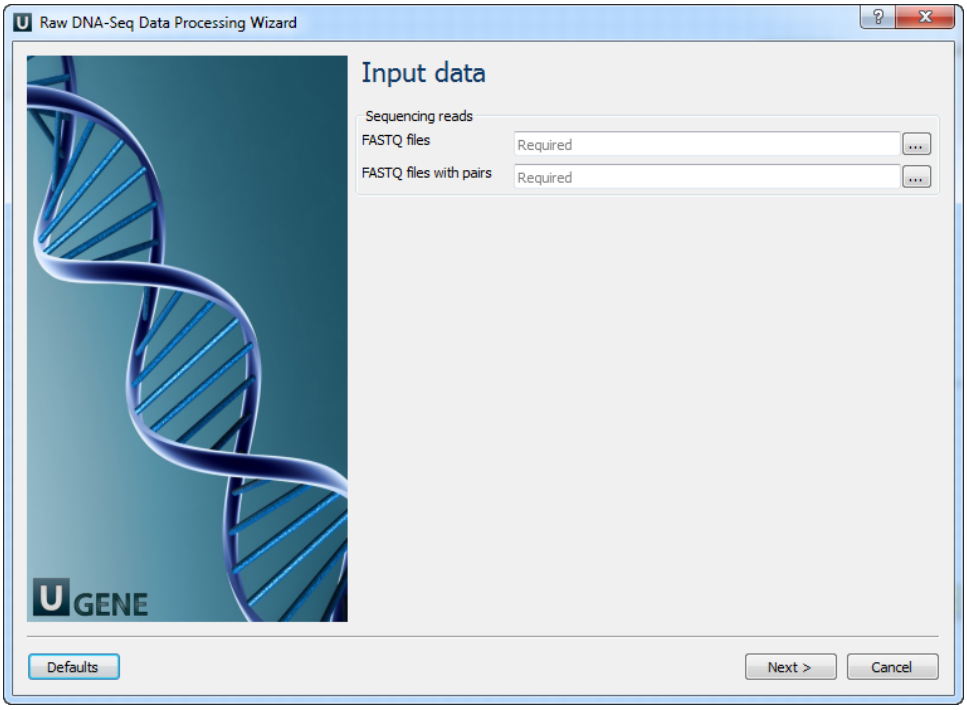
The workflow for paired-end short appearance is the following:



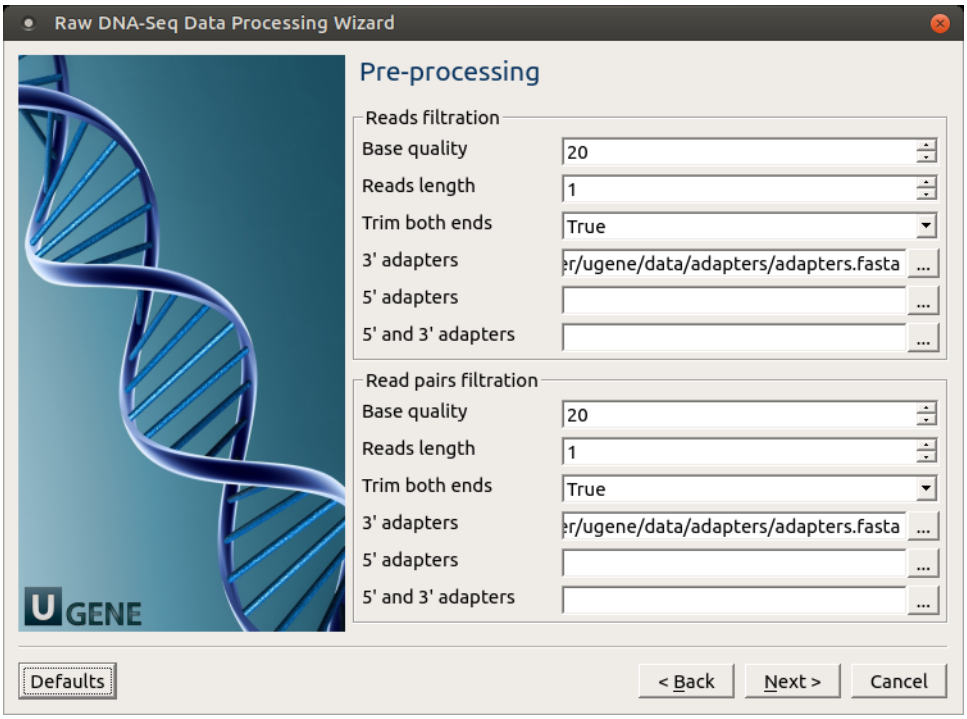
Workflow Wizard

The workflows have the similar wizards. The wizard for paired-end reads has 5 pages.

1. Input data: On this page you must input FASTQ file(s).



2. Pre-processing: On this page you can modify filtration parameters.

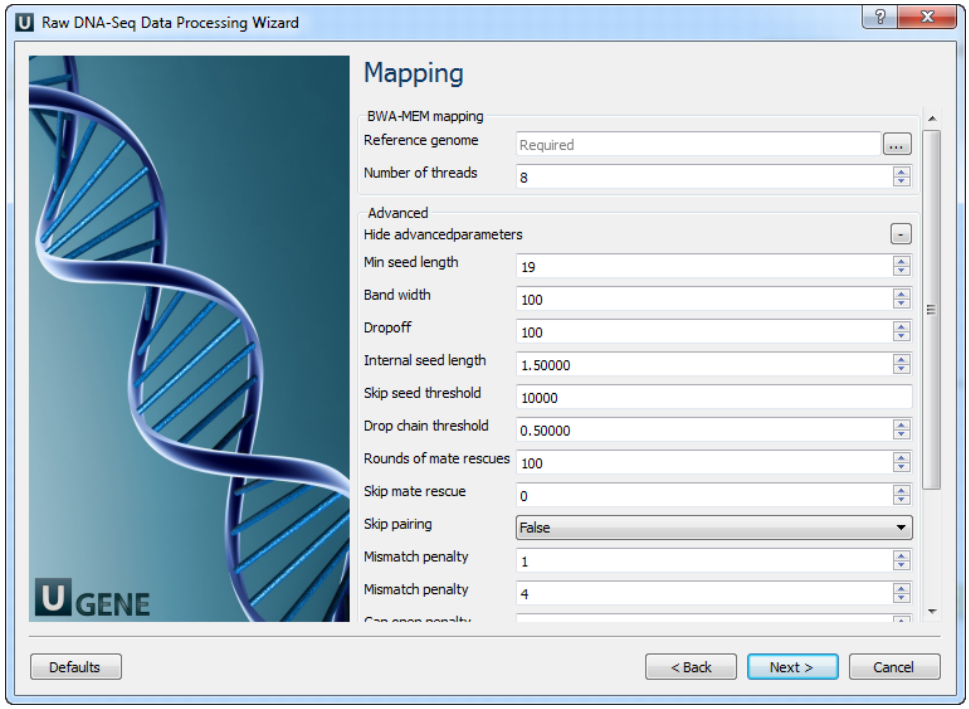


The following parameters are available for reads and reads pairs filtration:

Base quality	Quality threshold for trimming.
Reads length	Too short reads are discarded by the filter.
Trim both ends	Trim the both ends of a read or not. Usually, you need to set True for Sanger sequencing and False for NGS

3' adapters	A FASTA file with one or multiple sequences of adapter that were ligated to the 3' end. The adapter itself and anything that follows is trimmed. If the adapter sequence ends with the '$ character, the adapter is anchored to the end of the read and only found if it is a suffix of the read.
5' adapters	A FASTA file with one or multiple sequences of adapters that were ligated to the 5' end. If the adapter sequence starts with the character ^, the adapter is 'anchored'. An anchored adapter must appear in its entirety at the 5' end of the read (it is a prefix of the read). A non-anchored adapter may appear partially at the 5' end, or it may occur within the read. If it is found within a read, the sequence preceding the adapter is also trimmed. In all cases, the adapter itself is trimmed.
5' and 3' adapters	A FASTA file with one or multiple sequences of adapter that were ligated to the 5' end or 3' end.

3. Mapping: On this page you must input reference and optionally modify advanced parameters.

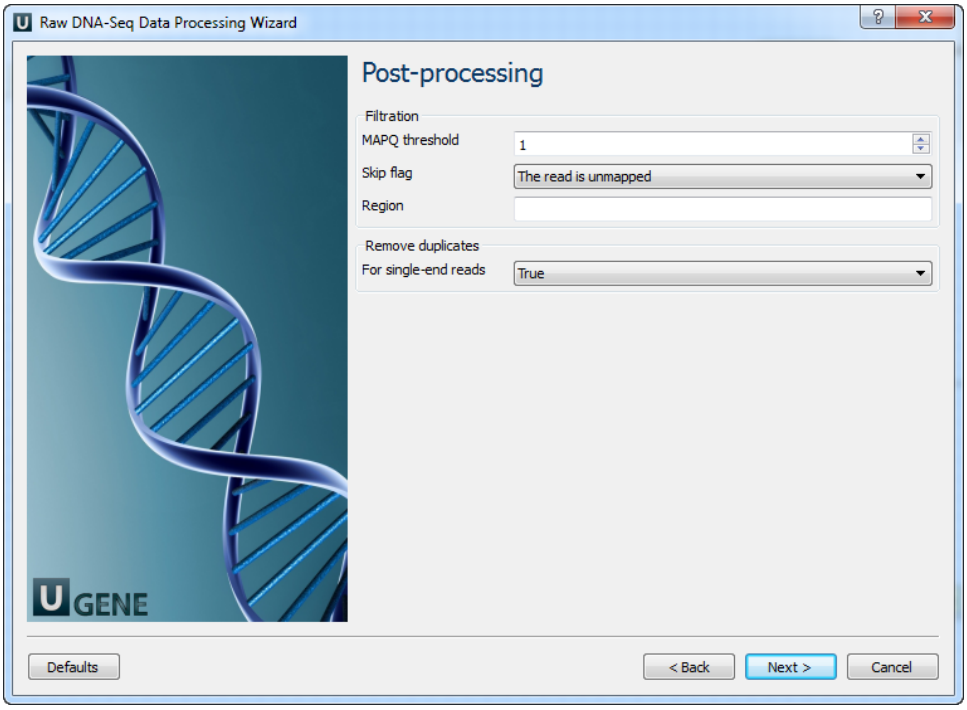


The following parameters are available:

Reference genome	Path to indexed reference genome.
Number of threads	Number of threads (-t).
Min seed length	Path to indexed reference genome (-k).
Band width	Band width for banded alignment (-w).
Dropoff	Off-diagonal X-dropoff (-d).
Internal seed length	Look for internal seeds inside a seed longer than {-k} (-r).
Skip seed threshold	Skip seeds with more than INT occurrences (-c).
Drop chain threshold	Drop chains shorter than FLOAT fraction of the longest overlapping chain (-D).
Rounds of mate rescues	Perform at most INT rounds of mate rescues for each read (-m).
Skip mate rescue	Skip mate rescue (-S).
Skip pairing	Skip pairing; mate rescue performed unless -S also in use (-P).
Mismatch penalty	Score for a sequence match (-A).
Mismatch penalty	Penalty for a mismatch (-B).

Gap open penalty	Gap open penalty (-O).
Gap extention penalty	Gap extension penalty; a gap of size k cost {-O} (-E).
Penalty for clipping	Penalty for clipping (-L).
Penalty unpaired	Penalty for an unpaired read pair (-U).
Score threshold	Minimum score to output (-T).

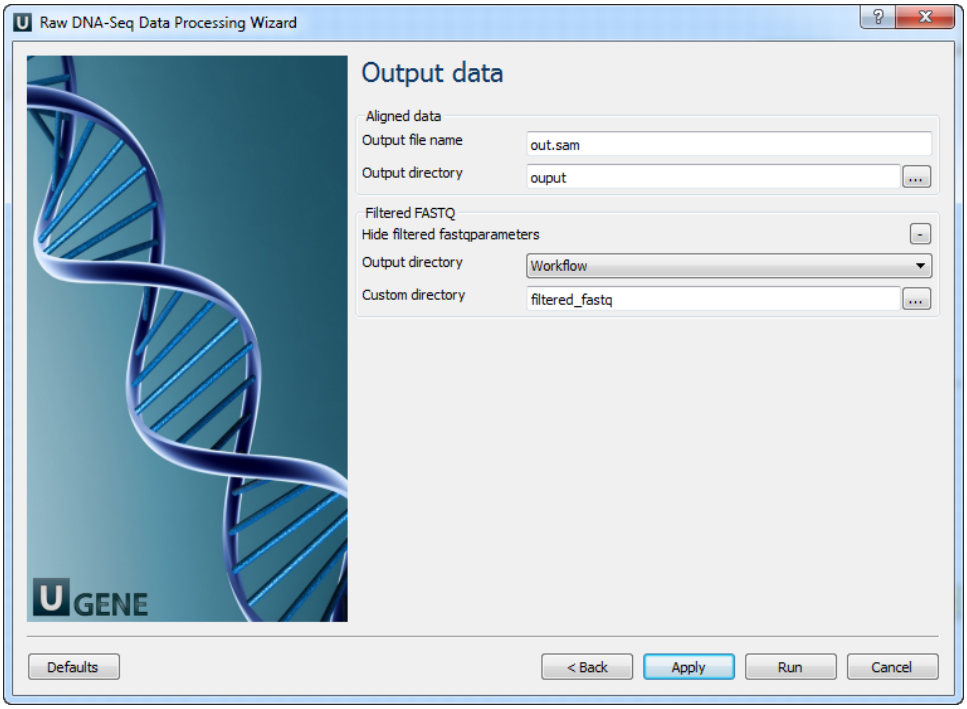
4. Post-processing: On this page you can modify post-processing parameters.



The following parameters are available:

MAPQ threshold	Minimum MAPQ quality score.
Skip flag	Skip alignment with the selected items. Select the items in the combobox to configure bit flag. Do not select the items to avoid filtration by this parameter.
Region	Regions to filter. For BAM output only. chr2 to output the whole chr2. chr2:1000 to output regions of chr 2 starting from 1000. chr2:1000-2000 to ouput regions of chr2 between 1000 and 2000 including the end point. To input multiple regions use the space seprator (e.g. chr1 chr2 chr3:1000-2000).
For single-end reads	Remove duplicates for single-end reads.

5. Output data: On this page you must input output parameters.



Raw RNA-Seq Data Processing

Use this workflow sample to process raw RNA-seq next-generation sequencing (NGS) data from the Illumina platform. The processing includes:

- *Filtration:*
  - Filtering of the NGS short reads by the CASAVA 1.8 header;
  - Trimming of the short reads by quality;
- *[Optionally] Mapping:*
  - Mapping of the short reads to the specified reference sequence (the TopHat tool is used in the sample);

The result output of the workflow contains the filtered and merged FASTQ files. In case the TopHat mapping has been done, the result also contains the TopHat output files: the accepted hits BAM file and tracks of junctions, insertions and deletions in BED format. Other intermediate data files are also output by the workflow.

How to Use This Sample

If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

What's Next?

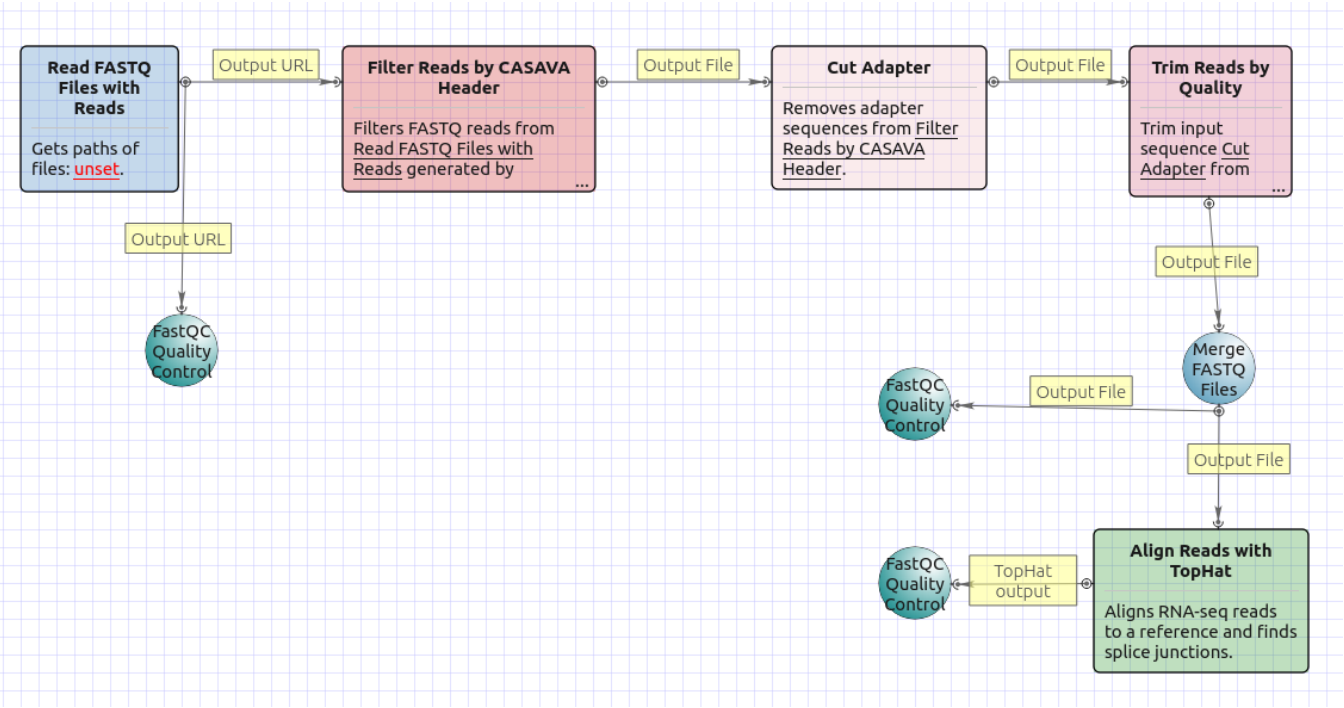
The Tuxedo workflow can be used to analyze the filtered RNA-seq data. In this case the mapping step of this workflow can be skipped, as it also present in the Tuxedo pipeline.

Workflow Sample Location

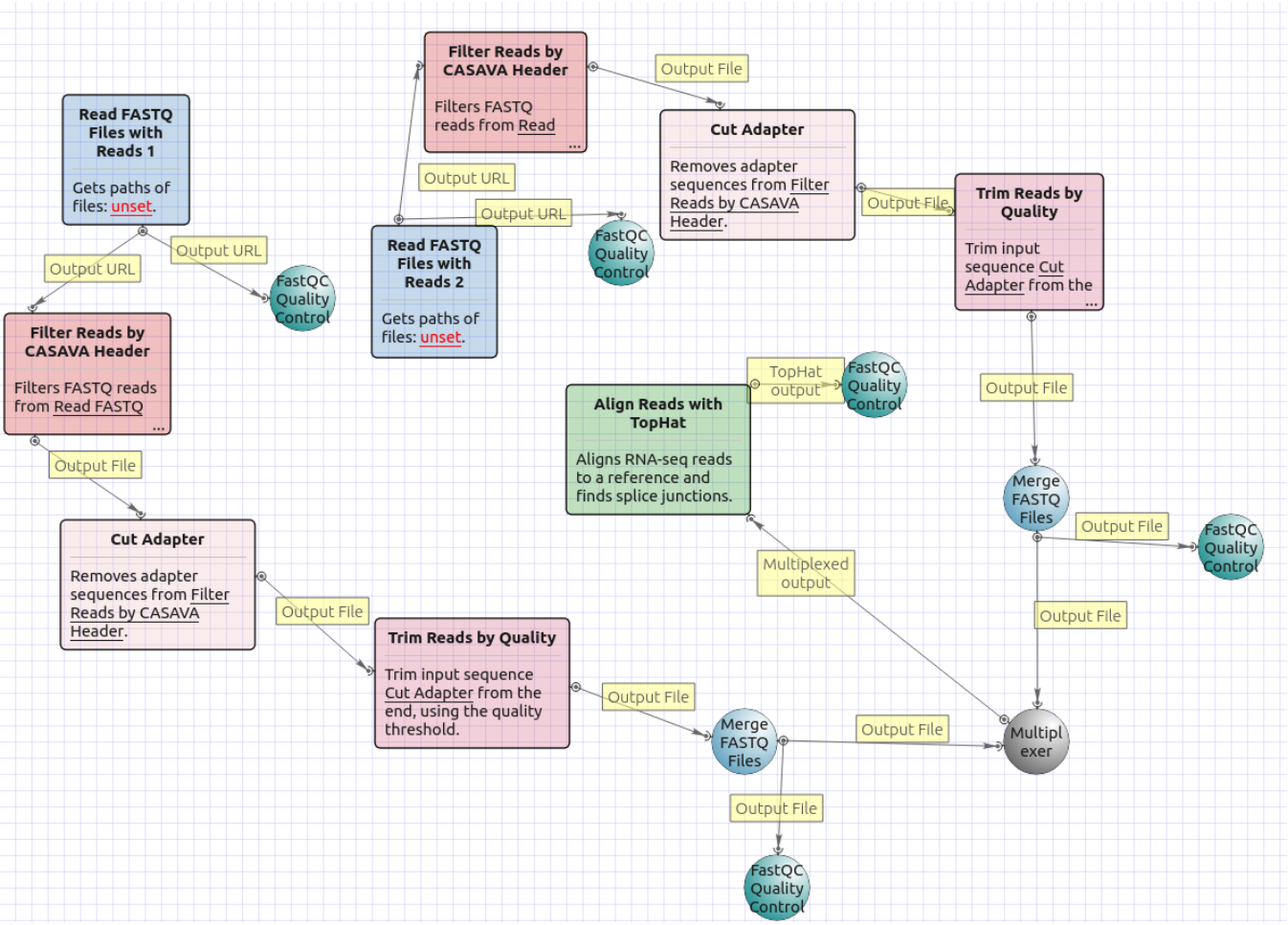
The workflow sample "Raw DNA-Seq processing" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

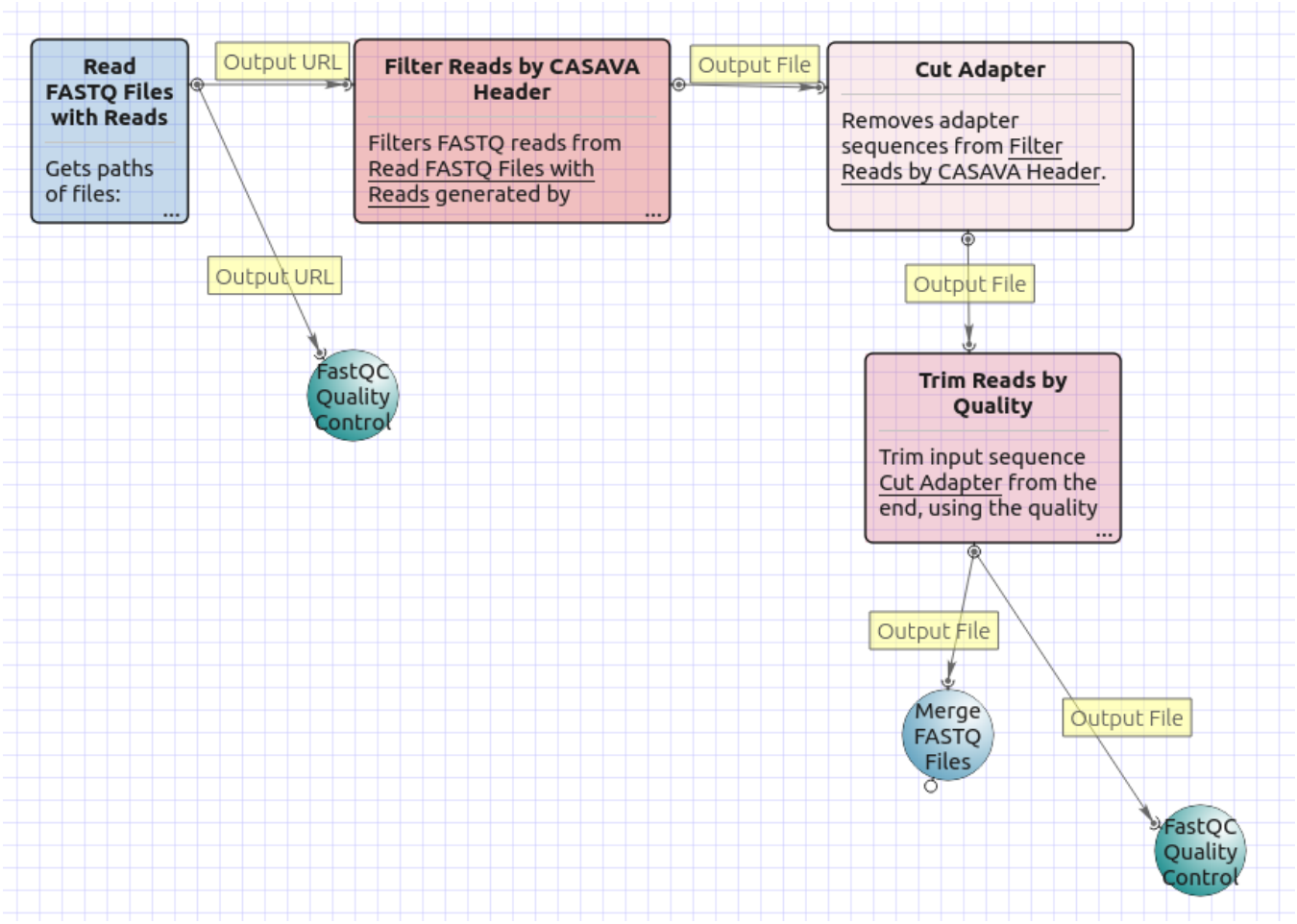
There are four versions of the workflow available. The workflow with mapping for single-end reads looks as follows:



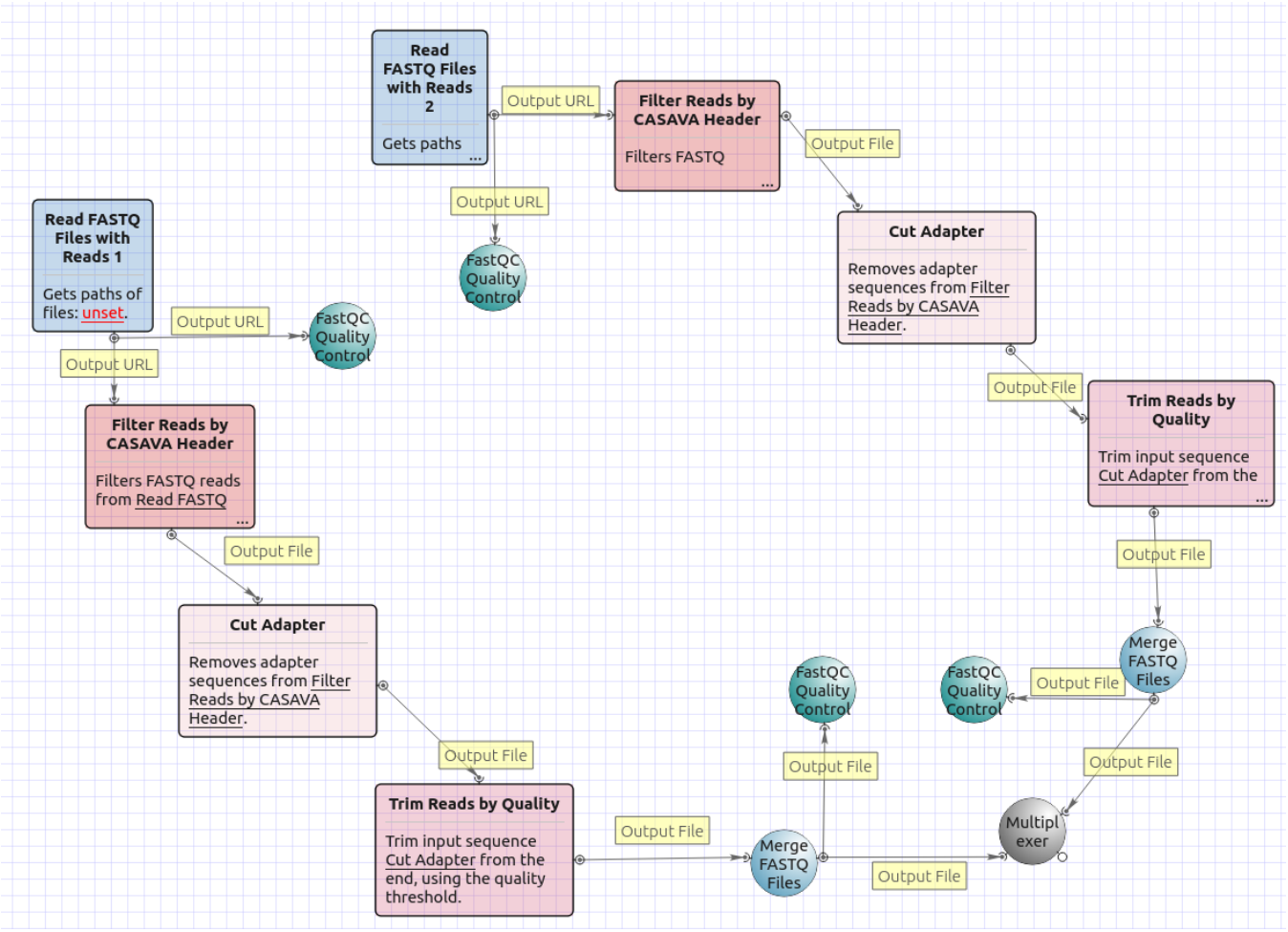
The workflow with mapping for paired-end short appearance is the following:



The workflow without mapping for single-end short appearance is the following:



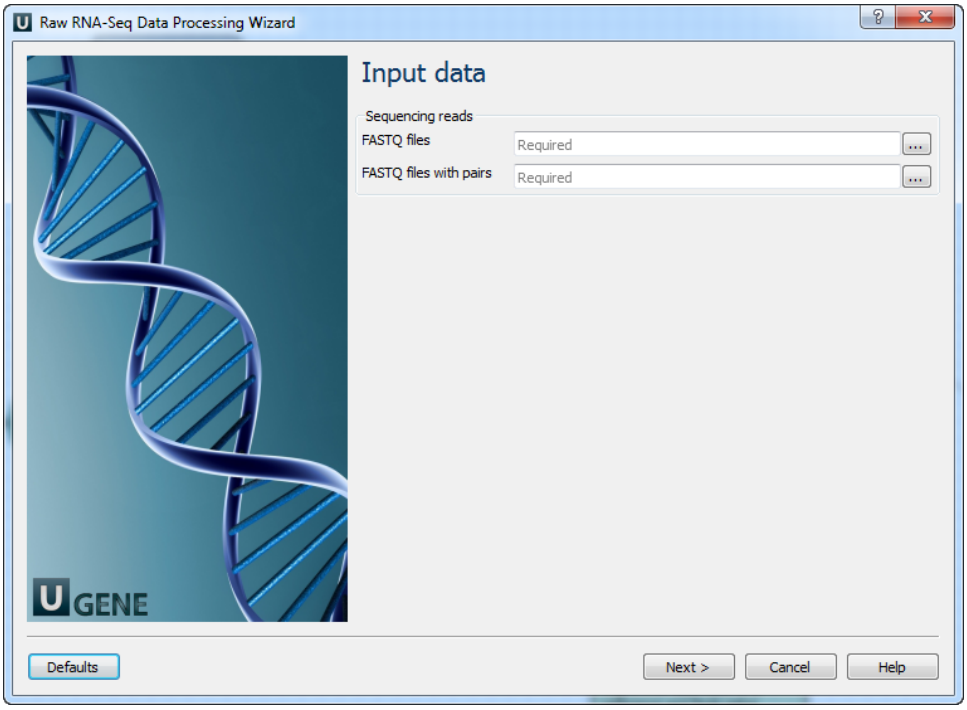
The workflow without mapping for paired-end short appearance is the following:



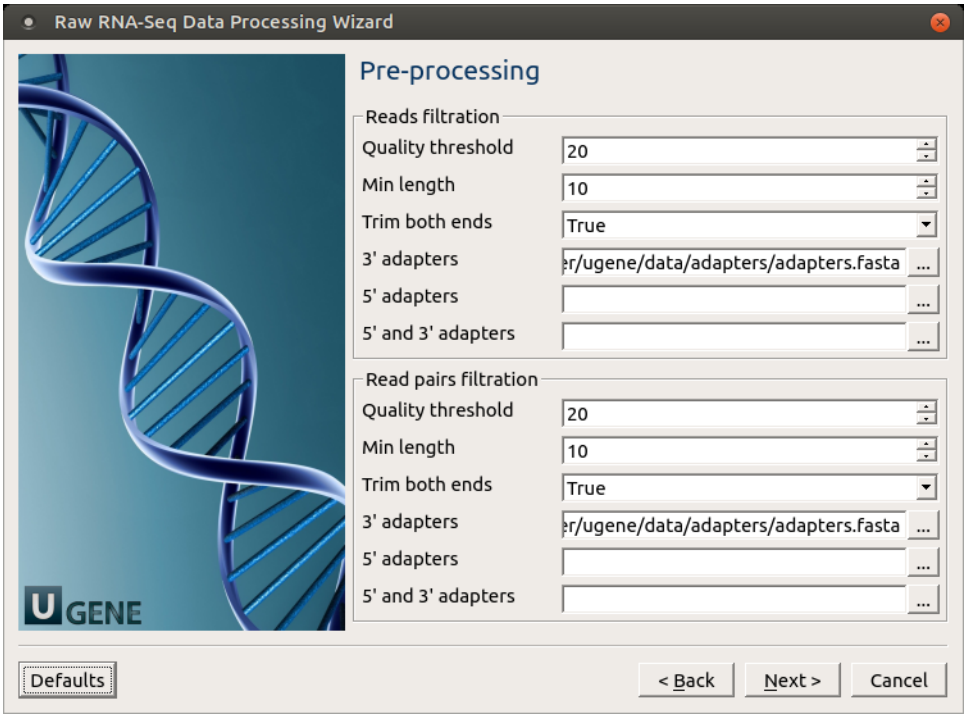
Workflow Wizard

The workflows have the similar wizards. The wizard for paired-end reads with mapping has 4 pages.

1. Input data: On this page you must input FASTQ file(s).



2. Pre-processing: On this page you can modify filtration parameters.

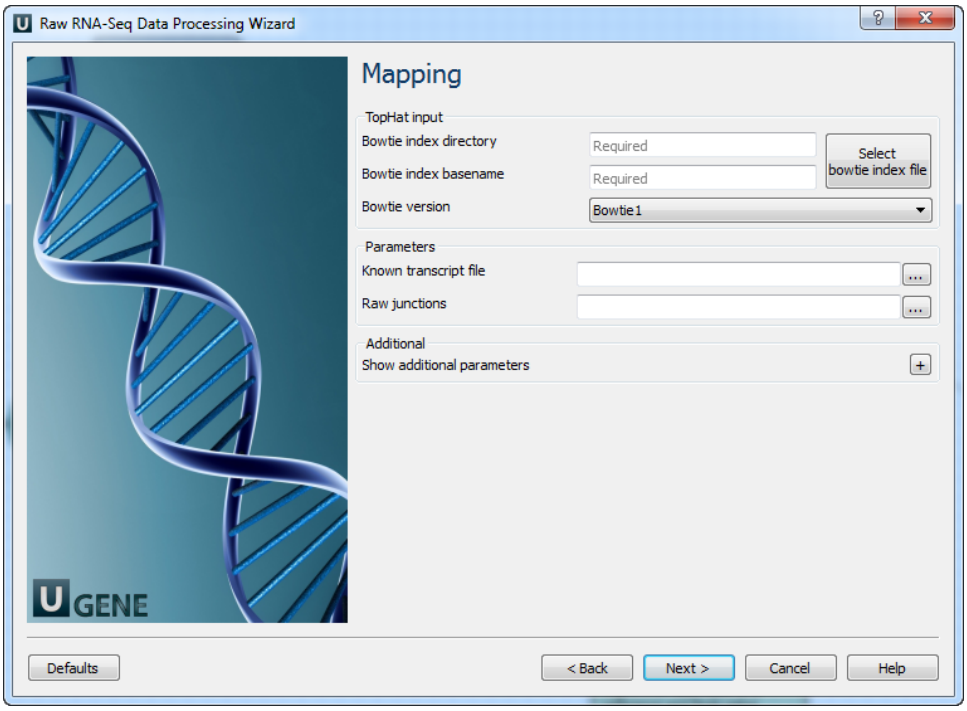


The following parameters are available for reads and reads pairs filtration:

Base quality	Quality threshold for trimming.
Reads length	Too short reads are discarded by the filter.
Trim both ends	Trim the both ends of a read or not. Usually, you need to set True for Sanger sequencing and False for NGS
3'	A FASTA file with one or multiple sequences of adapter that were ligated to the 3' end. The adapter itself and anything that follows is

adapters	trimmed. If the adapter sequence ends with the '$ character, the adapter is anchored to the end of the read and only found if it is a suffix of the read.
5' adapters	A FASTA file with one or multiple sequences of adapters that were ligated to the 5' end. If the adapter sequence starts with the character ^, the adapter is 'anchored'. An anchored adapter must appear in its entirety at the 5' end of the read (it is a prefix of the read). A non-anchored adapter may appear partially at the 5' end, or it may occur within the read. If it is found within a read, the sequence preceding the adapter is also trimmed. In all cases, the adapter itself is trimmed.
5' and 3' adapters	A FASTA file with one or multiple sequences of adapter that were ligated to the 5' end or 3' end.

3. Mapping: On this page you must input reference and optionally modify advanced parameters.



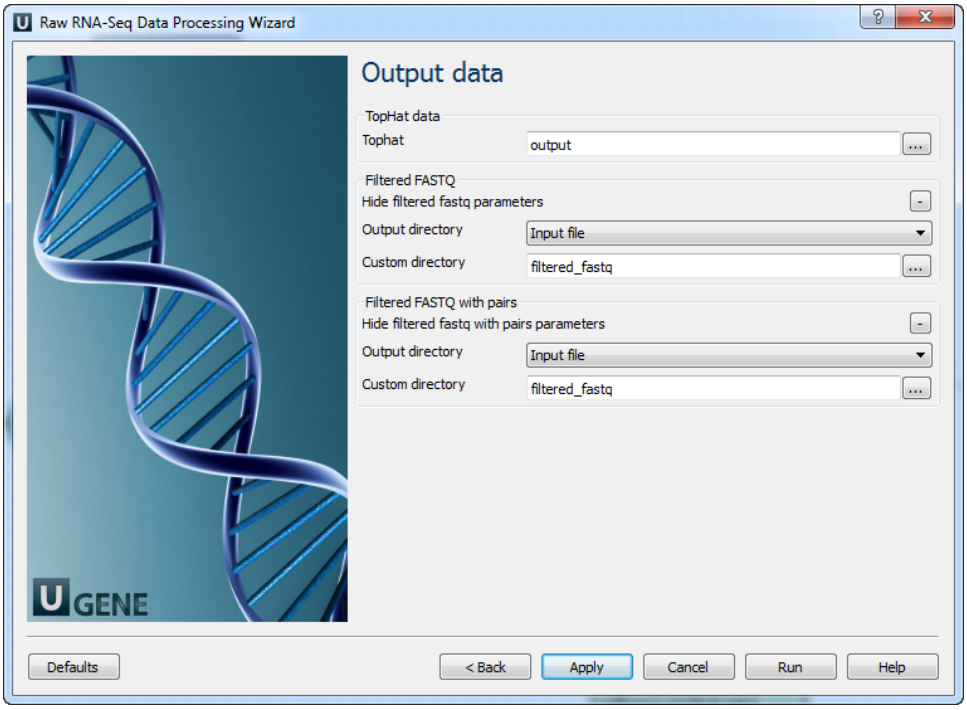
The following parameters are available:

Bowtie index directory	The directory with the Bowtie index for the reference sequence.
Bowtie index basename	The basename of the Bowtie index for the reference sequence.
Bowtie version	Specifies which Bowtie version should be used.
Known transcript file	A set of gene model annotations and/or known transcripts.
Raw junctions	The list of raw junctions.
	Expected (mean) inner distance between mate pairs.

Mate inner distance	
Mate standard deviation	Standard deviation for the distribution on inner distances between mate pairs.
Library type	Specifies RNA-seq protocol.
No novel junctions	Only look for reads across junctions indicated in the supplied GFF or junctions file. This parameter is ignored if Raw junctions or Known transcript file is not set.
Max multihits	Instructs TopHat to allow up to this many alignments to the reference for a given read, and suppresses all alignments for reads with more than this many alignments.
Segment length	Each read is cut up into segments, each at least this long. These segments are mapped independently.
Fusion search	Turn on fusion mapping.
Transcritome max hits	Only align the reads to the transcriptome and report only those mappings as genomic mappings.
Prefilter multihits	When mapping reads on the transcriptome, some repetitive or low complexity reads that would be discarded in the context of the genome may appear to align to the transcript sequences and thus may end up reported as mapped to those genes only. This option directs TopHat to first align the reads to the whole genome in order to determine and exclude such multi-mapped reads (according to the value of the Max multihits option).
Min anchor length	The anchor length. TopHat will report junctions spanned by reads with at least this many bases on each side of the junction. Note that individual spliced alignments may span a junction with fewer than this many bases on one side. However, every junction involved in spliced alignments is supported by at least one read with this many bases on each side.
Splice mismatches	The maximum number of mismatches that may appear in the anchor region of a spliced alignment.
Read mismatches	Final read alignments having more than these many mismatches are discarded.
Segment mismatches	Read segments are mapped independently, allowing up to this many mismatches in each segment alignment.
Solexa 1.3 quals	As of the Illumina GA pipeline version 1.3, quality scores are encoded in Phred-scaled base-64. Use this option for FASTQ files from pipeline 1.3 or later.
Bowtie version	specifies which Bowtie version should be used.
Bowtie -n mode	TopHat uses -v in Bowtie for initial read mapping (the default), but with this option, -n is used instead. Read segments are always mapped using -v option.
Bowtie tool path	The path to the Bowtie external tool.

SAMtools tool path	The path to the SAMtools tool. Note that the tool is available in the UGENE External Tool Package.
TopHat tool path	The path to the TopHat external tool in UGENE.
Temporary directory	The directory for temporary files.

4. Output data: On this page you must input output parameters.



Get Unmappet Reads

Use this workflow sample to extract unmapped reads from an input SAM/BAM file.

How to Use This Sample

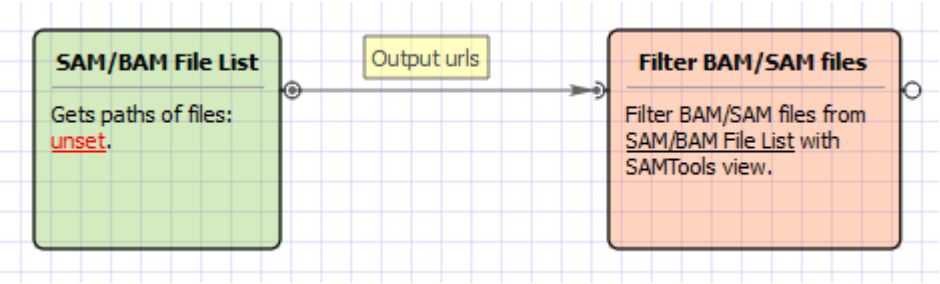
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Get Unmapped Reads" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

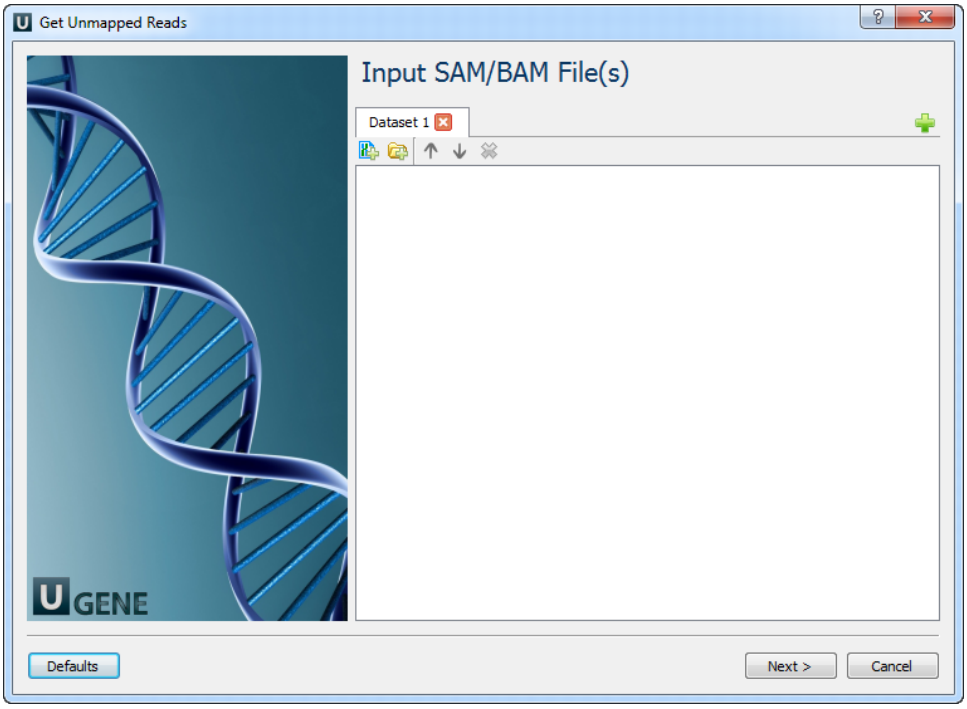
The workflow looks as follows:



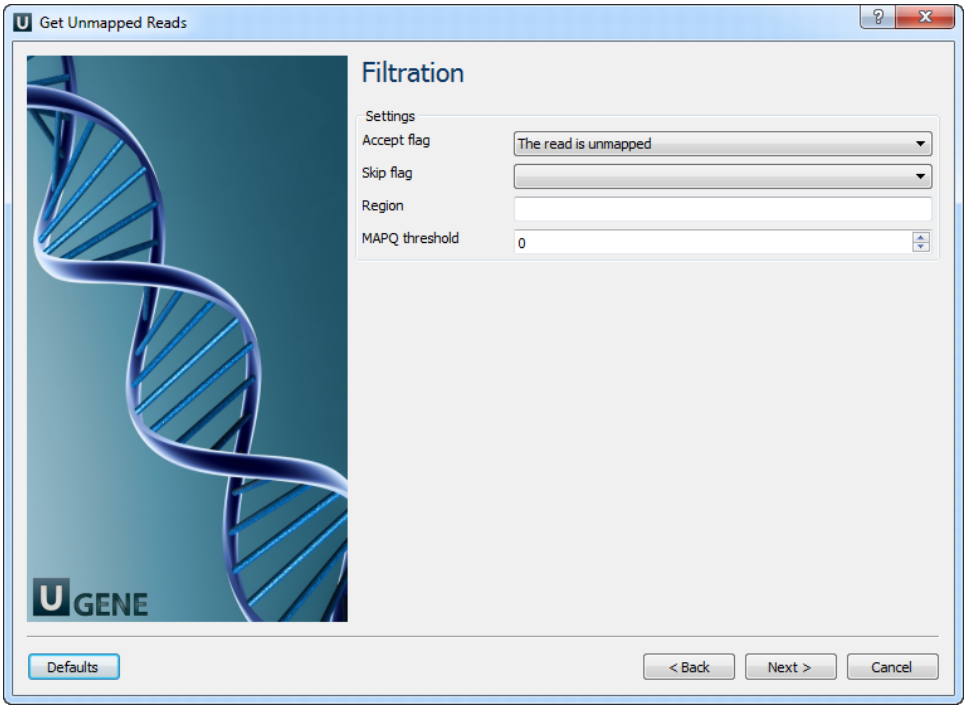
Workflow Wizard

The wizard has 3 page.

1. Input SAM/BAM File(s): On this page you need input SAM/BAM file(s).



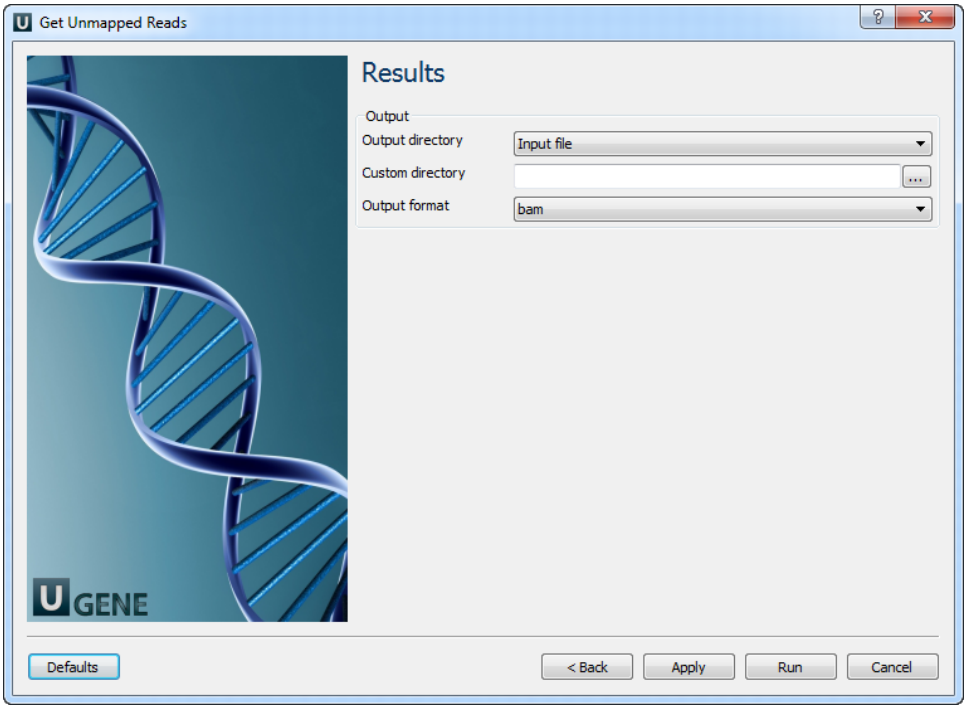
2. Filtration: On this page you can change the filtration parameters.



The following parameters are available:

Accept flag	Only output alignments with the selected items. Select the items in the combobox to configure bit flag. Do not select the items to avoid filtration by this parameter.
Skip flag	Skip alignment with the selected items. Select the items in the combobox to configure bit flag. Do not select the items to avoid filtration by this parameter.
Region	Regions to filter. For BAM output only. chr2 to output the whole chr2. chr2:1000 to output regions of chr 2 starting from 1000. chr2:1000-2000 to ouput regions of chr2 between 1000 and 2000 including the end point. To input multiple regions use the space seprator (e.g. chr1 chr2 chr3:1000-2000).
MAPQ threshold	Minimum MAPQ quality score.

3. Results: On this page you need input output parameters.



Sanger Sequencing

- Trim and Align Sanger Reads

Trim and Align Sanger Reads

The workflow does the following things:

1) Reads a set of Sanger sequencing reads from ABI files.
2) Trims ends of the reads by the quality value.
3) Filter the short trimmed reads.
4) Aligns the filtered trimmed reads to a reference sequence.

You can change the workflow parameters:

1) Quality threshold for the trimming.
2) Minimum read length. If length of a trimmed read is less than the minimum value than the read is filtered.

The output data are:

1) Multiple sequence alignment file. The first sequence of the alignment is the reference and other ones are the reads.
2) Annotated reference sequence file. The annotations are the aligned reads.

How to Use This Sample

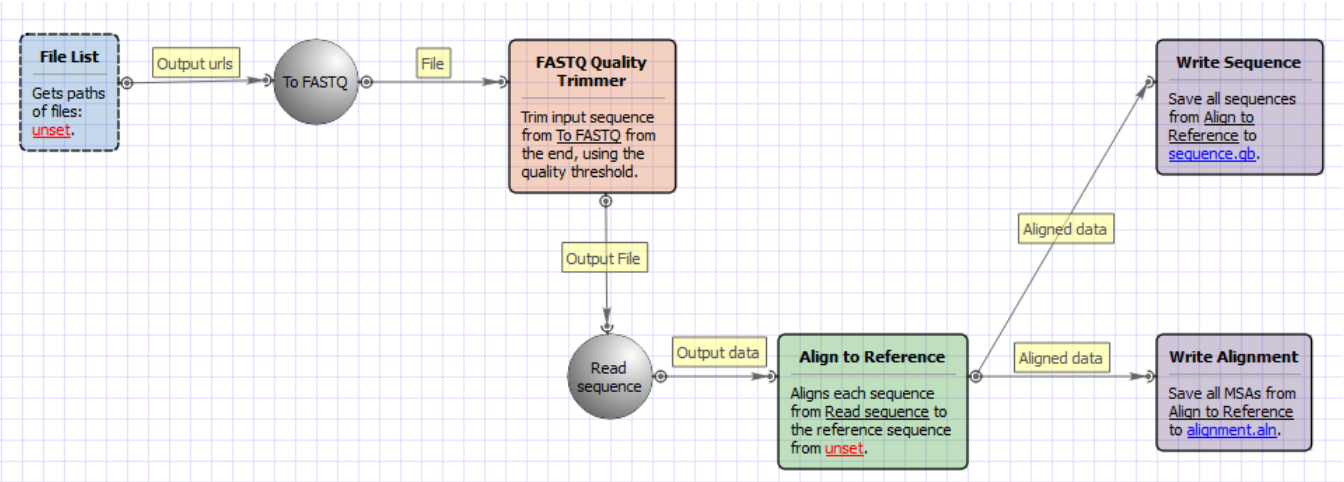
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Trim and Align Sanger Reads" can be found in the "Sanger Sequencing" section of the Workflow Designer samples.

Workflow Image

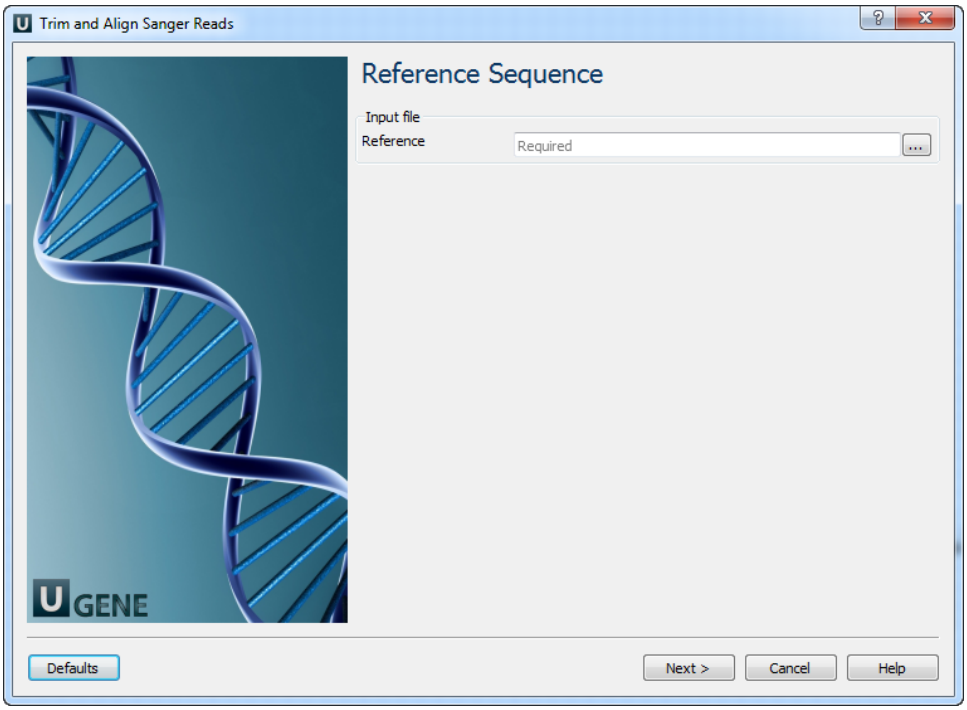
The opened workflow looks as follows:



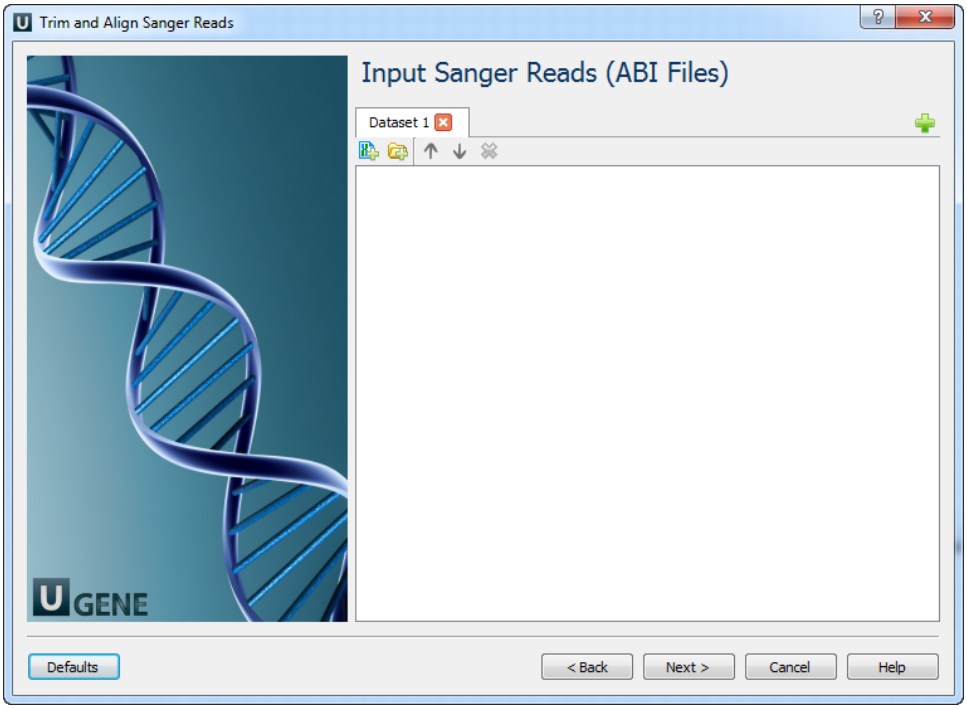
Workflow Wizard

The wizard has 4 pages.

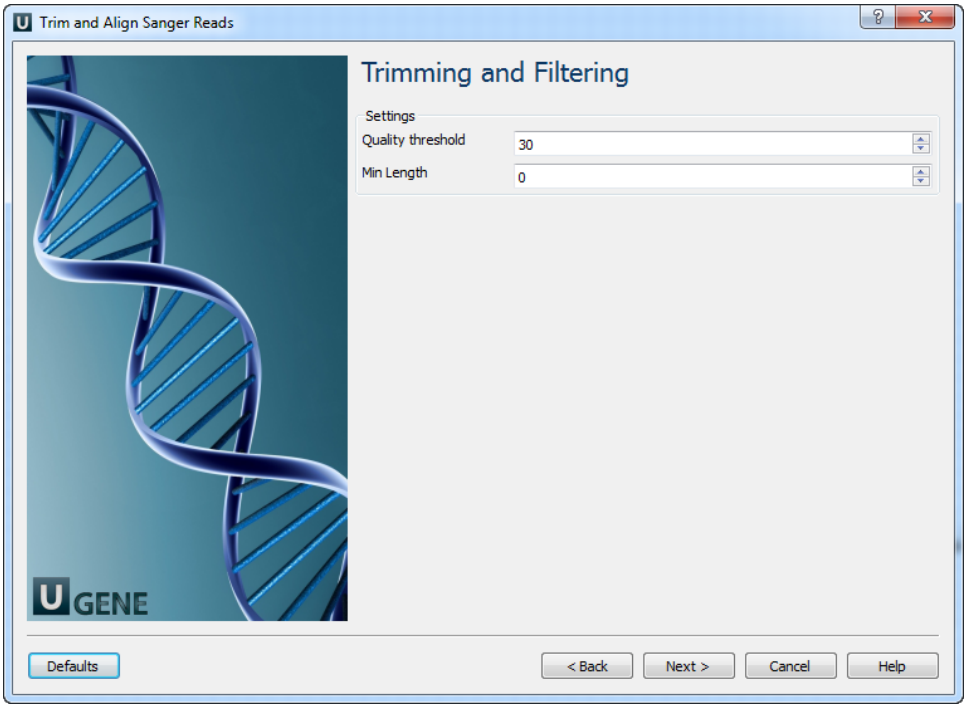
1. Reference Sequence: On this page you must input reference sequence.



2. Input Sanger Reads (ABI Files): On this page you must input ABI file(s).



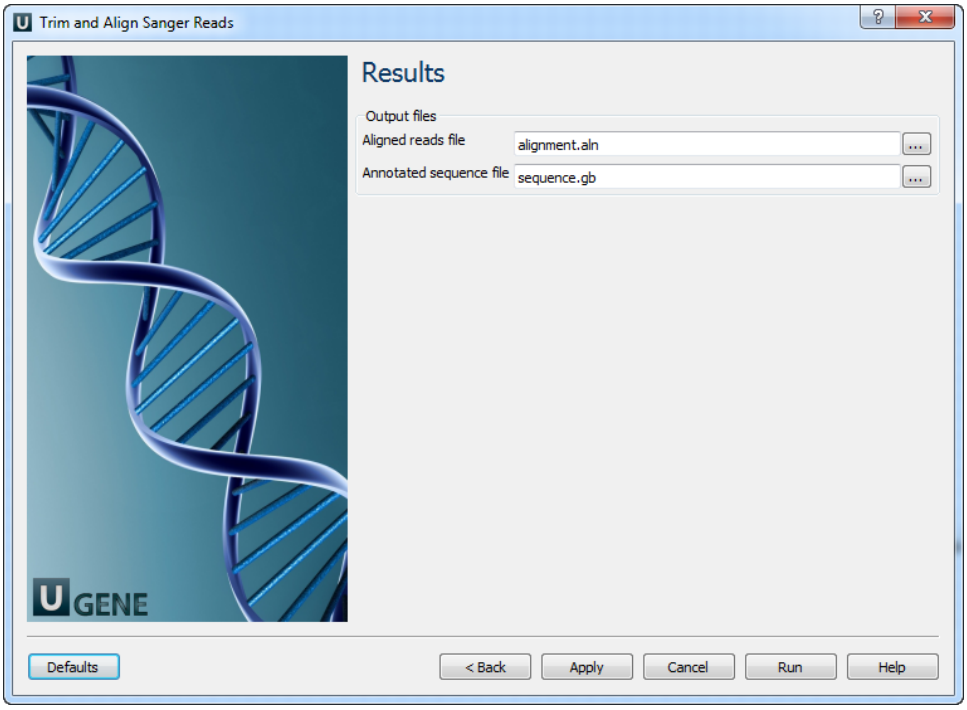
3. Trimming and Filtering: On this page you can modify trimming and filtering settings.



The following parameters are available:

Quality threshold	Quality threshold for trimming.
Min Length	Too short reads are discarded by the filter.

4. Results: On this page you can modify output files settings.



Scenarios

- Filter Sequence That Match a Pattern
- Search for Inverted Repeats
- Find Patterns
- Gene-by-gene Approach for Characterization of Genomes
- Group Primer Pairs
- Intersect Annotations
- Filter out Short Sequences
- Merge Sequences and Annotations
- In Silico PCR Sample
- Remote BLASTing
- Get Amino Translations of a Sequence

Filter Sequence That Match a Pattern

Using thist workflow you can select (or reject) only those sequence that match any pattern you input.

How to Use This Sample

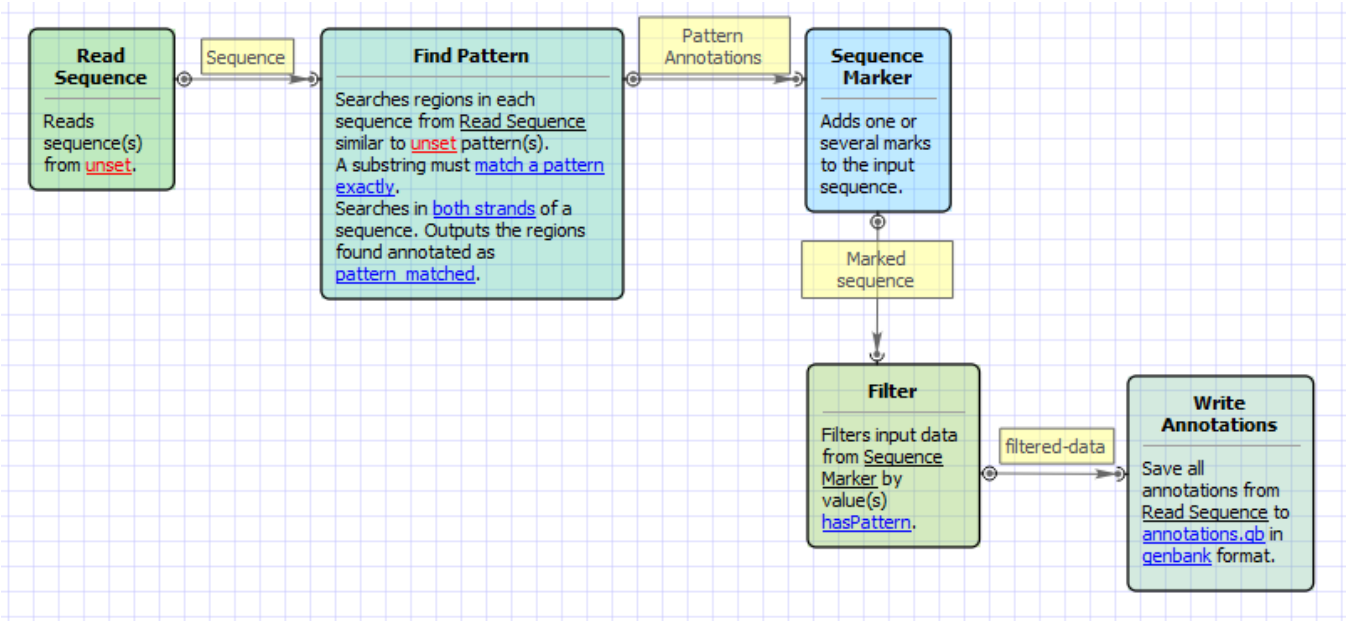
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Filter Sequence That Match a Pattern" can be found in the "Scenarios" section of the Workflow Designer samples.

Workflow Image

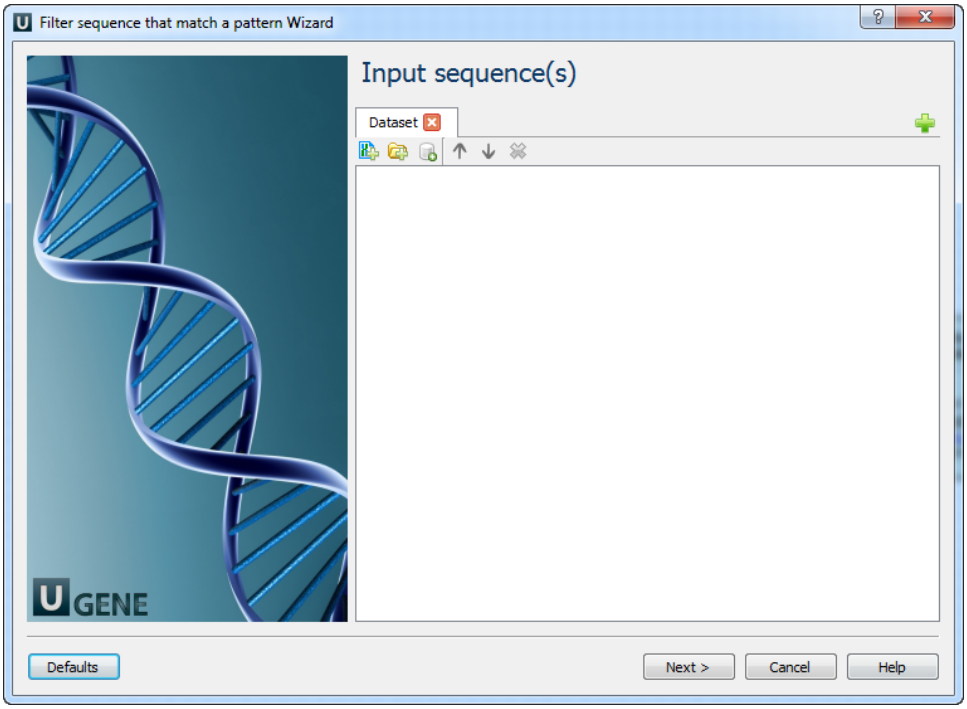
The workflow looks as follows:



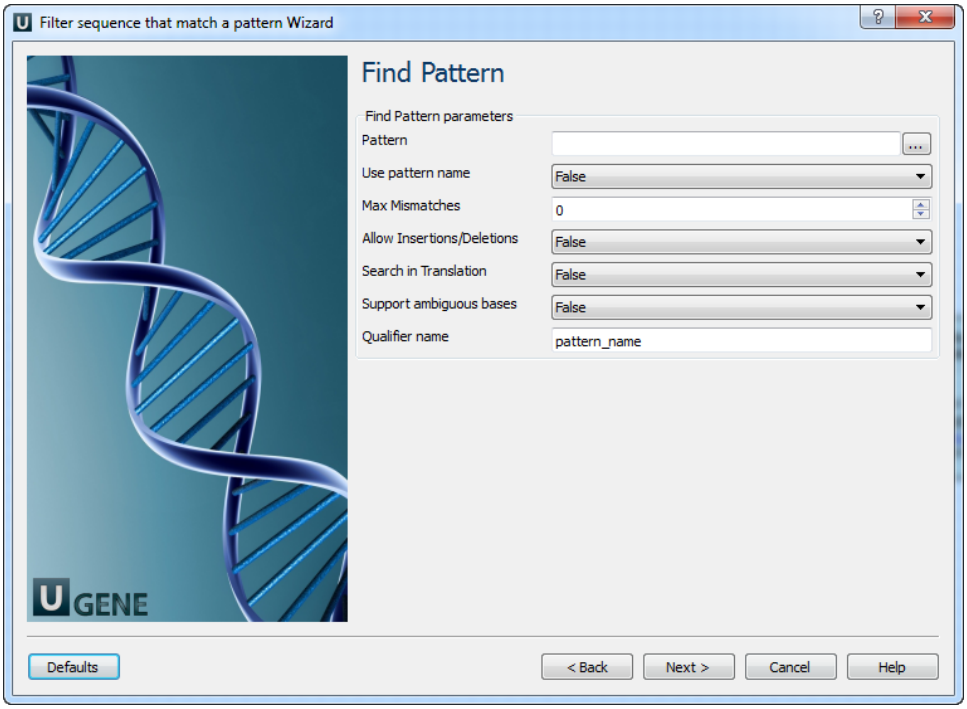
Workflow Wizard

The wizard has 3 pages.

1. Input sequence(s): On this page you must input sequence(s).



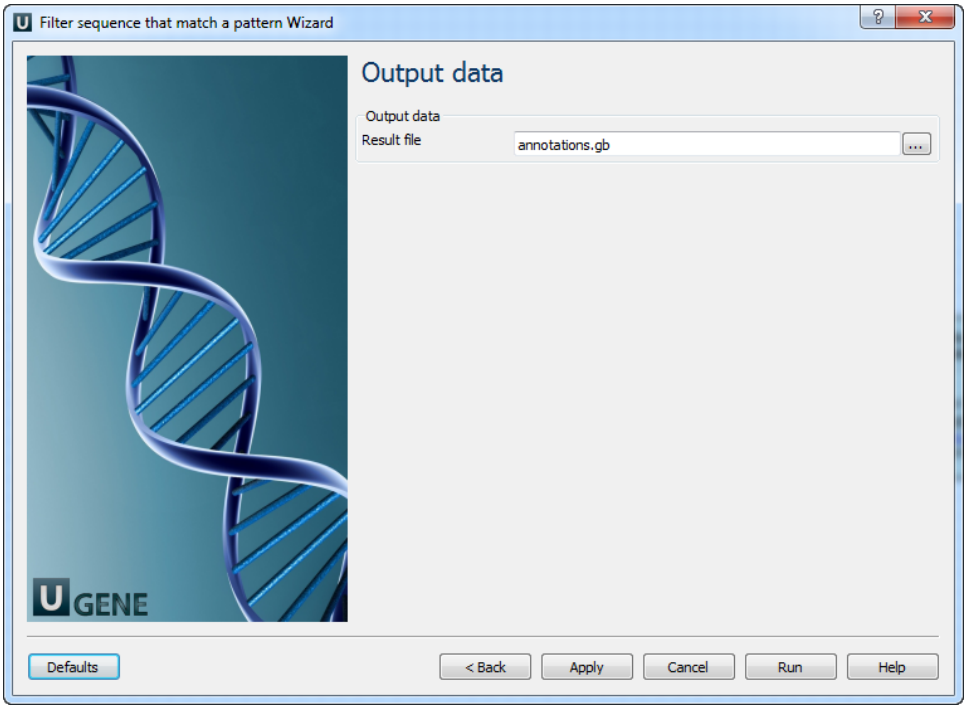
2. Find pattern: On this page you must input pattern(s) and you can modify searching parameters.



The following parameters are available:

Pattern	Semicolon-separated list of patterns to search for.
Use pattern name	If patterns are loaded from a file, use names of pattern sequences as annotation names. The name from the parameters is used by default.
Max Mismatches	Maximum number of mismatches between a substring and a pattern.
Allow Insertions /Deletions	Takes into account possibility of insertions/deletions when searching. By default substitutions are only considered.
Search in Translation	Translates a supplied nucleotide sequence to protein and searches in the translated sequence.
Support ambiguous bases	Performs correct handling of ambiguous bases. When this option is activated insertions and deletions are not considered.
Qualifier name	Name of qualifier in result annotations which is containing a pattern name.

3. Output data: On this page you can modify output parameters.



Search for Inverted Repeats

For each input sequence the workflow performs a search of inverted repeats.

Then it saves the repeats found on the direct strand to the "direct_strand_repeat_units.fa" file and the complement ones to the "compl_strand_repeat_units.fa" file.

How to Use This Sample

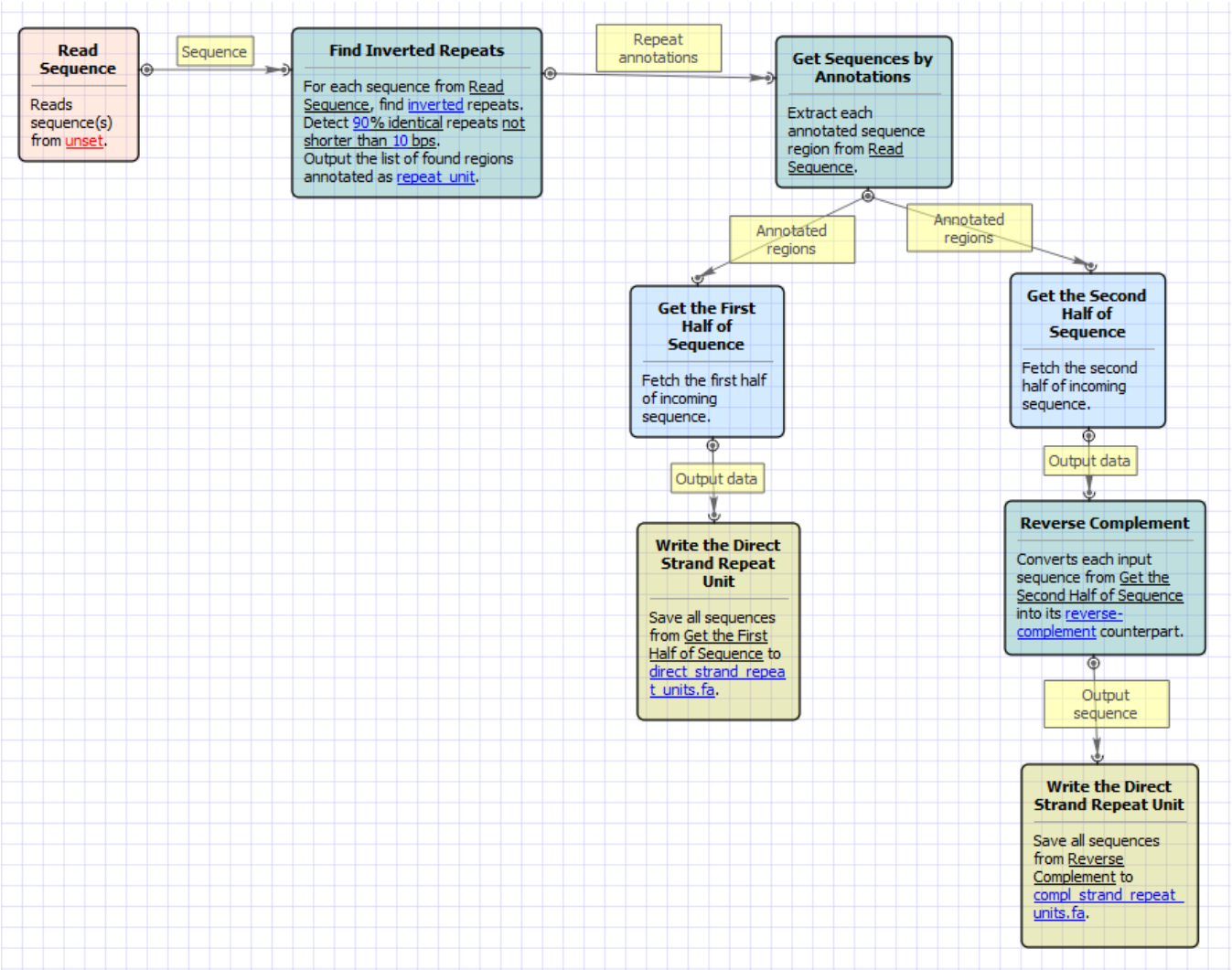
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Search for Inverted Repeats" can be found in the "Scwnarios" section of the Workflow Designer samples.

Workflow Image

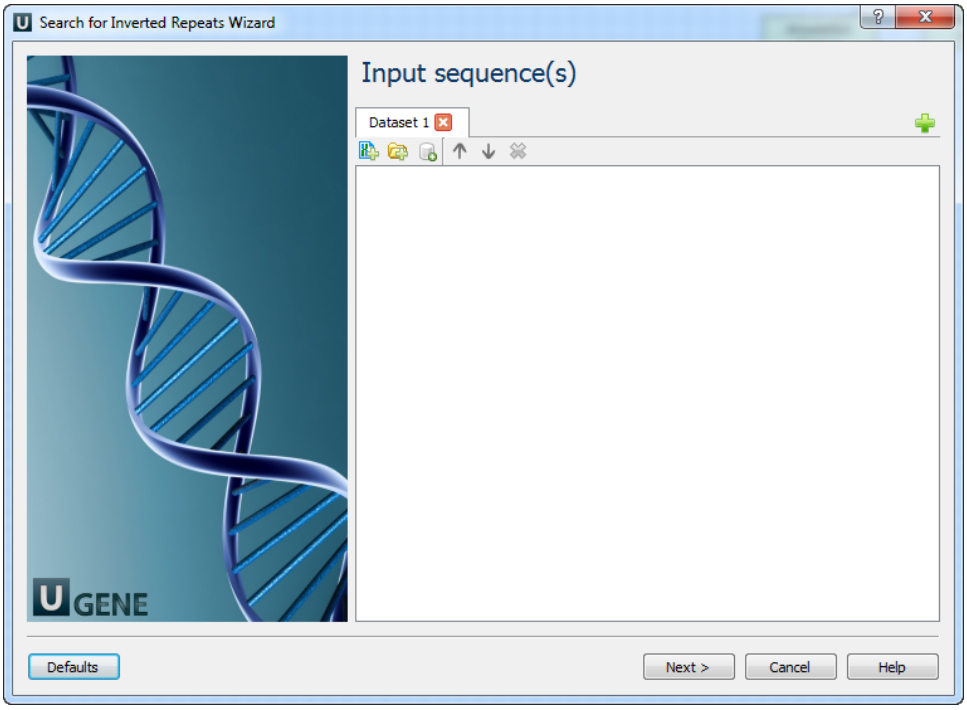
The opened workflow looks as follows:



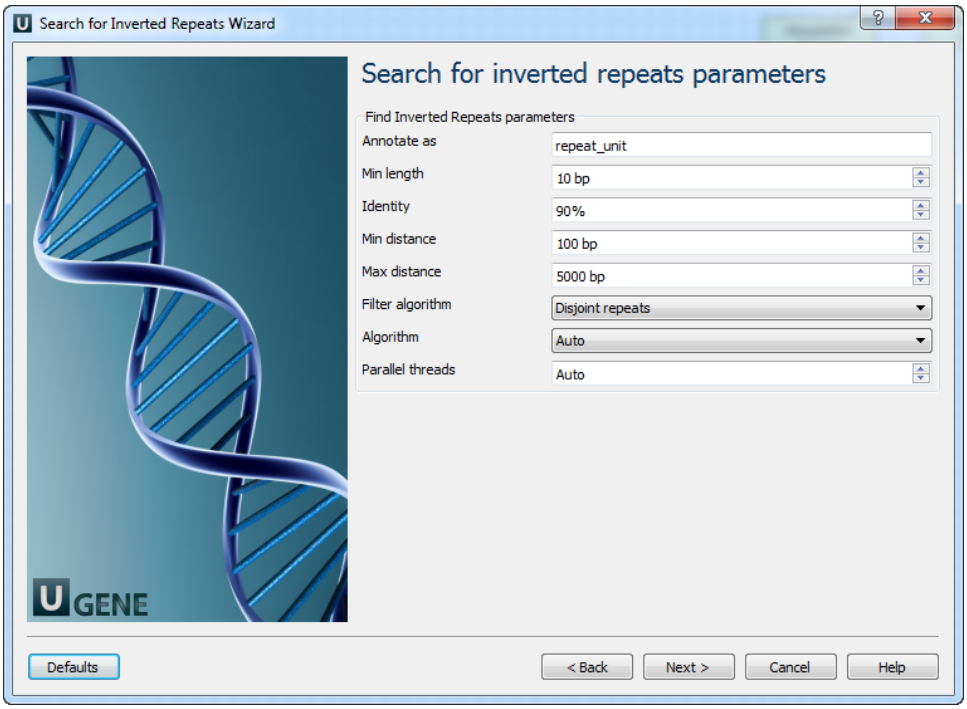
Workflow Wizard

The wizard has 3 pages.

1. Input sequence(s): On this page you must input sequence(s).



2. Search for inverted repeats parameters: On this page you can modify inverted repeats parameters.

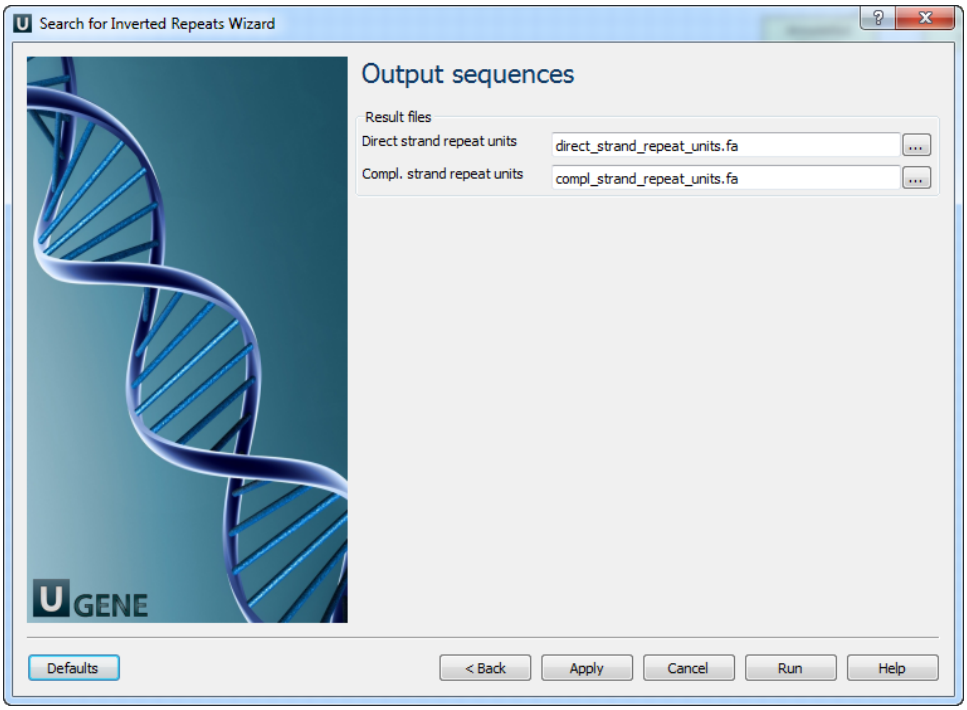


The following parameters are available:

Annotate as	Name of the result annotations marking found repeats.
Min length	Minimum length of repeats.
Identity	Repeats identity.
Min distance	Minimum distance between repeats.
Max distance	Maximum distance between repeats.
Filter algorithm	Filter repeats algorithm.
Algorithm	Control over variations of algorithm.

Parallel threads	Number of parallel threads used for the task.

3. Output Sequences: On this page you can modify result file(s) settings.



Find Patterns

This simple workflow finds patterns in you sequences and save them as annotations. You can use the workflow to map primers, regulatory signals, genes, etc. It loads any set of sequences from your files or folders and finds patterns in them.

How to Use This Sample

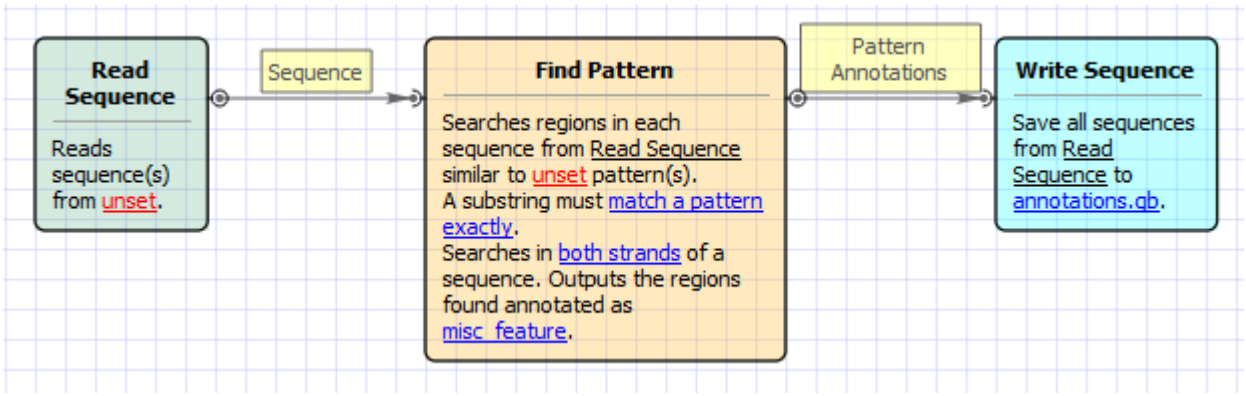
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Find Patterns" can be found in the "Scenarios" section of the Workflow Designer samples.

Workflow Image

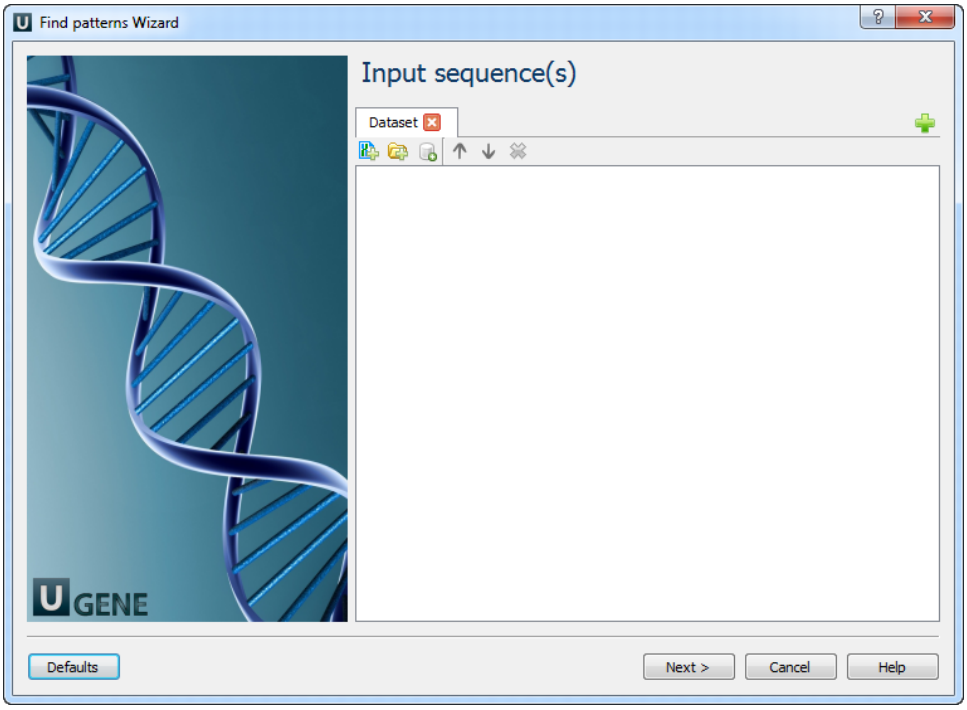
The workflow looks as follows:



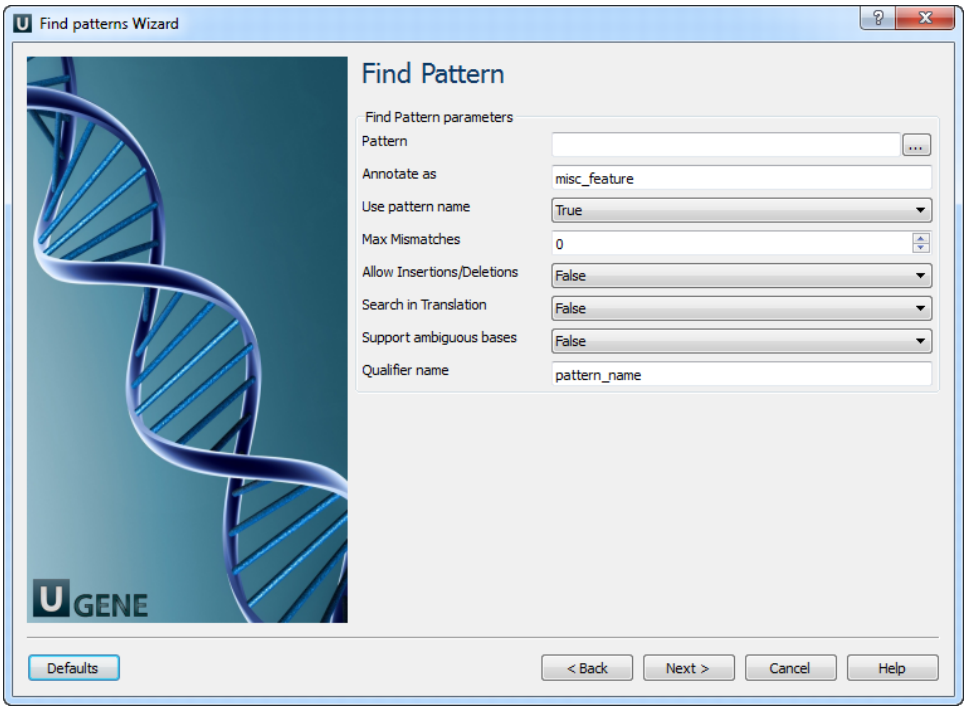
Workflow Wizard

The wizard has 3 pages.

1. Input sequence(s): On this page you must input sequence(s).



2. Find pattern: On this page you must input pattern(s) and you can modify searching parameters.

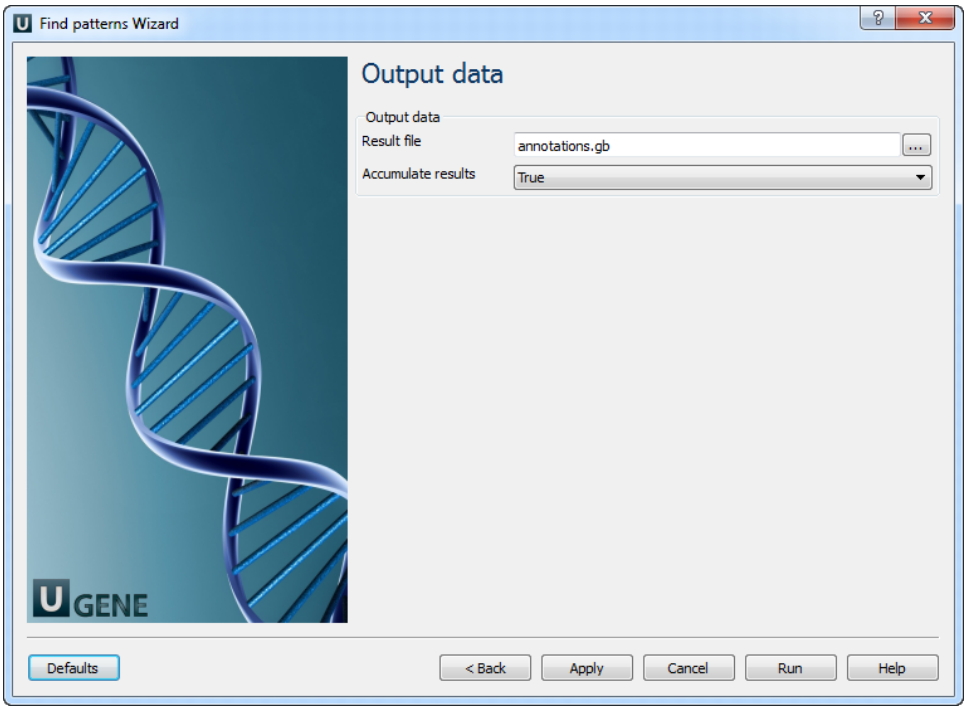


The following parameters are available:

Pattern	Semicolon-separated list of patterns to search for.
Annotate as	Name of the result annotations.
Use pattern name	If patterns are loaded from a file, use names of pattern sequences as annotation names. The name from the parameters is used by default.
Max Mismatches	Maximum number of mismatches between a substring and a pattern.
Allow Insertions	Takes into account possibility of insertions/deletions when searching. By default substitutions are only considered.

/Deletions	
Search in Translation	Translates a supplied nucleotide sequence to protein and searches in the translated sequence.
Support ambiguous bases	Performs correct handling of ambiguous bases. When this option is activated insertions and deletions are not considered.
Qualifier name	Name of qualifier in result annotations which is containing a pattern name.

3. Output data: On this page you can modify output parameters.



Gene-by-gene Approach for Characterization of Genomes

Suppose you have genomes and you want to characterize them. One of the ways to do that is to build a table of what genes are in each genome and what are not there.

1. Create a local BLAST db of your genome sequence/contigs. One db per one genome.
2. Create a file with sequences of genes you what to explore. This file will be the input file for the workflow.
3. Setup location and name of BLAST db you created for the first genome.
4. Setup output files: report location and output file with annotated (with BLAST) sequence. You might want to delete the "Write Sequence" element if you do not need output sequences.
5. Run the workflow.
6. Run the workflow on the same input and output files changing BLAST db for each genome that you have.

As the result you will get the report file. With "Yes" and "No" field. "Yes" answer means that the gene is in the genome. "No" answer MIGHT mean that there is no gene in the genome. It is a good idea to analyze all the "No" sequences using annotated files. Just open a file and find a sequence with a name of a gene that has "No" result.

How to Use This Sample

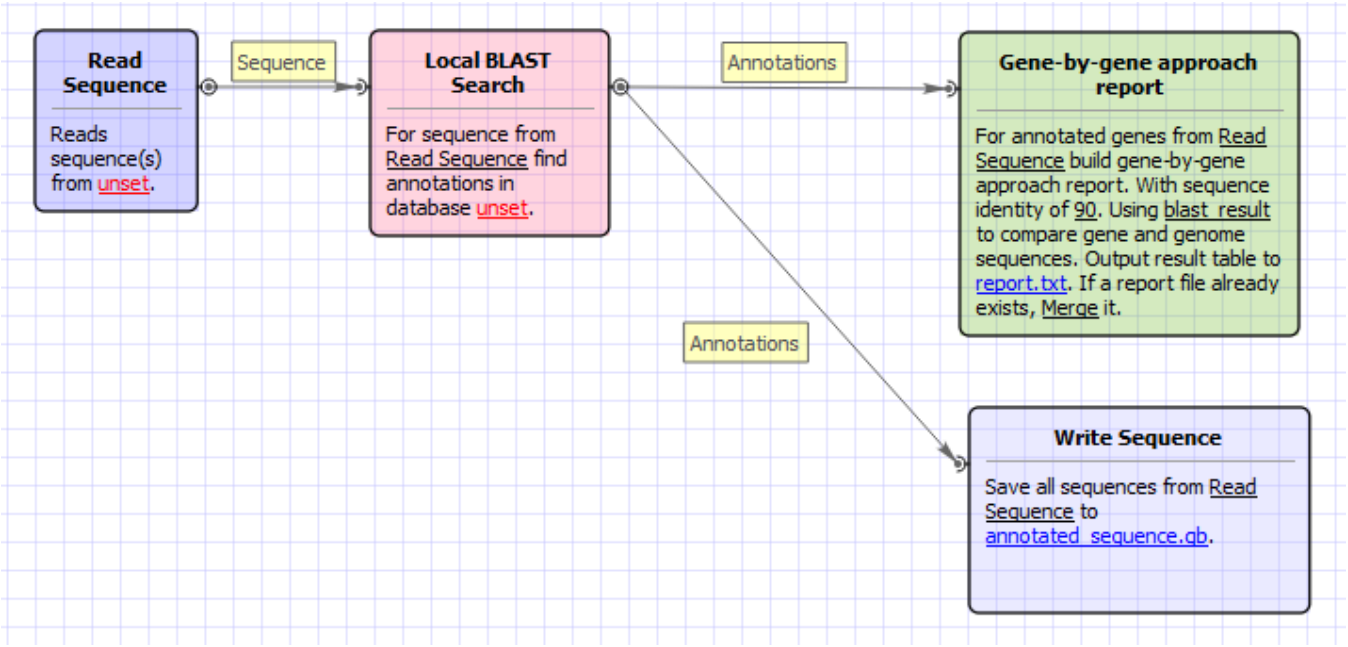
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Gene-by-gene Approach for Characterization of Genomes" can be found in the "Scenarios" section of the Workflow Designer samples.

Workflow Image

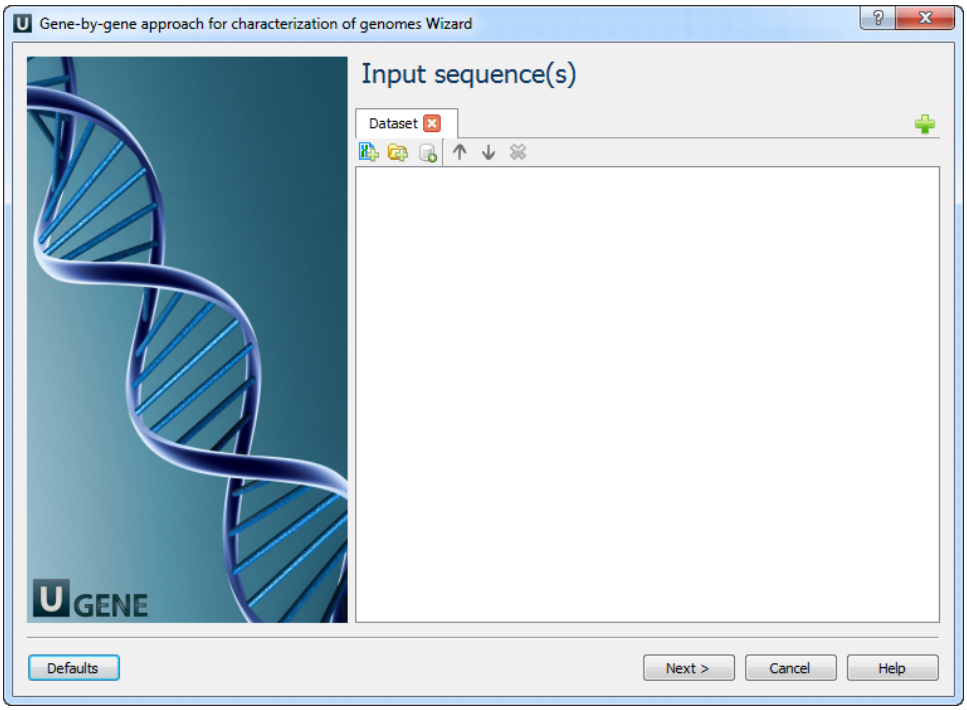
The workflow looks as follows:



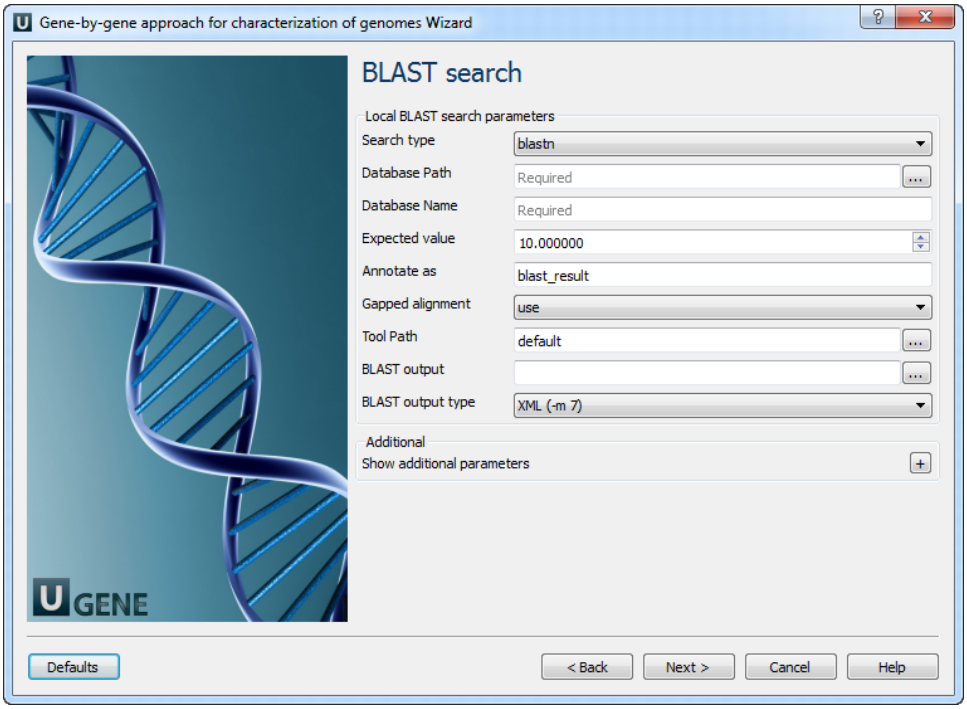
Workflow Wizard

The wizard has 3 pages.

1. Input sequence(s): On this page you must input sequence(s).



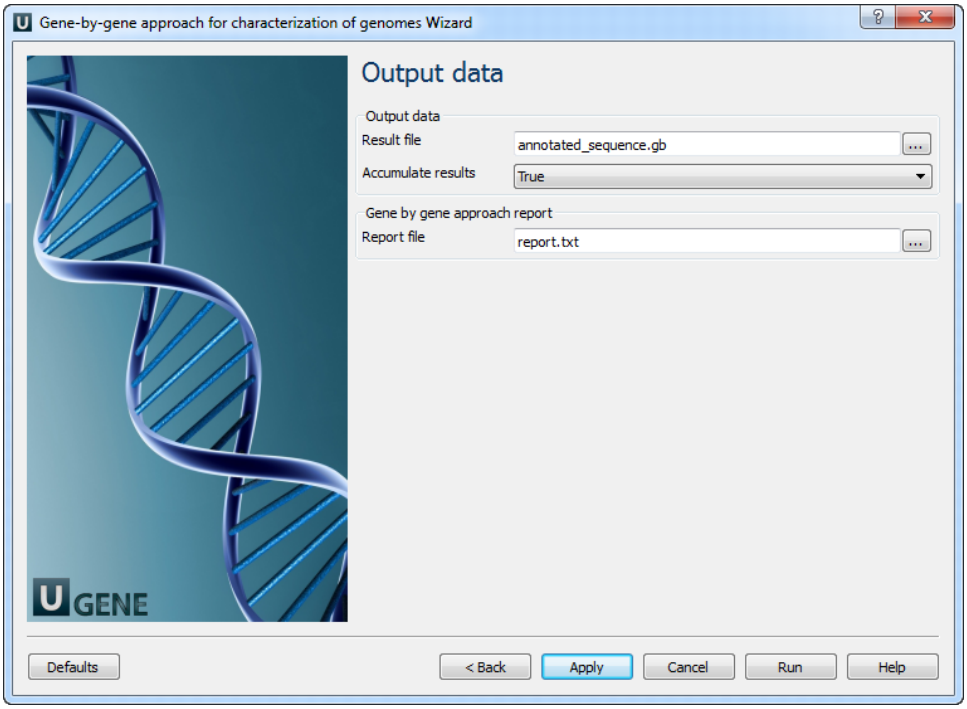
2. BLAST search: On this page you can modify BLAST search parameters.



The following parameters are available:

Search type	Select type of BLAST searches.
Database Path	Path with database files.
Database Name	Base name for BLAST DB files.
Expected value	This setting specifies the statistical significance threshold for reporting matches against database sequences.
Annotate as	Name for annotations.
Gapped alignment	Perform gapped alignment.
Tool Path	External tool path.
BLAST output	Location of BLAST output file.
BLAST output type	Type of BLAST output file.
Temporary directory	Directory for temporary files.
Gap costs	Cost to create and extend a gap in an alignment.
Match scores	Reward and penalty for matching and mismatching bases.

3. Output data: On this page you can modify output parameters.



Group Primer Pairs

The workflow helps determining different primer pairs that can be used in the same experiment.

First, you input a set of primers' sequences in the following order: pair1_direct_primer, pair1_reverse_primer, pair2_direct_primer, pair2_reverse_primer, etc. This could be a multifasta file, for example.

Second, the primers are checked for heterodimer formations. If there is no such formations between all primers in two or more primer pairs, it means that these pairs can be put simultaneously in the same reaction tube, so the workflow GROUPS these primer pairs.

However, please note that this workflow doesn't check the correctness of the primers themselves, for example for hairpins, selfdimers, etc.

The result report of the analysis is stored, by default, in the "report.html" file.

How to Use This Sample

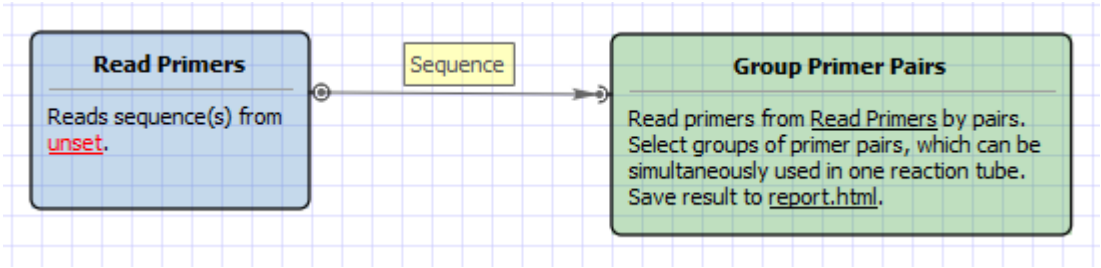
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Group Primer Pairs" can be found in the "Scenarios" section of the Workflow Designer samples.

Workflow Image

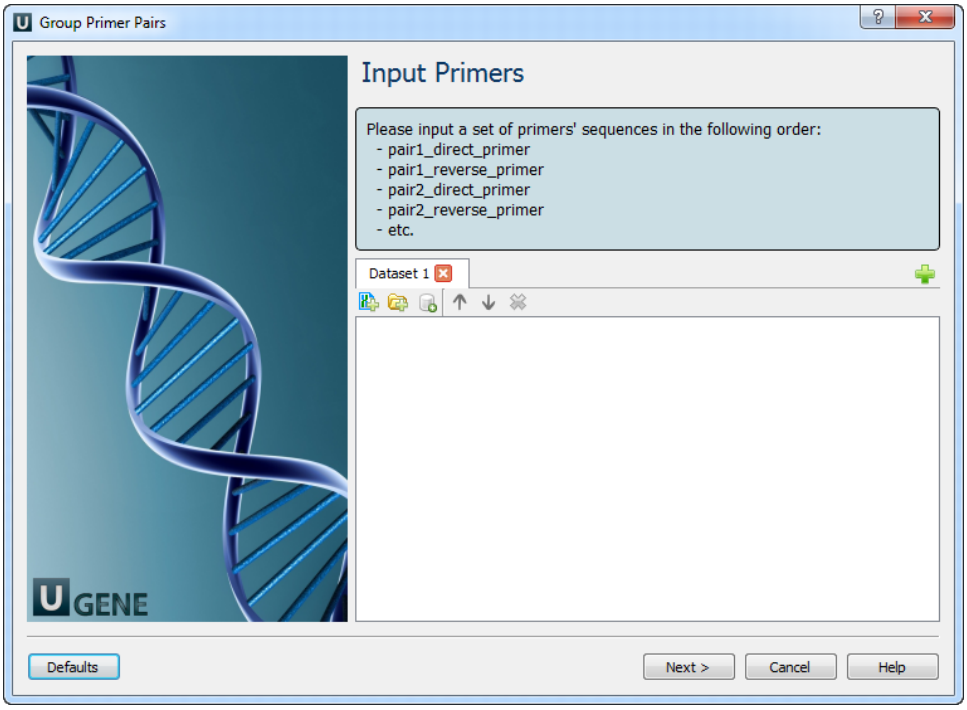
The workflow looks as follows:



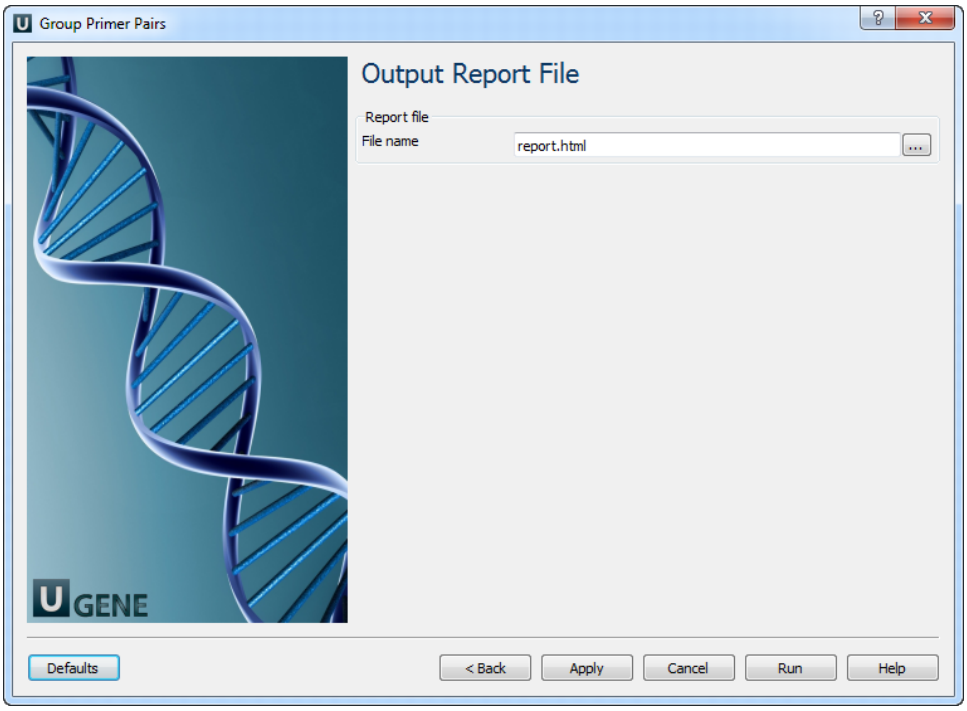
Workflow Wizard

The wizard has 2 pages.

1. Input primers: On this page you must input primers.



2. Output report file: On this page you can modify output parameters.



Intersect Annotations

The workflow takes two sets of annotations as input (denoted as A and B). It intersects the sets and outputs the result annotations.

How to Use This Sample

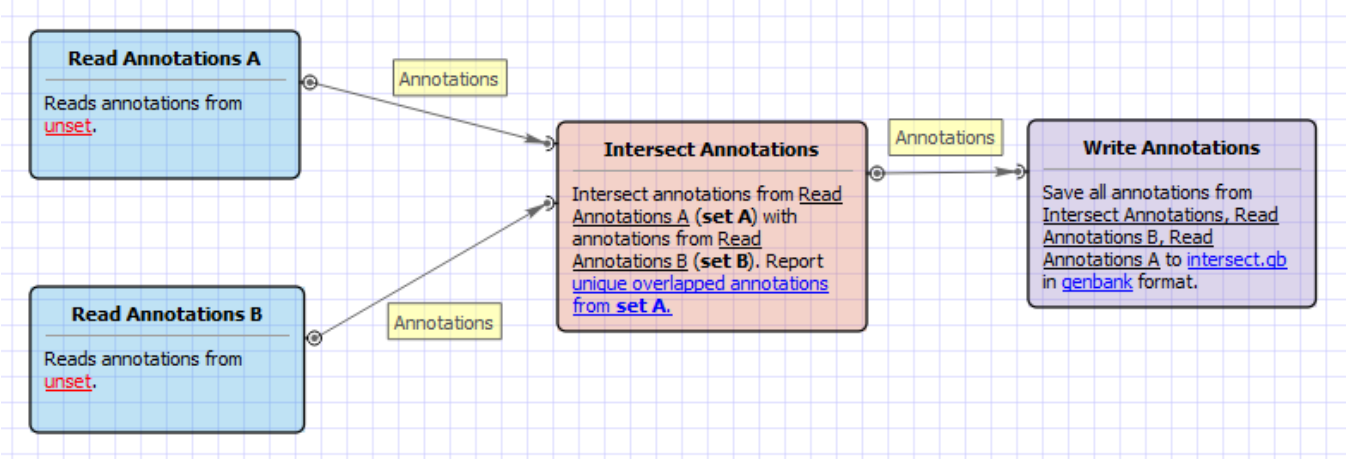
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Intersect Annotations" can be found in the "Scenarios" section of the Workflow Designer samples.

Workflow Image

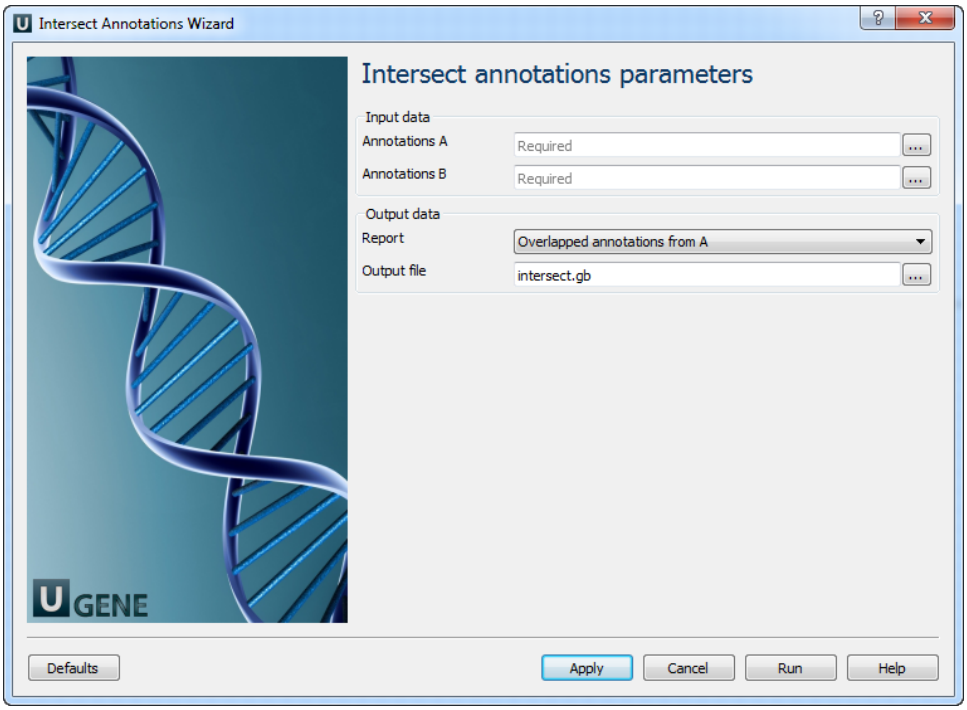
The opened workflow looks as follows:



Workflow Wizard

The wizard has 1 page.

1. Intersect annotations parameters: On this page you must input two sets of annotations and you can modify the output parameters.



Filter out Short Sequences

To use this workflow input a set of sequences and set a minimum sequence length. All sequences with length less than the specified value will be filtered out. The result will be written into a FASTA file by default.

How to Use This Sample

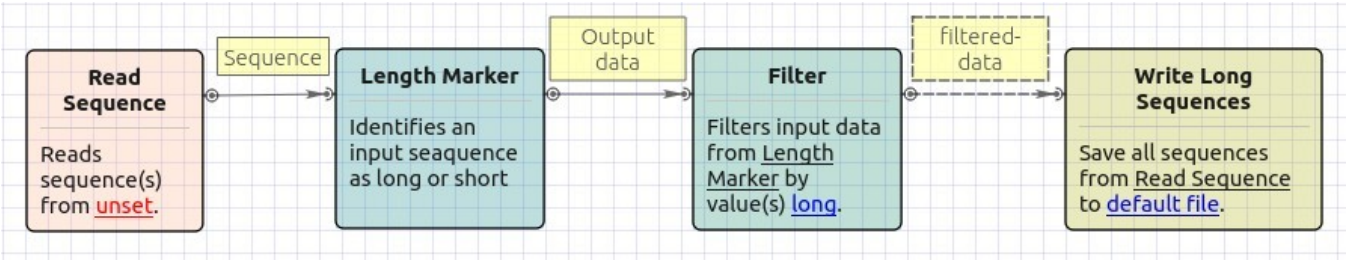
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Filter out Short Sequences" can be found in the "Scenarios" section of the Workflow Designer samples.

Workflow Image

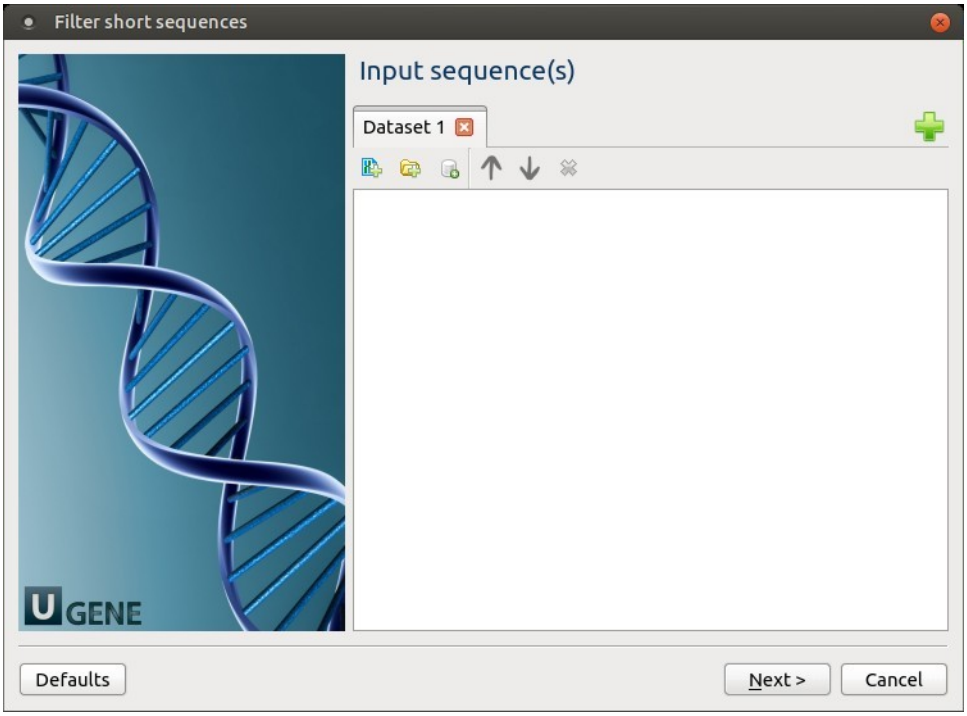
The opened workflow looks as follows:



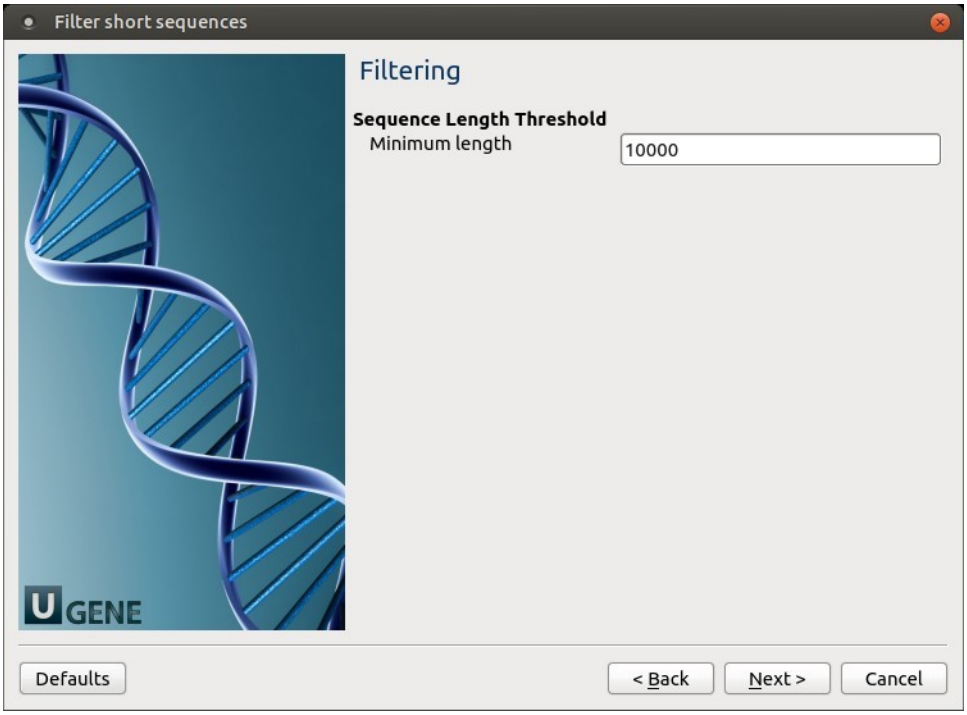
Workflow Wizard

The wizard has 3 pages.

1. Input sequence(s): On this page, input files must be set.



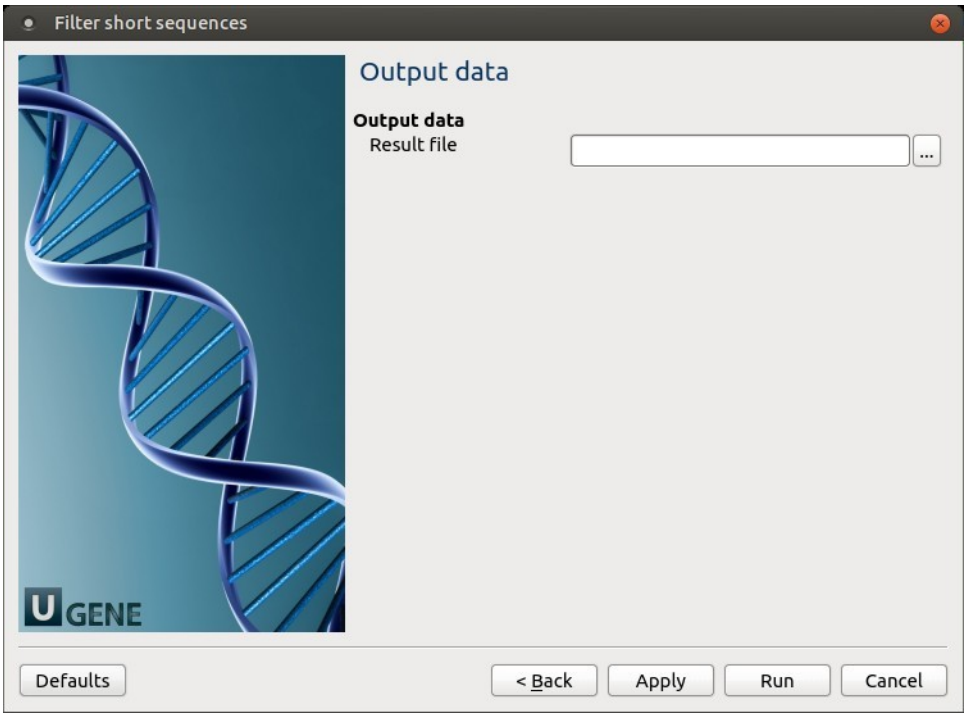
2. Filtering: The filtering parameters can be changed here.



The following parameters are available:

Minimum length	Minimum sequence length

3. Output data: On this page, the output file can be selected:



Merge Sequences and Annotations

This sample workflow shows how to merge input sequences with sets of annotations.

For example, you may have sequences in FASTA format and annotations in GFF format, and you would like to merge them and save the result into GenBank files.

The steps of the workflow are these:

1. The workflow reads sequences from the input sequence files, e.g. *sequence1*, *sequence2*, *sequence3*.
2. The workflow reads annotations from the input files with annotations, e.g. *ann_set1*, *ann_set2*, *ann_set3*.
3. The sequences and the annotations are multiplexed. The result is:
    - *sequence1 + ann_set1*
    - *sequence2 + ann_set2*
    - *sequence3 + ann_set3*
4. The result is written to the output files.

How to Use This Sample

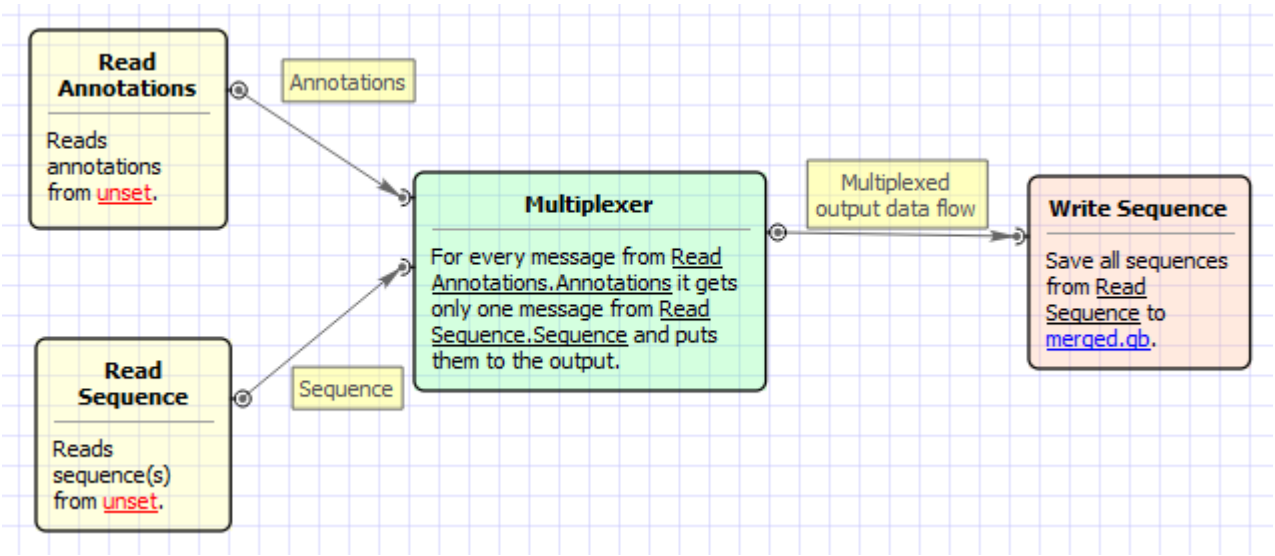
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Merge Sequences and Annotations" can be found in the "Scenarios" section of the Workflow Designer samples.

Workflow Image

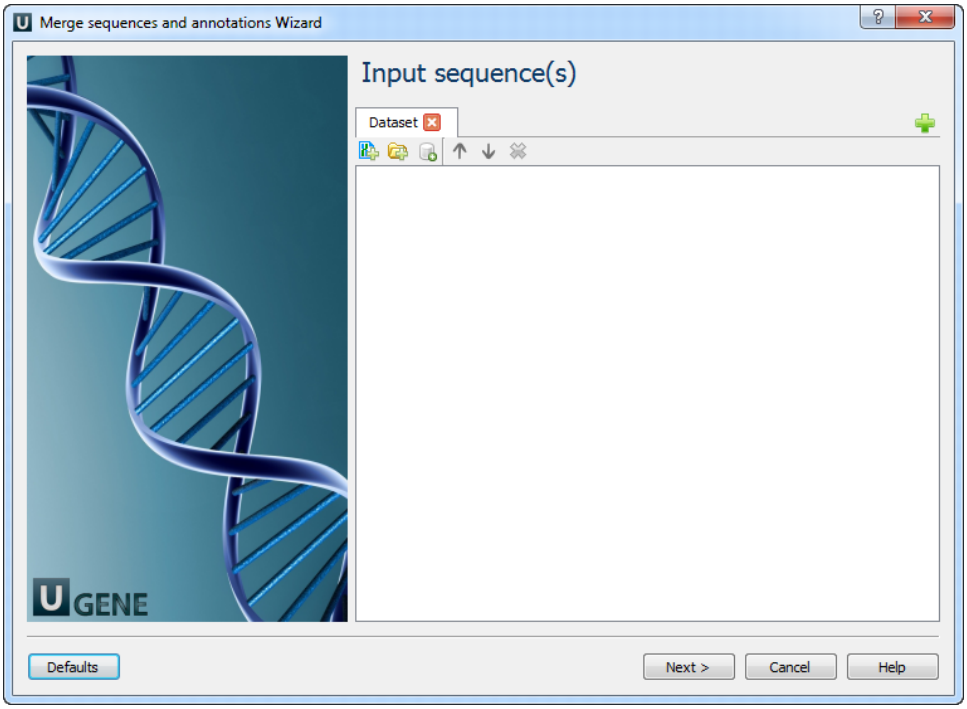
The workflow looks as follows:



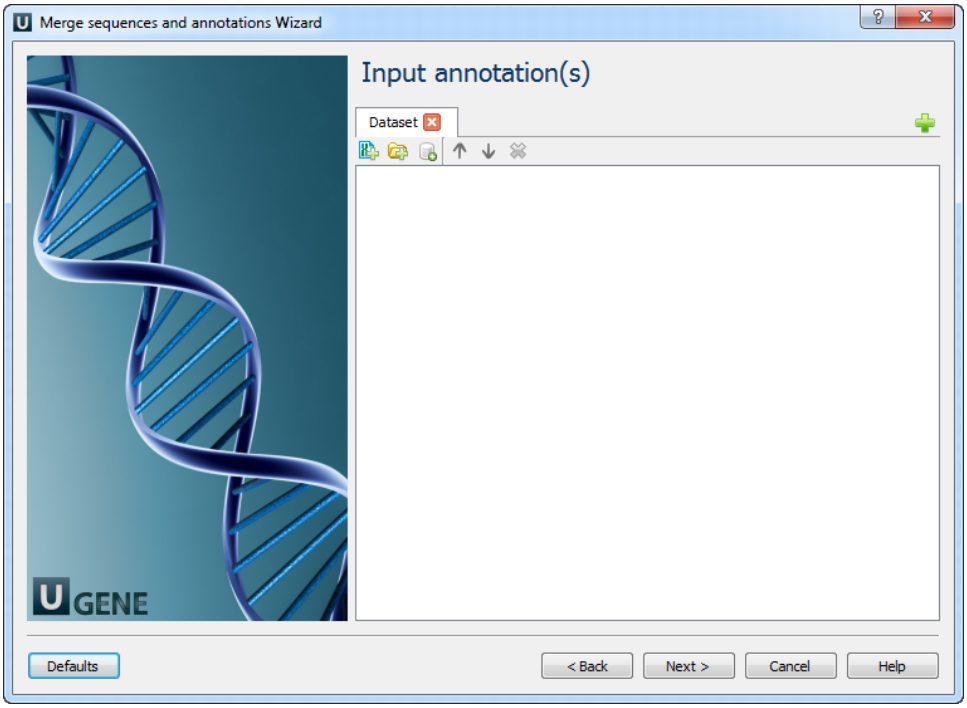
Workflow Wizard

The wizard has 3 pages.

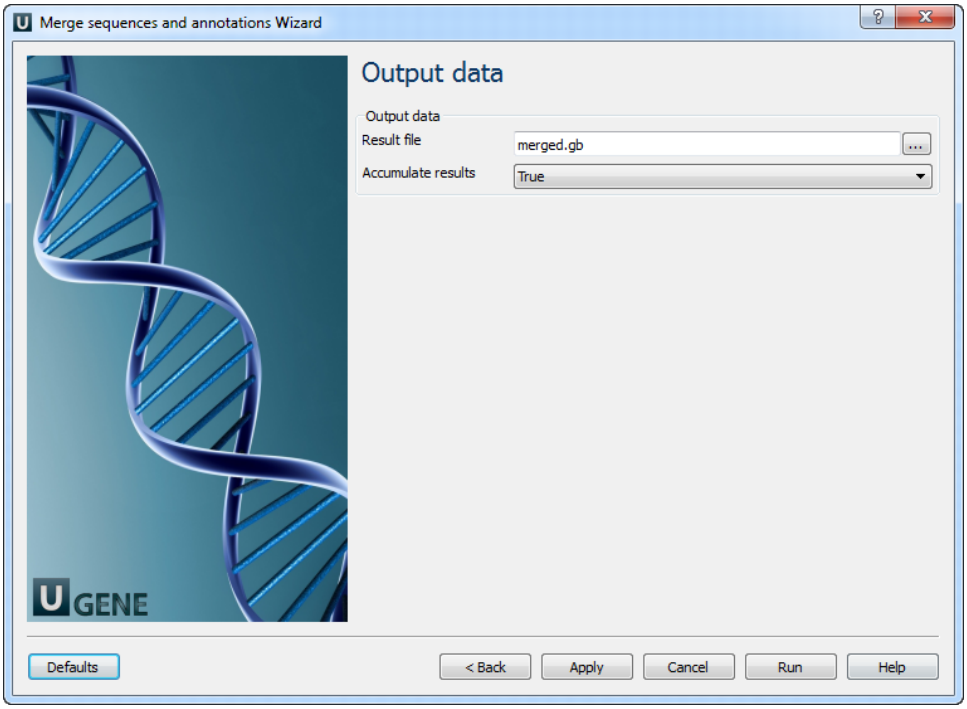
1. Input sequence(s): On this page you must input sequence(s).



2. Input annotation(s): On this page you must input annotation(s).



3. Output data: On this page you can modify output parameters.



In Silico PCR Sample

This workflow simulates the PCR process.

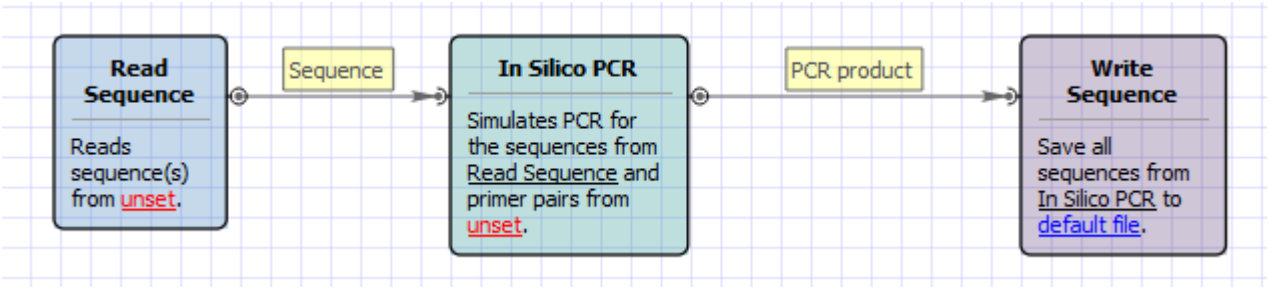
> **How to Use This Sample**
>
> If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "In Silico PCR" can be found in the "Scenarios" section of the Workflow Designer samples.

Workflow Image

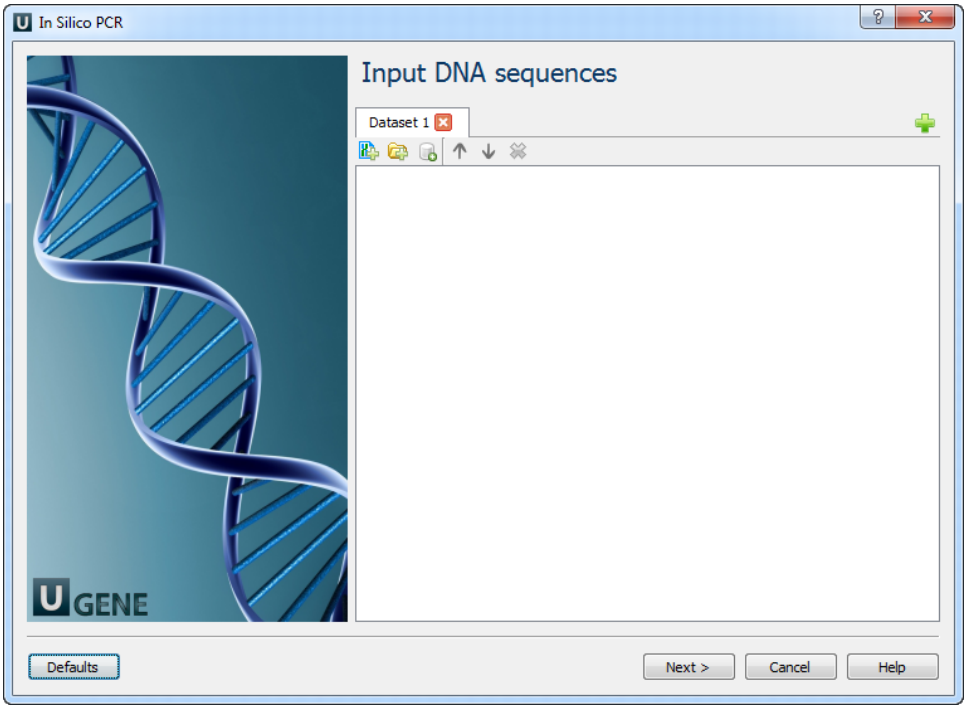
The opened workflow looks as follows:



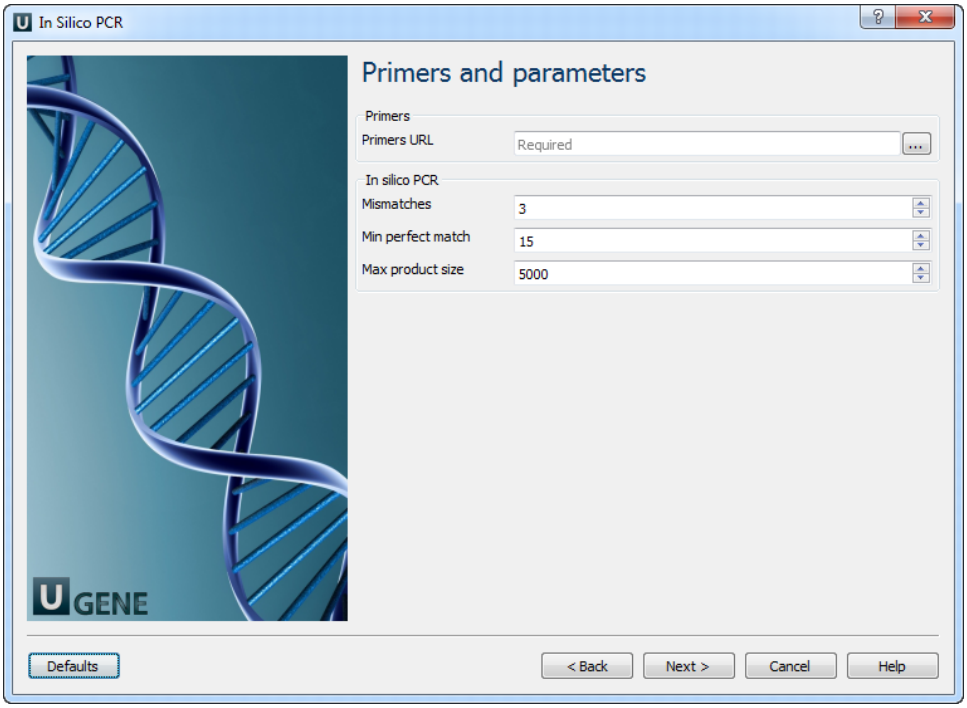
Workflow Wizard

The wizard has 3 pages.

1. Input DNA Sequences: On this page you must input DNA sequences.



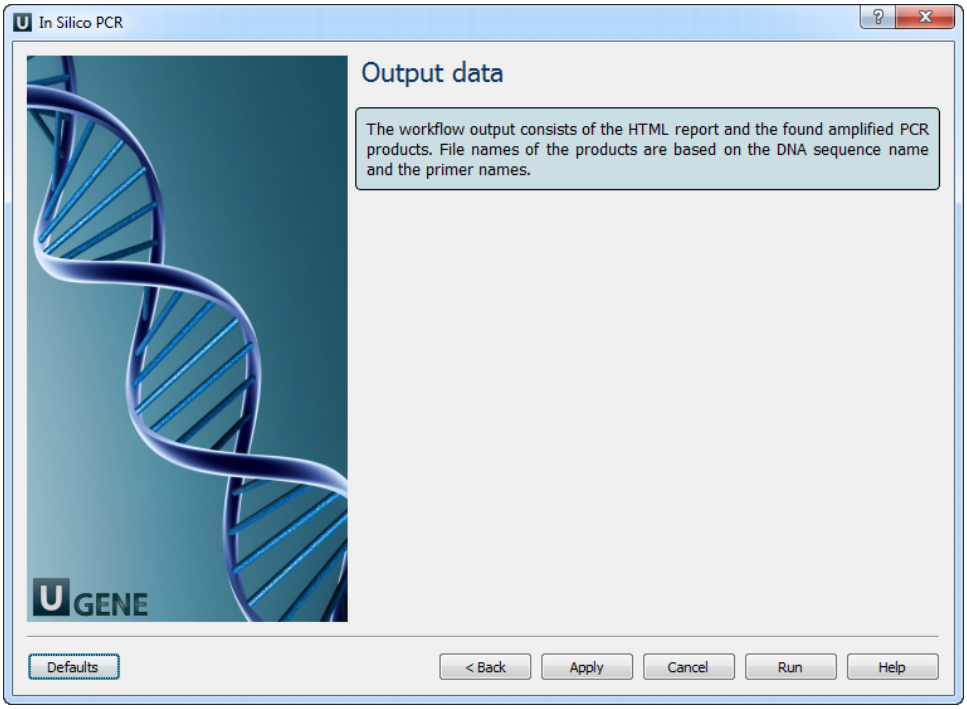
2. Primers and Parameters: Here you must input *Primers* and you can optionally modify *In Silico PCR* parameters.



The following parameters are available:

Primers URL	A URL to the input file with primer pairs.
Mismatches	Number of allowed mismatches.
Min perfect match	Number of bases that match exactly on 3' end of primers.
Max product size	Maximum size of amplified region.

3. Output data: Here you can see information about output data.



Remote BLASTing

The workflow sample, described below, allows one to do remote queries to the NCBI BLAST database to search for homologous nucleotide sequences for multiple input sequences at the same time.

As the result of the BLAST each input sequence is annotated with the "blast result" annotations. These annotations are used to fetch the corresponding homologous sequences from the NCBI database based on the identifiers specified in the "blast result" annotations. The output homologous sequences and the original sequences, annotated by BLAST, are grouped by folders.

Environment Requirements

Internet connection is required for running this workflow sample.

How to Use This Sample

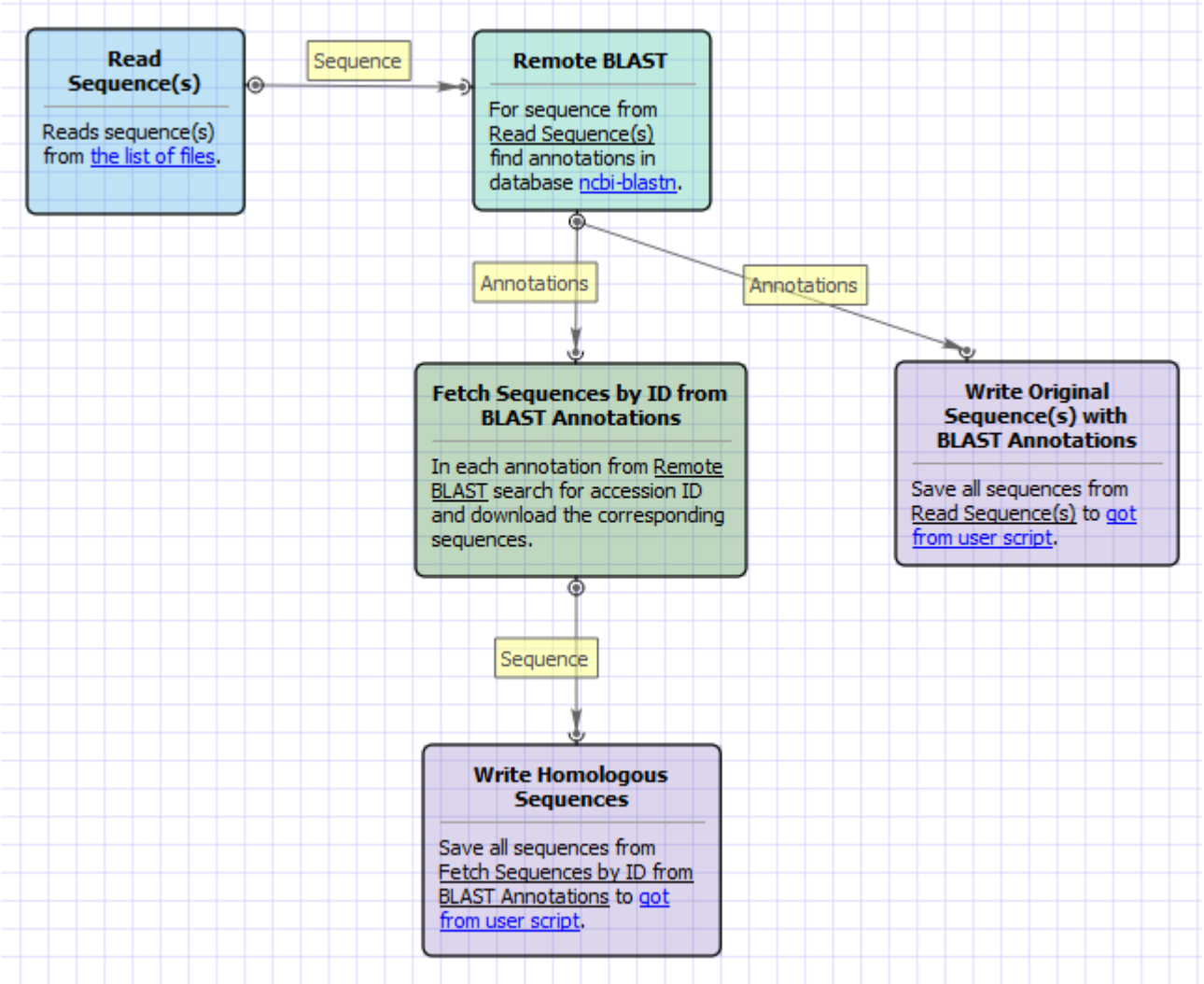
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Remote BLASTing" can be found in the "Scenarios" section of the Workflow Designer samples.

Workflow Image

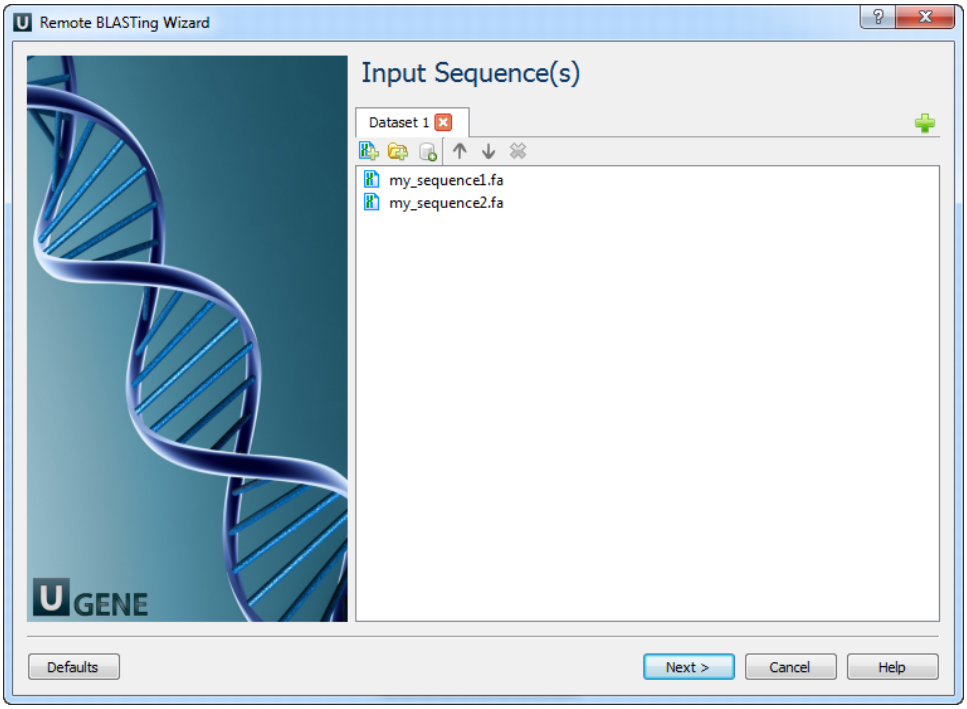
The opened workflow looks as follows:



Workflow Wizard

The wizard has 3 pages.

1. Input Sequence(s) Page: On this page you must input at least one nucleotide sequence.

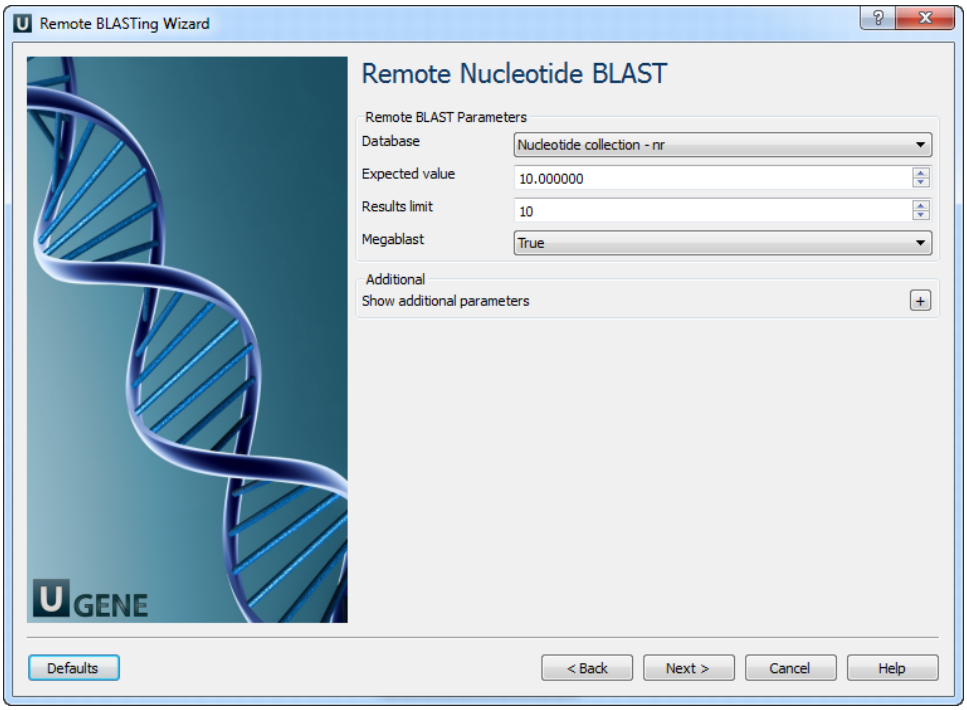


Example Input Data

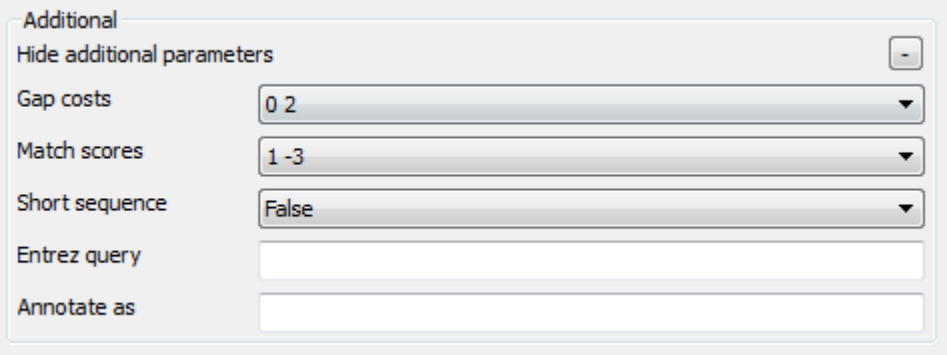
For example, you can use the following two files as an input to the workflow:

- my_sequence1.fa
- my_sequence2.fa

2. Remote Nucleotide BLAST Page: Here you can optionally modify parameters that should be used for the remote BLAST queries. For example, you can select the search database, correct the e-value and set the maximum number of results (i.e. "Max hits"). The "Megablast" option, applied by default, specifies to optimize the search for high similar sequences only. Selecting it decreases the search time, but some less similar results could be skipped by the search in this case. Note that the "Megablast" option is also applied by default in the NCBI BLAST web interface.



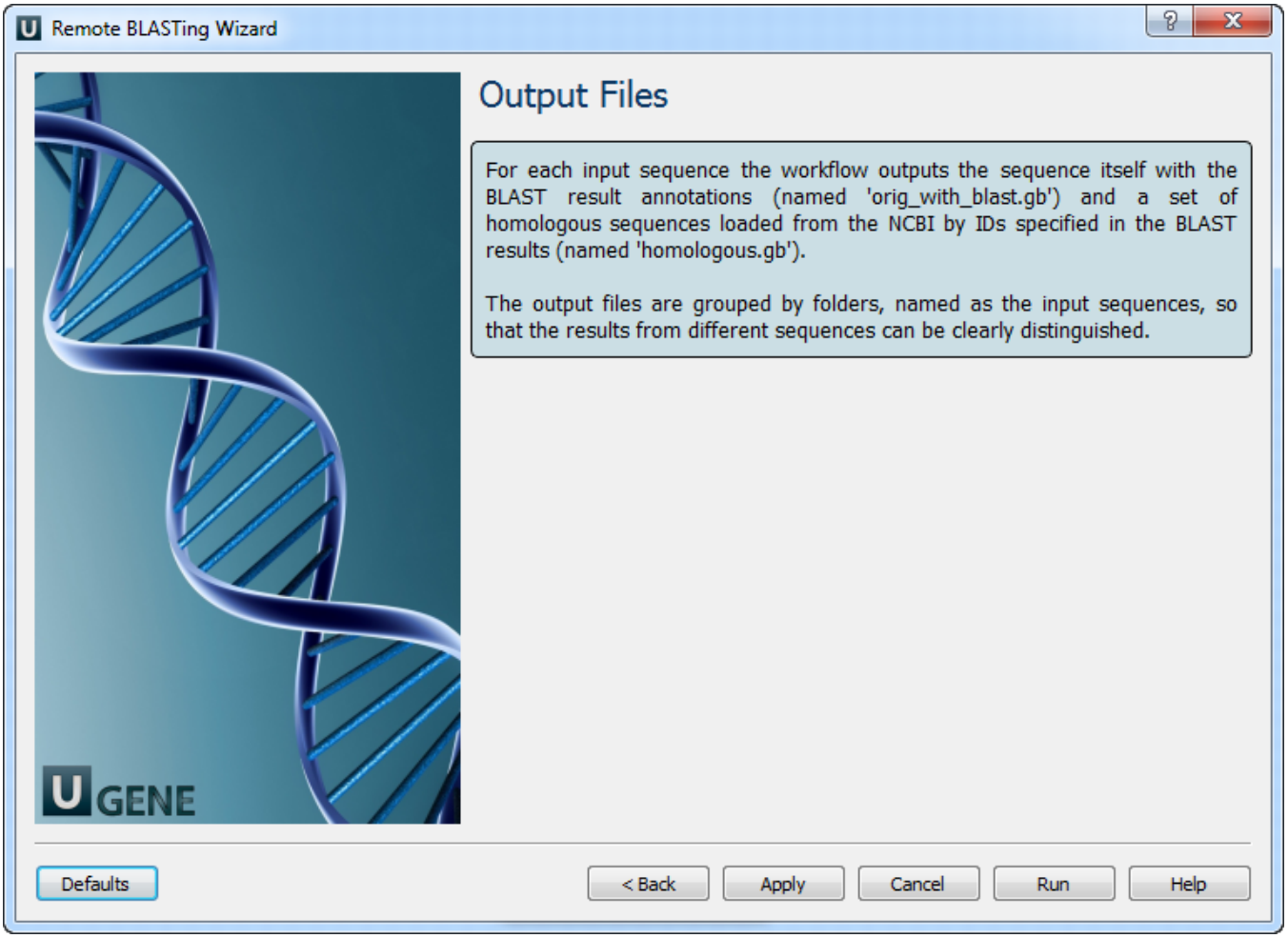
There are also some additional parameters. Description of them can be found in the Remote BLAST workflow element chapter of the documentation.



3. Output Files Page: this is an informational page. It states that this workflow has predefined names of the output files.
   For each input sequence the workflow outputs:
   - *"orig_with_blast.gb" file:* the file contains the input sequence itself and the "blast result" annotations;
   - "homologous.gb" file: the file contains the found homologous sequences loaded from the NCBI by identifiers, specified in the BLAST results.

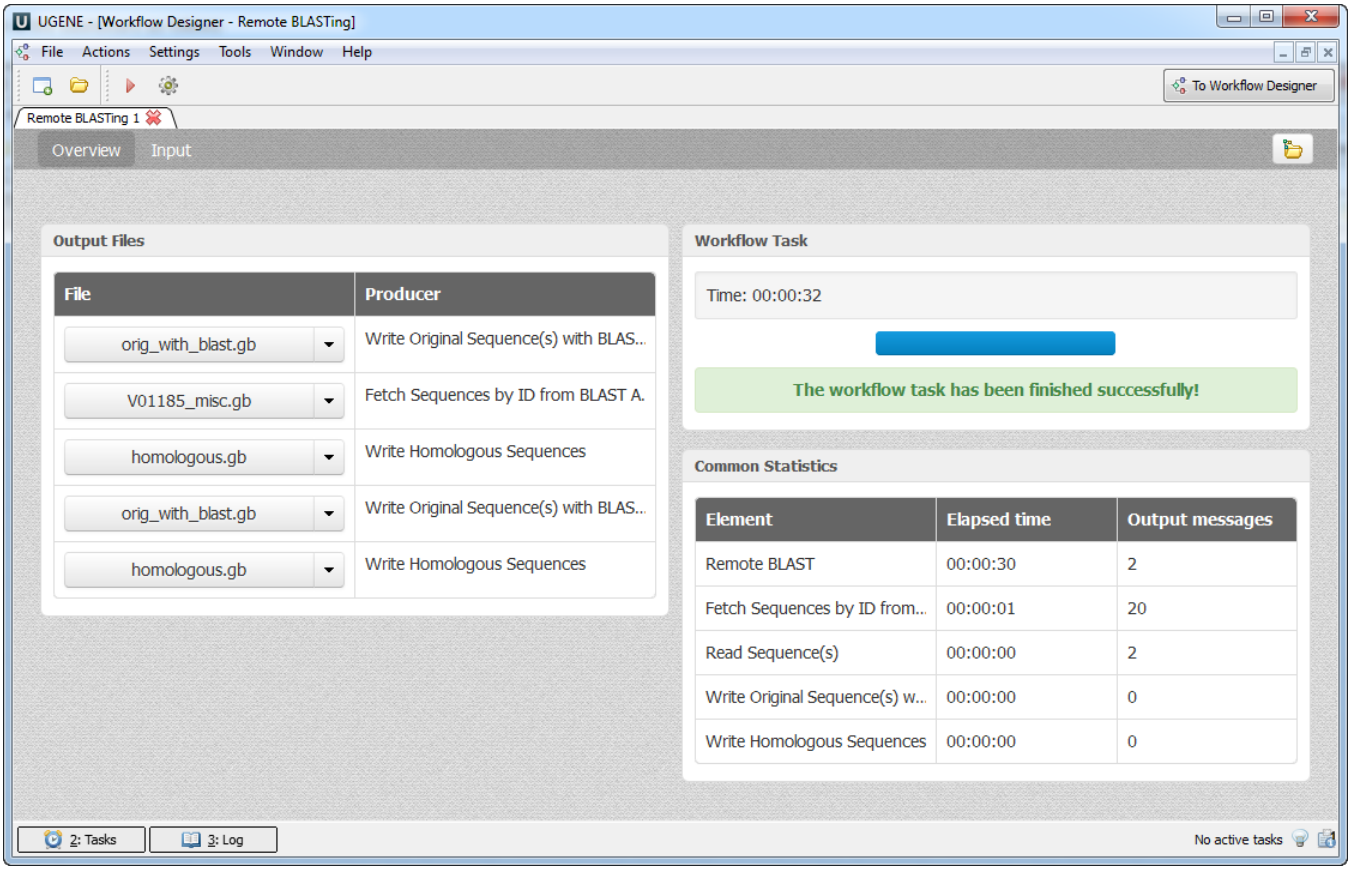
The results on the hard drive are grouped by folders (see below).

The wizard page looks as follows:



Workflow Result

The workflow output files are shown in the dashboard as follows:



Each file can be opened in the UGENE Sequence View by clicking on the corresponding link in the dashboard.

On the hard drive the output is grouped by folders with the names of the input sequences. For example, for the input sequences specified above, the output hierarchy will be the following:

- *my_sequence1.fa* folder with files:
  - *orig_with_blast.gb*
  - *homologous.gb*
- *my_sequence2.fa* folder with files:
  - *orig_with_blast.gb*
  - *homologous.gb*

Get Amino Translations of a Sequence

The workflow takes a nucleotide sequence as input and returns two files: translations of the sequence and translations of the complement sequence.

How to Use This Sample

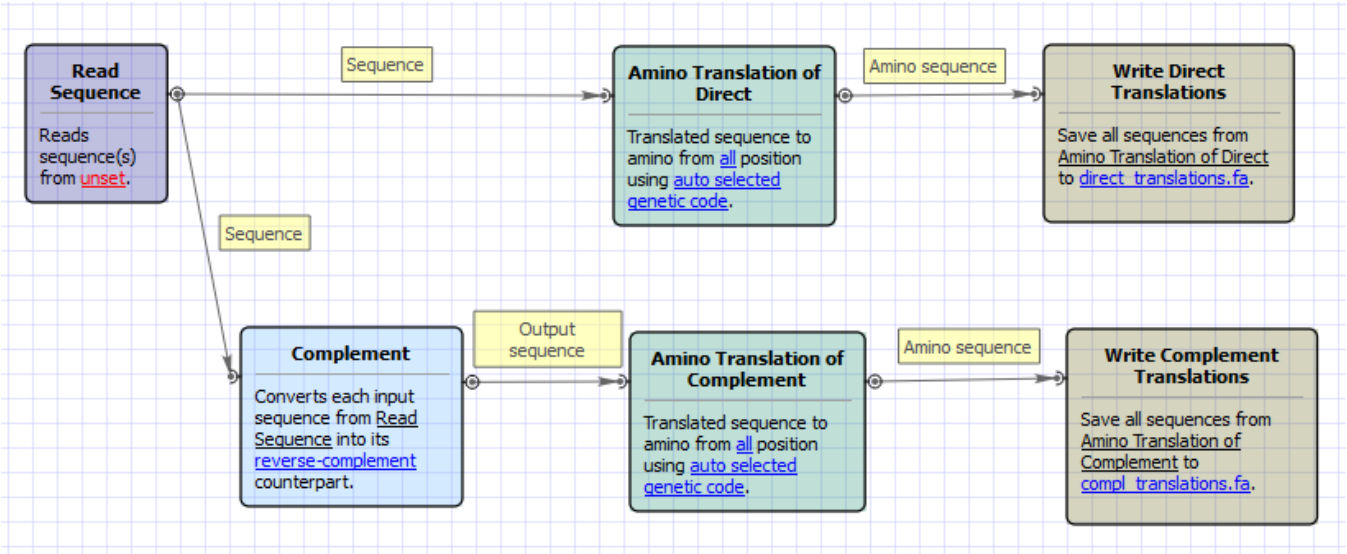
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Get Amino Translations of a Sequence" can be found in the "Scwnarios" section of the Workflow Designer samples.

Workflow Image

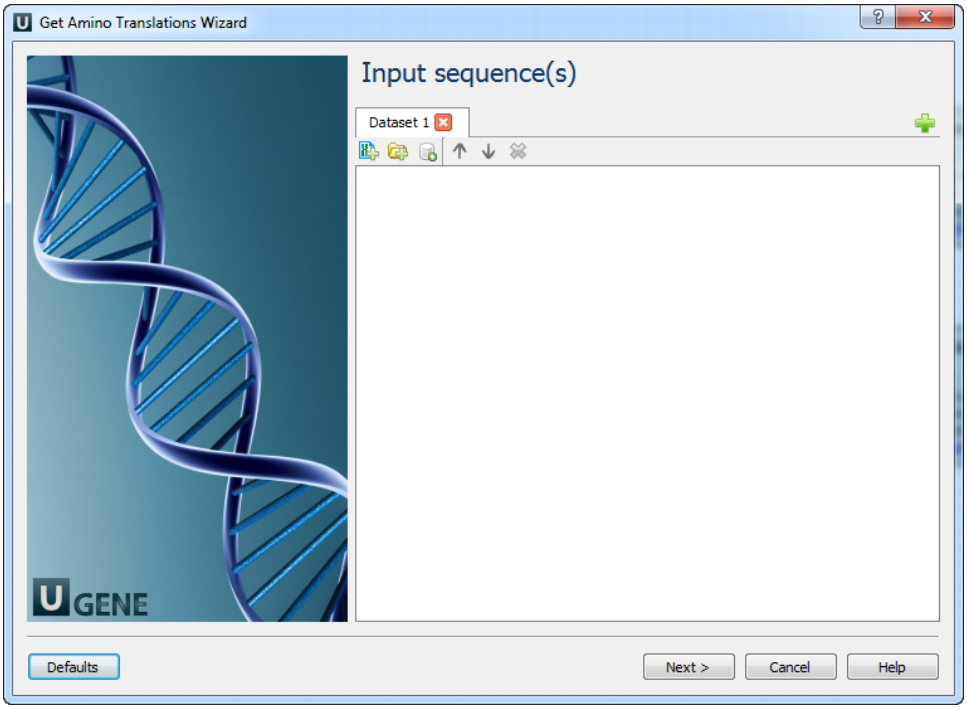
The opened workflow looks as follows:



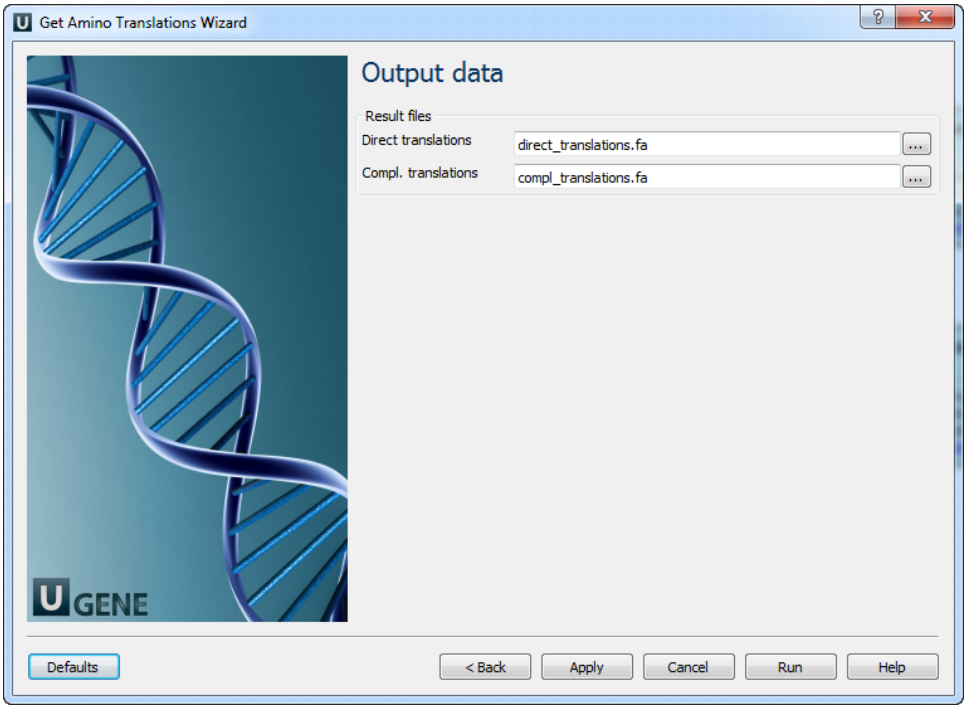
Workflow Wizard

The wizard has 2 pages.

1. Input sequence(s): On this page you must input sequence(s).



2. Output Data: On this page you can modify result files settings.



Transcriptomics

- Search for Transcription Factor Binding Sites (TFBS) in Genomic Sequences

Search for Transcription Factor Binding Sites (TFBS) in Genomic Sequences

This workflow predicts binding sites for number of transcription factors of interest using SITECON algorithm. The present workflow sample is designed for simultaneous recognition of binding sites for 3 different transcription factor types, you can expand it for recognition of any desired number of transription factor types.SITECON - is a program pakage for recognition of potential transcription factor binding sites basing on the data about conservative conformational and physicochemical properties revealed on the basis of the binding sites sets analysis.Citing SITECON Please cite:Oshchepkov D.Y., Vityaev E.E., Grigorovich D.A., Ignatieva E.V., Khlebodarova T.M.SITECON:a tool for detecting conservative conformational and physicochemicalproperties in transcription factor binding site alignments and for siterecognition. // Nucleic Acids Res. 2004 Jul 1;32(Web Server issue): W208-12.

How to Use This Sample

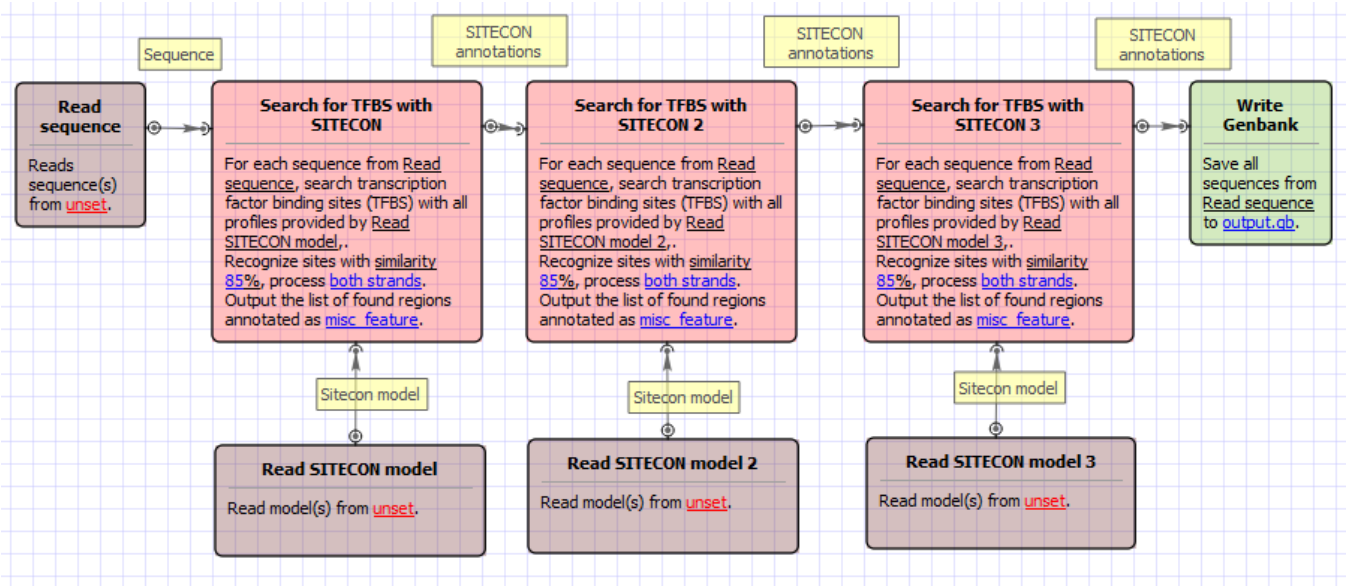
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Search for Transcription Factor Binding Sites (TFBS) in Genomic Sequences" can be found in the "Transcriptomics" section of the Workflow Designer samples.

Workflow Image

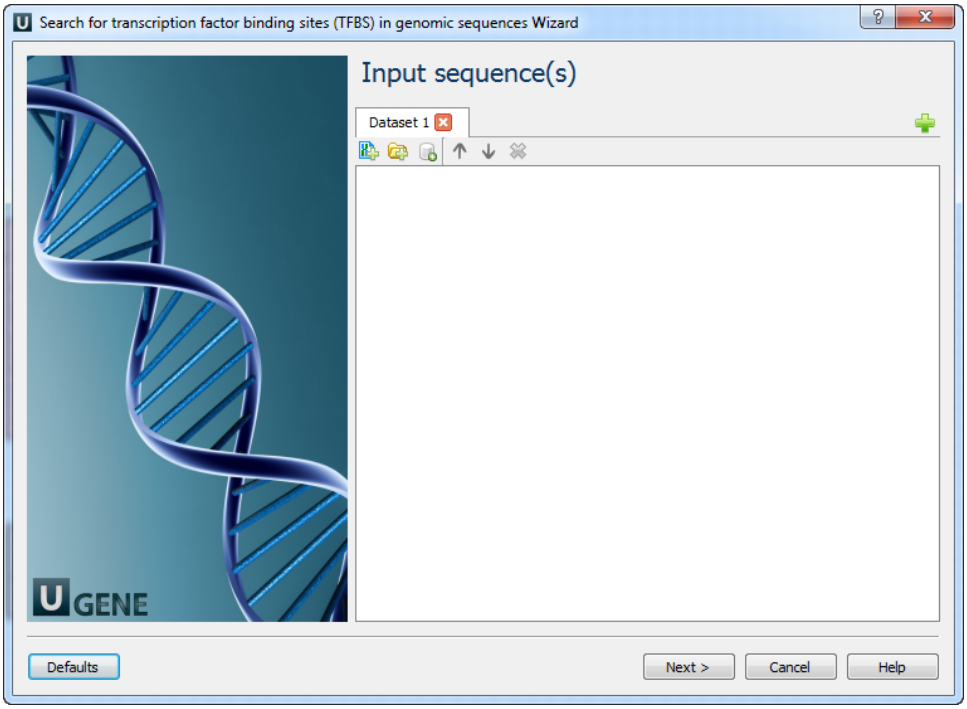
The workflow looks as follows:



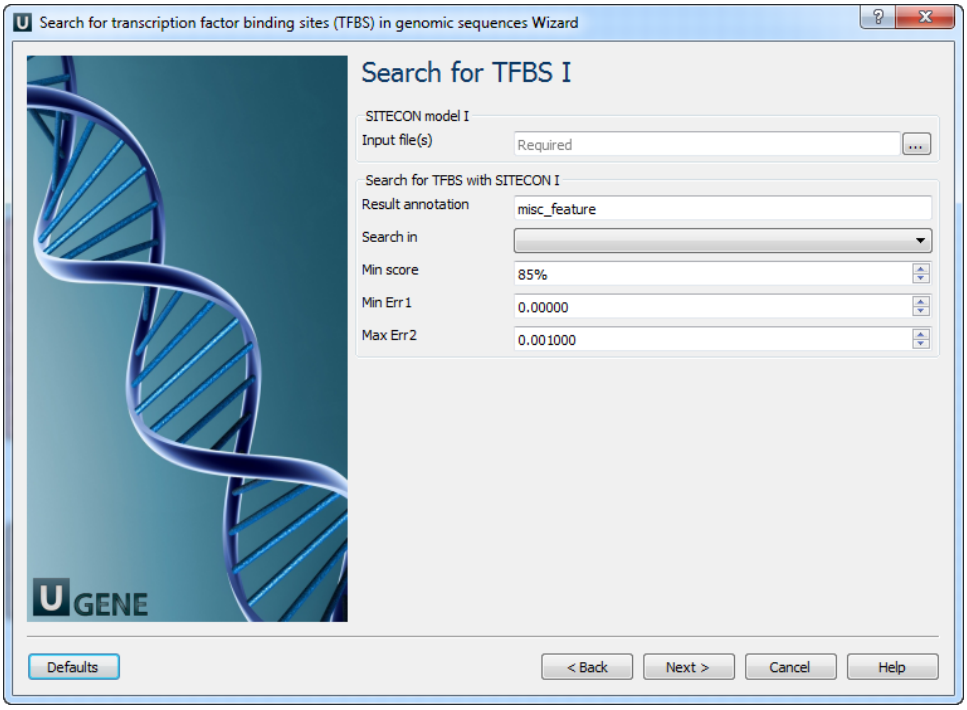
Workflow Wizard

The wizard has 5 pages.

1. Input sequence(s): On this page you must input sequence(s).



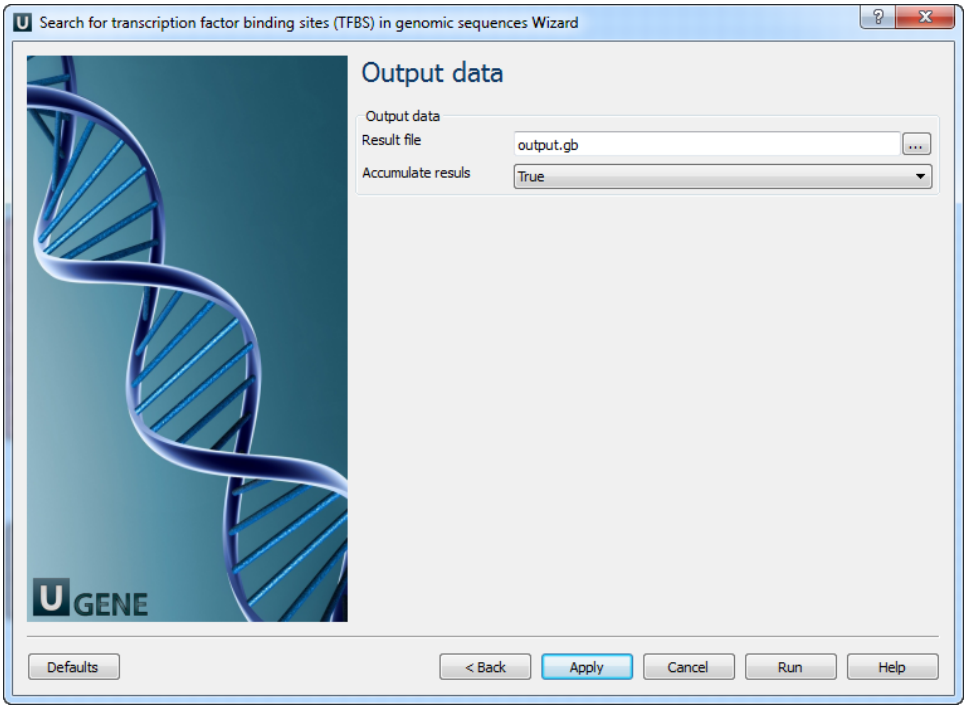
2. Search for TFBS 1, 2, 3: On these pages you can modify search for TFBS parameters.



The following parameters are available:

Input file(s)	Semicolon-separated list of paths to the input files.
Result annotation	Annotation name for marking found regions.
Search in	Which strands should be searched: direct, complement or both.
Min score	Recognition quality percentage threshold. If you need to switch off this filter choose the lowest value
Min Err 1	Alternative setting for filtering results, minimal value of Error type I. Note that all thresholds (by score, by err1 and by err2) are applied when filtering results. If you need to switch off this filter choose "0" value
Max Err 2	Alternative setting for filtering results, max value of Error type II. Note that all thresholds (by score, by err1 and by err2) are applied when filtering results. If you need to switch off this filter choose "1" value

3. Output data: On this page you can modify output parameters.



Query Designer

- About the Query Designer
- Query Designer Introduction
  - Launching the Query Designer
  - Terminology
    - Query Designer Window Components
    - Schema Elements
  - How to Create and Run Schema
- Manipulating Query Designer Element
  - Adding Algorithm Element
  - Adding Constraint Element
  - Renaming Algorithm Element
  - Resizing and Moving Algorithm Element
  - Managing of Elements Parameters
  - Changing Algorithm Element Appearance
  - Deleting Query Designer Element
- Manipulating Schema
  - Creating New Schema
  - Loading Schema
  - Saving Schema
  - Changing Schema Appearance
  - Setting Order of Algorithms Execution
  - Managing Strands
    - Element Direction in Schema
    - Querying Sequence Strands
  - Running Schema from the Query Designer
- Running Schema from the Sequence View
- Query Designer Schema File Format
  - Header Query Designer Element
  - Body Query Designer Element
    - Element Description
      - Algorithm Element Description
      - Constraint Element Description
    - Metainformation Query Designer Element
- Query Elements
  - Algorithm Elements
    - CDD Algorithm Element
    - Base Content Algorithm Element
    - Gc Content Algorithm Element
    - HMM3 Algorithm Element
    - ORF Algorithm Element
    - Primer Algorithm Element
    - Repeats Algorithm Element
    - Restriction Sites Algorithm Element

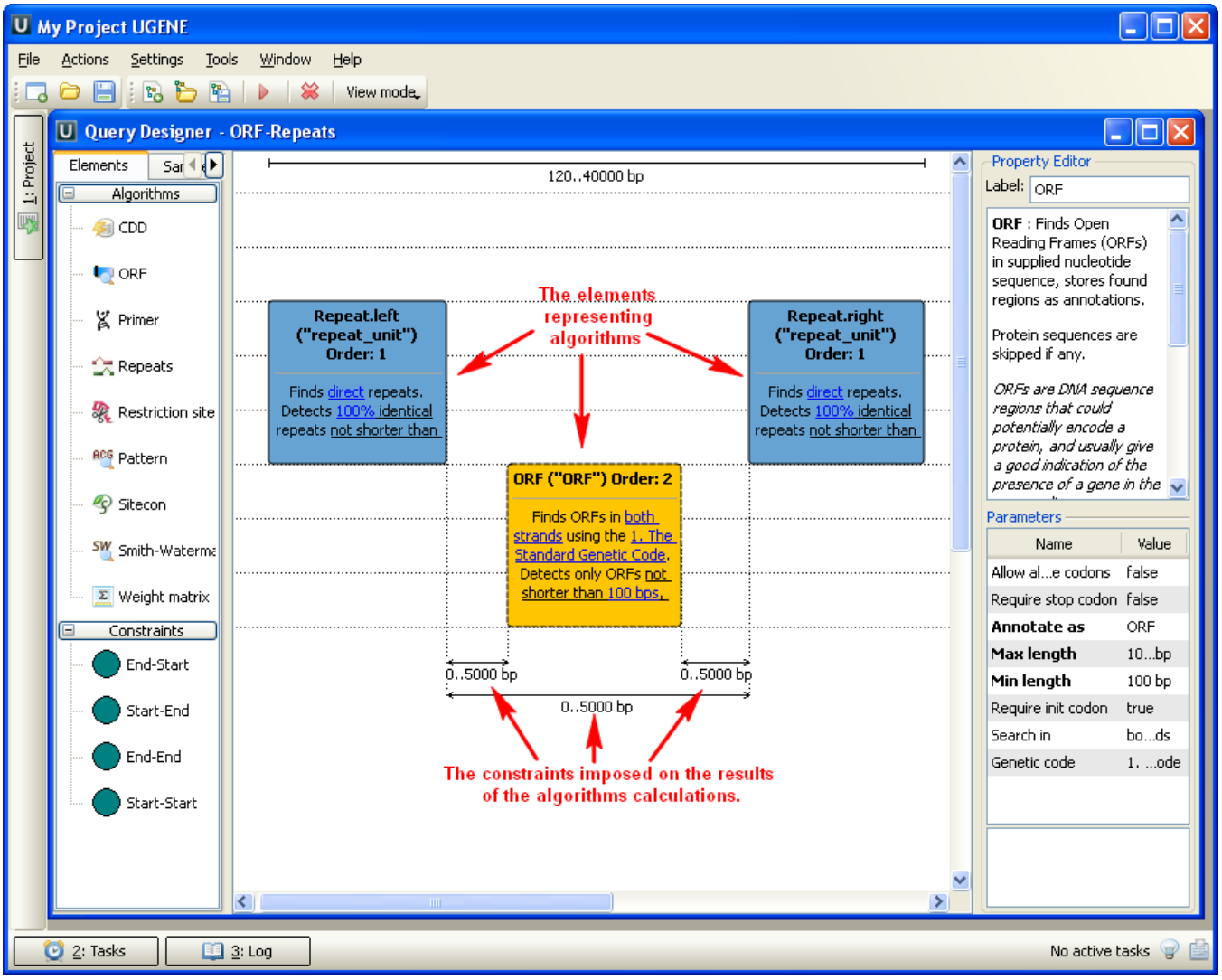
- Pattern Algorithm Element
- SITECON Algorithm Element
- Smith-Waterman Algorithm Element
- Tandem Repeats Algorithm Element
- Weight Matrix Algorithm Element
- HMM2 Algorithm Element

- Constraint Elements
  - End-Start Constraint Element
  - Start-End Constraint Element
  - End-End Constraint Element
  - Start-Start Constraint Element

About the Query Designer

The *Query Designer* allows a molecular biologist to analyze a nucleotide sequence using different algorithms (Repeats finder, ORF finder, Weight matrix matching, etc.) at the same time imposing constraints on the positional relationship of the results obtained from the algorithms.

A user-friendly interface is used to create a schema of the algorithms and constraints.



Alternatively, you can create/edit a schema using a text editor.

When the schema has been created and all its parameters have been set you can run it for a nucleotide sequence. The results are saved as a set of annotations to the specified file in the Genbank format. Also when you have a query designer scheme you can analyze a nucleotide sequence from the sequence view with the help of this schema. Call the *Analyze->Analyze with query schema* context menu item for this.

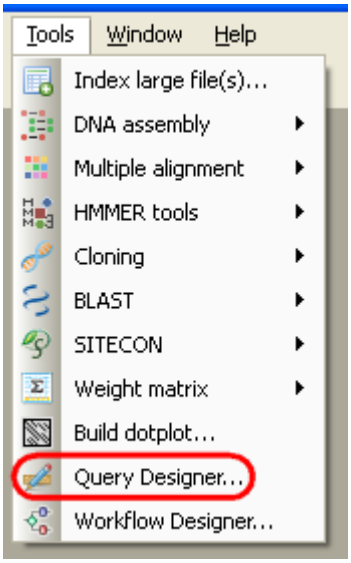
Query Designer Introduction

This chapter describes the *Query Designer* key elements and provides an example on how to create and run a simple *schema*.

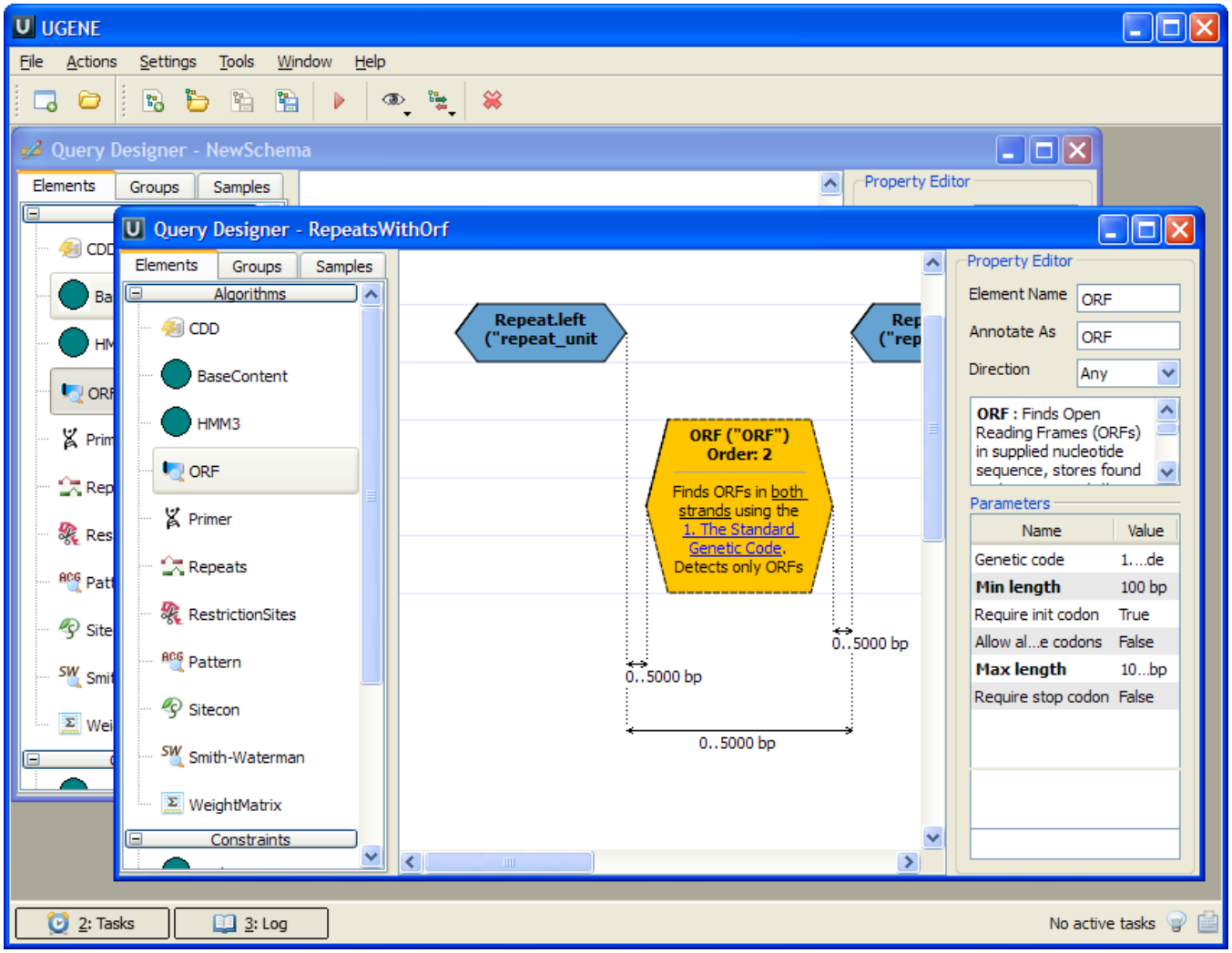
- Launching the Query Designer
- Terminology
  - Query Designer Window Components
  - Schema Elements
- How to Create and Run Schema

Launching the Query Designer

To launch the *Query Designer* select the *Tools Query Designer* item in the main menu:



The tool provides multi-window user interface, so you can open and use at the same time as many *Query Designer* windows as you need.



Terminology

- Query Designer Window Components
- Schema Elements

Query Designer Window Components

Each *Query Designer* window consists of:

Palette

The palette of the *elements*.

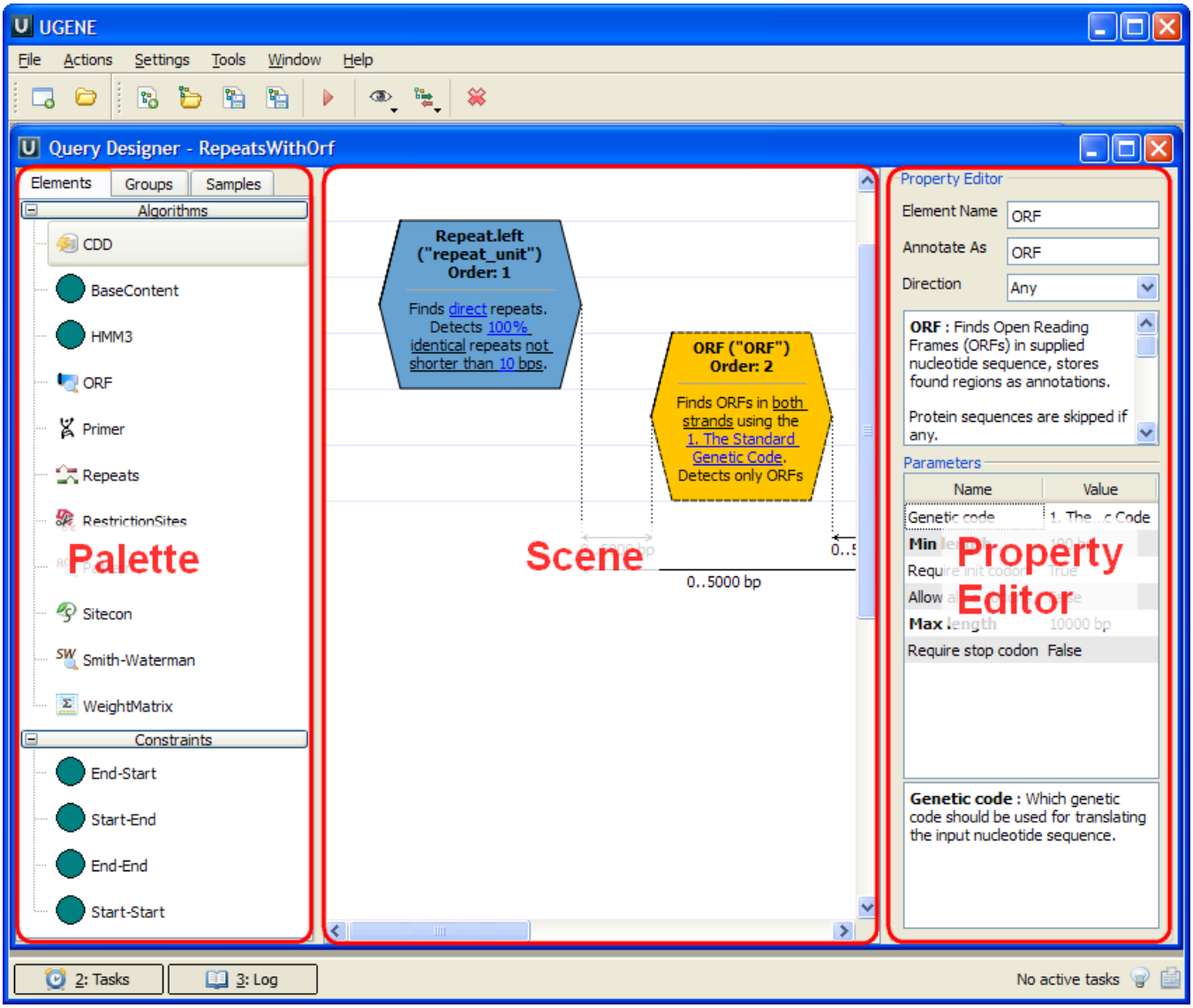
Scene

The main drawing scene is the place where *elements* are constructed into a *schema*.

Property Editor

Provides information about a currently selected *element* and allows configuring it.

On the image below you can see these components in a maximized *Query Designer* window:



All these components are resizable and can be adjusted to individual needs.

Schema Elements

The *Scene* is initially empty and you start with creating a *schema* on it:

schema

A schema is a visual representation of the query that would be used to analyze a nucleotide sequence. It consists of *elements*.

element

An element of a *schema*. There are two types of elements: *algorithm elements* and *constraint elements*.

algorithm element

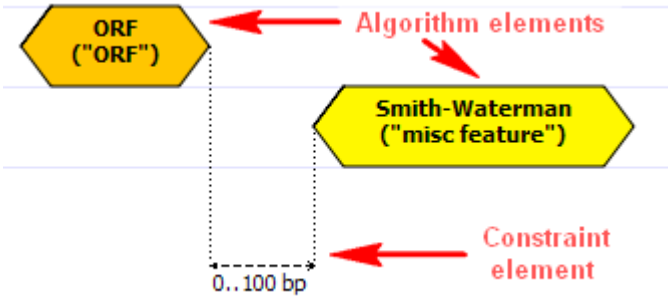
An element of a *schema* that represents an algorithm used to analyze a nucleotide sequence.

constraint element

An element of a *schema* used to impose constraints on the results obtained from *algorithm elements*.

Your first step is to *add necessary algorithm elements* to the *Scene* by dragging them to the *Palette*.

The next step is to *add constraints* on the algorithms results.

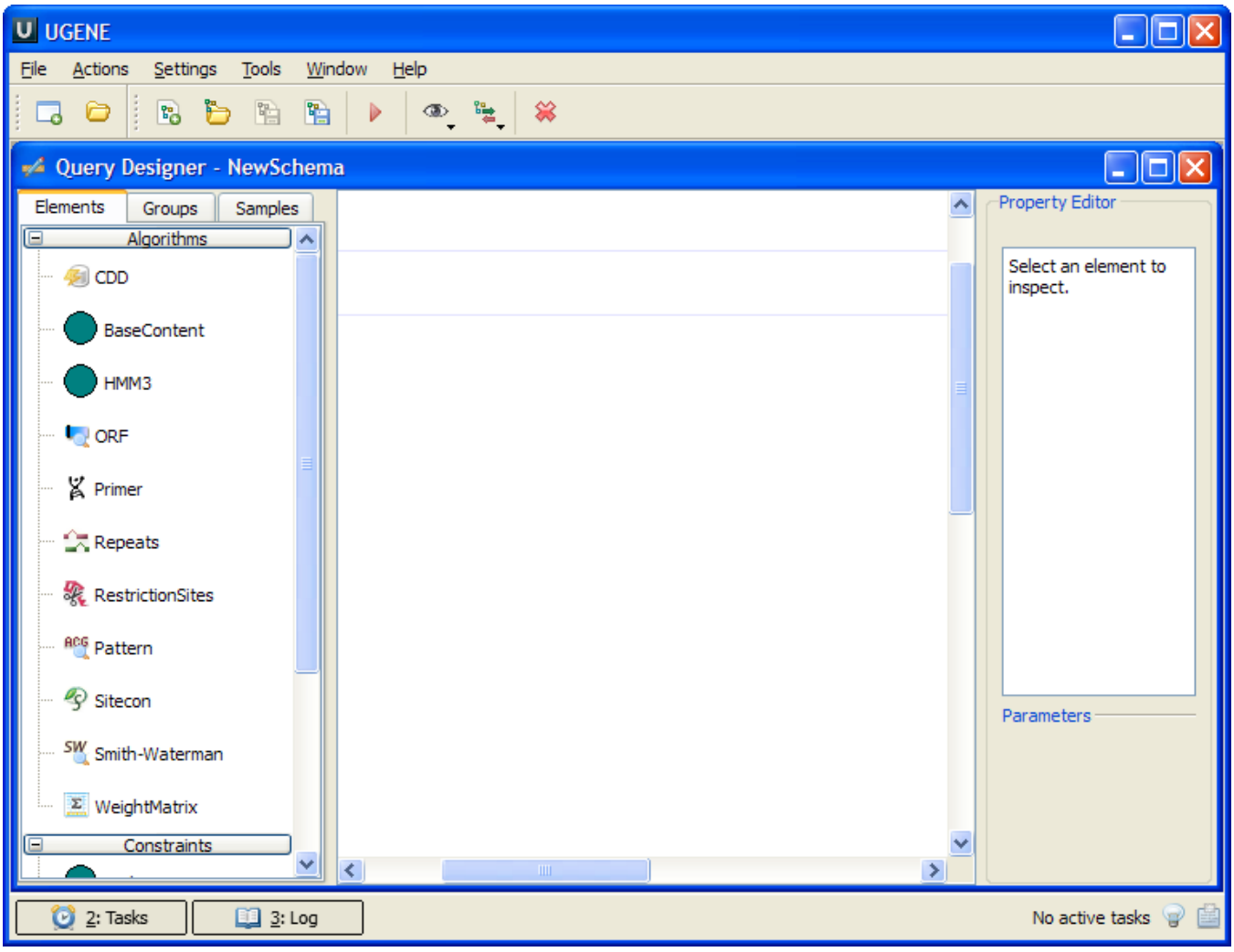


Read *Manipulating Element* chapter to learn the details or check the section below on *how to create and run a schema*.

How to Create and Run Schema

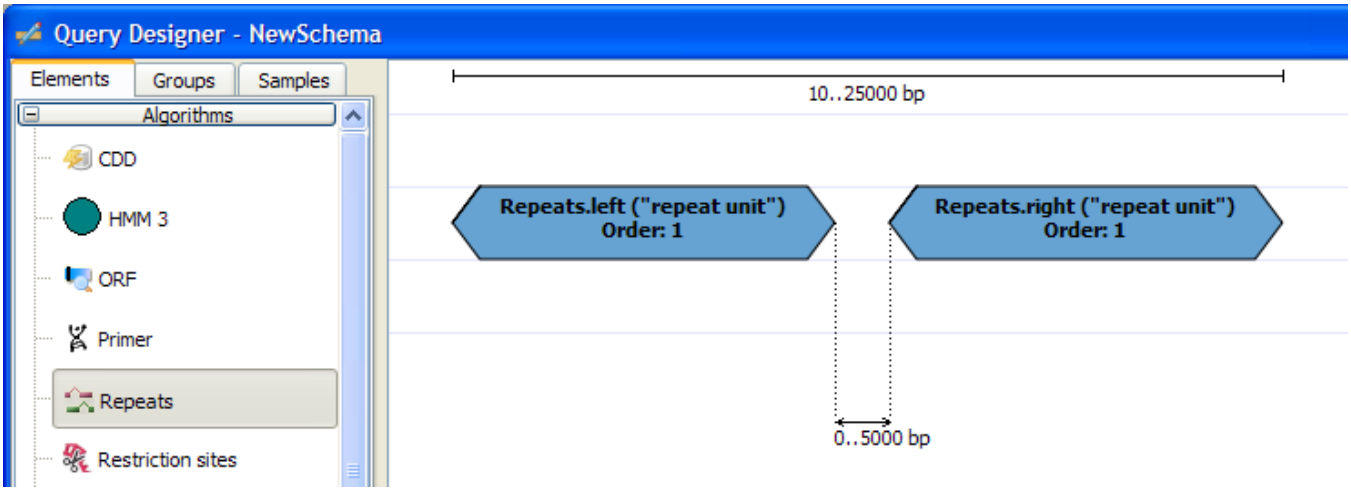
- Select the *Tools Query Designer* item in the main menu.

Result: The Query Designer window appears:



- Drag the *Repeats* element from the *Palette* to the *Scene*.

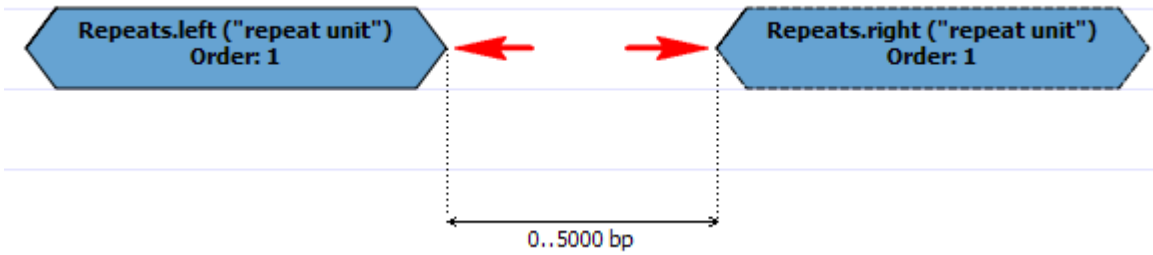
Result: The *Repeats* element subunits are presented on the *Scene*:



Note, that by default minimum distance between left and right repeats is 0bp, the maximum distance is 5000bp.

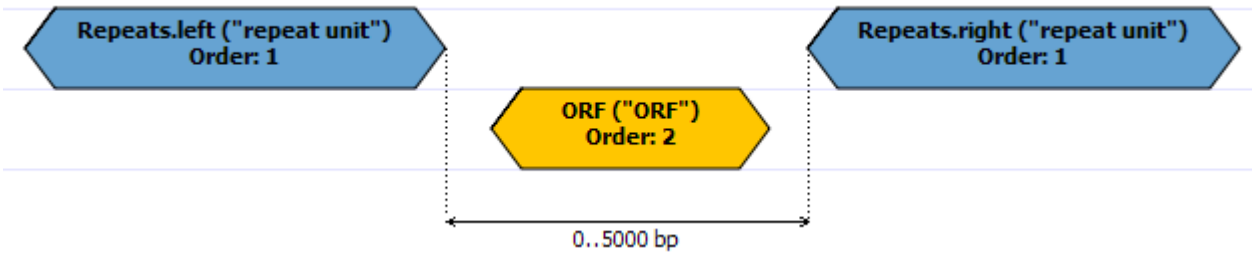
- Slide the *Repeats* element subunits apart.

Result:

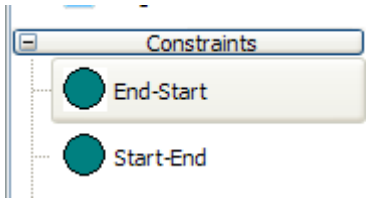


- Drag the *ORF* element from the *Palette* and drop it between the repeats.

Result: The *ORF* element is presented on the *Scene*:

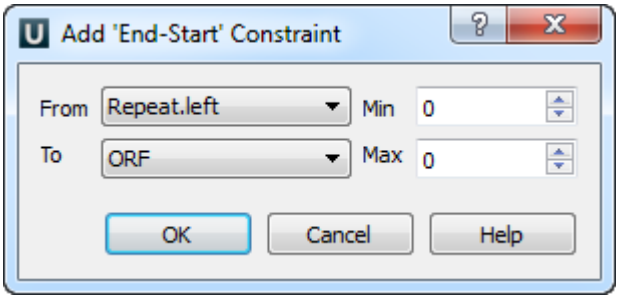


- Find the *End-Start* constraint on the *Palette*:



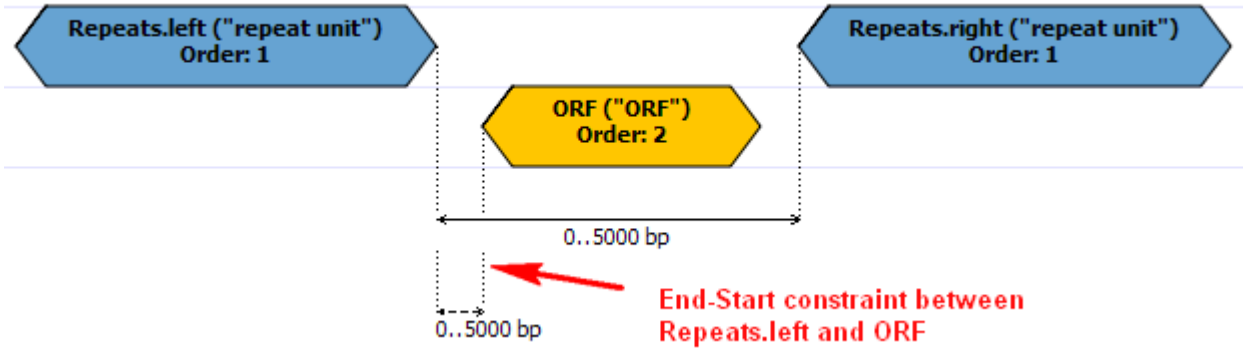
And drag it between the left *Repeats* element and *ORF*.

Result: The dialog appears:



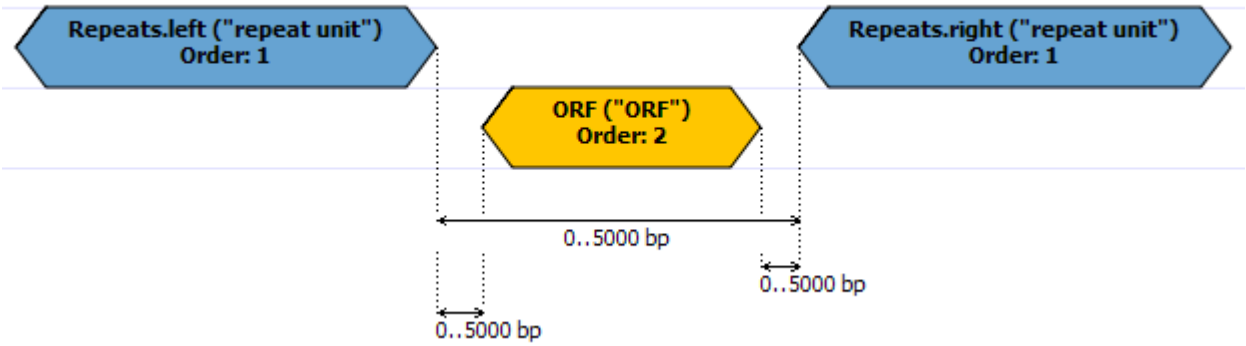
- Check that *From* equals to *Repeats.left* and *To* equals to *ORF*. Set *Max* to 5000. Press the *OK* button.

Result: The constraint has been added to the elements:

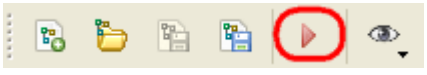


- Repeat steps 5-6 for *ORF* element and the right *Repeats* subunit.

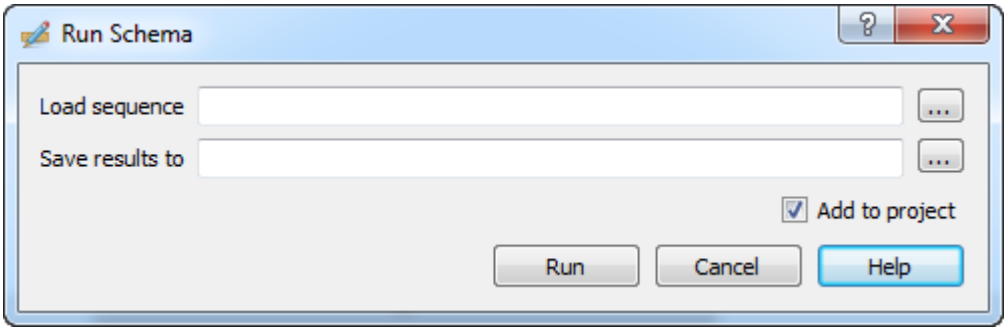
Result: The schema looks as follows:



- Press the *Run Schema* button on the toolbar:

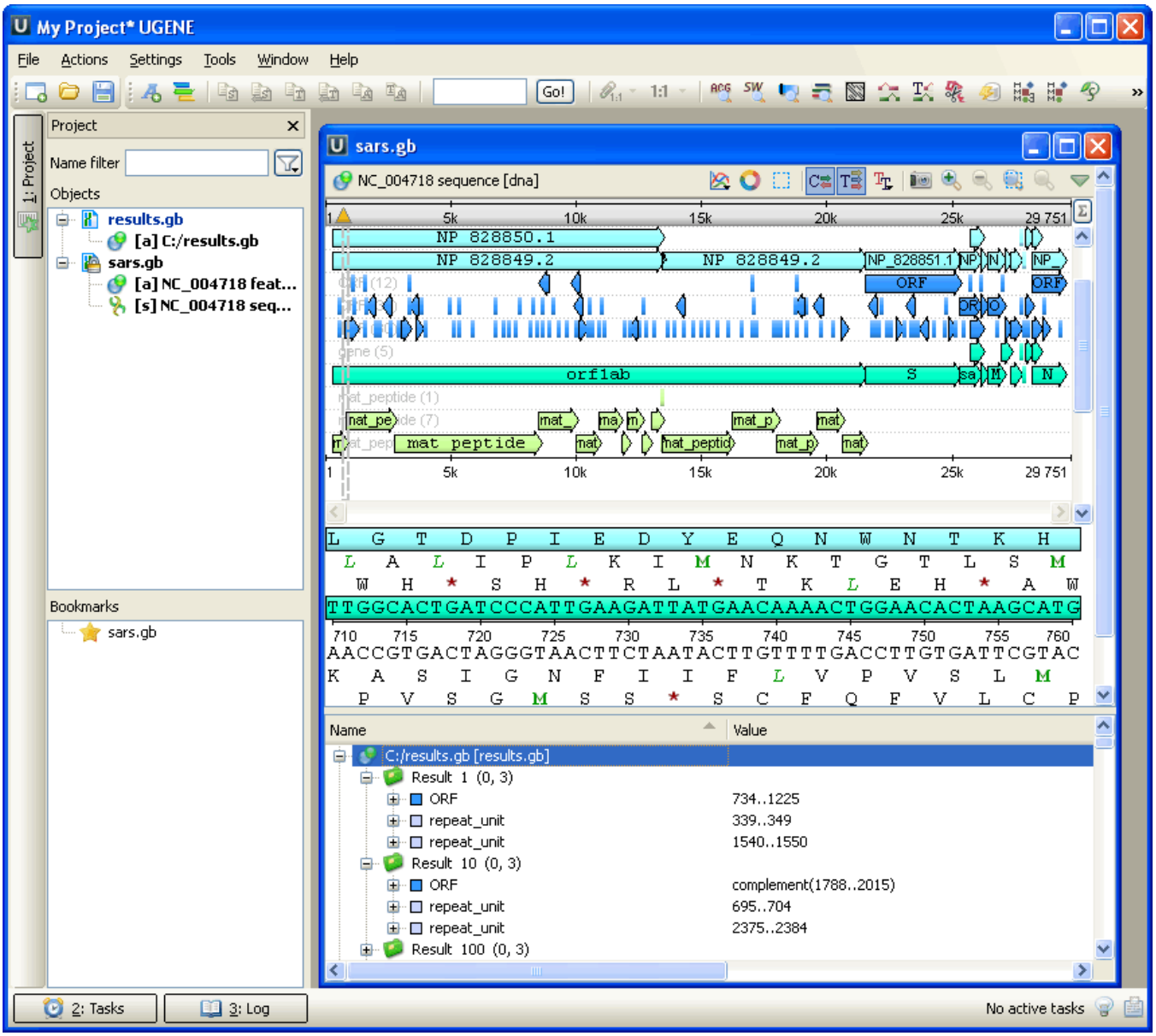


Result: The *Run schema* dialog appears:



- Browse for the sequence to analyze (the *Load sequence* field) and for a GenBank file to save results to (the *Save results to* field). Click the *Run* button.

Result: Both the sequence and the file with annotations are added to the current project. The sequence appears in the Sequence View window, for example:



To learn more about the Sequence View read main UGENE User Manual.

The schema described in this example is also available as sample schema. Select the *Samples* tab on the *Palette* and double-click the *ORF-Repeats* to open the sample schema.

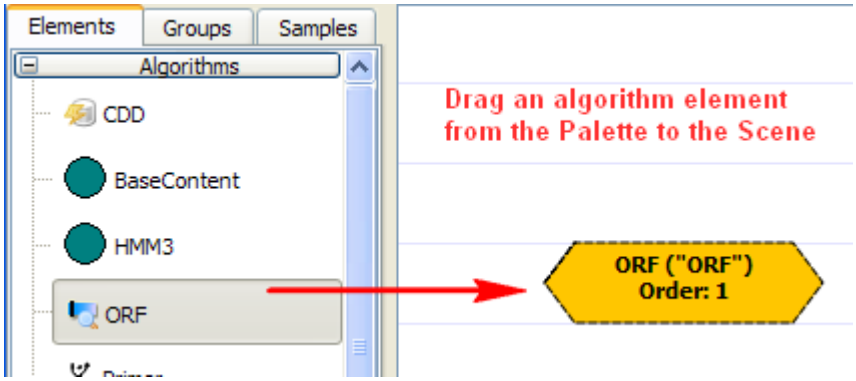
Manipulating Query Designer Element

This chapter describes in more detail how you can manipulate *elements* using the GUI.

- Adding Algorithm Element
- Adding Constraint Element
- Renaming Algorithm Element
- Resizing and Moving Algorithm Element
- Managing of Elements Parameters
- Changing Algorithm Element Appearance
- Deleting Query Designer Element

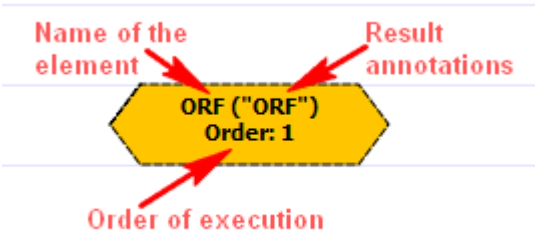
Adding Algorithm Element

To add an *algorithm element* drag it from the *Palette* to the *Scene*. Or you can just click on the element on the *Palette* and then click somewhere on the *Scene*.

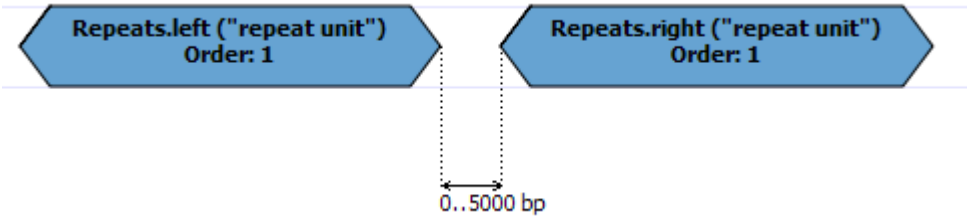


If the default elements appearance hasn't been *modified*, on the added element you can see:

1. The element name
2. The result annotations name, enclosed in parentheses
3. The order of the algorithm execution



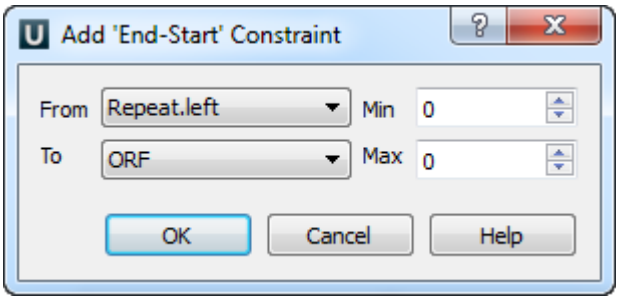
Some elements consist of two subunits. In this case the subunits are marked as **.left** and **.right**:



Adding Constraint Element

A constraint can be imposed on any two *algorithm elements*.

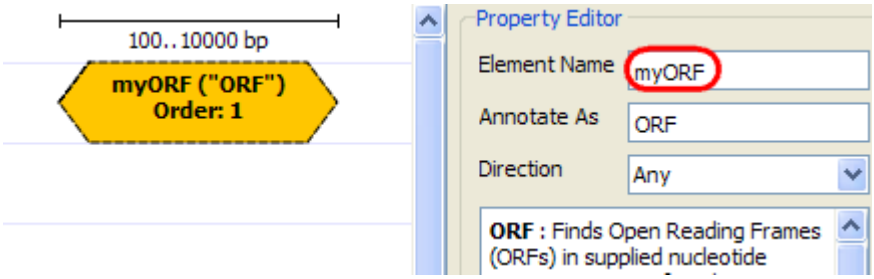
To add a *constraint element* drag it from the *Palette* to the *Scene*. The corresponding dialog appears:



Once the *algorithm elements* are selected and the *OK* button is pressed in the dialog, the *constraint element* is added to the *Scene*. From now the constraint is binded to the *algorithm elements* and it is not possible to modify this assignment.

Renaming Algorithm Element

To rename an *algorithm element*, select it and edit the *Element name* field in the *Property Editor*.



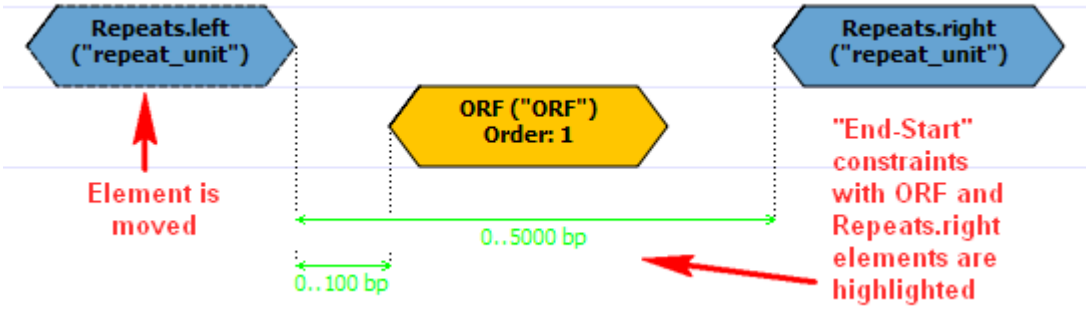
The name can contain only english letters, numbers, '_' (underscore) and '-' (hyphen) characters.

Resizing and Moving Algorithm Element

An *algorithm element* can be moved around on the *Scene* by dragging it and can be resized by dragging its borders.

But if some constraints are imposed on the element, then the element can only be moved / resized within the bounds of the constraints.

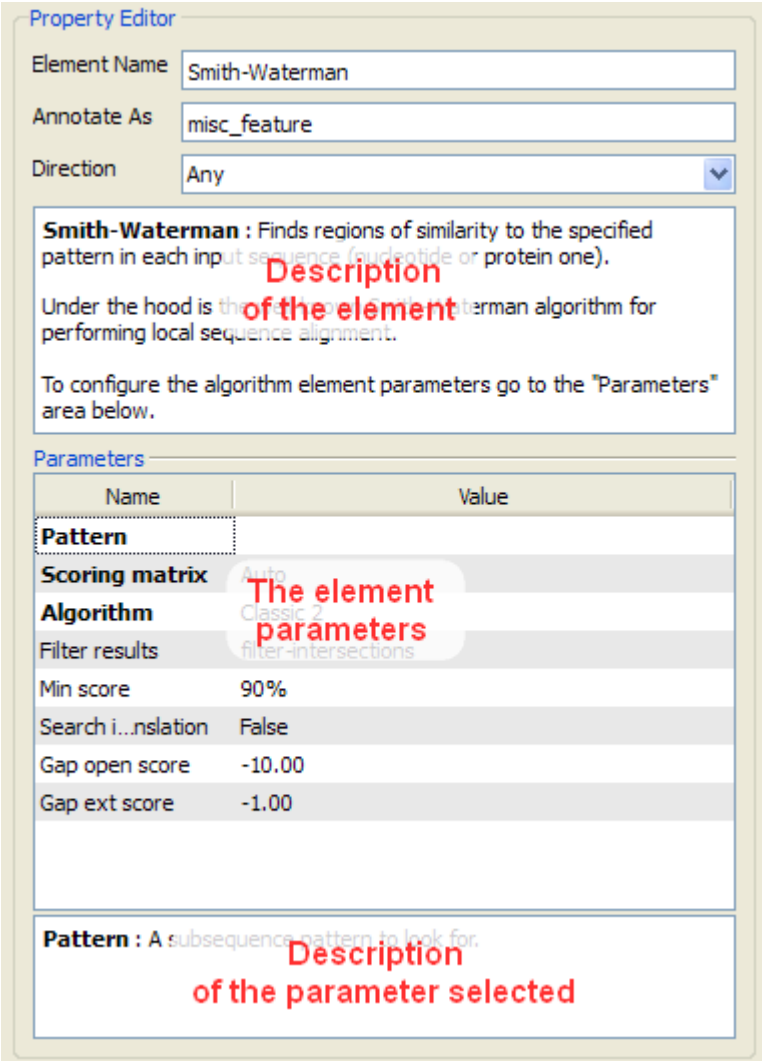
On the image below you can see constraints highlighted in green when an element is moved:



Managing of Elements Parameters

The parameters values can be edited for the both *algorithm* and *constraint* elements.

When you select an *element* on the *Scene*, it's description and parameters appear in the *Property Editor*.



There are some common parameters:

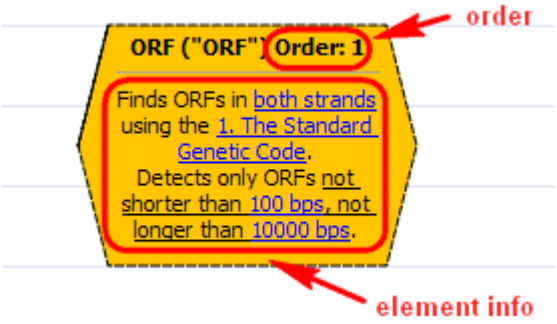
- *Element Name* sets the name of the current element.
- *Annotate As* sets the name of the result annotations.
- *Direction* specifies the direction of the current element relatively to other elements in the schema (i.e. either the result must be searched in the current strand of the input sequence or in the reverse complementary strand). To learn more read *Managing Strands*.

Other parameters can be found in the *Parameters* area. Depending on the parameter in question, the value is selected either from a drop-down list or a spinbox, etc.

Changing Algorithm Element Appearance

You can select to either show or hide an *algorithm element* **info** and **order** on the *Scene* by checking / unchecking the *Show element info* and *Show order* items in the *View Mode* menu (to open the menu select either *View Mode* toolbar button or select *Actions* *View Mode*).

On the image below you can see an ORF element when the both items are checked:



See also the *Setting Order of Algorithms Execution* section.

Deleting Query Designer Element

To delete one or more *algorithm* or *constraint* elements from a *schema*, select them on the *Scene* and press the Delete key.

Alternatively, you can use the *Actions Delete* item in the main menu or the *Delete* toolbar button.

Manipulating Schema

This chapter describes how to manipulate a *schema* using the GUI: how to create, to load, to save and, finally, to run one and so on.

- Creating New Schema
- Loading Schema
- Saving Schema
- Changing Schema Appearance
- Setting Order of Algorithms Execution
- Managing Strands
  - Element Direction in Schema
  - Querying Sequence Strands
- Running Schema from the Query Designer

Creating New Schema

A new *schema* is created each time you *launch the Query Designer*.

To create a new *schema* from the *Query Designer* select either the *Actions New Schema* item in the main menu or the *New Schema* toolbar button.

Loading Schema

To load an existing *schema* from the *Query Designer*, select the *Actions Load Schema* item in the main menu or use the *Load schema* toolbar button.

The *Load Schema* dialog appears. Browse for the required schema file with the *.uql extension.

Saving Schema

To save a *schema* from the *Query Designer* select the *Actions Save Schema* item in the main menu or the *Save Schema* toolbar button.

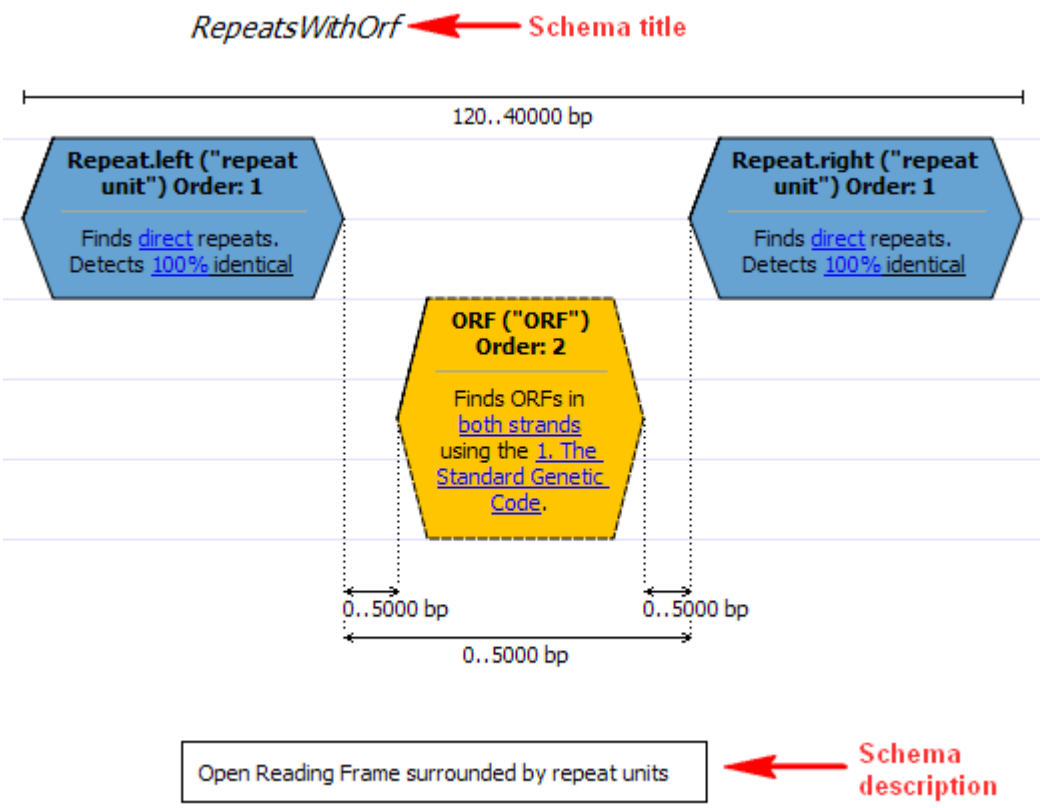
When you save a schema for the first time the *Save Schema* dialog appears where you should input a name and browse for a location of the schema file. The schema is saved to a file with the *.uql extension.

To save a schema to a different file select the *Save Schema As* item on the toolbar or in the *Actions* main menu.

Changing Schema Appearance

You can select to either show or hide a *schema* **title** and **description** on the *Scene* by checking / unchecking the *Show title* and *Show description* items in the *View Mode* toolbar menu.

On the image below the both **title** and **description** are displayed:



Selecting an element direction changes its appearance. See *Managing Strands* for details.

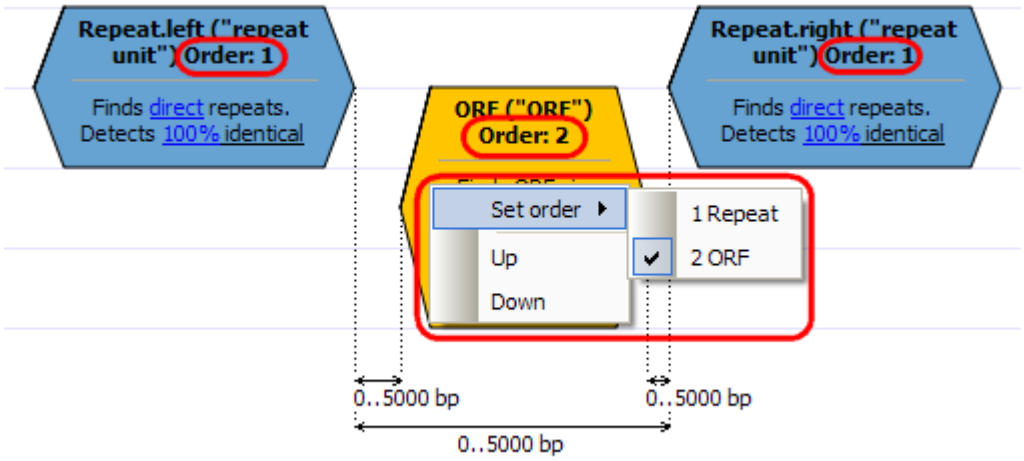
Setting Order of Algorithms Execution

By default, the algorithms are executed in the same order as corresponding *algorithm elements* have been added to the *schema*:

1. The element with order 1 is executed.
2. The element with order 2 is executed. It uses the results obtained from the step 1.
3. And so on.

As each following algorithm element uses the results obtained from the previous element, setting the order can affect efficiency. For example, the less results are obtained on the first step, the faster the second algorithm is executed.

To change the order use the *Set order* submenu or *Up* / *Down* items in an *algorithm element* context menu.



Managing Strands

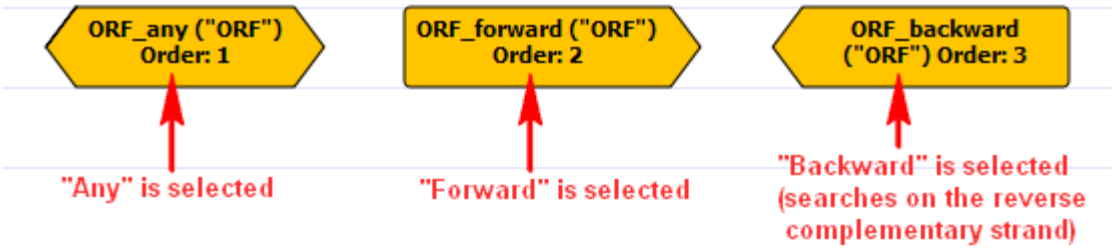
- Element Direction in Schema
- Querying Sequence Strands

Element Direction in Schema

For some *algorithm elements* (e.g. *ORF*) the *Direction* parameter is available in the *Property Editor*. It specifies the direction of the current element relatively to other elements in the schema and can take the following values:

- *Direct* — specifies to search the results for the element in the current strand.
- *Forward* — specifiesd to search the results for the element in the reverse complementary strand.
- *Any* — the results for the element are seached in the both strands.

Notice that an element changes it's appearance on the *Scene* when different values are selected:

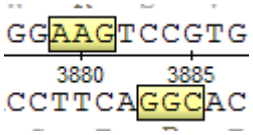


Example1:

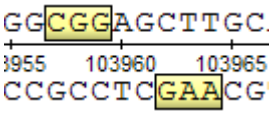
Create the following schema:

1. The Smith-Waterman *algorithm element* with *AAG* patern and the *Forward* direction.
2. The Smith-Waterman element with *CGG* pattern and the *Backward* direction.
3. Add a *constraint* to these elements.
4. Run the schema for a sequence.

By default, if the *Query Sequence Mode* hasn't been *modified*, the followings results will be found:



and



Querying Sequence Strands

As the **Example1** above shows, both sequence strands are queried by default.

To modify this behavior select the *Actions* *Query Sequence Mode* item in the main menu or *Query Sequence Mode* toolbar button. You can choose between the following values:

- *Direct strand* — the search is performed for the direct strand only.

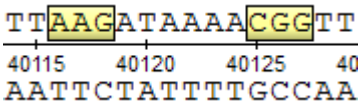
  Note, that the results can still be found in the complement strand if you have set the *Any* or *Backward* direction for an element.
- *Reverse complementary strand* — the search is performed for the reverse complementary strand.
- *Both strands* — the search is performed for the both strands.

Example2:

Create the following schema:

1. The Smith-Waterman *algorithm element* with *AAG* patern and the *Forward* direction.
2. The Smith-Waterman element with *CGG* pattern and the *Forward* direction.
3. Add a *constraint* to these elements.
4. Set the *Query Sequence Mode* to *Direct strand*.
5. Run the schema for a sequence.

Only the following results will be found:



Running Schema from the Query Designer

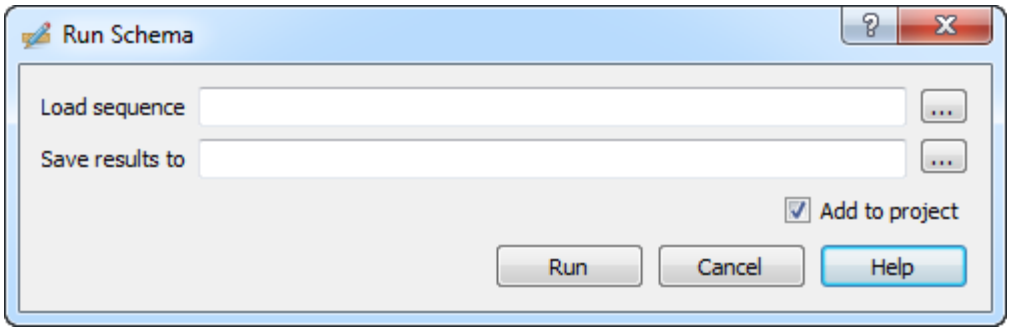
As soon as a *schema* construction is finished:

- all required *algorithm elements are added*
- all required *constraints are set*
- all required *parameters are set*

the schema can be run.

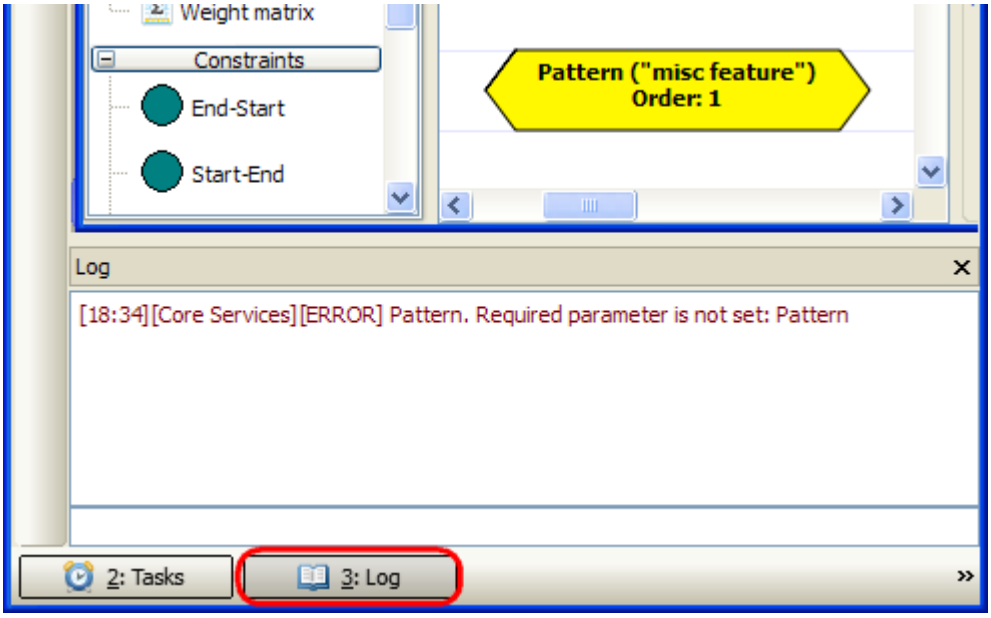
To run the schema select the *Actions* *Run Schema* item in the main menu or select the *Run Schema* toolbar button.

The *Run Schema* dialog will appear:



Select the *Sequence* to analyze with the schema and browse for the file in GenBank format to *Save results to*. Optionally, check the *Add to project* button to add the created result file to the active project. Press the *Run* button.

In case of an error check the Log View:



Note also that the schema execution task can be canceled from the Task View.

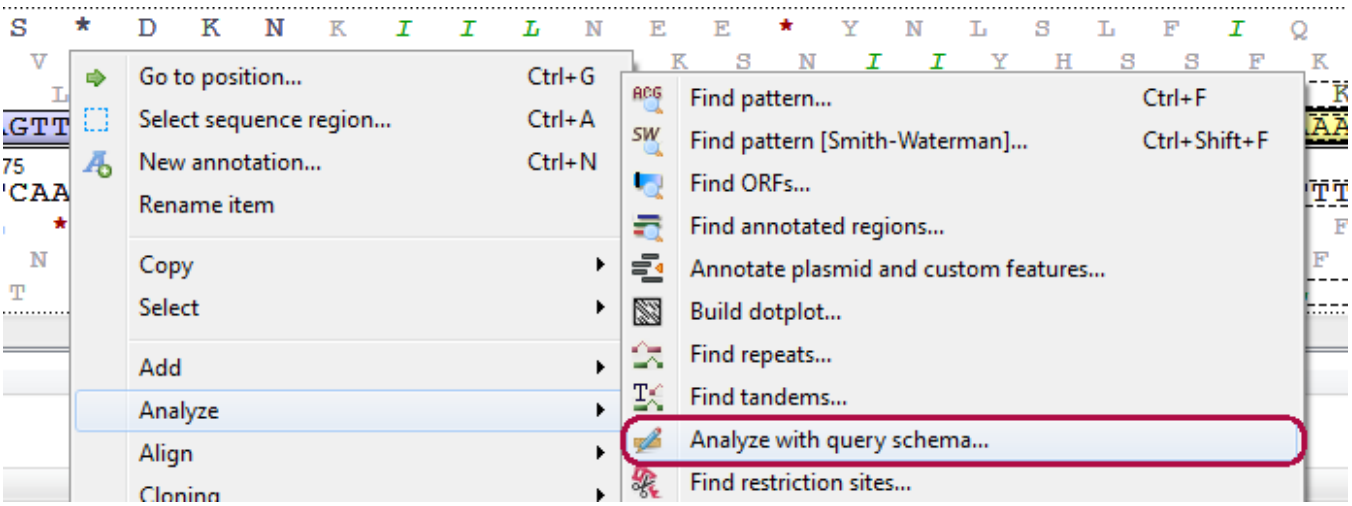
Running Schema from the Sequence View

Prepare a query *schema* and *save* it to a file.

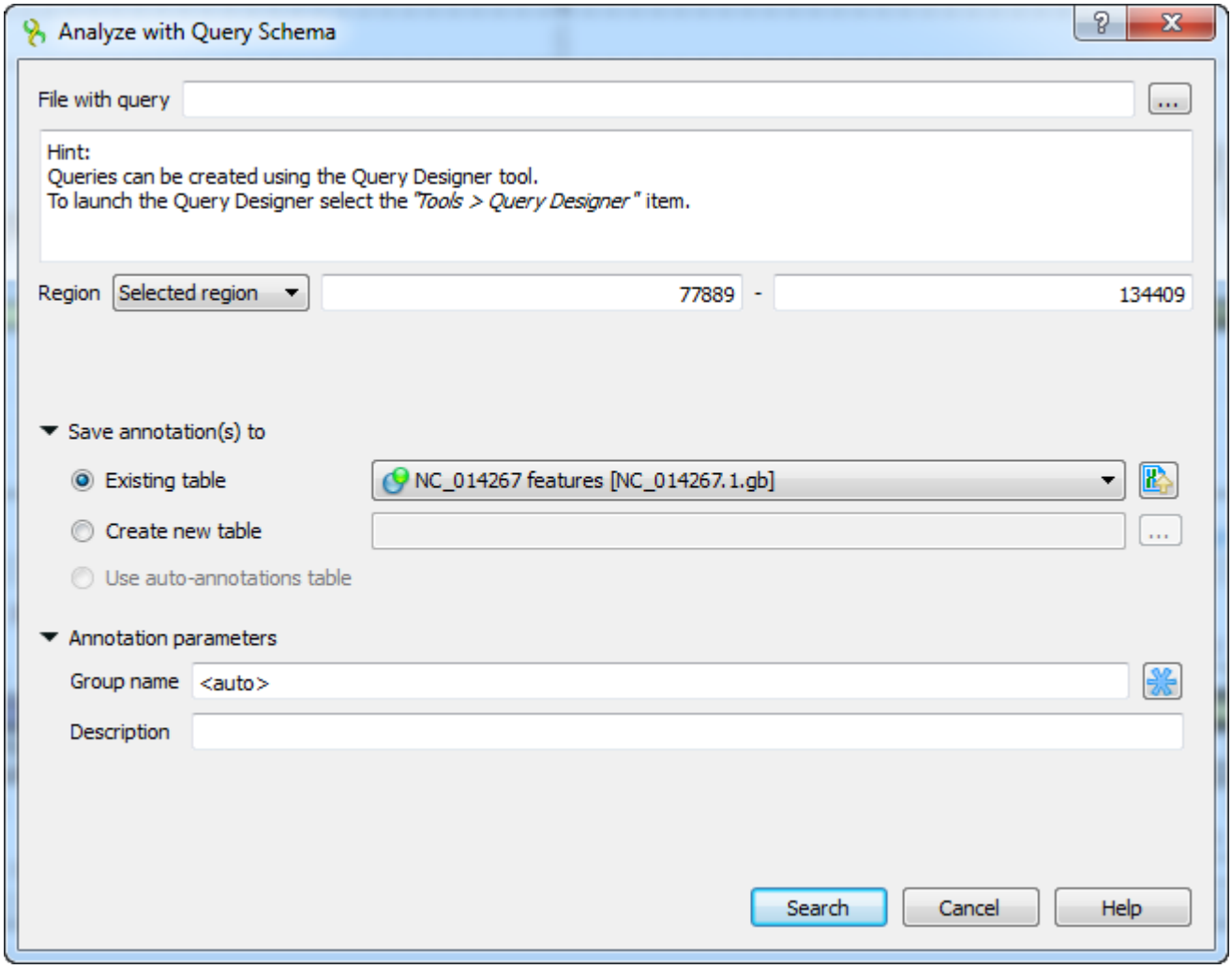
Open a nucleotide sequence that you want to analyze with this query schema. You can see the sequence displayed in the **Sequence view**.

> To learn more about the **Sequence view** read the main UGENE User Manual.

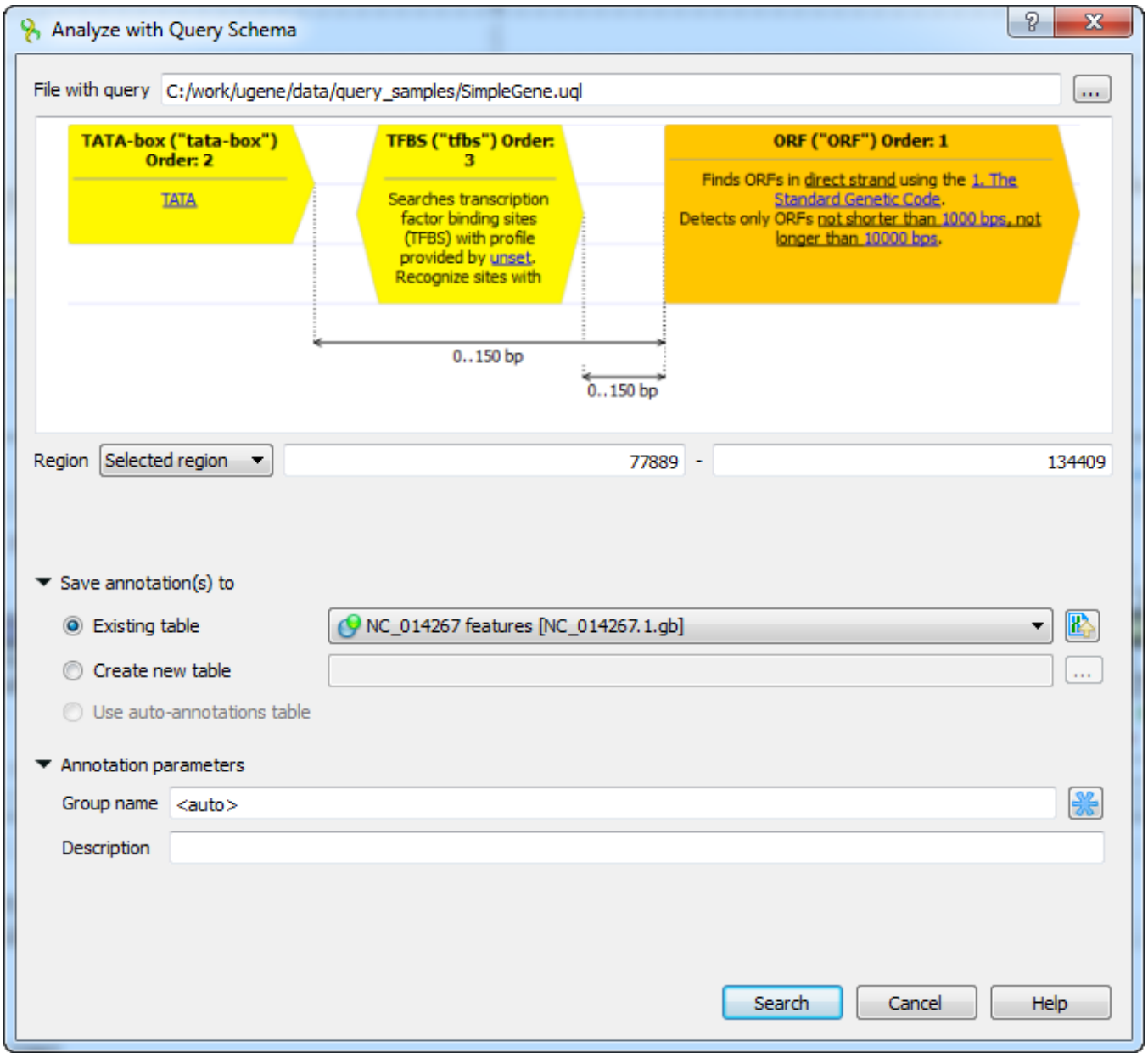
Select the *Analyze Find query designer pattern* item in the *Actions* main menu or in the context menu:



The *Analyze with query schema* dialog appears:



Browse for the file with a query schema. The selected schema preview appears in the dialog, for example:



You can also adjust other parameters:

Region — the sequence range to analyze with the query schema, you can select:

- *Whole sequence* — to analyze the whole sequence.
- *Selected range* — to analyze the currently selected sequence region. This item is disabled if there is no region selected.
- *Custom range* — to specify manually a range to analyze.

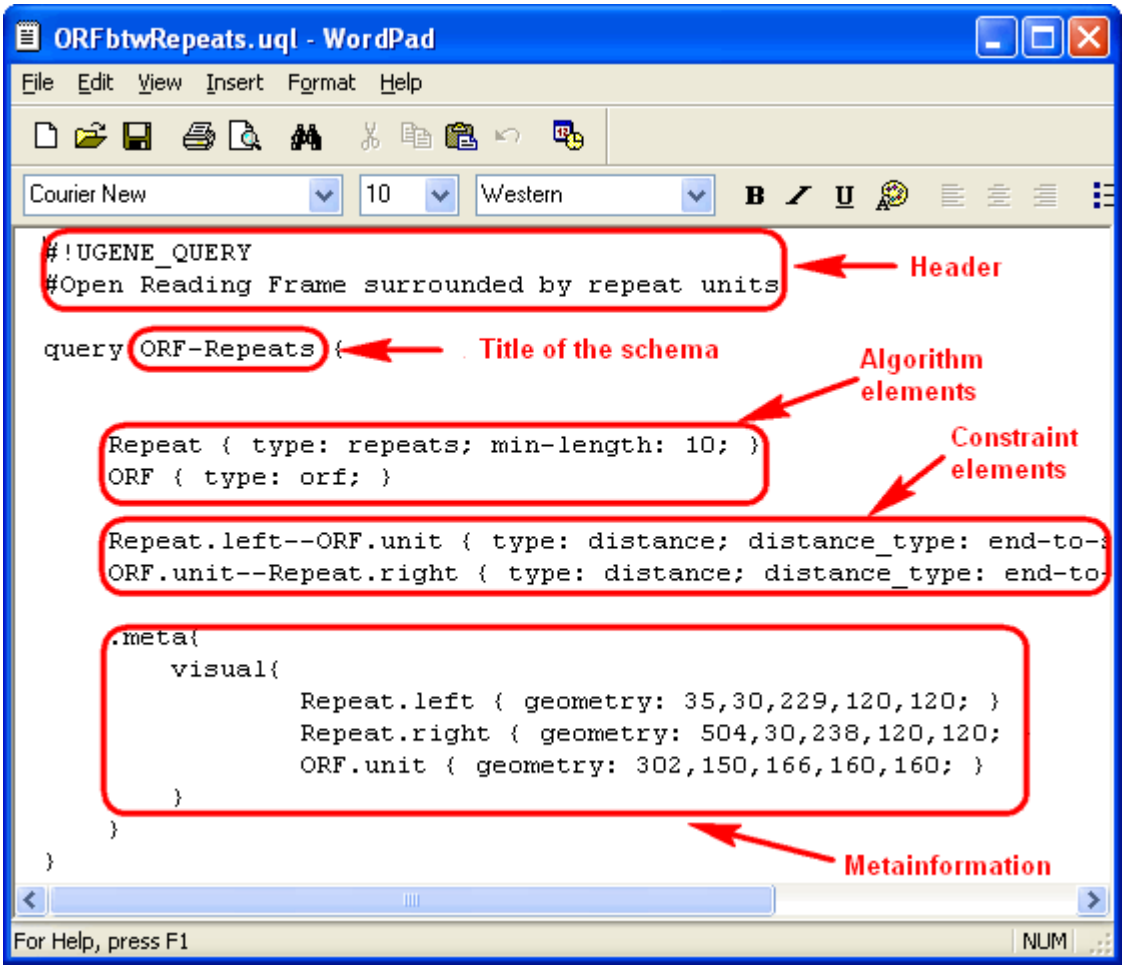
In the *Save annotation(s) to* group you can set up a file to store annotations. It could be either an existing annotation table object, a new annotation table or auto-annotations table (if it is available).

In the *Annotation parameters* group you can specify the name of the group. If the group name is set to <auto> UGENE will use the group name as the name for the group. Also you can add a description in the corresponding text field.

Query Designer Schema File Format

Using the GUI is not the only way to create / edit a *schema*. As specified *earlier* a schema is saved to a file with *.uql extension. This chapter describes the format of the file and explains how you can create / edit a schema file using a text editor.

The best way to learn schema file format is to study an existent *.uql file. For example, let's take the sample schema file described in the *example*. Open the file in a text editor. On the image below you can see the file opened in the MS WordPad.



The file consists of the header and the body. Check the description of each part below.

- Header Query Designer Element
- Body Query Designer Element
  - Element Description
    - Algorithm Element Description
    - Constraint Element Description
  - Metainformation Query Designer Element

Header Query Designer Element

The header consists of the following key string:

```
#!UGENE_QUERY
```

And the description of the schema:

```
# Write here the description of the schema.
```

Body Query Designer Element

The body begins with the **query** keyword followed by the title of the schema and curly braces:

```
query schema_title {
```

```
  # Description of algorithm and constraint elements

  # Metainformation

}
```

Within the body you describe the required *algorithm* and *constraint* elements. The metainformation block can also be presented automatically added by the *Query Designer*.

- Element Description
  - Algorithm Element Description
  - Constraint Element Description
- Metainformation Query Designer Element

Element Description

An *element* description consists of the element name and a set of parameters enclosed in curly braces. A parameter and the value are separated by ':', different parameters are separated by ';':

```
element_name {

    parameter1: value1;
    parameter2: value2;
    ...
}
```

A parameter's type can be one of the following:

string

A string.

numeric

A number.

boolean

A boolean data type.

Available values are *true* / *false*, *0* / *1* and *yes* / *no*.

- Algorithm Element Description
- Constraint Element Description

Algorithm Element Description

When you describe an algorithm element you must specify the **element_name** and the element type.

The element name corresponds to the element's *Element Name* in the GUI. The name can contain only english letters, numbers, '_' (underscore) and '-' (hyphen) characters.

See, for example, the description of an ORF element:

```
myORF {

   type: orf;

   # Other parameters
}
```

To find the type and the parameters available for the required algorithm element refer to *Algorithm Elements* chapter.

Constraint Element Description

When you describe a constraint element the **element_name** consists of two parts separated by two hyphens.

```
part1--part2
```

Each part represents one of the algorithms the constraint is imposed on.

If the algorithm is presented as one element on a schema (like *ORF*, *Pattern*) the algorithm's part has format:

```
algorithm_element_name.unit
```

If the algorithm is presented as two subelements on a schema (like *Repeats*, *Primer*) the algorithm's part has format:

```
algorithm_element_name.left
```

or:

```
algorithm_element_name.right
```

depending on the subelement the constraint is imposed on.

Also you should specify the constraint type parameter (currently the only available type is *distance*):

```
type: distance;
```

And specify one of the distance types, for example:

```
distance-type: end-to-start;
```

Example1: The constraint is imposed on **myORF** and **myPattern** algorithm elements:

```
myORF.unit--myPattern.unit {

   type: distance;
   distance-type: start-to-start;

   # Other parameters
}
```

Example2: The constraint is imposed on **myORF** algorithm element and the left **myRepeats** algorithm subelement:

```
myORF.unit--myRepeats.left {

   type: distance;
   distance-type: start-to-end;

   # Other parameters
}
```

The available constraint elements are described in the *Constraint Elements* chapter.

Metainformation Query Designer Element

The metainformation is added when you create / edit the schema with the *Query Designer*. It is not required for running the schema and is skipped when the schema is created manually, for example:

```
#Open Reading Frame surrounded by repeat units
```

```
query ORF-Repeats {

    Repeat {
        type: repeats;
        min-length: 10;
    }

    ORF { type: orf; }

    Repeat.left--ORF.unit {
        type: distance;
        distance_type:
        end-to-start;
        min: 0;
        max: 5000;
    }
    ORF.unit--Repeat.right {
        type: distance;
        distance_type: end-to-start;
        min: 0;
        max: 5000;
    }
}
```

Query Elements

- Algorithm Elements
    - CDD Algorithm Element
    - Base Content Algorithm Element
    - Gc Content Algorithm Element
    - HMM3 Algorithm Element
    - ORF Algorithm Element
    - Primer Algorithm Element
    - Repeats Algorithm Element
    - Restriction Sites Algorithm Element
    - Pattern Algorithm Element
    - SITECON Algorithm Element
    - Smith-Waterman Algorithm Element
    - Tandem Repeats Algorithm Element
    - Weight Matrix Algorithm Element
    - HMM2 Algorithm Element
- Constraint Elements
    - End-Start Constraint Element
    - Start-End Constraint Element
    - End-End Constraint Element
    - Start-Start Constraint Element

Algorithm Elements

- CDD Algorithm Element
- Base Content Algorithm Element
- Gc Content Algorithm Element
- HMM3 Algorithm Element
- ORF Algorithm Element
- Primer Algorithm Element
- Repeats Algorithm Element
- Restriction Sites Algorithm Element
- Pattern Algorithm Element
- SITECON Algorithm Element
- Smith-Waterman Algorithm Element
- Tandem Repeats Algorithm Element
- Weight Matrix Algorithm Element
- HMM2 Algorithm Element

CDD Algorithm Element

When the *element* is used the input nucleotide sequence is translated into 6 amino sequences. The translated sequences are used to query the NCBI Conserved Domain Database (CDD).

Parameters in GUI

Parameter	Description	Default value
Annotate As	Name of the result annotations.	CDD result
Expected value	It is also called E-value. Describes the number of hits one can “expect” to see by chance when searching a database of a particular size.	10
Min length	Minimum result length.	50bp
Max length	Maximum result length.	5000bp
Pattern	Filters the result annotations: the qualifier value must contain the specified pattern value.	No default value.

Parameters in Schema File

Type: CDD

Parameter	Parameter in the GUI	Type
key	**Annotate As**	*string*
evalue	**Expected value**	*numeric*
min-length	**Min length**	*numeric*
max-length	**Max length**	*numeric*
pattern	**Pattern**	*string*

Base Content Algorithm Element

Searches regions in a sequence that contain a specified percentage of a certain base.

Parameters in GUI

Parameter	Description	Default value
Annotate As	Name of the result annotations.	misc_feature
Base	Specifies the base, i.e. A, C, G or T.	You must specify a value!
Percentage	Percentage of the base in a region. The value must be >= 50%.	90%
Min Length	Minimum length of a region. The value must be >= 5.	5 bp
Max Length	Maximum length of a region.	5 bp
Direction	See the description *here*.	Any

Parameters in Schema File

Type: base-content

Parameter	Parameter in the GUI	Type
key	**Annotate As**	*string*
base	**Base**	*string*
percent	**Percentage**	*numeric*
min-len	**Min Length**	*numeric*
max-len	**Max Length**	*numeric*

Gc Content Algorithm Element

Searches regions in a sequence with GC content in the specified range.

Parameters in GUI

Parameter	Description	Default value
Min GC content	Min GC content value in percents.	20%
Max GC content	Max GC content value in percents.	40%
Min length	Minimum length of a region.	50bp
Max length	Maximum length of a region.	1000bp

Parameters in Schema File

Type: find-gc

Parameter	Parameter in the GUI	Type
region-start	**Min GC content**	*numeric*
region-end	**Max GC content**	*numeric*
min-len	**Min length**	*numeric*
max-len	**Max length**	*numeric*

HMM3 Algorithm Element

Searches a sequence for significantly similar sequence matches with one or more profile HMM and saves the results as annotations.

The search is performed using HMMER3 **hmmsearch** tool integrated into UGENE.

Parameters in GUI

General parameters:

Parameter	Description	Default value
Annotate As	Name of the result annotations.	hmm_signal
Profile HMM	Semicolon-separated list of input HMM files.	You must specify a value!
Min Length	Minimum length of a result region.	30
Max Length	Maximum length of a result region.	5000

Parameters controlling reporting threshold:

Reporting thresholds controls which hits are reported.

Parameter	Description	Default value
Use E-value	Filter by E-value if true. Otherwise, filters by score.	True
Filter by High E-value	Reports domains <= this E-value threshold in output (**hmmsearch–domE** option).	1e+1
Filter by Low Score	Reports domains >= this score cutoff in output (**hmmsearch–domT** option).	0.01

Parameters controlling the acceleration pipeline:

HMMER3 searches are accelerated in a three-step filter pipeline: the MSV filter, the Viterbi filter, and the Forward filter. The first filter is the fastest and most approximate; the last is the full Forward scoring algorithm. There is also a bias filter step between MSV and Viterbi.

Parameter	Description	Default value
Max	Turns off all acceleration heuristic filters. This increases sensitivity somewhat, at a large cost in speed.	False
MSV Filter Threshold	P-value threshold for the MSV filter step.	0.02
Viterbi Filter Threshold	P-value threshold for the Viterbi filter step.	0.001
Forward Filter Threshold	P-value threshold for the Forward filter step.	1e-5
No Bias Filter	Turns off composition bias filter. This increases sensitivity somewhat, but can come at a high cost in speed.	False

Other parameters:

Parameter	Description	Default value
No Null2	Turns off the null2 score corrections for biased composition.	False
Number of Sequences	Specifies number of significant sequences. It is used for domain E-value calculations (**hmmsearch–domZ** option).	1 (i.e. one input sequence)
Seed	Random number seed. The default is to use a fixed seed(42), so that results are exactly reproducible. Any other positive integer will give different (but also reproducible) results. A choice of 0 uses a randomly chosen seed.	42

Parameters in Schema File

Type: hmm3

Parameter	Parameter in the GUI	Type
key	**Annotate As**	*string*
min-len	**Min Length**	*string*
max-len	**Max Length**	*string*
hmm-profile	**Profile HMM**	*string*
use-e-val	**Use E-value**	*boolean*
e-val	**Filter by High E-value**	*numeric*
score	**Filter by Low Score**	*numeric*
do-max	**Max**	*boolean*
msv-filter-threshold	**MSV Filter Threshold**	*numeric*
viterbi-filter-threshold	**Viterbi Filter Threshold**	*numeric*
forward-filter-threshold	**Forward Filter Threshold**	*numeric*
no-bias-filter	**No Bias Filter**	*boolean*
no-score-corrections	**No Null2**	*boolean*
seqs-num	**Number of Sequences**	*numeric*
random-generator-seed	**Seed**	*numeric*

ORF Algorithm Element

The *element* searches for open reading frames (ORFs) in the supplied sequence.

Parameters in GUI

Parameter	Description	Default value

Annotate As	Name of the result annotations.	ORF
Direction	See the description *here*.	Any
Allow alternative codons	Allows / disallows ORFs starting with alternative initiation codons, accordingly to the current translation table.	False
Require init codons	Allows / disallows ORFs starting with any codon other than terminator.	True
Require stop codons	Ignores or takes into account boundary ORFs which last beyond the search region.	False
Min length	Ignores ORFs shorter than the specified length.	100bp
Max length	Maximum length of annotation allowed.	10000bp
Genetic code	Genetic code that should be used to translate the input nucleotide sequence.	The standard genetic code

Parameters in Schema File

Type: orf

Parameter	Parameter in the GUI	Type
key	**Annotate As**	*string*
strand	**Direction**	*string* Available values are: • complement • direct • both
alt-start	**Allow alternative codons**	*boolean*
starts-with-init	**Require init codons**	*boolean*
ends-with-stop	**Require stop codons**	*boolean*
min-length	**Min length**	*numeric*
max-length	**Max length**	*numeric*
trans-id	**Genetic code**	*string* Available values are: • “NCBI-GenBank #1” • “NCBI-GenBank #2” • etc.

Primer Algorithm Element

The *element* searches primers against the input sequence.

Parameters in GUI

Parameter	Description	Default value
Annotate As	Name of the result annotations.	top primers
Direction	See the description *here*.	Any
Excluded regions	The regions should be avoided for primer selection.	No default value.
Max repeat mispriming	The maximum allowed weighted similarity with any sequence in Mispriming Library.	12

Max template mispriming	The maximum allowed similarity to ectopic sites in the sequence from which you are designing the primers.	12
Number to return	The maximum number of primer pairs to return.	5
Pair max repeat mispriming	The maximum allowed sum of similarities of a primer pair (one similarity for each primer) with any single sequence in Mispriming Library.	24
Pair max template mispriming	The maximum allowed summed similarity of both primers to ectopic sites in the sequence from which you are designing the primers.	24
Product size ranges	List of product size ranges. Primer first tries to pick primers in the first range. If that is not possible, it goes to the next range and tries again. It continues in this way until it has either picked all necessary primers or until there are no more ranges.	150-200 100-300 301-400 401-500 501-600 601-700 701-850 851-1000
Max 3' stability	The maximum stability for the last five 3' bases of a left or right primer. Bigger numbers mean more stable 3' ends.	9
Targets	If one or more Targets is specified then a legal primer pair must flank at least one of them.	No default value.

Parameters in Schema File

Type: primer

Parameter	Parameter in the GUI	Type
key	**Annotate As**	*string*
excluded_regions	**Excluded region**	*string*
max_mispriming	**Max repeat mispriming**	*numeric*
max_template_mispriming	**Max template mispriming**	*numeric*
num_return	**Number to return**	*numeric*
pair_max_mispriming	**Pair max repeat mispriming**	*numeric*
pair_max_template_mispriming	**Pair max template mispriming**	*numeric*
size_ranges	**Product size range**	*string*
stability	**Max 3' stability**	*numeric*
targets	**Targets**	*string*

Repeats Algorithm Element

The *element* searches for repeats in the input sequence.

Parameters in GUI

Parameter	Description	Default value
Annotate As	Name of the result annotations.	repeat_unit
Algorithm	Algorithm used to search for repeats. Available values are:	Suffix index

	• Suffix index • Diagonals • Auto	
Filter nested	If *True*, filters nested repeats.	True
Identity	Repeats percentage identity.	100%
Inverted	If *True*, searches for inverted repeats.	False
Min length	Minimum length of repeats.	5bp
Max length	Maximum length of repeats.	10000bp
Parallel threads	Number of parallel threads used to execute the calculations.	Auto

Parameters in Schema File

Type: repeats

Parameter	Parameter in the GUI	Type
key	**Annotate As**	*string*
algorithm	**Algorithm**	*string* Available values are: • suffix • diagonals • auto
filter-nested	**Filter nested**	*boolean*
identity	**Identity**	*numeric*
invert	**Invert**	*boolean*
min-length	**Min length**	*numeric*
max-length	**Max length**	*numeric*
threads	**Parallel threads**	*numeric*

Restriction Sites Algorithm Element

The *element* searches restriction sites in the input sequence.

Parameters in GUI

Parameter	Description	Default value
Annotate As	Name of the result annotations.	Special value <rsite> is used. It specifies to use the enzymes names as names of the result annotations.
Circular	If *True* allows to search for restriction sites between the end and the beginning of the sequence.	False
Enzymes	Restriction enzymes used to recognize the restriction sites.	You must specify a value!

Parameters in Schema File

Type: rsite

Parameter	Parameter in the GUI	Type
key	**Annotate As**	*string*
circular	**Circular**	*boolean*
enzymes	**Enzymes**	*string*

Pattern Algorithm Element

The *element* searches for the specified pattern in the supplied sequence.

Parameters in GUI

Parameter	Description	Default value
Annotate As	Name of the result annotations.	misc_feature
Pattern	The pattern to search for.	You must specify a value!
Direction	See the description *here*.	Any

Parameters in Schema File

Type: search

Parameter	Parameter in the GUI	Type
key	**Annotate As**	*string*
pattern	**Pattern**	*string*

SITECON Algorithm Element

The *element* searches the input sequence for transcription factor binding sites (TFBSs) significantly similar to the specified SITECON profiles.

Parameters in GUI

Parameter	Description	Default value
Annotate As	Name of the result annotations.	misc_feature
Direction	See the description *here*.	Any
Min Err1	Filters the results by minumum value of Error type I.	0
Max Err2	Filters the results by maximum value of Error type II.	0.001
Model	Semicolon-separated list of SITECON profiles.	You must specify a value!
Min score	Recognition quality percentage threshold. Choosing too low threshold will lead you to recognition of too many TFBS. Choosing too high threshold will lead to no TFBS recognized.	85%

Parameters in Schema File

Type: sitecon

Parameter	Parameter in the GUI	Type
key	**Annotate As**	*string*
err1	**Min Err1**	*numeric*
err2	**Max Err2**	*numeric*
profile	**Model**	*string*
score	**Min score**	*numeric*
strand	**Direction**	*string* Available values are: • complement • direct • both

Smith-Waterman Algorithm Element

The *element* uses the Smith-Waterman algorithm to search in the input sequence for regions similar to the specified pattern.

Parameters in GUI

Parameter	Description	Default value
Annotate As	Name of the result annotations.	misc_feature
Direction	See the description *here*.	Any
Algorithm	Algorithm version. Depending on the computer configuration the following values may be available: • Classic 2 • SSE2 • CUDA	Classic 2
Filter results	Results filtering strategy. The values available are: • filter-intersections • none	filter-intersections
Gap ext score	Gap extension score.	-1.00
Gap open score	Gap open score.	-10.00
Scoring matrix	Specifies the scoring matrix to use.	Auto
Min score	Percentage of matching between the pattern and the searched sequence region.	90%
Pattern	The pattern to search for.	You must specify a value!
Search in translation	Translates the nucleotide sequence supplied to a protein sequence and searches in the translated sequence.	False

Parameters in Schema File

Type: ssearch

Parameter	Parameter in the GUI	Type
key	**Annotate As**	*string*
algorithm	**Algorithm**	*string* Depending on the computer configuration the values available are: • “Classic 2” • SSE2 • CUDA
filter	**Filter results**	*string* The values available are: • filter-intersections • none
gap-ext-score	**Gap ext score**	*numeric*
gap-open-score	**Gap open score**	*numeric*
matrix	**Scoring matrix**	*string*
min-score	**Min score**	*numeric*

pattern	**Pattern**	*string*
strand	**Direction**	*string* Available values are: • complement • direct • both
translate	**Search in translation**	*boolean*

Tandem Repeats Algorithm Element

The *Tandem repeats element* finds tandem repeats in a supplied sequence, stores found regions as annotations.

Parameters in GUI

Parameter	Description	Default value
Annotate As	Name of the result annotations.	repeate_unit
Direction	See the description *here*.	Any
Algorithm	The algorithm parameter allows one to select the search algorithm. The default and a fast one is the optimized suffix array algorithm.	Suffix index (optimized)
Min period	Minimum acceptable repeat length measured in base symbols.	1 n
Max period	Maximum acceptable repeat length measured in base symbols.	1000000 n
Min tandem size	The minimum tandem size sets the limit on minimum acceptable length of the tandem, i.e. the minimum total repeats length of the searched tandem.	9
Min repeat count	The minimum number of repeats of a searched tandem.	x3
Search for overlapped tandems	If this parameter is set to *True* then overlapped tandems should be included into the result.	
Parallel threads	Number of parallel threads used for the task.	Auto

Parameters in Schema File

Type: tandems

Parameter	Parameter in the GUI	Type
key	**Annotate As**	*string*
algorithm	**Algorithm**	*string* Available values are: • Suffix index • Suffix index (optimized)
min-period	**Min period**	*numeric*
max-period	**Max period**	*numeric*
min-tandem-size	**Min tandem size**	*numeric*
min-repeat-count	**Min repeat count**	*numeric*
show-overlapped-tandems	**Search for overlapped tandems**	*boolean*
strand	**Direction**	*string* Available values are: • complement • direct

		• both
n-threads	**Parallel threads**	*string*

Weight Matrix Algorithm Element

The *element* searches the input sequence for transcription factor binding sites (TFBSs) significantly similar to the specified weight matrix.

Parameters in GUI

Parameter	Description	Default value
Annotate As	Name of the result annotations.	misc_feature
Direction	See the description *here*.	Any
Matrix	Path to the profile.	You must specify a value!
Min score	Minimum score to detect TFBS.	85%

Parameters in Schema File

Type: wsearch

Parameter	Parameter in the GUI	Type
key	**Annotate As**	*string*
matrix	**Matrix**	*string*
min-score	**Min score**	*numeric*
strand	**Direction**	*string* Available values are: • complement • direct • both

HMM2 Algorithm Element

Searches HMM signals in a sequence with one or more profile HMM2 and saves the results as annotations.

Parameters in GUI

Parameter	Description	Default value
Profile HMM	Semicolon-separated list of input HMM files.	You must specify a value!
Min Length	Minimum length of a result region.	10
Max Length	Maximum length of a result region.	1000
Filter by High E-value	Reports domains <= this E-value threshold in output (**hmmsearch–domE** option).	1e+1
Filter by Low Score	Reports domains >= this score cutoff in output (**hmmsearch–domT** option).	0.01
Number of Sequences	Specifies number of significant sequences. It is used for domain E-value calculations (**hmmsearch–domZ** option).	1 (i.e. one input sequence)

Parameters in Schema File

Type: hmm2

Parameter	Parameter in the GUI	Type
min-len	**Min Length**	*string*
max-len	**Max Length**	*string*
hmm-profile	**Profile HMM**	*string*
e-val	**Filter by High E-value**	*numeric*
score	**Filter by Low Score**	*numeric*
seqs-num	**Number of Sequences**	*numeric*

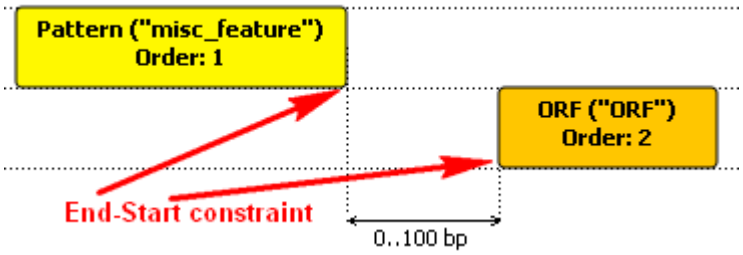
Constraint Elements

To add a constraint element add the required algorithm elements to the scene and drug&drop the required constraint element.

When a *constraint element is added*, two *algorithm elements* are selected.

There are four types of constraints that you can impose on the positional relationship of the results obtained from the algorithms calculations: *End-Start*, *Start-End*, *End-End* and *Start-Start*.

On the image below you can see a *schema* with *Pattern* and *ORF* algorithm elements and an *End-Start* constraint:



This means the following:

1. The algorithm elements specify to analyze the sequence with Pattern and ORF algorithms. The results of these analyses are the sets of annotations.

2. The condition says that the distance between a “Pattern annotation end” and an “ORF annotation start” should be within the specified bounds, i.e.:

Let:

pattern_annot_end := the last nucleotide of an annotation obtained from the Pattern algorithm calculations

orf_annot_start := the first nucleotide of an annotation obtained from the ORF algorithm calculations

The constraint is:

0bp <= Distance(**pattern_annot_end**, **orf_annot_start**) <= 100bp

Find the details on each constraint element below.

- End-Start Constraint Element
- Start-End Constraint Element
- End-End Constraint Element
- Start-Start Constraint Element

End-Start Constraint Element

Add the End-Start constraint to some two *algorithm elements*. Lets denote these elements as **alg1** and **alg2**.

Parameters in GUI

Parameter	Description	Default value
Min distance	Minimum distance between an **alg1** annotation end and an **alg2** annotation start.	0bp
Max distance	Maximum distance between an **alg1** annotation end and an **alg2** annotation start.	0bp

Constraint Explanation:

Let:

alg1_annot_end := the last nucleotide of an annotation obtained from the **alg1**.

alg2_annot_start := the first nucleotide of an annotation obtained from the **alg2**.

The result annotations should comply with the rule:

Min distance <= Distance(**alg1_annot_end**, **alg2_annot_start**) <= **Max distance**

Parameters in Schema File

Type: distance

Distance-type: end-to-start

Parameter	Parameter in the GUI	Type
min	**Min distance**	*numeric*
max	**Max distance**	*numeric*

Start-End Constraint Element

Add the Start-End constraint to some two *algorithm elements*. Lets denote these elements as **alg1** and **alg2**.

Parameters in GUI

Parameter	Description	Default value
Min distance	Minimum distance between an **alg1** annotation start and an **alg2** annotation end.	0bp
Max distance	Maximum distance between an **alg1** annotation start and an **alg2** annotation end.	0bp

Constraint Explanation:

Let:

alg1_annot_start := the first nucleotide of an annotation obtained from the **alg1**.

alg2_annot_end := the last nucleotide of an annotation obtained from the **alg2**.

The result annotations should comply with the rule:

Min distance <= Distance(**alg1_annot_start**, **alg2_annot_end**) <= **Max distance**

Parameters in Schema File

Type: distance

Distance-type: start-to-end

Parameter	Parameter in the GUI	Type
min	**Min distance**	*numeric*
max	**Max distance**	*numeric*

End-End Constraint Element

Add the End-End constraint to some two *algorithm elements*. Lets denote these elements as **alg1** and **alg2**.

Parameters in GUI

Parameter	Description	Default value
Min distance	Minimum distance between an **alg1** annotation end and an **alg2** annotation end.	0bp
Max distance	Maximum distance between an **alg1** annotation end and an **alg2** annotation end.	0bp

Constraint Explanation:

Let:

alg1_annot_end := the last nucleotide of an annotation obtained from the **alg1**.

alg2_annot_end := the last nucleotide of an annotation obtained from the **alg2**.

The result annotations should comply with the rule:
Min distance <= Distance(**alg1_annot_end**, **alg2_annot_end**) <= **Max distance**

Parameters in Schema File

Type: distance

Distance-type: end-to-end

Parameter	Parameter in the GUI	Type
min	**Min distance**	*numeric*
max	**Max distance**	*numeric*

Start-Start Constraint Element

Add the Start-Start constraint to some two *algorithm elements*. Lets denote these elements as **alg1** and **alg2**.

Parameters in GUI

Parameter	Description	Default value
Min distance	Minimum distance between an **alg1** annotation start and an **alg2** annotation start.	0bp
Max distance	Maximum distance between an **alg1** annotation start and an **alg2** annotation start.	0bp

Constraint Explanation:

Let:

alg1_annot_start := the first nucleotide of an annotation obtained from the **alg1**.

alg2_annot_start := the first nucleotide of an annotation obtained from the **alg2**.

The result annotations should comply with the rule:
Min distance <= Distance(**alg1_annot_start**, **alg2_annot_start**) <= **Max distance**

Parameters in Schema File

Type: distance

Distance-type: start-to-start

Parameter	Parameter in the GUI	Type
min	**Min distance**	*numeric*
max	**Max distance**	*numeric*

Extensions

- DNA Annotator
- DNA Flexibility
  - Configuring Dialog Settings
  - Result Annotations
- DNA Statistics
- DNA Generator
- ORF Marker
- Remote BLAST
  - Exporting BLAST Results to Alignment
  - Fetching Sequences from Remote Database
- BLAST+
  - Creating Database
  - Making Request to Database

- Fetching Sequences from Local BLAST+ Database
- Repeat Finder
  - Repeats Finding
  - Tandem Repeats Finding
    - Tandem Repeats Search Result
- Restriction Analysis
  - Selecting Restriction Enzymes
  - Using Custom File with Enzymes
  - Filtering by Number of Hits
  - Excluding Region
  - Circular Molecule
  - Results
- Molecular Cloning in silico
  - Digesting into Fragments
  - Creating Fragment
  - Constructing Molecule
    - Available Fragments
    - Fragments of the New Molecule
    - Changing Fragments Order in the New Molecule
    - Removing Fragment from the New Molecule
    - Editing Fragment Overhangs
    - Reverse Complement a Fragment
    - Other Constuction Options
    - Output
  - Creating PCR Product
- In Silico PCR
  - Primers Details
  - Primer Library
- Secondary Structure Prediction
- SITECON
  - SITECON Searching Transcription Factors Binding Sites
  - Types of SITECON Models
    - Eukaryotic
    - Prokaryotic
  - Building SITECON Model
- Smith-Waterman Search
- HMM2
  - Building HMM2 Model
  - Calibrating HMM2 Model
  - Searching Sequence Using HMM2 Profile
- HMM3
  - Building HMM Model
  - Searching Sequence Using HMM Profile
  - Searching Sequence Against Sequence Database
- uMUSCLE
  - MUSCLE Aligning
  - Aligning Profile to Profile with MUSCLE
  - Aligning Sequences to Profile with MUSCLE
- ClustalW
- MAFFT
- T-Coffee
- Bowtie
  - Bowtie Aligning Short Reads
  - Building Index for Bowtie
- Bowtie 2
  - Bowtie 2 Aligning Short Reads
  - Building Index for Bowtie 2
- BWA
  - Aligning Short Reads with BWA
  - Building Index for BWA
- BWA-SW
  - Aligning Short Reads with BWA-SW
  - Building Index for BWA-SW
- BWA-MEM
  - Aligning Short Reads with BWA-MEM
  - Building Index for BWA-MEM
- UGENE Genome Aligner
  - Aligning Short Reads with UGENE Genome Aligner
  - Building Index for UGENE Genome Aligner
  - Converting UGENE Assembly Database to SAM Format
- CAP3
- SPAdes
- Weight Matrix
  - Searching JASPAR Database
  - Building New Matrix
- Primer3
  - RTPCR Primer Design
- Spliced Alignment mRNA and cDNA
- External Tools Plugin

- Configuring External Tool
- Plasmid Auto Annotation
- ClustalO
- Kalign Aligning

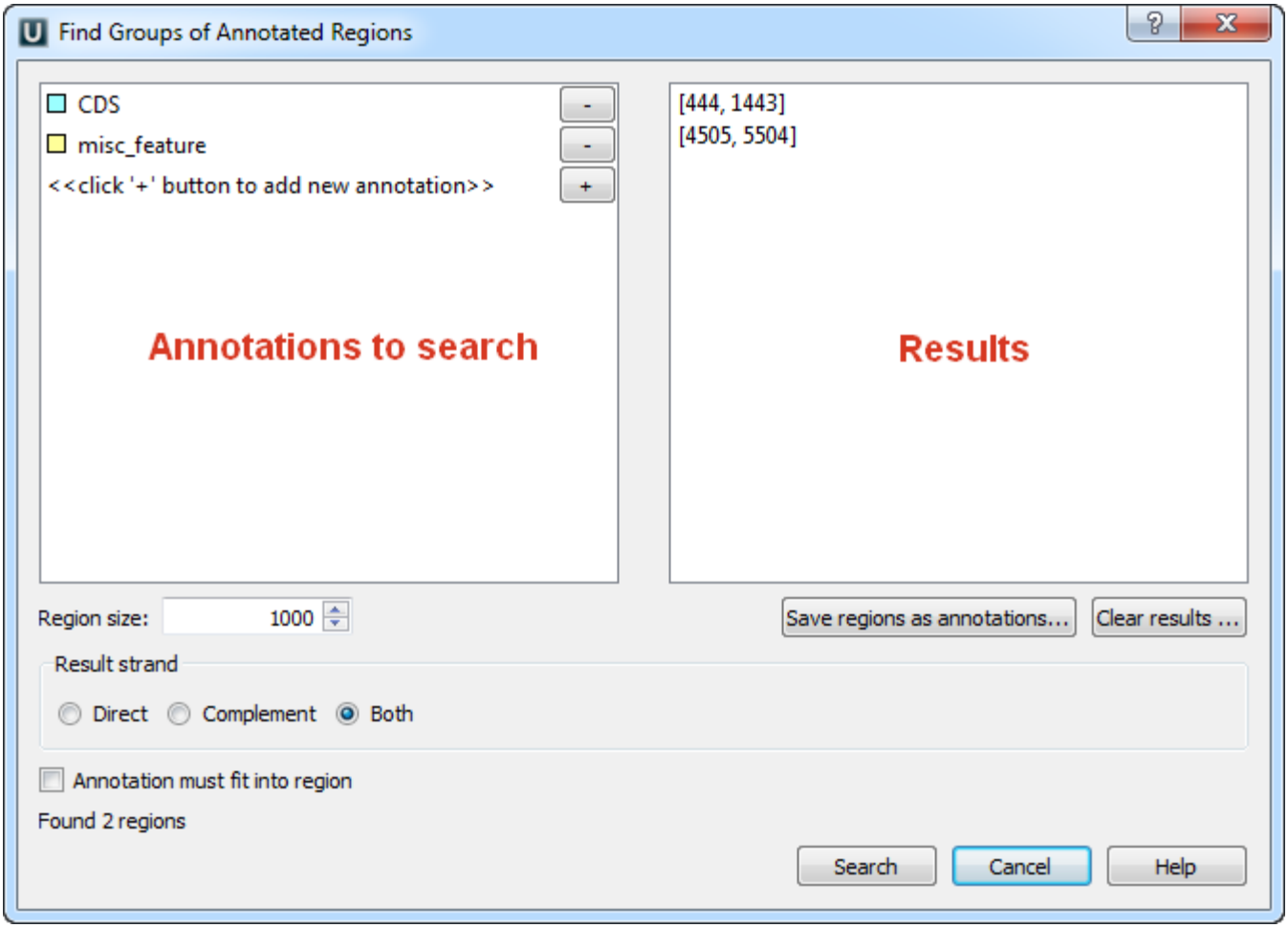
DNA Annotator

The *DNA Annotator* plugin provides an algorithm to search for sequence regions that contain a predefined set of annotations.

Usage example:

Open the *Sequence View* for a sequence that has annotations. A good candidate here could be any file in Genbank format with a rich set of annotations.

Select the *Analyze Find annotated regions* item in the context menu. The dialog will appear:



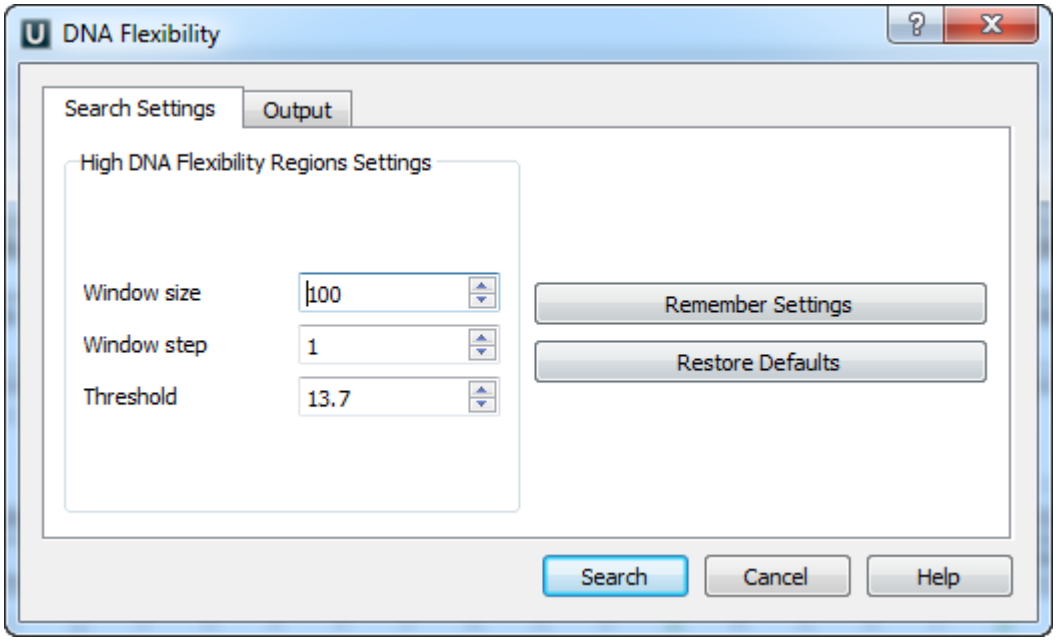
Using this dialog you can search for DNA sequence regions that contain every annotation from the list on the left side. The found regions are displayed on the right side of the dialog.

Use the *Save regions as annotations* button to store the regions as new annotations to the sequence.

DNA Flexibility

To search for regions of high DNA helix flexibility in a DNA sequence, open the sequence in the *Sequence View* and select the *Analyze Find high DNA flexibility regions* item in the context menu. Note that only standard DNA alphabet is supported, i.e. the sequence should consist of characters A, C, G, T and N.

The following dialog appears:



The calculation is made for overlapping windows along a given sequence. If there are two or more consecutive windows with an average flexibility threshold (in each window) greater than the specified *Threshold* parameter, such area is marked by an *annotation*.

The average threshold in a window is calculated by the following formula:

```
(average window threshold) = (sum of flexibility angles in the window) / (the window size - 1)
```

The following flexibility angles are used during the calculation:

Dinucleotide	Angle	Dinucleotide	Angle
AA	7.6	**CA**	14.6
AC	10.9	**CC**	7.2
AG	8.8	**CG**	11.1
AT	12.5	**CT**	8.8
GA	8.2	**TA**	25
GC	8.9	**TC**	8.2
GG	7.2	**TG**	14.6
GT	10.9	**TT**	7.6

A minimum value is used when N characters is present in a dinucleotide:

- **CN**, **NC**, **GN**, **NG**, **NN**: 7.2
- **AN**, **NA**, **TN**, **NT** : 7.6

- Configuring Dialog Settings
- Result Annotations

Configuring Dialog Settings

In the dialog you can setup the corresponding parameters:

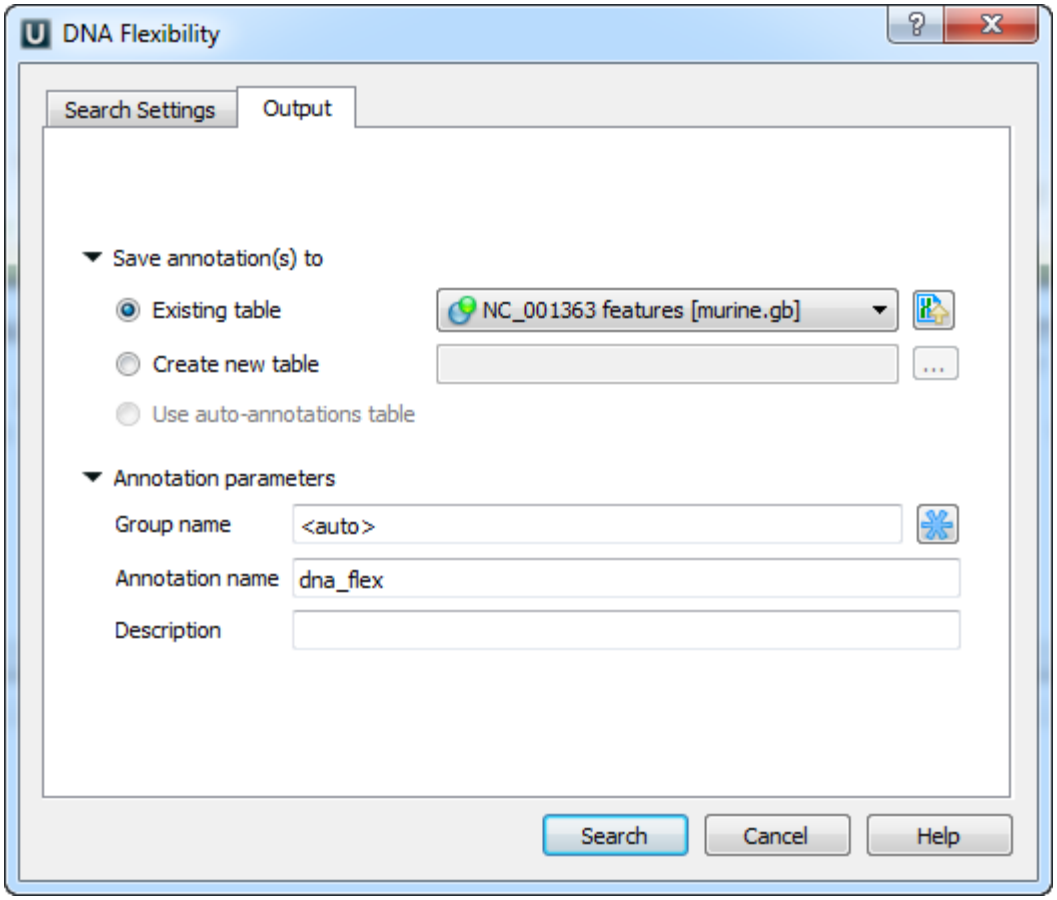
Window size — the number of bases in a window. The window size should be greater than 2. The default value is 100 bp.

Window step — the number of bases used to shift a window. The *Window step* should be a positive integer. The default value is 1 bp.

Threshold — the threshold value of the twist angle (see above). The default value is 13.7.

You can remember the input values or restore the default values using the *Remember Setting* and the *Restore Defaults* buttons.

The annotations names and other parameters can be changed on the *Output* tab of the dialog:

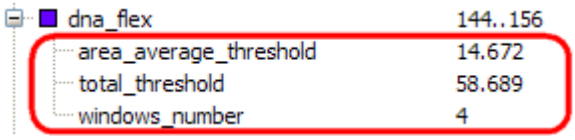


Once the *Search* button has been pressed, the *annotations* for the regions of the high DNA flexibility are created.

Result Annotations

Each annotation has the following qualifiers:

- *area_average_threshold* — average window threshold in the area (i.e. *total_threshold* / *windows_number*)
- *total_threshold* — sum of all window thresholds in the area
- *windows_number* — number of windows in the area



Using the *DNA Graphs Package* you can see the flexibility graph of a DNA sequence.

DNA Statistics

The *DNA Statistics* plugin provides exportable statistic reports.

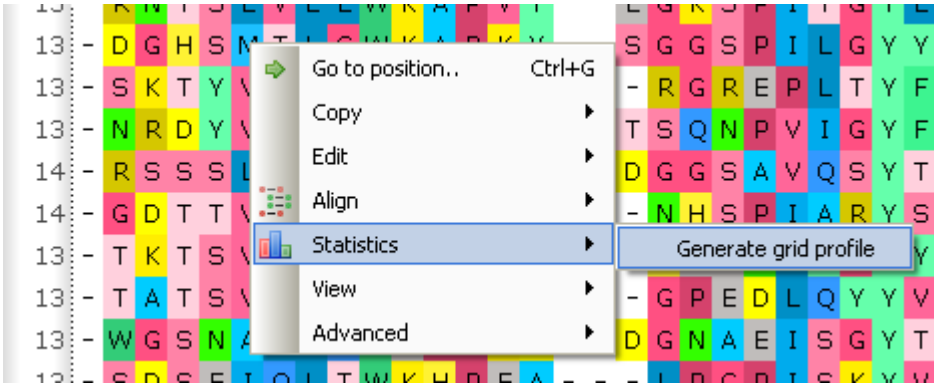
In the current UGENE version, the *DNA Statistics* plugin provides only *Alignment Grid Profile* report. The *Alignment Grid Profile* shows positional amino acid or nucleotide counts highlighted according to the frequency of symbols in a row.

The original idea of the MSA Grid Profile is described in the following paper:

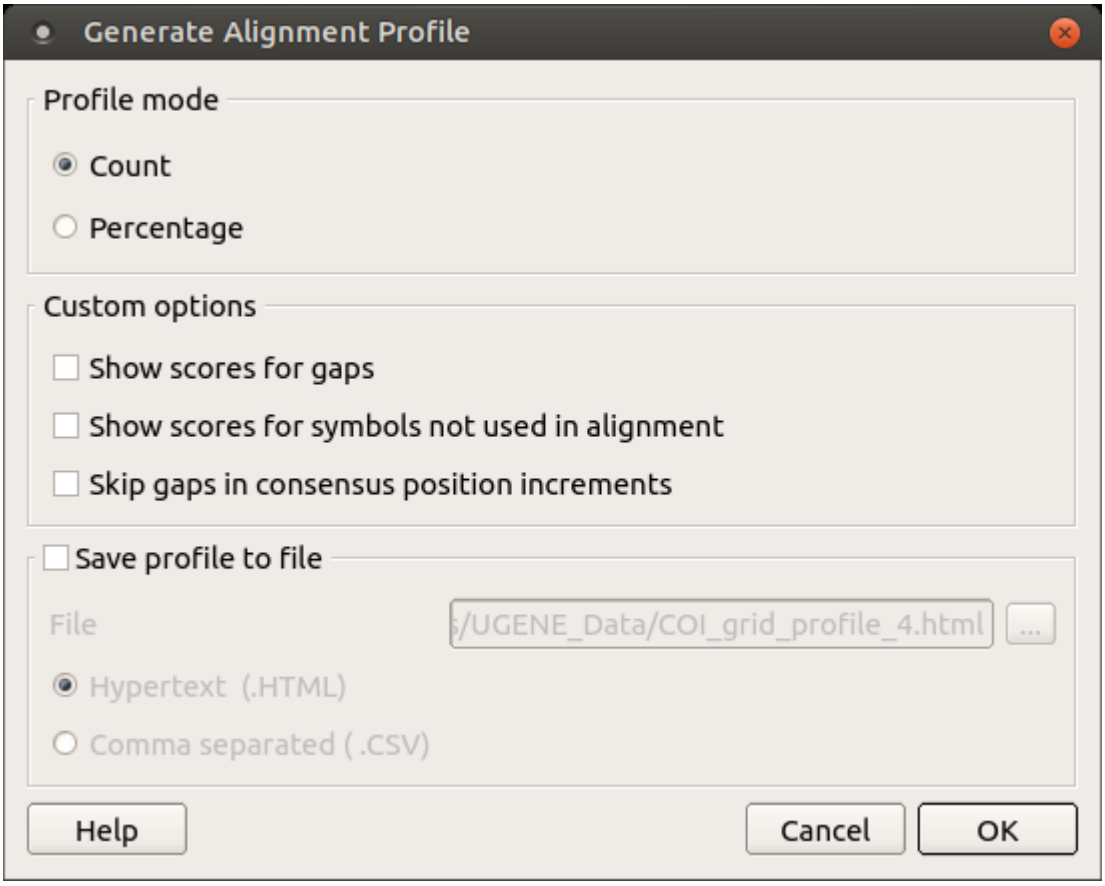
“Alberto Roca, Albert Almada and Aaron C Abajian: ProfileGrids as a new visual representation of large multiple sequence alignments: a case study of the RecA protein family, BMC Bioinformatics 2008, 9:554”

Usage example:

Open a sequence alignment in the *Alignment Editor* and use the *Statistics Generate grid profile* context menu item.



The dialog will appear:



Here is a brief description of the options that can be set in the dialog:

Profile mode: Counts/Percents — select the *Percents* to have scores shown as percents in the report.

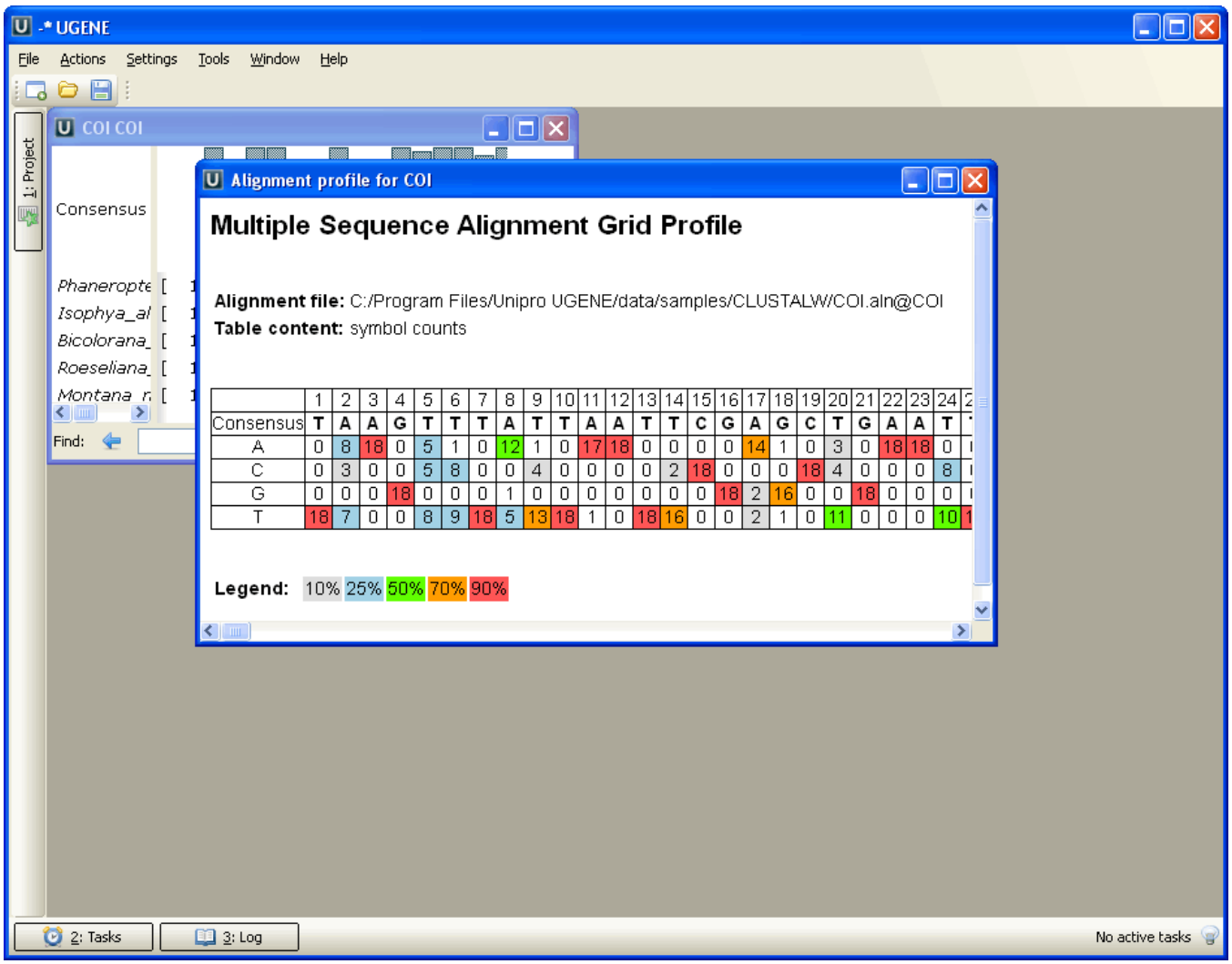
Show scores for gaps — check this item if you want gap characters ('—') statistics to be shown in the report.

Show scores for symbols not used in alignment — if a symbol is not used in the alignment at all it won't be shown in the report. Check this item to make all symbols of alignment alphabet reported.

Skip gaps in consensus position increments — consensus ruler configuration. If checked the gaps in consensus will not lead to ruler increments.

Save profile to file — allows saving profile to a file in the HTML or CSV format. The CSV format is convenient for further processing in worksheets editors like Excel.

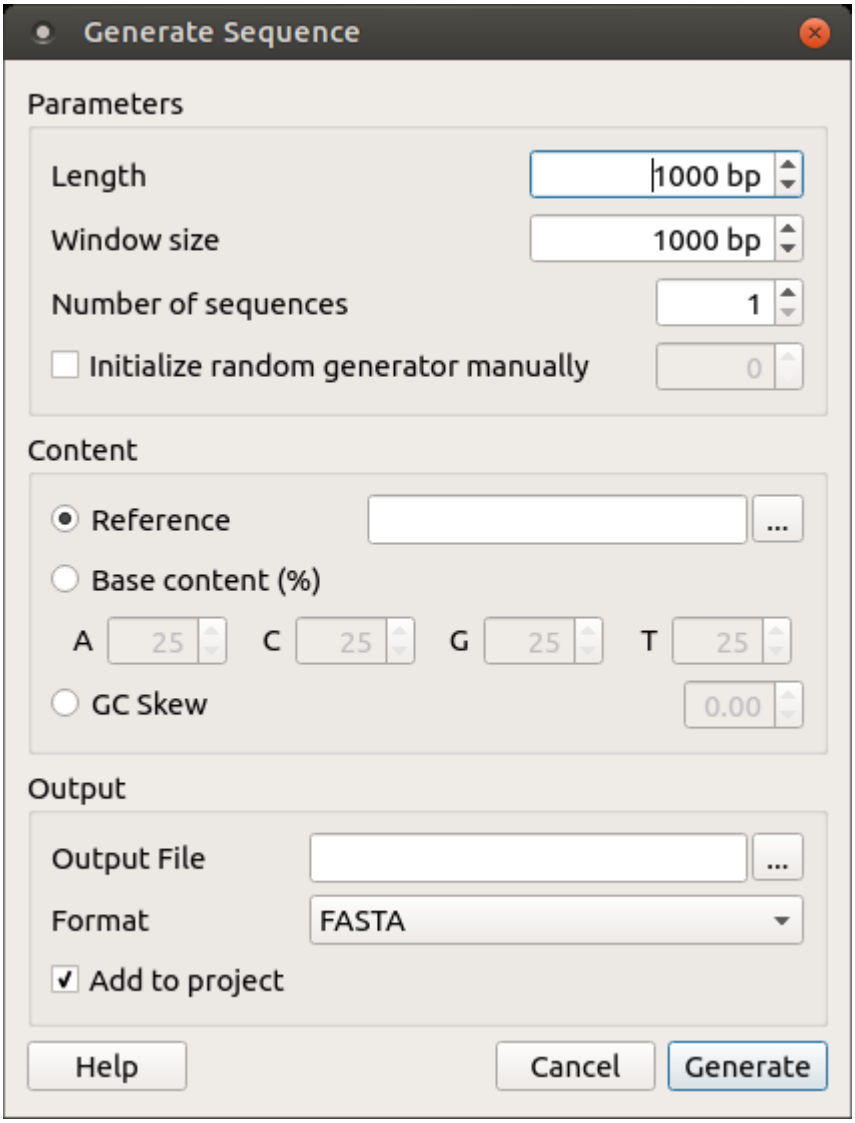
The resulting profile in the HTML mode:



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Consensus	T	A	A	G	T	T	T	A	T	T	A	A	T	T	C	G	A	G	C	T	G	A	A	T
A	0	8	18	0	5	1	0	12	1	0	17	18	0	0	0	0	14	1	0	3	0	18	18	0
C	0	3	0	0	5	8	0	0	4	0	0	0	0	2	18	0	0	0	18	4	0	0	0	8
G	0	0	0	18	0	0	0	1	0	0	0	0	0	0	0	18	2	16	0	0	18	0	0	0
T	18	7	0	0	8	9	18	5	13	18	1	0	18	16	0	0	2	1	0	11	0	0	0	10

DNA Generator

DNA sequence generator is a tool that generates a random DNA sequence with specified nucleotide content. To generate a random DNA sequence select the *Tools->Generate* sequence item in the main menu. The dialog will appear:



The following parameters are available:

Length - length of the resulted sequence(s) (using '1000' bp by default).

Window size - the size of a window where set content (using '1000' by default).

Number of sequences - the number of sequences to generate (using '1' by default).

Initialize random generator manually - value to initialize the random generator.

Reference - path to the reference file (could be a sequence or an alignment).

Base content - set the base content percents manually.

GC Skew - set the GC skew of the resulted sequence.

Output file - output file.

Format - output file format (using 'fasta' by default).

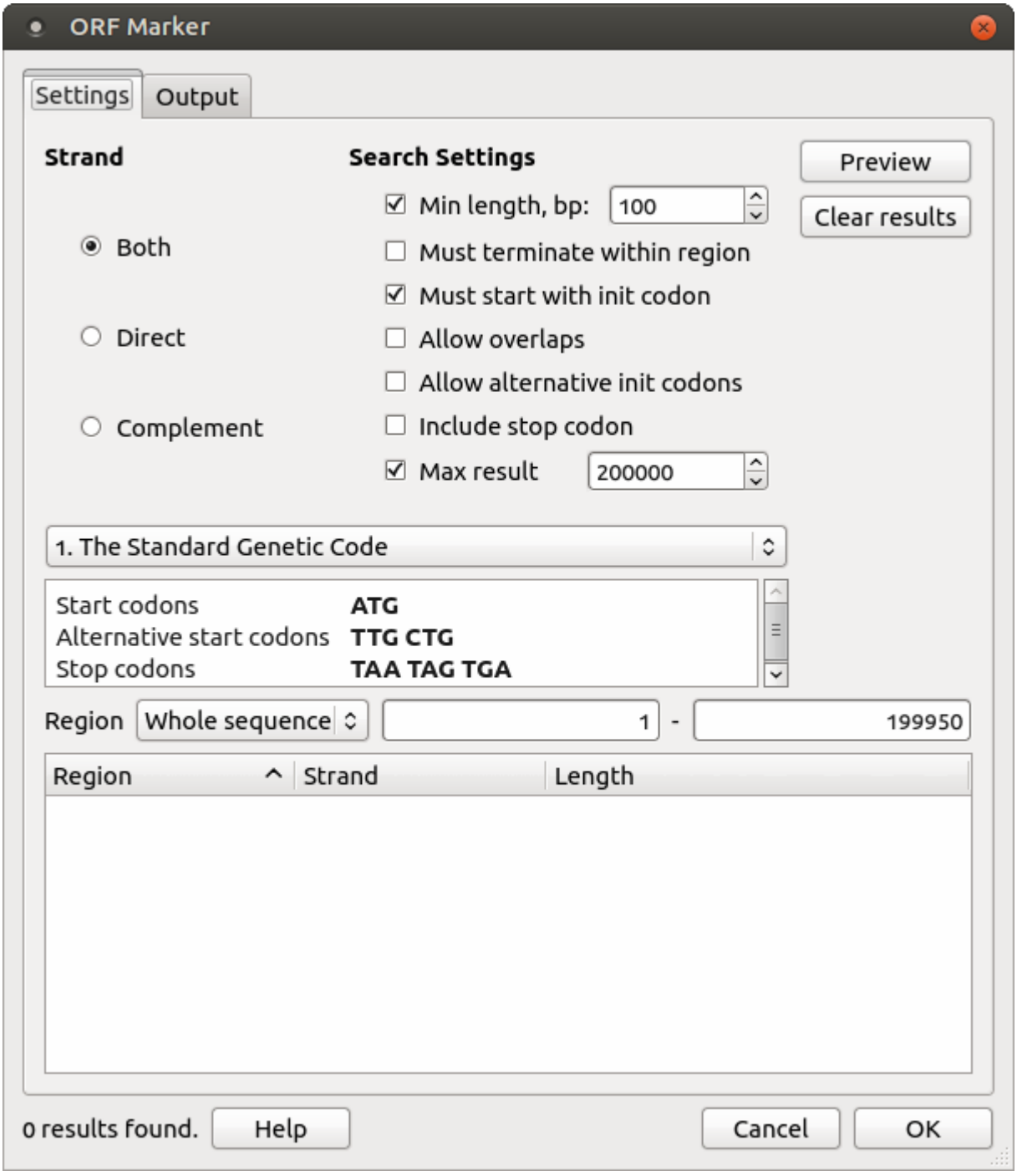
Add to project - adds the generated sequence(s) to project.

Once the *Search* button has been pressed, the sequence(s) are created.

ORF Marker

From this chapter you can learn how to search for Open Reading Frames (ORF) in a DNA sequence. The ORFs found are stored as automatic annotations. This means that if the automatic annotations highlighting has been enabled then ORFs are searched and highlighted for each sequence opened. Refer *Automatic Annotations Highlighting* to learn more.

To open the *ORF Marker* dialog, select the *Analyze Find ORFs* item in the context menu.



The following search settings are available:

Min length — ORFs with length lower than *Min length* value will not be found.

Must terminate within region — this option ignores boundary ORFs located beyond the search region.

Must start with init codon — item switches the ORF Marker algorithm to the mode when any non-stop amino acid code is interpreted as region start position.

Allow overlaps — alternative (downstream) initiators, when another start codon is located within a longer ORF, i.e. all possible ORFs will be found, not only the longest ones.

Allow alternative init codon — option includes ORFs starting with alternative initiation codons, accordingly to the current translation table.

Include stop codon — includes stop codons into resulting annotations.

The other available parameters are:

DNA-to-Amino translation table defines the way start, alternative start and stop codons are encoded.

Strand — where to search the ORFs: in the direct strand, in the complement strand or in both strands.

Preview — allow to preview the regions, strands and lengths of the found ORFs.

Clear results — becomes available when some results have been found, clears these results.

To set the saving parameters go to the *Output* tab of the dialog.

Here you can modify *the annotations saving* parameters (*Group name*, *Description* and a file to save the annotation to).

Results:

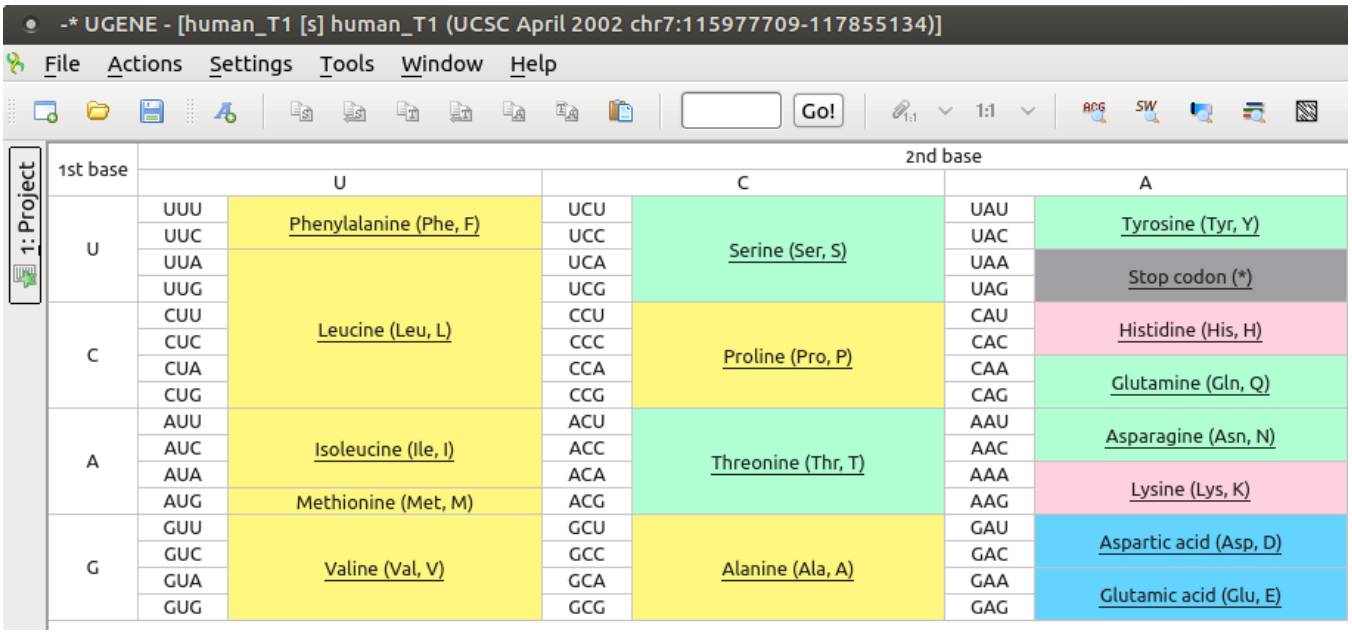
When the search parameters has been selected and the *OK* button has been pressed in the dialog, the *auto-annotating* becomes enabled. In the *Annotatio ns editor* the ORFs annotations can be found in the Auto-annotations\orf group.

After the search has been finished you can browse the results, sort them by length, strand or start position and save as annotations to the original sequence in the Genbank format.

For more information about codons use the codon table. To show or hide the table use *Ctrl+T* shortcut or click the *Show codon table* toolbar button menu:



The codon table will appear:



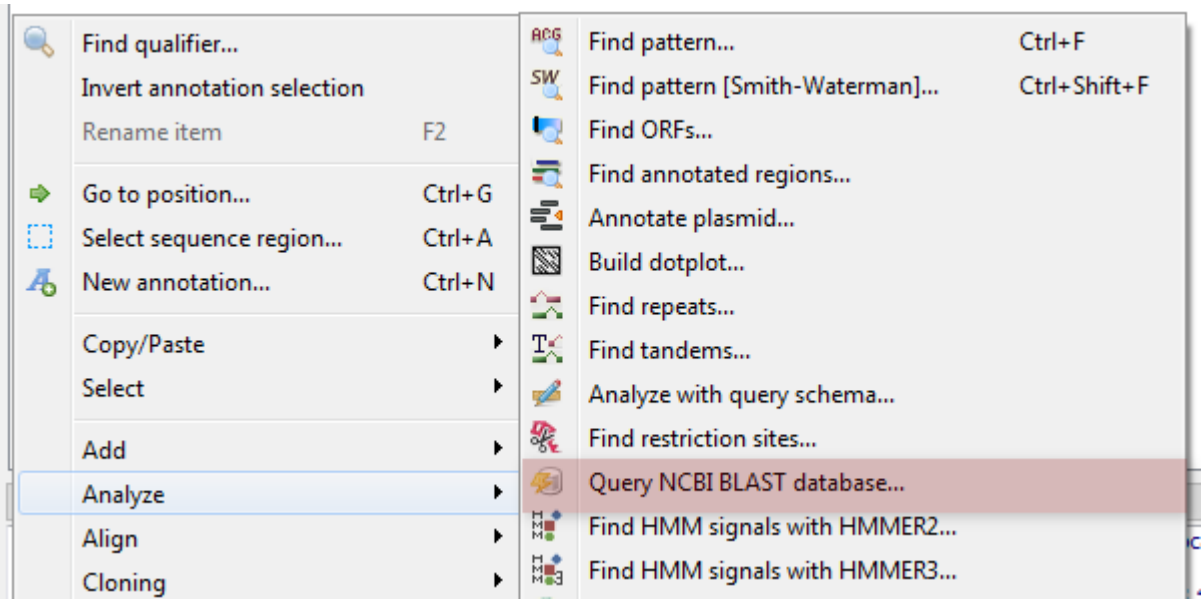
1st base	2nd base U		2nd base C		2nd base A	
U	UUU	Phenylalanine (Phe, F)	UCU	Serine (Ser, S)	UAU	Tyrosine (Tyr, Y)
	UUC		UCC		UAC	
	UUA	Leucine (Leu, L)	UCA		UAA	Stop codon (*)
	UUG		UCG		UAG	
C	CUU		CCU	Proline (Pro, P)	CAU	Histidine (His, H)
	CUC		CCC		CAC	
	CUA		CCA		CAA	Glutamine (Gln, Q)
	CUG		CCG		CAG	
A	AUU	Isoleucine (Ile, I)	ACU	Threonine (Thr, T)	AAU	Asparagine (Asn, N)
	AUC		ACC		AAC	
	AUA		ACA		AAA	Lysine (Lys, K)
	AUG	Methionine (Met, M)	ACG		AAG	
G	GUU	Valine (Val, V)	GCU	Alanine (Ala, A)	GAU	Aspartic acid (Asp, D)
	GUC		GCC		GAC	
	GUA		GCA		GAA	Glutamic acid (Glu, E)
	GUG		GCG		GAG	

Clicking on a codon name redirects you to Wikipedia to give you a brief description of the corresponding amino acid. Cells of the table are colored according to classes of amino acids.

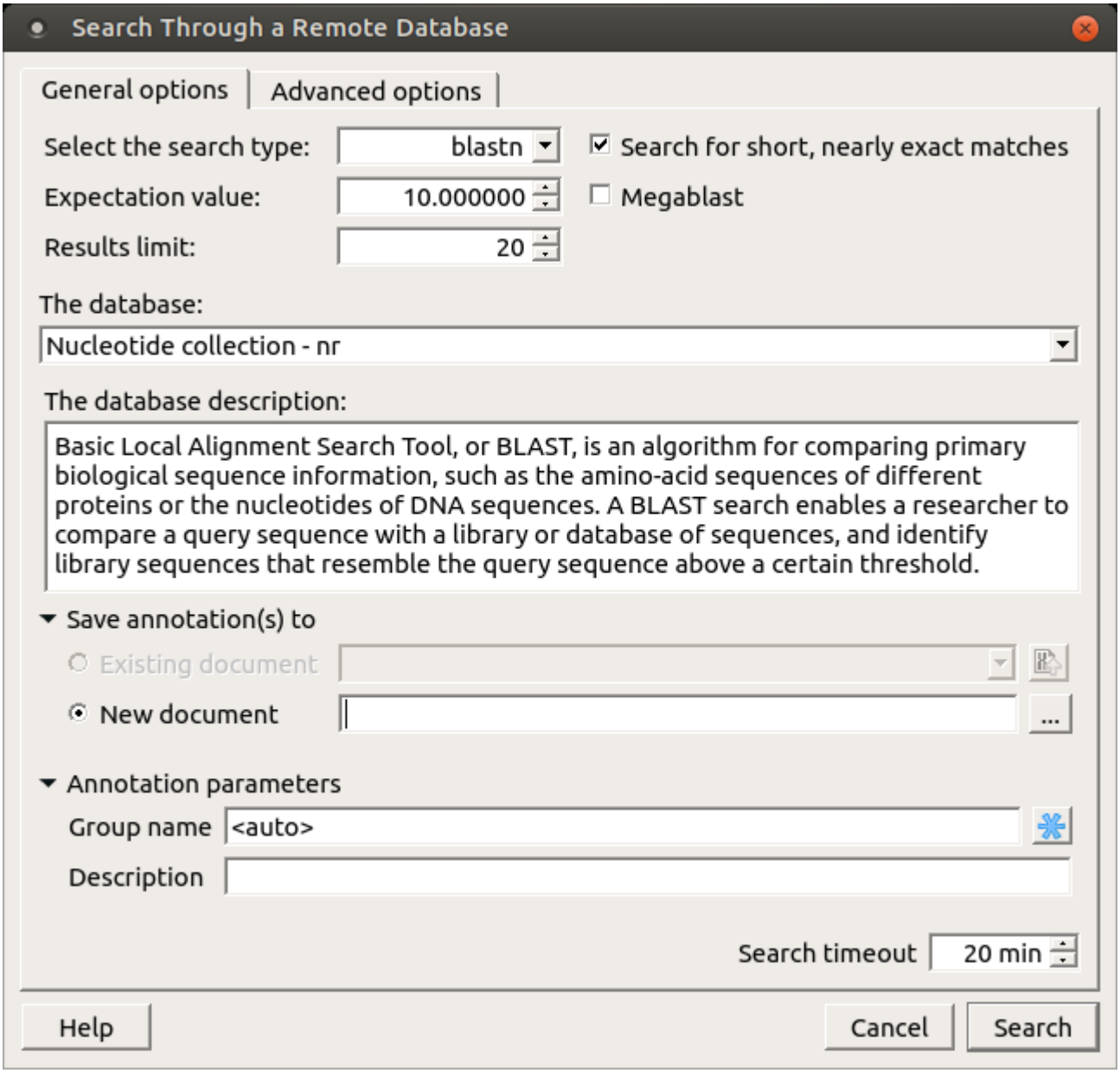
Remote BLAST

The *Remote BLAST* plugin provides a capability to annotate sequences with information stored in the NCBI BLAST remote database.

To perform a remote database search open a *Sequence View*, select a sequence region to analyze and click the *Analyze Query NCBI BLAST database* co ntext menu item. If a region is not selected the whole sequence will be analyzed.



The following dialog will appear where you can choose the search options:



General options are:

Select the search type — in the remote databases the *blastn* search is used for nucleotide sequences, *blastp* and *cdd* searches are used for amino sequences.

UGENE also provides a way to use *blastp* and *cdd* searches for nucleotide sequences. This is achieved by translating the nucleotide sequence into the amino sequences.

When a sequence is translated the translation table from the active *Sequence View* is used. Finally, all 6 translations are used to query the remote database with the selected *blastp* or *cdd* search.

Expectation value — this option specifies the statistical significance threshold for reporting matches against database sequences. Lower expect thresholds are more stringent, leading to fewer chance matches being reported.

Max hits — the maximum number of hits that will be shown (not equal to number of annotations). The maximum availablle number is 5000.

Database — the target database.

Search for short, nearly exact matches — automatically adjusts the word size and other parameters to improve results for short queries.

Megablast — select this option to compare query with closely related sequences. It works best if the target percent identity is 95% or more, but it is very fast.

You can see the description of the annotation saving parameters *here*.

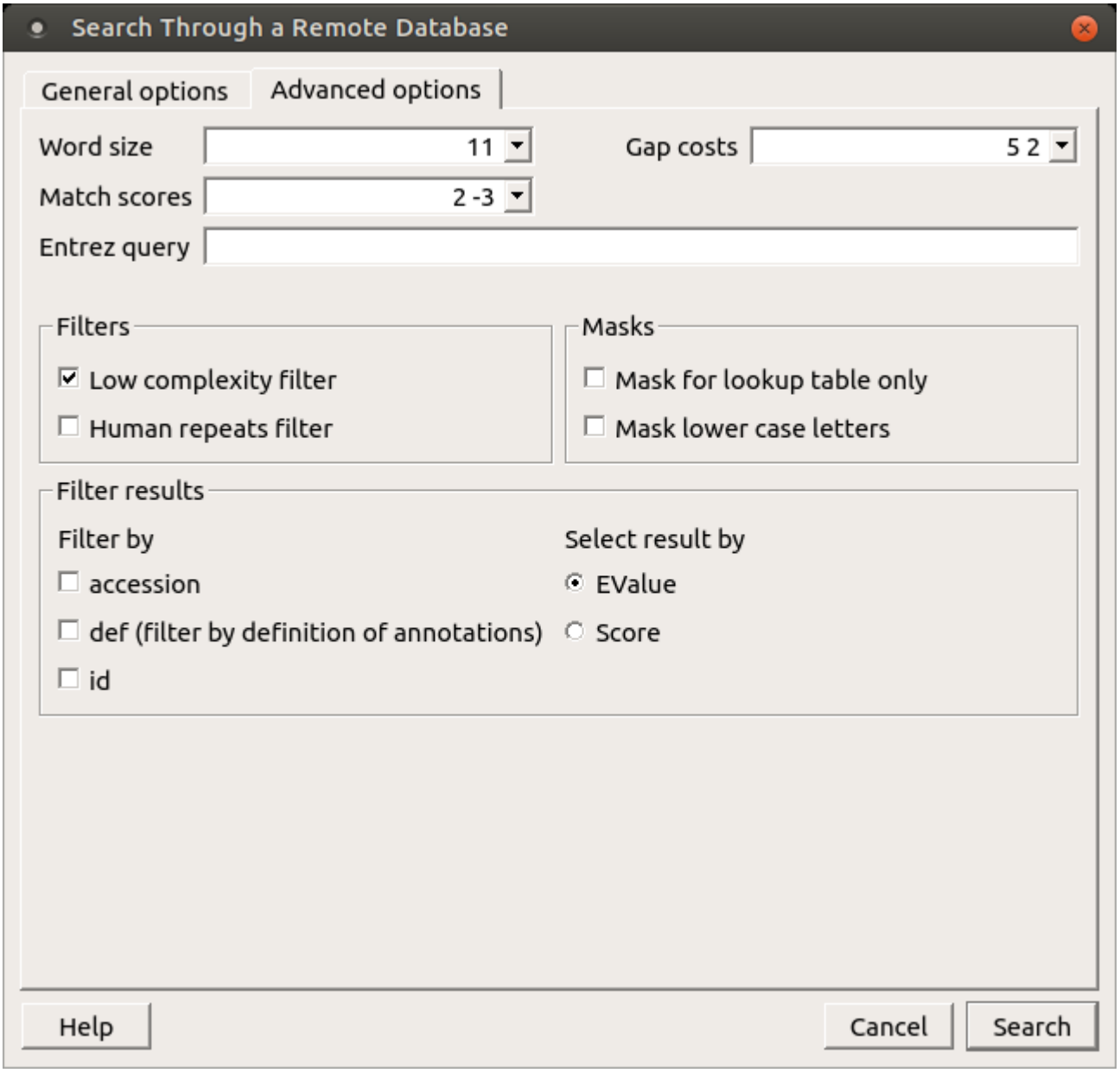
Search timeout — the remote task terminated if the timeout is reached.

There is a little difference in default values of parameters between NCBI Nucleotide BLAST web interface and UGENE:

- The web interface uses the *megablast* option by default: the search is fast, but only highly similar sequences are found.
- UGENE ignores the option by default: the search may take more time, but all somewhat similar sequences are found.

Check the *Megablast* option, if you want exactly the same results to be found in UGENE as you had in the NCBI web interface.

Also there is *Advanced options* tab:



The view of the *Advanced options* tab depends on the selected search. For the *blastn* search it looks like on the picture above.

Word size — the size of the subsequence parameter for the initiated search.

Gap costs — costs to create and extend a gap in an alignment. Increasing the Gap costs will result in alignments which decrease the number of Gaps introduced.

Match scores — reward and penalty for matching and mismatching bases.

Entrez query — a BLAST search can be limited to the result of an *Entrez query* against the database chosen. This restricts the search to a subset of entries from that database fitting the requirement of the *Entrez query*. Examples are given below:

protease NOT hiv1[organism] — this will limit a BLAST search to all proteases, except those in HIV 1.

1000:2000[slen] — this limits the search to entries with lengths between 1000 to 2000 bases for nucleotide entries, or 1000 to 2000 residues for protein entries.

Mus musculus[organism] AND biomol_mrna[properties] — this limits the search to mouse mRNA entries in the database. For common organisms, one can also select from the pulldown menu.

10000:100000[mlwt] — this is yet another example usage, which limits the search to protein sequences with calculated molecular weight between 10 kD to 100 kD.

src specimen voucher[properties] — this limits the search to entries that are annotated with a /specimen_voucher qualifier on the source feature.

all[filter] NOT enviromnental sample[filter] NOT metagenomes[orgn] — this excludes sequences from metagenome studies and uncultured sequences from anonymous environmental sample studies.

For help in constructing *Entrez queries* see the Entrez Help document.

Filters — filters for regions of low compositional complexity and repeat elements of the human's genome.

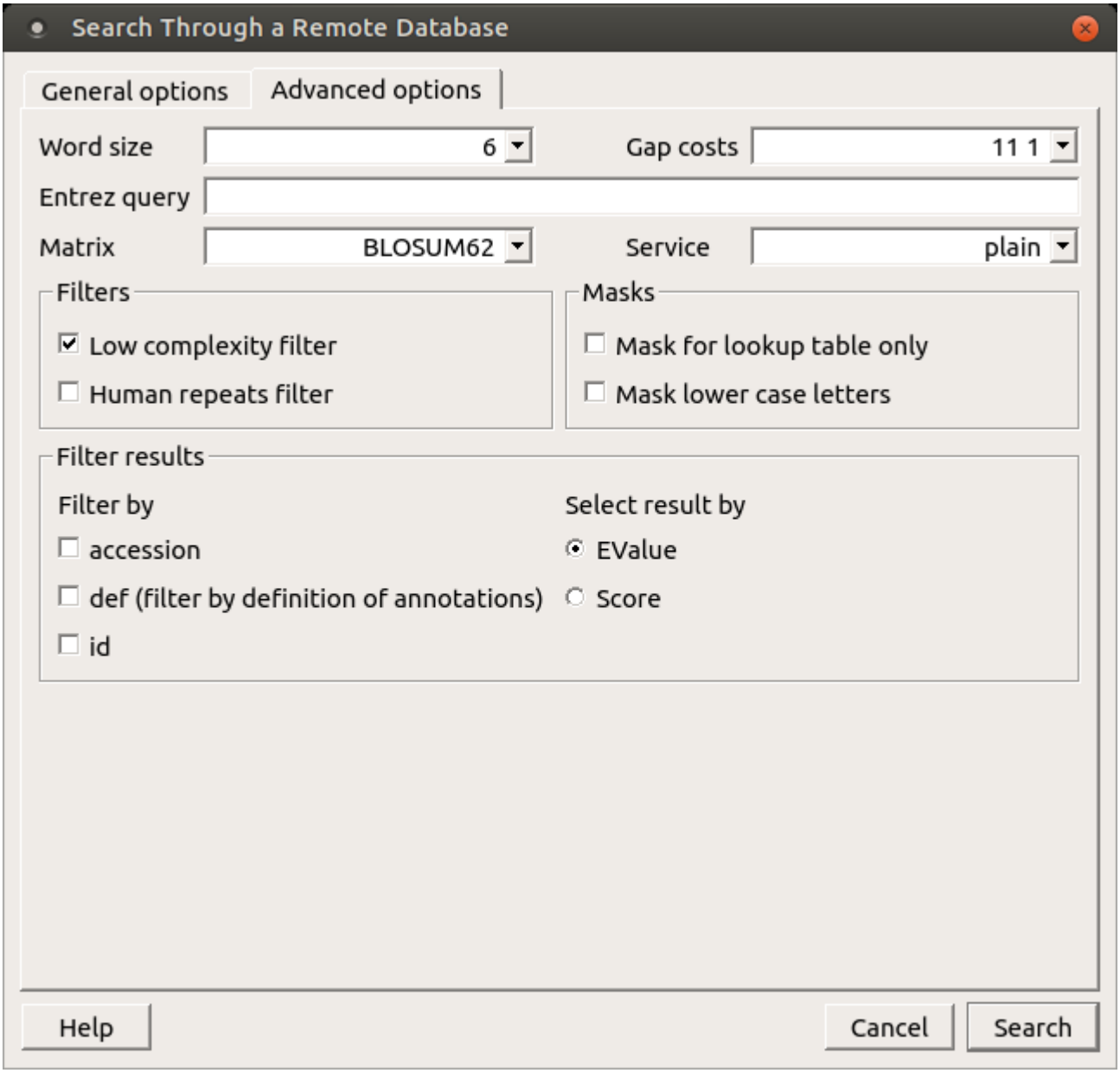
Masks for lookup table only — this option masks only for purposes of constructing the lookup table used by BLAST so that no hits are found based upon low-complexity sequence or repeats (if repeat filter is checked).

Mask lower case letters — with this option selected you can cut and paste a FASTA sequence in upper case characters and denote areas you would like filtered with lower case.

Filter by — filters results by accession, by definition of annotations or by id.

Select result by — selects results by EValue or by score.

When the *blastp* search is selected in the general options, the view of the *Advanced options* tab is the following:



As you can see there is no *Match scores* option, but there are *Matrix* and *Service* options.

Matrix — key element in evaluating the quality of a pair-wise sequence alignment is the “substitution matrix”, which assigns a score for aligning any possible pair of residues.

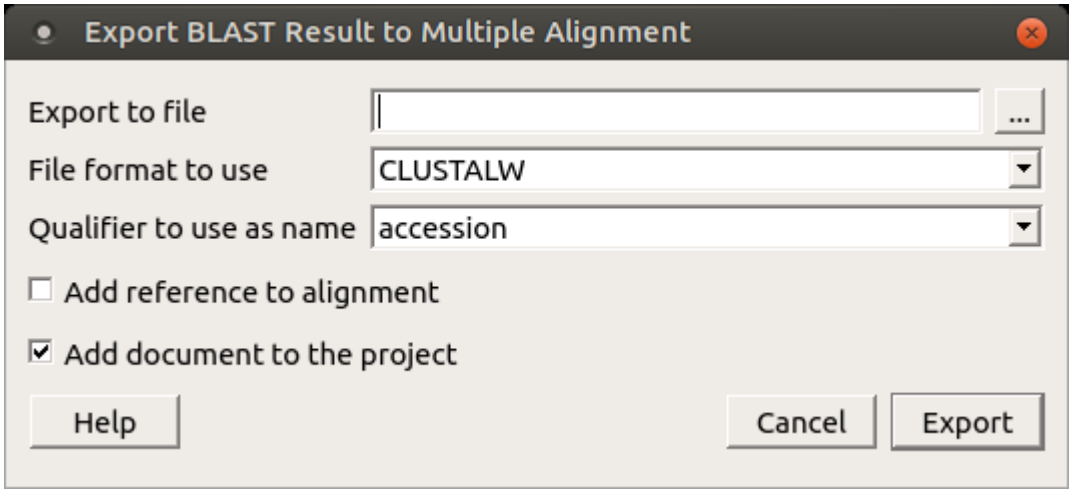
Service — blastp service which needs to be performed: plain, psi or phi.

The *Advanced options* tab is not available when the *cdd* search is selected.

- Exporting BLAST Results to Alignment
- Fetching Sequences from Remote Database

Exporting BLAST Results to Alignment

To export BLASt results as alignment select the results in the *Annotations Editor* and call the *Export->Export BLAST result to alignment* context menu item. The following dialog will appear:



The following parameters are available:

Export to file - name of the new file.

File format to use - format of the new file. The following formats are available: CLUSTALW, FASTA, MSF, MEGA, NEXUS, PHYLIP Interleaved, PHYLIP Sequential, Stockholm.

Qualifier to use as name - name of the qualifier. The following qualifiers are available: accession, def, id.

Add reference to alignment - adds a reference to alignment.

Add document to the project - adds the new document to the project.

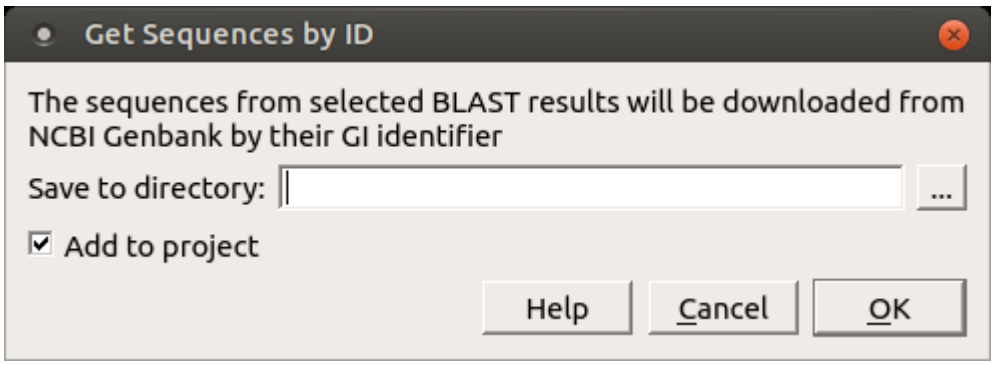
Select the options and click on the *Export* button.

Fetching Sequences from Remote Database

Each result annotation found with the *remote BLAST* in UGENE has "accession" and "id" qualifiers that can be used to fetch the corresponding sequences from the NCBI. The prompt way to fetch the sequences of several annotations is the following:

- Select the annotations in the *Annotations Editor*.
- Open the context menu.
- Choose the *Fetch sequences from remote database->Fetch sequences by 'id' from 'blast result'* item or *Fetch sequences from remote database->Fetch sequences by 'accession' from 'blast result'* item.

The following dialog will appear:



Select an output path in the dialog and click the *OK* button.

BLAST+

The Basic Local Alignment Search Tool (BLAST) finds regions of local similarity between sequences. The program compares nucleotide or protein sequences to sequence databases and calculates the statistical significance of matches. BLAST+ can be used to infer functional and evolutionary relationships between sequences as well as help identify members of gene families.

From UGENE you can use the following tools of the old BLAST+ package:

- **blastall** — the old program developed and distributed by the NCBI for running BLAST searches.
- **formatdb** — formats protein or nucleotide source databases before these databases can be searched by **blastall**.
- **blastn** — searches a nucleotide database using a nucleotide query.
- **blastp** — searches a protein database using a protein query.
- **blastx** — searches a protein database using a translated nucleotide query.
- **tblastn** — compares a protein query against a translated nucleotide database (the all six reading frames).
- **tblastx** — translates the query nucleotide sequence in all six possible frames and compares it against the six-frame translations of a nucleotide sequence database.
- **makeblastdb** — — formats protein or nucleotide source databases before these databases can be searched by other *BLAST+* tools.

BLAST home page: http://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastHome

To make *BLAST+* tools available from UGENE:

1. Install the required version of *BLAST+* on your system.
2. Set the paths to the executables, you are going to use, on the *External tools* tab of UGENE *Application Settings* dialog.

After you've finished this configuration you can access the tools from the *Tools BLAST +* submenu of the main menu.

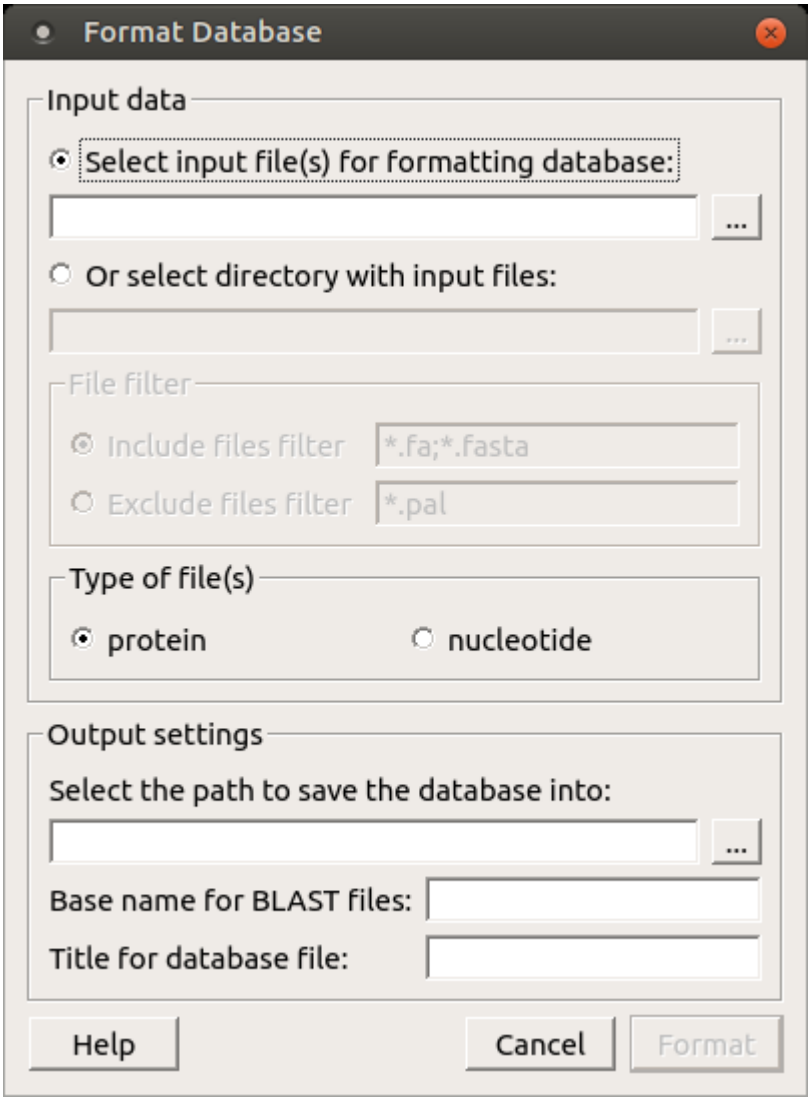
- Creating Database
- Making Request to Database
- Fetching Sequences from Local BLAST+ Database

Creating Database

To format a BLAST+ database do the following:

- Open *Tools BLAST+ BLAST+ make DB*.

The *Format database* dialog appears:



Here you must select the input files. If all the files you want to use are located in one directory, you can simply select the directory with the files. By default, only the files are taken into account with *.fa and *.fasta extensions. You can change this by specifying either *Include files filter* or exclude *files filter*.

You can choose either *protein* or *nucleotide* type of the files.

Then you must select the path to save the database file and specify a *Base name for BLAST files* and a *Title for the database file*.

Making Request to Database

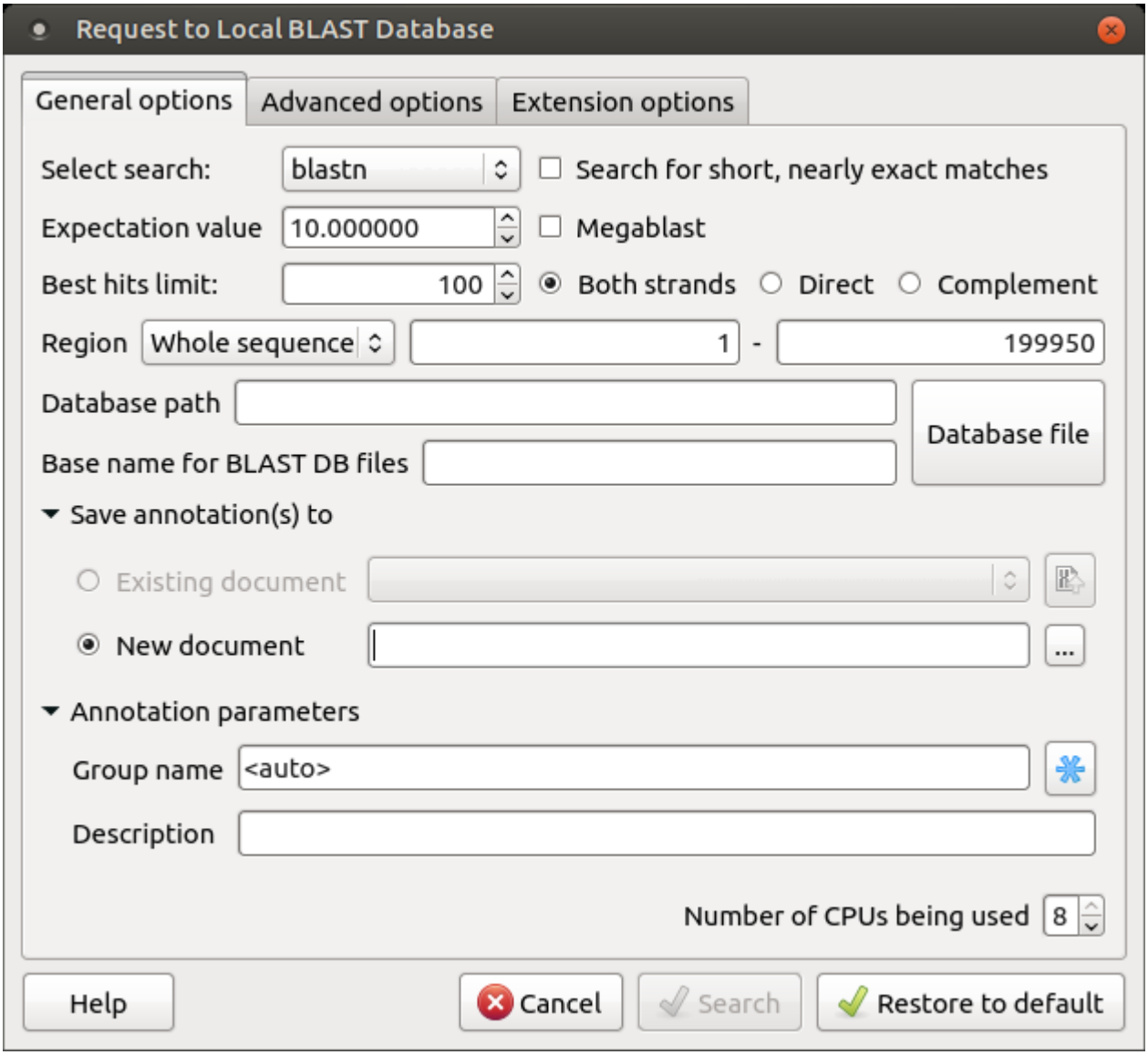
To make a request to a local BLAST database do the following:

- Open *Tools BLAST+ BLAST+ Search*.

If there is a sequence opened you can also initiate the request to a local BLAST database from the *Sequence View*:

- Select the *Analyze Query with BLAST+* item in the context menu or in the *Actions* the main menu.

The *Request to local BLAST+ database* dialog will appear:



The following general options are available:

Select search - here you should select the tool you would like to use. If the query sequence is a nucleotide sequence then *blastn*, *blastx* and *tblastx* it ems are available. For a protein sequence, the items are *blastp* and *tblastn*.

Expectation value - this option specifies the statistical significance threshold for reporting matches against database sequences. Lower expect thresholds are more stringent, leading to fewer chance matches being reported.

Culling limit - the maximum number of hits that will be shown (not equal to a number of annotations). The maximum available number is 5000.

Search for short, nearly exact matches - automatically adjusts the word size and other parameters to improve results for short queries.

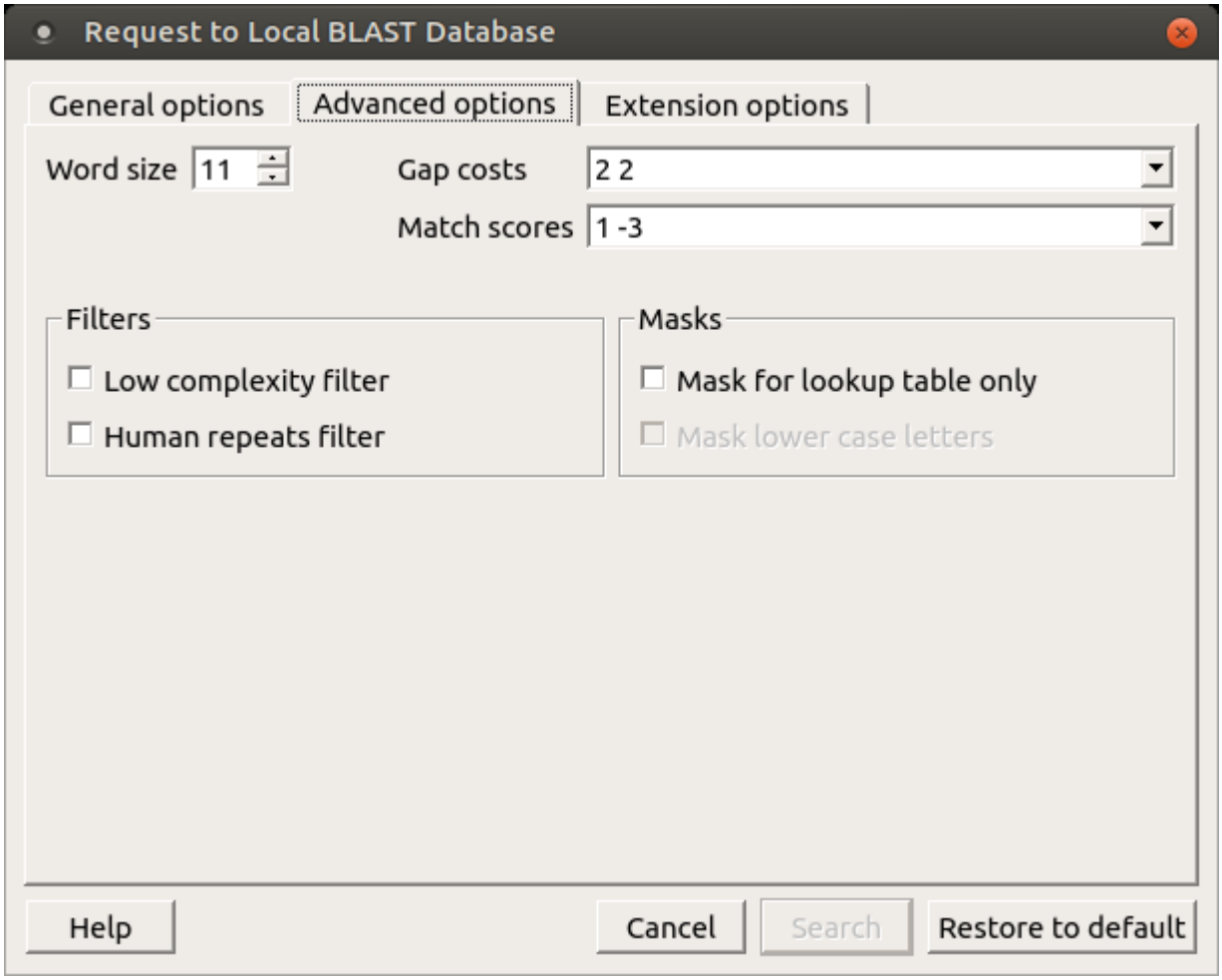
Megablast - select this option to compare queries with closely related sequences. It works best if the target percent identity is 95% or more, but it is very fast.

Database path - path to the database files.

Base name for BLAST DB files - the base name for the BLAST database files.

You can see the description of the annotation saving parameters *here*.

The following advanced parameters are available:



Word size - the size of the subsequence parameter for the initiated search.

Gap costs - costs to create and extend a gap in an alignment. Increasing the Gap costs will result in alignments which decrease the number of Gaps introduced.

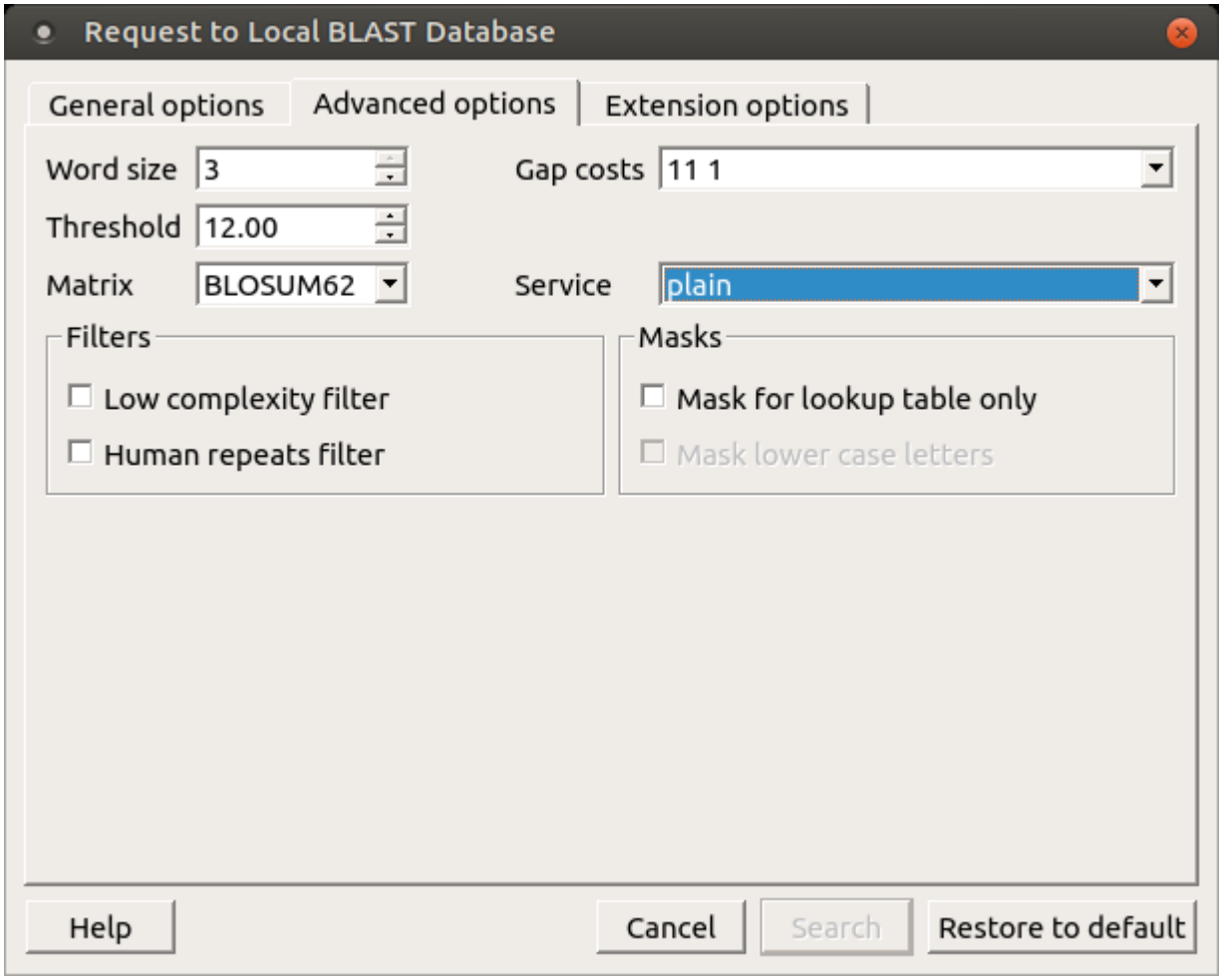
Match scores - reward and penalty for matching and mismatching bases.

Filters - filters for regions of low compositional complexity and repeat elements of the human's genome.

Masks for lookup table only — this option masks only for purposes of constructing the lookup table used by BLAST so that no hits are found based upon low-complexity sequence or repeats (if repeat filter is checked).

Mask lower case letters — with this option selected you can cut and paste a FASTA sequence in upper case characters and denote areas you would like filtered with lower case.

The view of the *Advanced options* tab depends on the selected search. For the *blastn* search, it looks like in the picture above. When the *blastx* search is selected in the general options, the view of the *Advanced options* tab is the following:



As you can see there is no *Match scores* option, but there are *Threshold*, *Matrix, Composition-based statistics* and *Service* options.

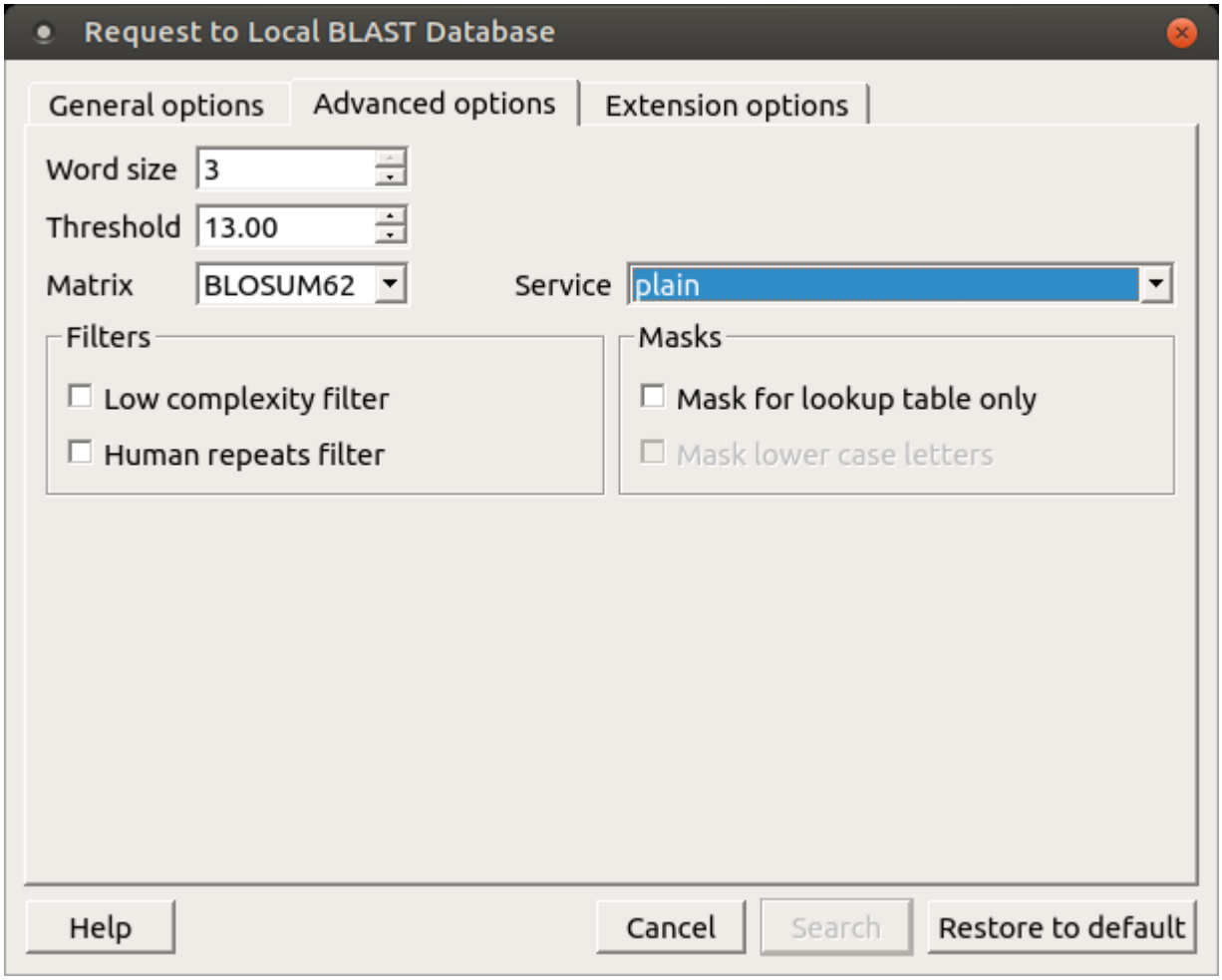
Threshold - threshold for extending hits.

Matrix — key element in evaluating the quality of pair-wise sequence alignment is the “substitution matrix”, which assigns a score for aligning any possible pair of residues.

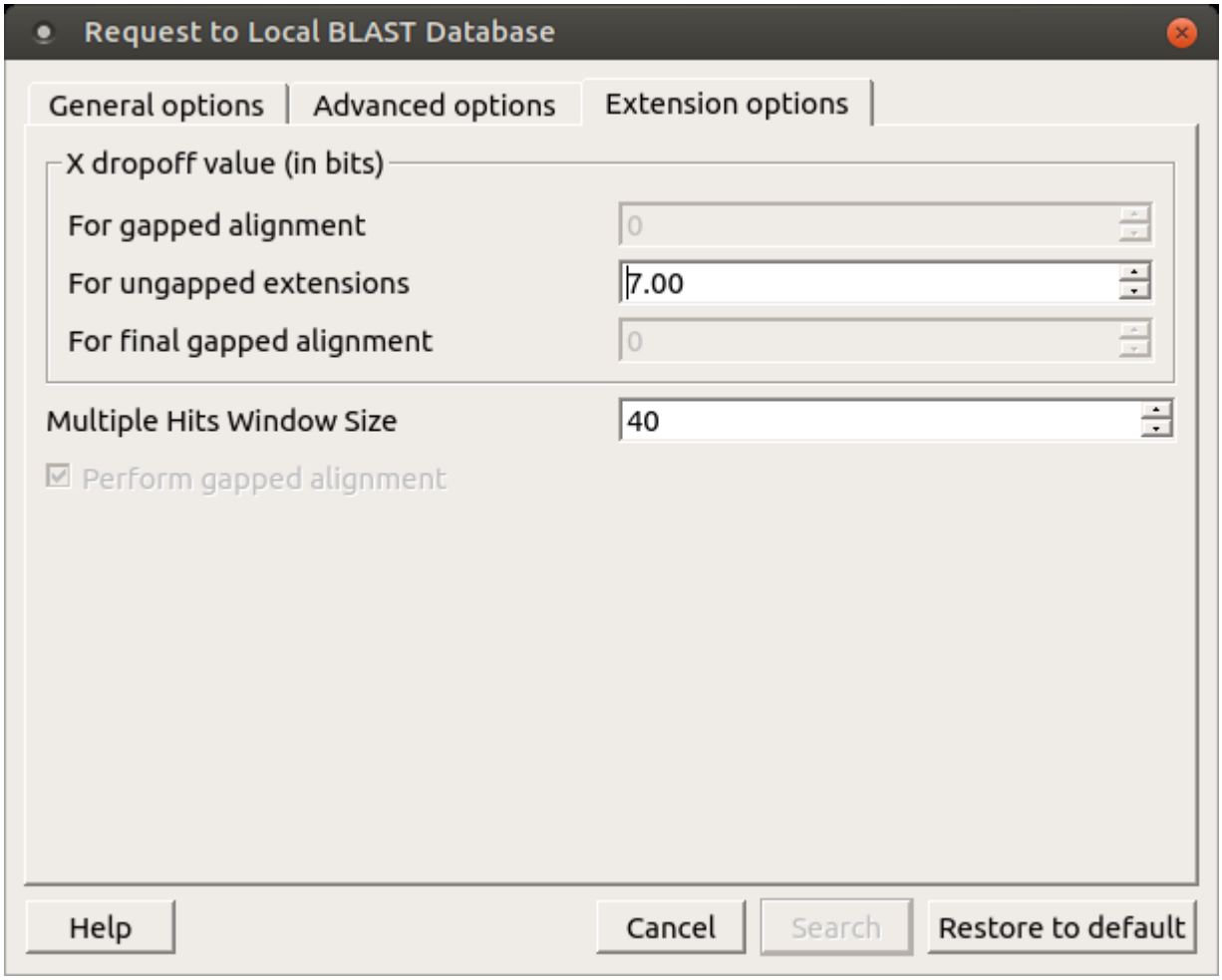
Service — blastp service which needs to be performed: plain, psi or phi.

Composition-based statistics - composition-based statistics.

When the *tblastx* search is selected in the general options, the view of the *Advanced options* tab is the following:



The following extension options are available:



For gapped alignment - X dropoff value (in bits) for gapped alignment.

For ungapped alignment - X dropoff value (in bits) for ungapped alignment.

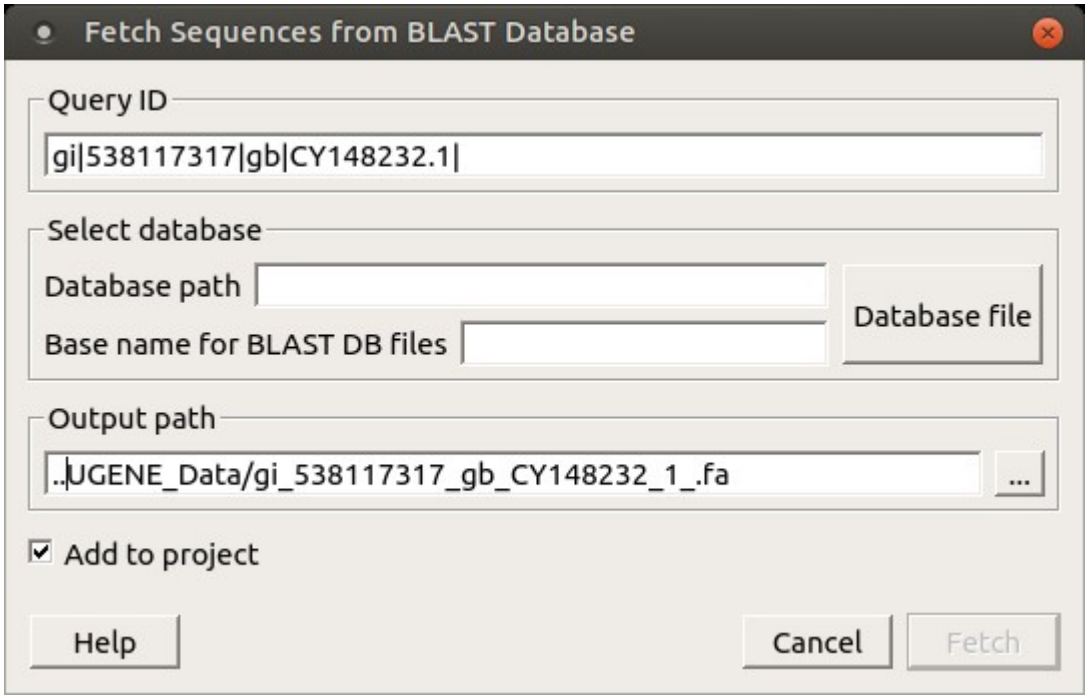
For final gapped alignment - X dropoff value (in bits) for final gapped alignment.

Multiple hits window size - multiple hits window size.

Perform gapped alignment - performs gapped alignment.

Fetching Sequences from Local BLAST+ Database

To fetch sequences from the local BLAST+ database use the *Fetch sequences from local BLAST+ database->Fetch sequences by 'id' from 'blast result'* co ntext menu item of the blast result. The following dialog will appear:



Here you need to select a query ID, database, type of file(s) and output path. After that click on the *Fetch* button. To fetch sequences for several annotations at the same time select the blast results with *Ctrl* key and call the *Fetch sequences from local BLAST database->Fetch sequences by 'id' from 'blast result'* context menu item.

Repeat Finder

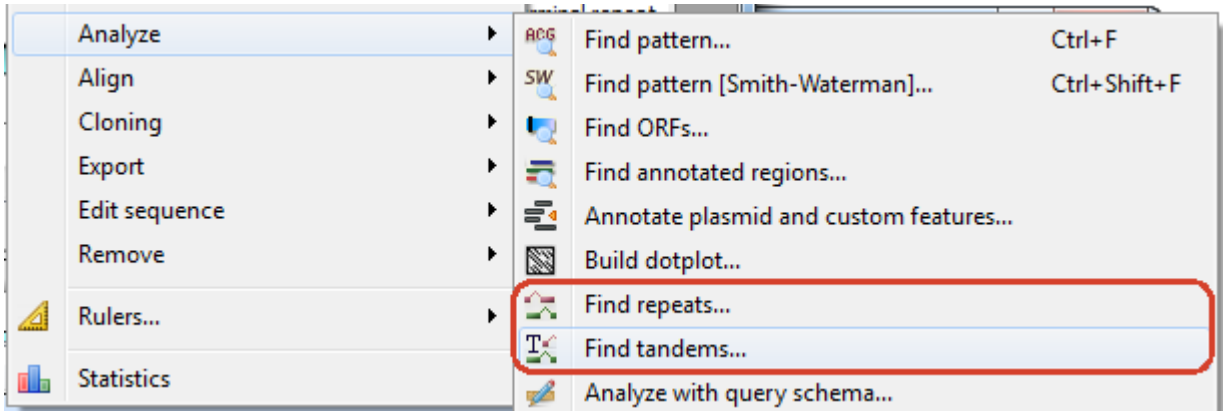
The *Repeat Finder* plugin provides a tool to search for direct and invert repeats in a DNA sequence. Also it allows to search for tandem repeats.

- Repeats Finding
- Tandem Repeats Finding
  - Tandem Repeats Search Result

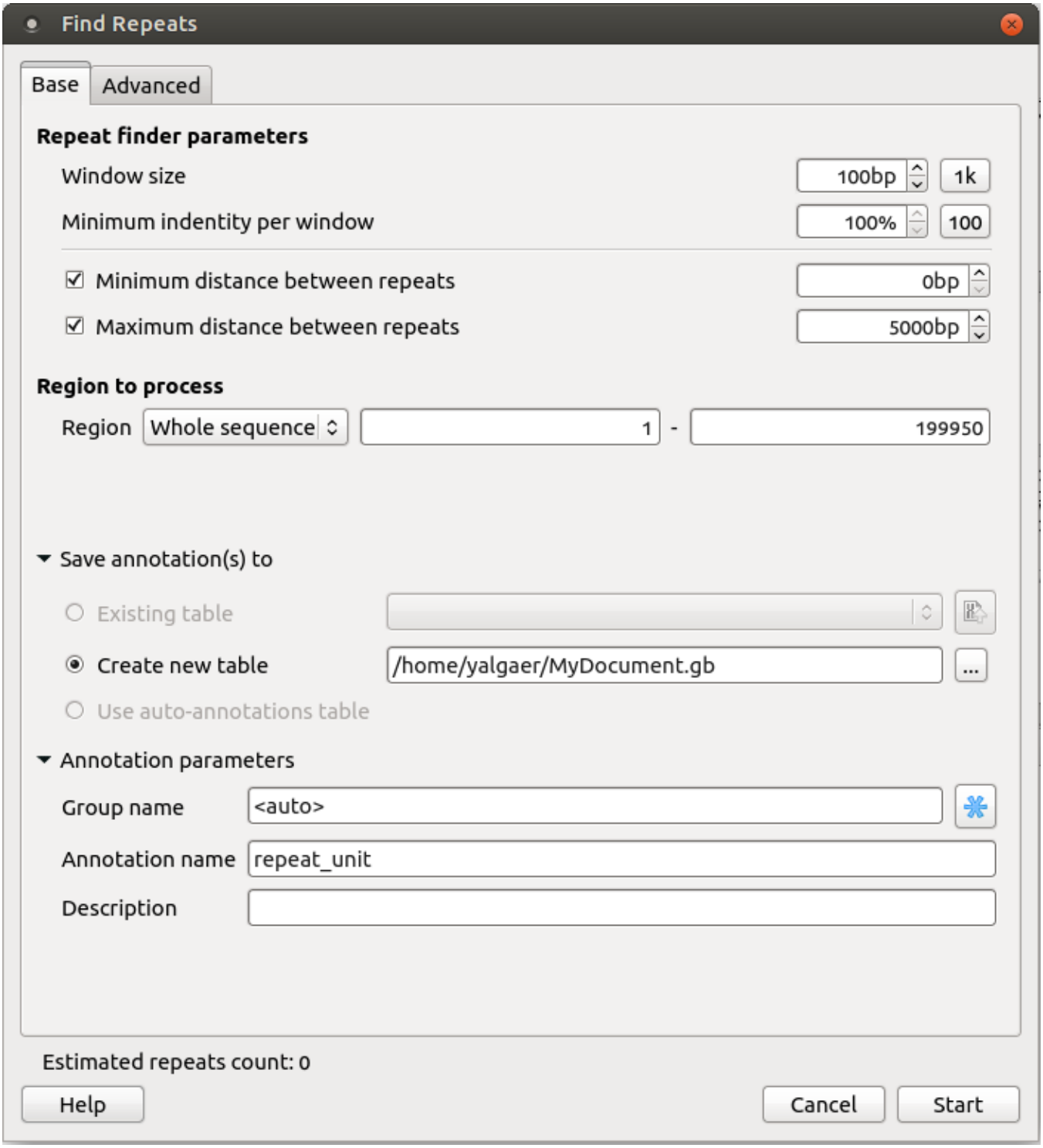
Repeats Finding

Usage example:

Open a DNA sequence in the *Sequence View* and select the *Analyze Find repeats...* context menu item:

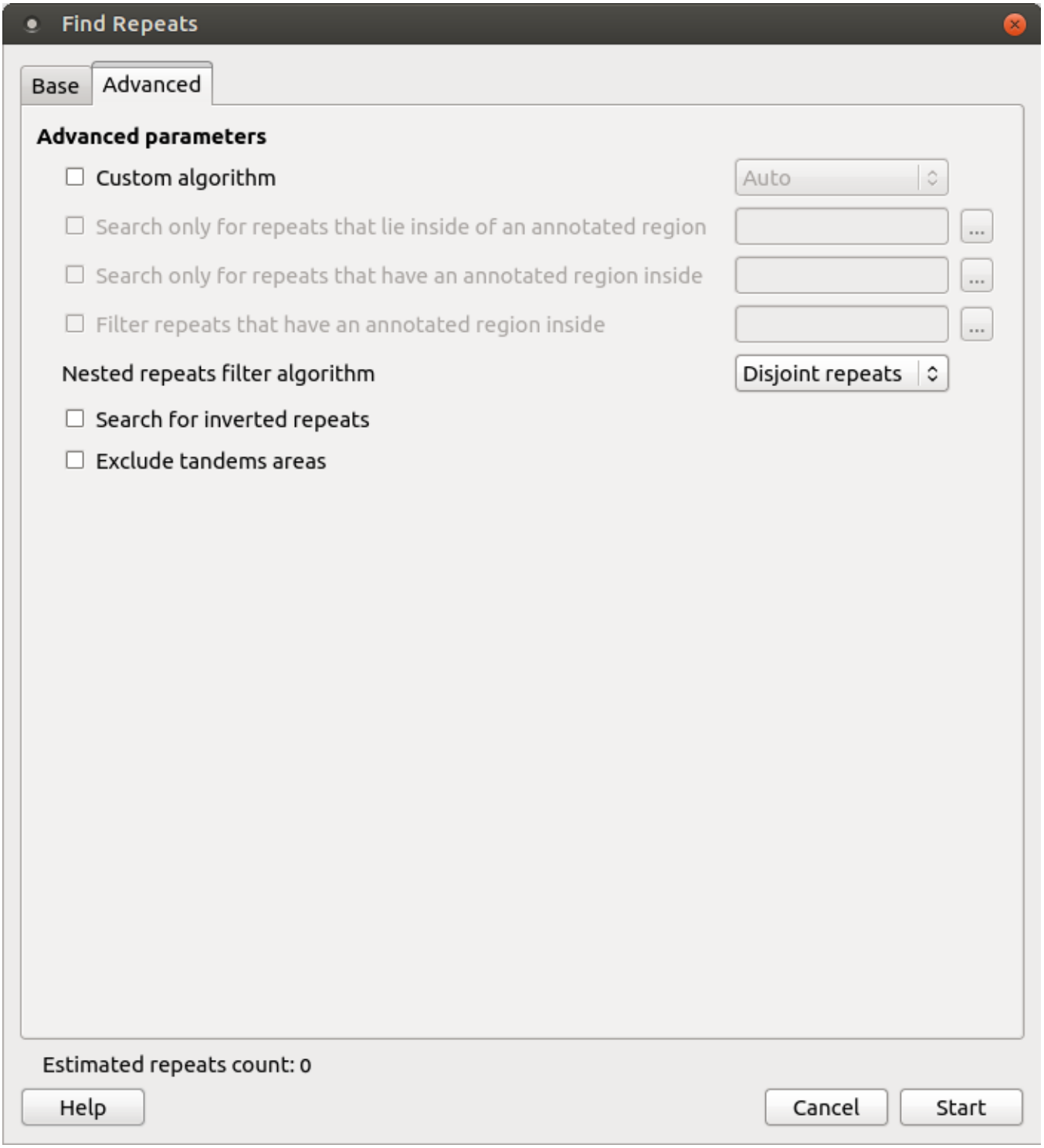


The dialog will appear that allows specifying repeat parameters and the annotations table document to save the results into:

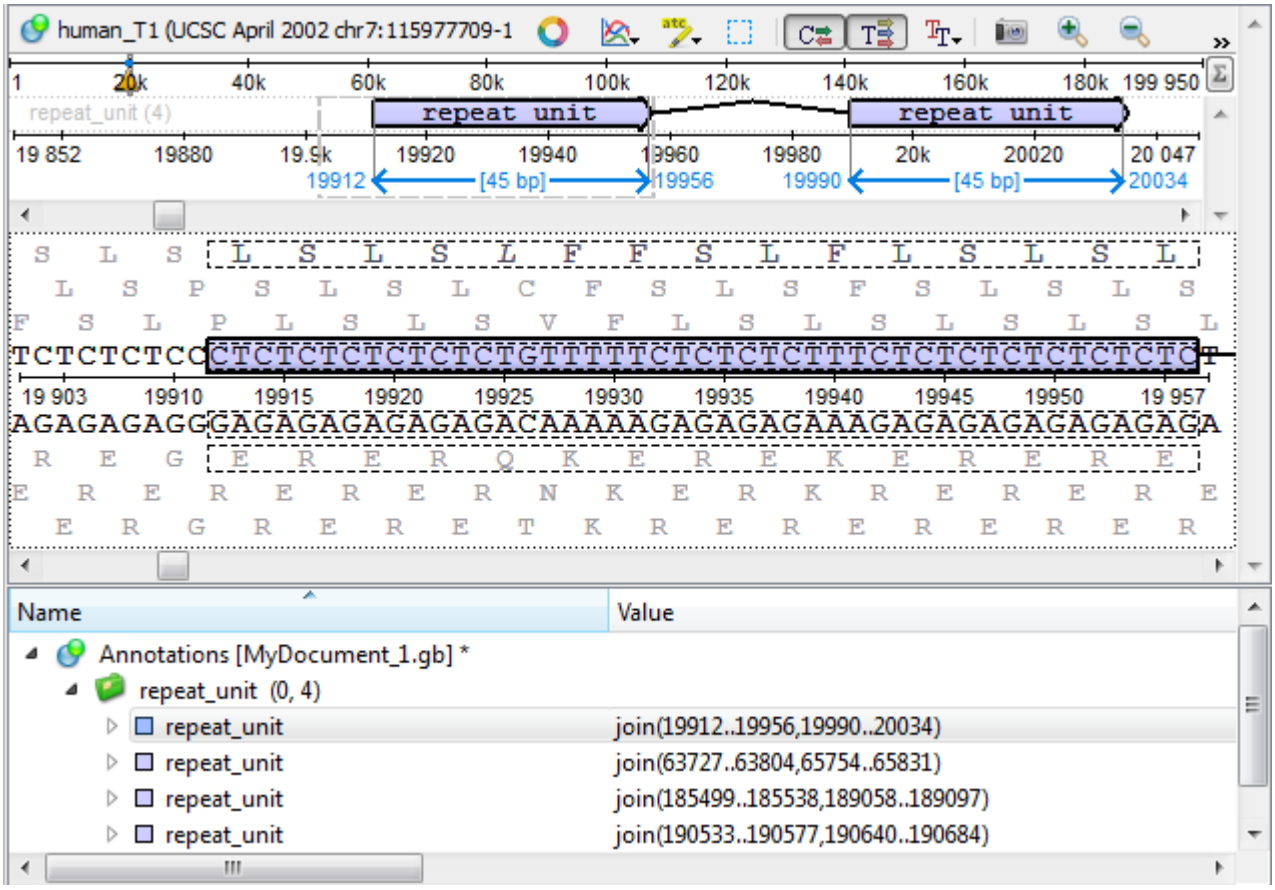


The dialogues status line displays approximate repeats number that will be found with the current settings.

The *Advanced* tab provides additional repeats finding options:



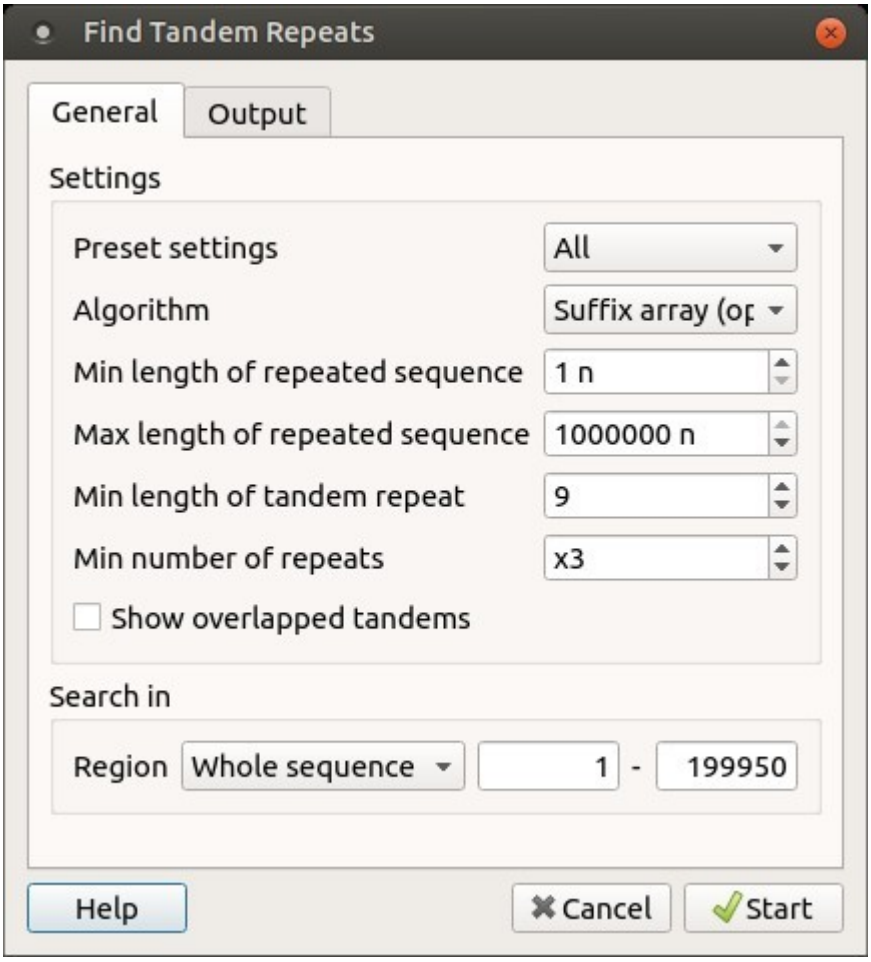
The found repeats are saved and displayed as annotations to the DNA sequence:



Tandem Repeats Finding

To find tandem repeats, select the *Analyze Find tandem repeats...* context menu item in the *Sequence View* window.

In the opened dialog you can specify the tandem search parameters, the region to search in and the result parameters:



The dialog parameters:

Present Settings — specify the tandem repeats parameters with predefined values by selecting the available preset.

Algorithm — the algorithm parameter allows to select the search algorithm. The default and a fast one is optimized suffix array algorithm.

Min length of repeated sequence - the minimum acceptable repeat length measured in base symbols.

Max length of repeated sequence - the mzximum acceptable repeat length measured in base symbols.

Min length of tandem repeat - the minimum tandem size sets the limit on the minimum acceptable length of the tandem, i.e. the minimum total repeats length of the searched tandem.

Min number of repeats - the minimum number of repeats of a searched tandem.

Show overlapped tandems - check if the plugin should search for the overlapped tandems, otherwise keep unchecked.

Search in - specify the region to search in the whole sequence, a custom region or the region of the current selection (if any).

The output settings can be found in the *Output* tab:

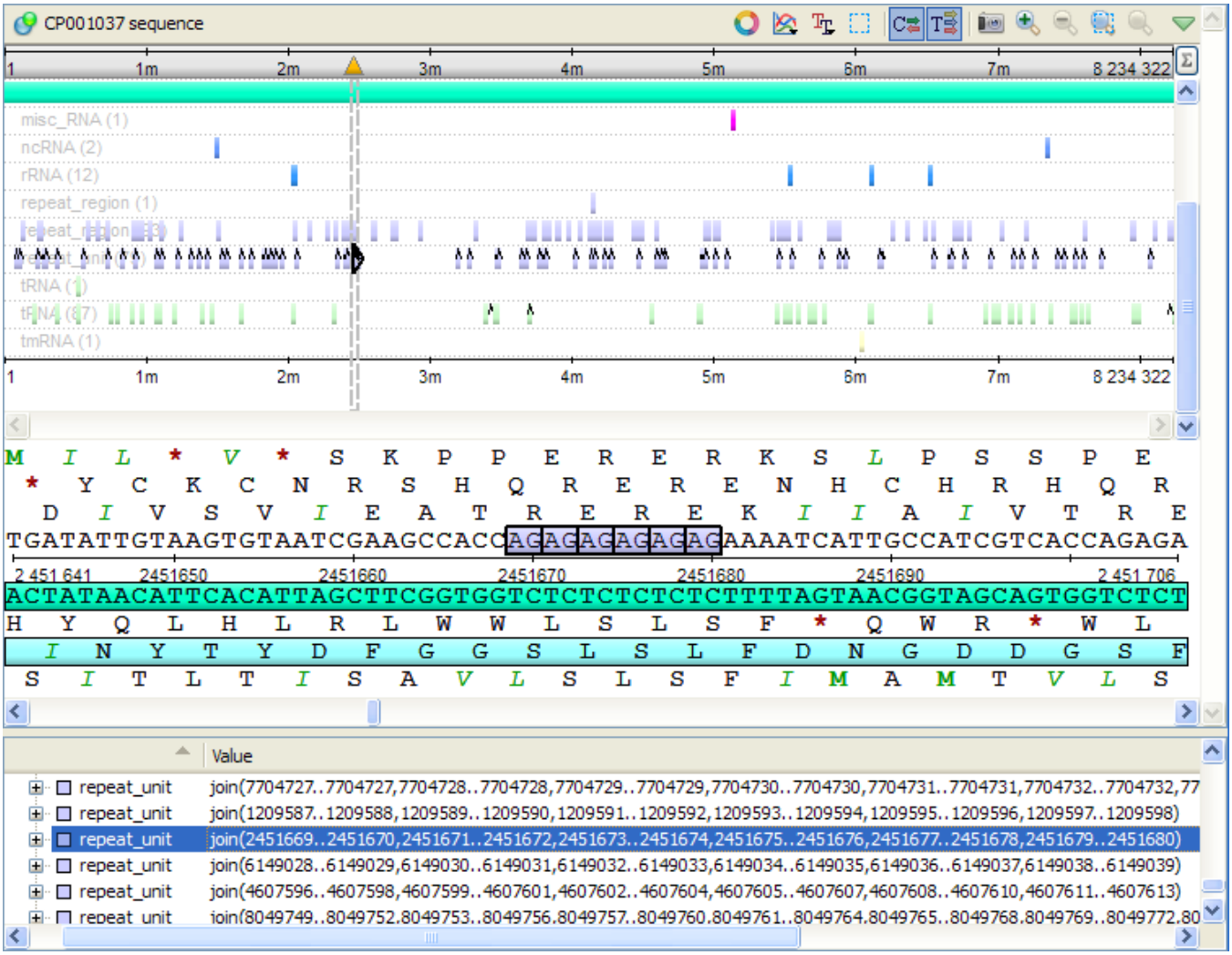
In the *Save annotation(s) to* group, you can set up a file to store annotations. It could be either an existing annotation table object, a new annotation table or auto-annotations table (if it is available).

In the *Annotation parameters* group, you can specify the name of the group and the name of the annotation. If the group name is set to <auto> UGENE will use the group name as the name for the group. You can use the '/' characters in this field as a group name separator to create subgroups. If the annotation name is set to *by type* UGENE will use the annotation type from the *Annotation type:* table as the name for the annotation. Also, you can add a description in the corresponding text field.

- Tandem Repeats Search Result

Tandem Repeats Search Result

An example of the search results for the micro-satellite preset:



The tandem repeats annotations are located side by side.

Restriction Analysis

From this chapter you can learn how to search for restriction sites on a DNA sequence.

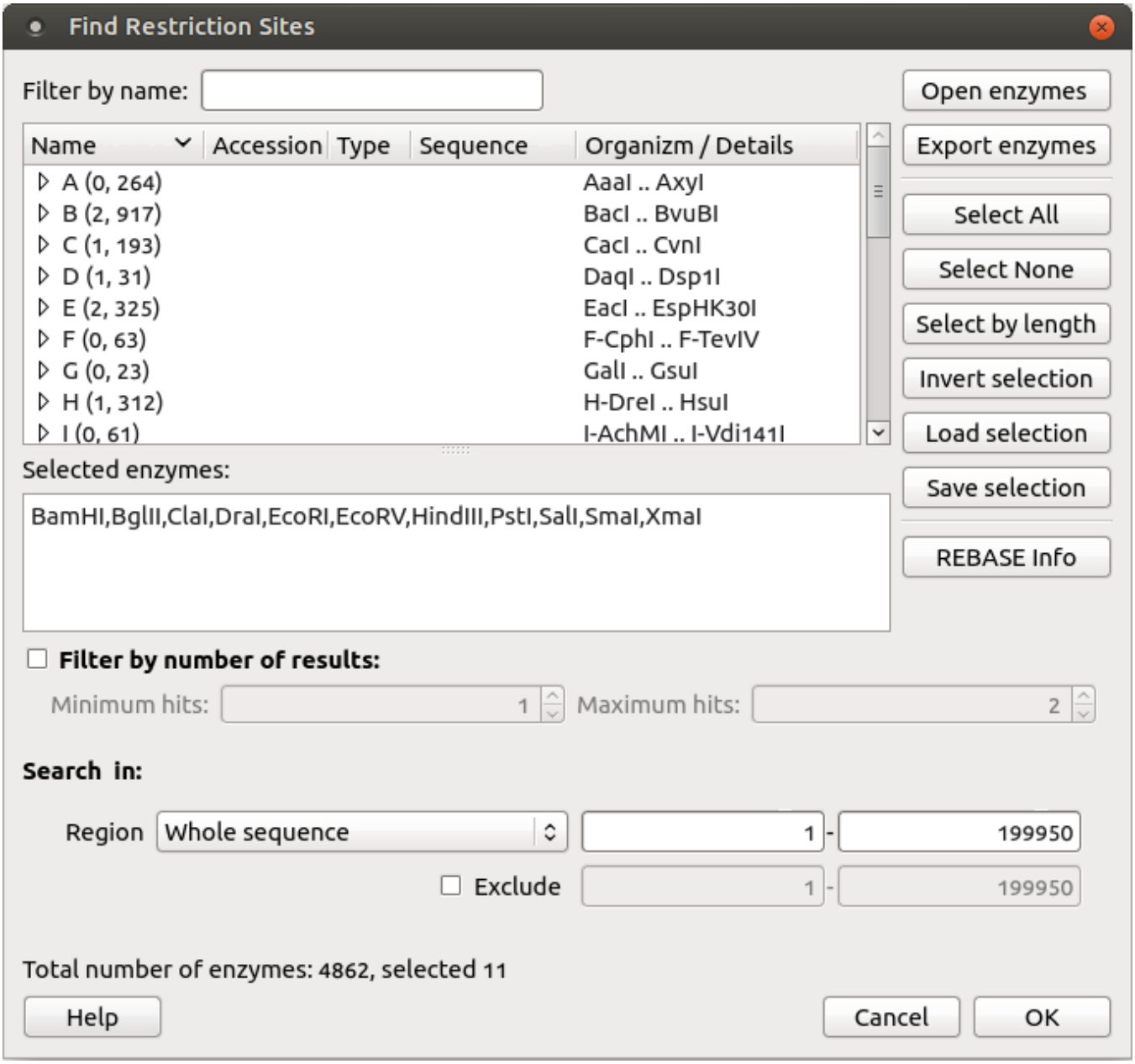
The restriction sites found are stored as automatic annotations. This means that if the automatic annotations highlighting is enabled then the restiction sites are searched and highlighted for each nucleotide sequence opened. Refer *Automatic Annotations Highlighting* to learn more.

Open a DNA sequence in and click the following button on the *Sequence View* toolbar:



Alternatively, select either the *Actions Analyze Find restriction sites* item in the main menu or the *Analyze Find restriction sites* item in the context menu.

The *Find restriction sites* dialog appears:



You can see the list of restriction enzymes that can be used to search for restriction sites. Also you can set a region to search for. The information about enzymes was obtained from the REBASE database. For each enzyme in the list a brief description is available (the accession ID in the database, the recognition sequence, etc.). If you're online you can get more detailed information about an enzyme selected by clicking the *REBASE Info* button.

- Selecting Restriction Enzymes
- Using Custom File with Enzymes
- Filtering by Number of Hits
- Excluding Region
- Circular Molecule
- Results

Selecting Restriction Enzymes

To select an enzyme check it in the list. Notice that the enzyme appears in the *Selected enzymes* area of the dialog.

You can also use the *Select All* button to select all the enzymes available, the *Select None* button to deselect all the enzymes.

To select all enzymes with recognition sequence length shorter than the specified value click the *Select by length* button and input the minimum length in the dialog appeared.

To invert selection click the *Invert selection* button.

As soon as enzymes are selected you can click the *OK* button to search for corresponding restriction sites in the sequence.

Using Custom File with Enzymes

To load a custom file with enzymes click the *Enzymes file* button and browse for the file. The file must be of the Bairoch format.

For details about the format refer http://rebase.neb.com/rebase/rebase.f19.html.

To export enzymes use the *Export enzymes* button. You can also save the currently selected enzymes to a file and load saved selection. Click the *Save selection* and *Load selection* buttons correspondingly to do it.

Filtering by Number of Hits

To filter the results by the number of restriction sites found for an enzyme check the *Filter by number of results* check box and input the minimum value and the maximum value of hits.

Excluding Region

To exclude a sequence region from the search check the *Exclude region* check box and input the start and the end positions of the region. If a subsequence has been selected before opening the dialog you can click the *Selected* button to automatically fill the values with the selected subsequence' s start and end positions.

Circular Molecule

To consider the sequence as circular and be able to search for restriction sites between the end and the beginning of the sequence check the *Circular molecule* option.

Example: Let's consider:

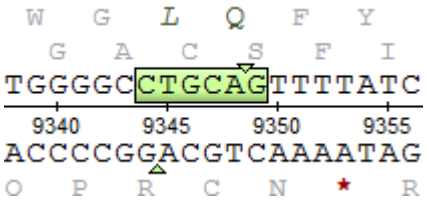
- The sequence is "CTGC ... CAC".
- *AarI* restriction enzyme (with recognition sequence "CACCTGC") has been checked.

In this case if the *Circular molecule* option has been checked, the restriction site will be found. If it hasn't been checked, the restriction site won't be found (in this position).

Results

When at least one enzyme has been selected and the *OK* button has been pressed in the dialog, the *auto-annotating* becomes enabled. In the *Annotations editor* the Restriction Sites annotations can be found in the Auto-annotations\enzyme group.

The direct and complement cut site positions are visualized as triangles on an annotation in the *Sequence details view*:



Molecular Cloning in silico

This chapter describes a set of tools in UGENE to perform molecular cloning experiments *in silico*.

This allows you to digest a molecule into fragments, create a fragment from a sequence region and ligate fragments into a new molecule.

- Digesting into Fragments
- Creating Fragment
- Constructing Molecule
  - Available Fragments
  - Fragments of the New Molecule
  - Changing Fragments Order in the New Molecule
  - Removing Fragment from the New Molecule
  - Editing Fragment Overhangs
  - Reverse Complement a Fragment
  - Other Constuction Options
  - Output
- Creating PCR Product

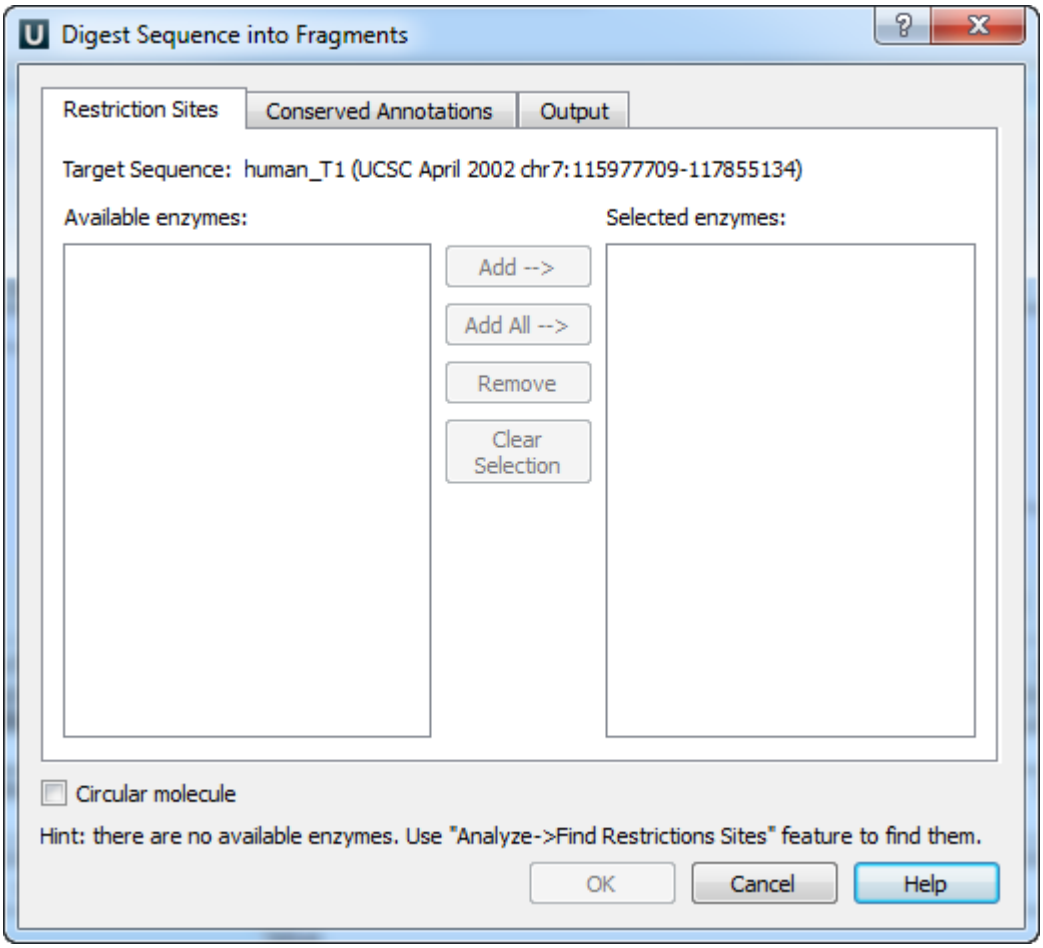
Digesting into Fragments

Open a DNA molecule you want to cut into fragments.

Digestion into fragments is performed using restriction enzymes. So before continuing make sure that the restriction analysis has been performed. Refer chapter *Restriction Analysis* for details.

Select either the *Tools Cloning Digest into Fragments* item or the *Actions Cloning Digest into Fragments* item in the main menu or the *Cloning Digest into Fragments* item in the context menu.

The *Digest Sequence into Fragments* dialog appears:



On the *Restriction Sites* tab of the dialog you can see the name of the molecule, the list of restriction enzymes found during the restriction analysis that can cut the molecule and the list of enzymes selected to perform the digestion.

To digest the sequence into fragments you should select at least one enzyme.

To move an enzyme to the *Selected enzymes* list click on it in the *Available enzymes* list and press the *Add* button. Note that you can select several items in a list by holding the Ctrl key while clicking on the items.

To select all available enzymes press the *Add All* button.

To remove enzymes from the *Selected enzymes* list select them in the list and press the *Remove* button.

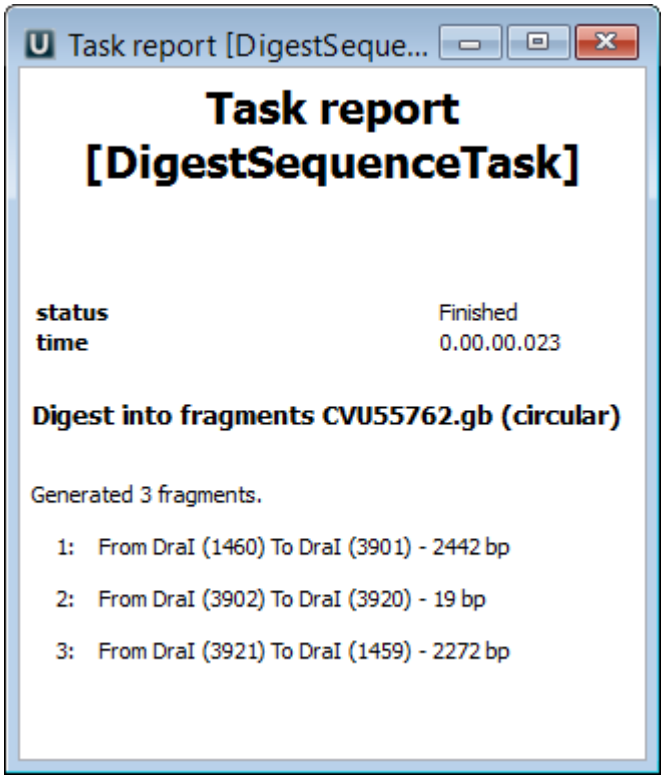
To remove all items from the *Selected enzymes* list press the *Clear Selection* button.

On the *Conserved Annotations* tab of the dialog you can select the annotations that must not be disrupted during cloning.

On the *Output* tab of the dialog you can select the file to save the new molecule to.

As soon as the required parameters are selected press the *OK* button. The fragments will be saved as annotations.

Also all the generated fragments are available in the task report:

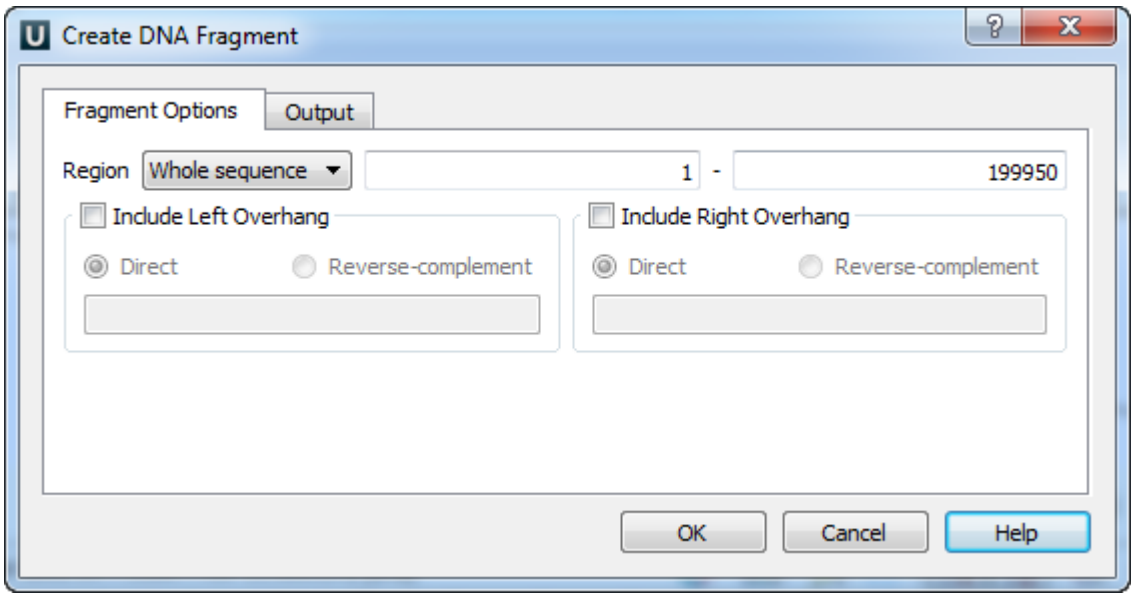


Refer to *Notifications* to learn more about task reports.

Creating Fragment

To create a DNA fragment from a sequence region activate the *Sequence View* window and select either the *Actions Cloning Create Fragment* item in the main menu or the *Cloning Create Fragment* item in the context menu.

The *Create DNA Fragment* dialog appears:



If a region has been selected you can choose to create the fragment from this region. Otherwise you can either choose to create the fragment from the whole sequence or choose the *Custom* item and input the custom region.

To add a 5' overhang to the direct strand check the *Include Left Overhang* check box and input the required nucleotides. To add a 5' overhang to the reverse strand in addition to the described steps select the *Reverse-complement* item in the same group box.

Similarly, to add a 3' overhang check the *Include Right Overhang* check box, input the required overhang and select either the direct or the reverse-complement strand.

On the *Output* tab of the dialog you can optionally modify the annotations output settings.

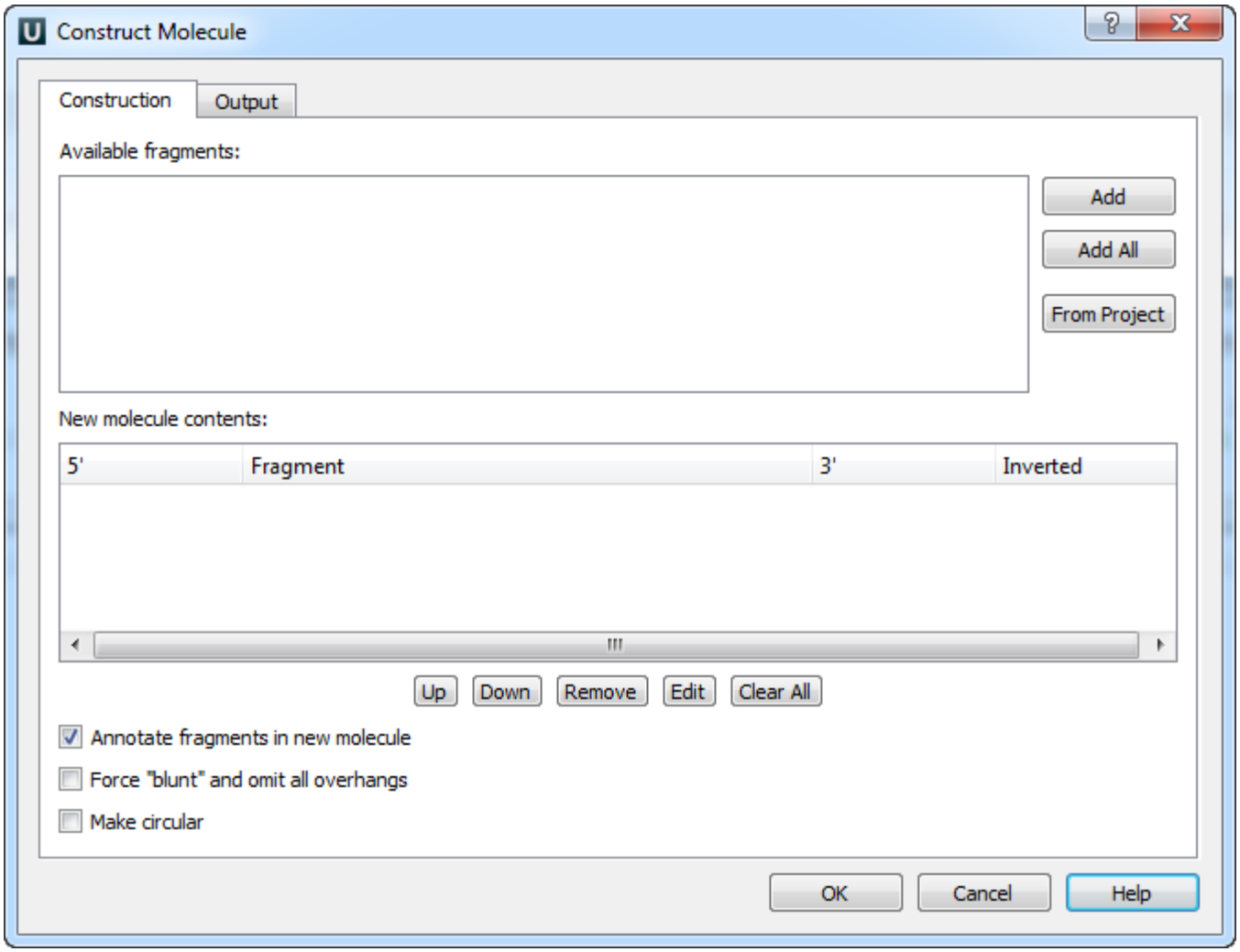
Finally, press the *OK* button to create the fragment. The fragment will be saved as an annotation.

Constructing Molecule

To construct a new molecule from fragments select the *Tools Cloning Construct Molecule* item in the main menu.

If a *Sequence View* window is active you can also select either the *Actions Cloning Construct Molecule* item in the main menu or the *Cloning Construct Molecule* item in the context menu.

The *Construct Molecule* dialog appears:



- Available Fragments
- Fragments of the New Molecule
- Changing Fragments Order in the New Molecule
- Removing Fragment from the New Molecule
- Editing Fragment Overhangs
- Reverse Complement a Fragment
- Other Constuction Options
- Output

Available Fragments

All the fragments available in the current project are shown in the *Available fragments* list.

You can automatically create a fragment from a DNA molecule from the current UGENE *project*. Click the *From Project* button to do so. The *Select Item* dia log appears with the sequence objects available. Select a sequence and press the *OK* button. After that create a fragment in the appeared *Create DNA Fragment* dialog as described in the *Creating Fragment* paragraph. The fragment created from the sequence appears in the list of available fragments.

Fragments of the New Molecule

The next step is to add required fragments to the new molecule contents.

To add fragments select them in the list of available fragments and click the *Add* button or by double-click on a fragment.

To add all the fragments click the *Add All* button.

Changing Fragments Order in the New Molecule

To change the order of fragments in the new molecule select a fragment in the new molecule contents list and click either the *Up* or the *Down* button to move the fragment in the corresponding direction.

Removing Fragment from the New Molecule

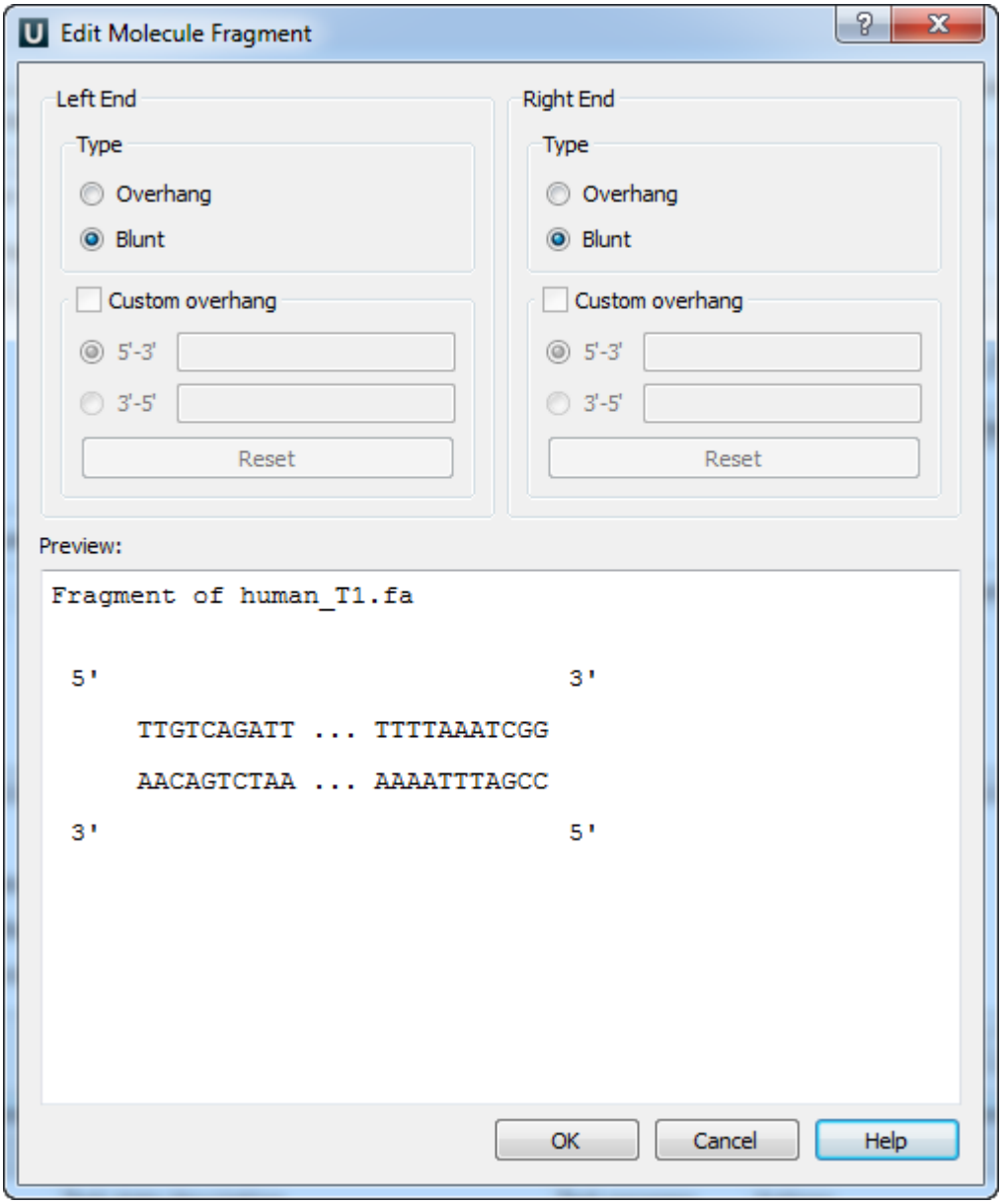
To remove a fragment from the new molecule select it in the new molecule contents list and click the *Remove* button.

To remove all the fragments click the *Clear All* button.

Editing Fragment Overhangs

To edit a fragment's overhangs select the fragment in the new molecule contents list and click the *Edit* button.

The *Edit Molecule Fragment* dialog appears:



Here you can select the type of each DNA end and even input a custom overhang.

The changes you've made are shown in the *Preview* area of the dialog.

To confirm the changes and close the dialog click the *OK* button.

Reverse Complement a Fragment

To reverse complement a fragment check the *Inverted* check box for the fragment in the new molecule contents list.

Other Constuction Options

To save the fragments of the new molecule as annotations check the *Annotate fragments in new molecule* check box.

To make all DNA ends blunt check the *Force "blunt" and omit all overhangs* check box. All overhangs would be cut in this case.

Check the *Make circular* check box to make the new molecule circular.

Output

On the *Output* tab of the dialog you can select the file to save the new molecule to.

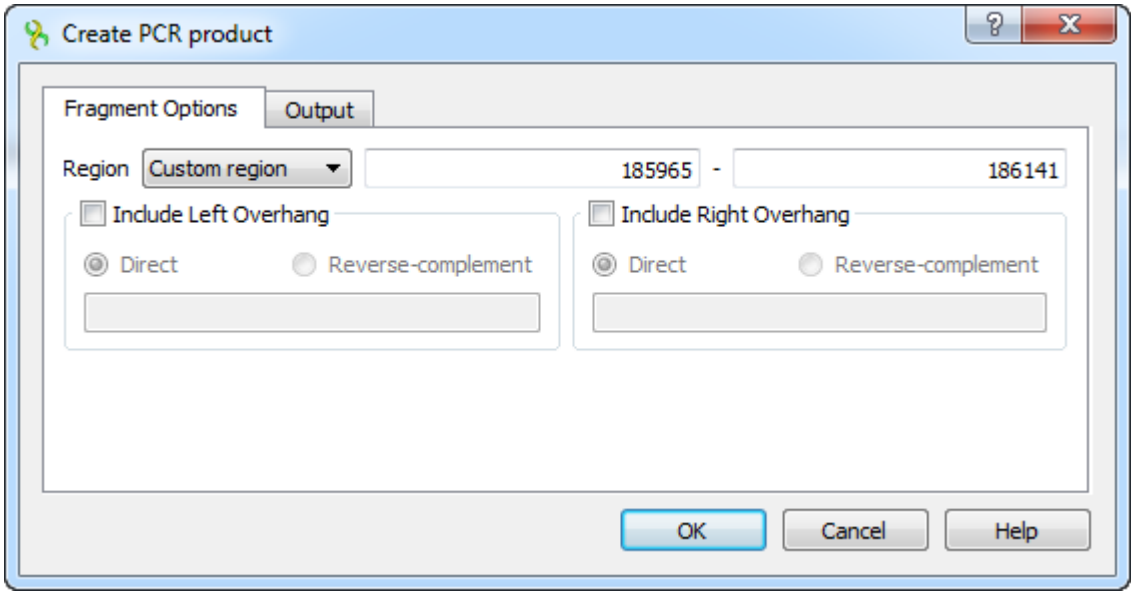
The molecule is opened by default as soon as it is created. To modify this behavior uncheck the *Open view for new molecule* check box on the same tab.

To save the molecule file to the hard disk immediately after it is created check the *Save immediately* check box. Otherwise it would be stored in memory until you save or remove it.

Creating PCR Product

To create a PCR product from a primer use the *Cloning->Create PCR product* context menu of primer annotation.

The *Create PCR Product* dialog appears:



If a primer has been selected you can choose to create the PCR product from this primer. Otherwise you can either choose to create the PCR from the whole sequence or choose the *Custom* item and input the custom region.

To add a 5' overhang to the direct strand check the *Include Left Overhang* check box and input the required nucleotides. To add a 5' overhang to the reverse strand in addition to the described steps select the *Reverse-complement* item in the same group box.

Similarly, to add a 3' overhang check the *Include Right Overhang* check box, input the required overhang and select either the direct or the reverse-complement strand.

On the *Output* tab of the dialog you can optionally modify the annotations output settings.

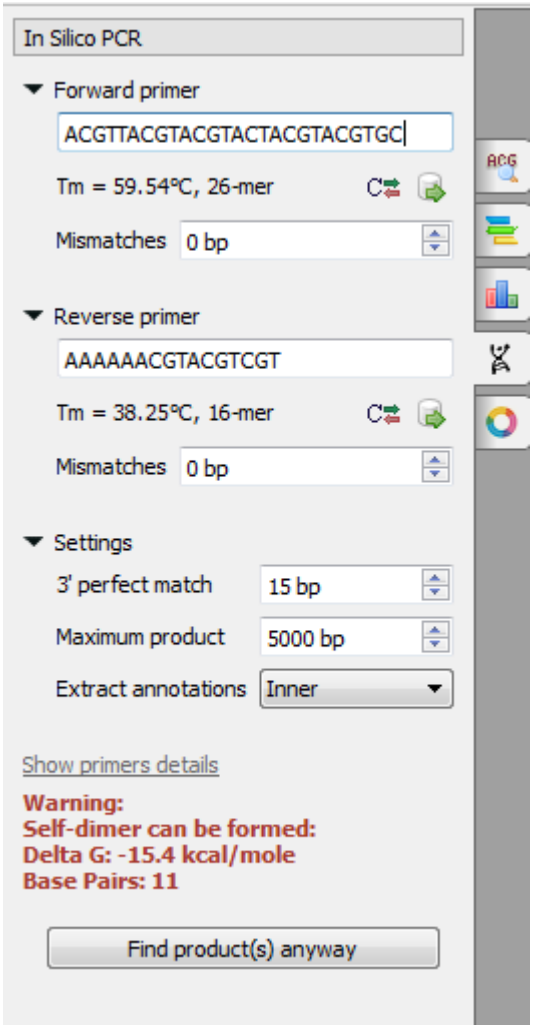
Finally, press the *OK* button to create the PCR product. The PCR product will be saved as an annotation.

In Silico PCR

In Silico PCR Overview

In silico PCR is used to calculate theoretical polymerase chain reaction (PCR) results using a given set of primers (probes) to amplify DNA sequences.

UGENE provides the In silico PCR feature only for nucleic sequences with Standard DNA and Extended DNA alphabets. To use it in UGENE open a DNA sequence and go to the *In silico PCR* tab of the Options Panel:



There are the following parameters:

Forward primer - forward primer.

Reverse primer - on the opposite strand from the forward primer.

Mismatches - mismatches limit.

3' perfect match - specify the number of nucleotides at the 3' end that must not have mismatches.

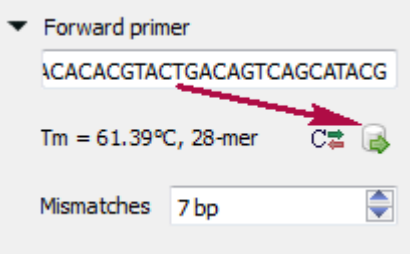
Maximum product - maximum size of the amplified sequence.

Extract annotations - specify the type of extracted annotations: *Inner, All intersected* or *None*.

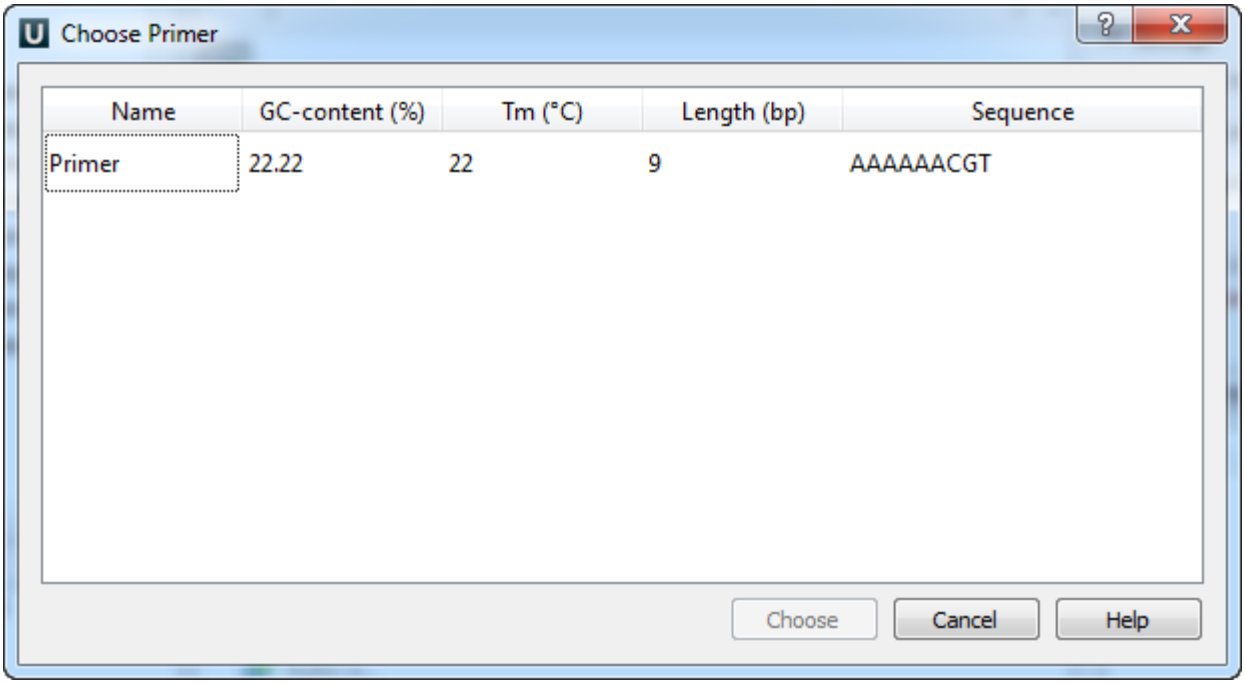
- Value *Inner* is selected by default. When this value is selected, the extracted PCR product contains annotations from the original sequence, located within the extracted region.
- Value *All intersected* specifies that all annotations of the original sequence that intersect the extracted region must be extracted as well.
- Value *None* specifies that annotations from the original sequence must not be extracted.

Choosing primers

Type two primers for running In Silico PCR. If the primers pair is invalid for running the PCR process then the warning is shown. Also, primers for the running In silico PCR can be chosen from a primer library. Click the following button to choose a primer from the primers library:

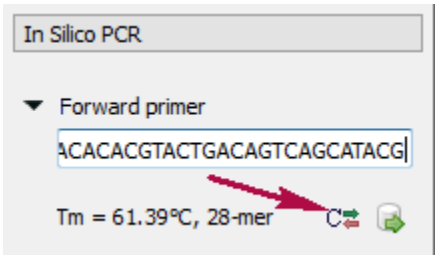


The following dialog will appear:



The table consists of the following columns: name, GC-content (%), Tm, Length (bp) and sequence. Select primer in the table and click the *Choose* button.

Click the *Reverse-complement* button for making a primer sequence reverse-complement:



Click *Show primers details* for seeing statistic details about primers.

When you run the process, the predicted PCR products appear in the products table.

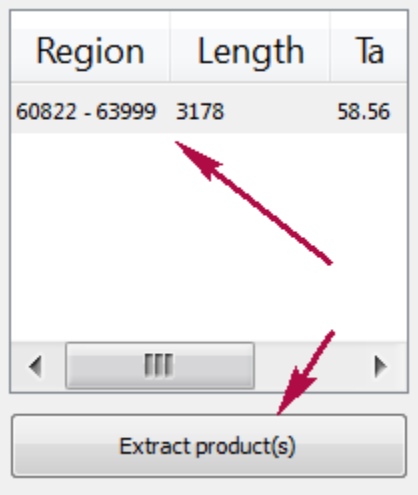
Products table

There are three columns in the table:

- region of product in the sequence
- product length
- preferred annealing temperature

Click the product for navigating to its region in the sequence.

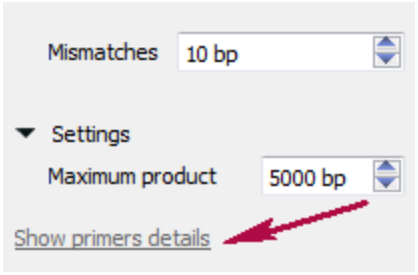
Click the *Extract product(s)* button for exporting a product(s) in a file or use double click for that.



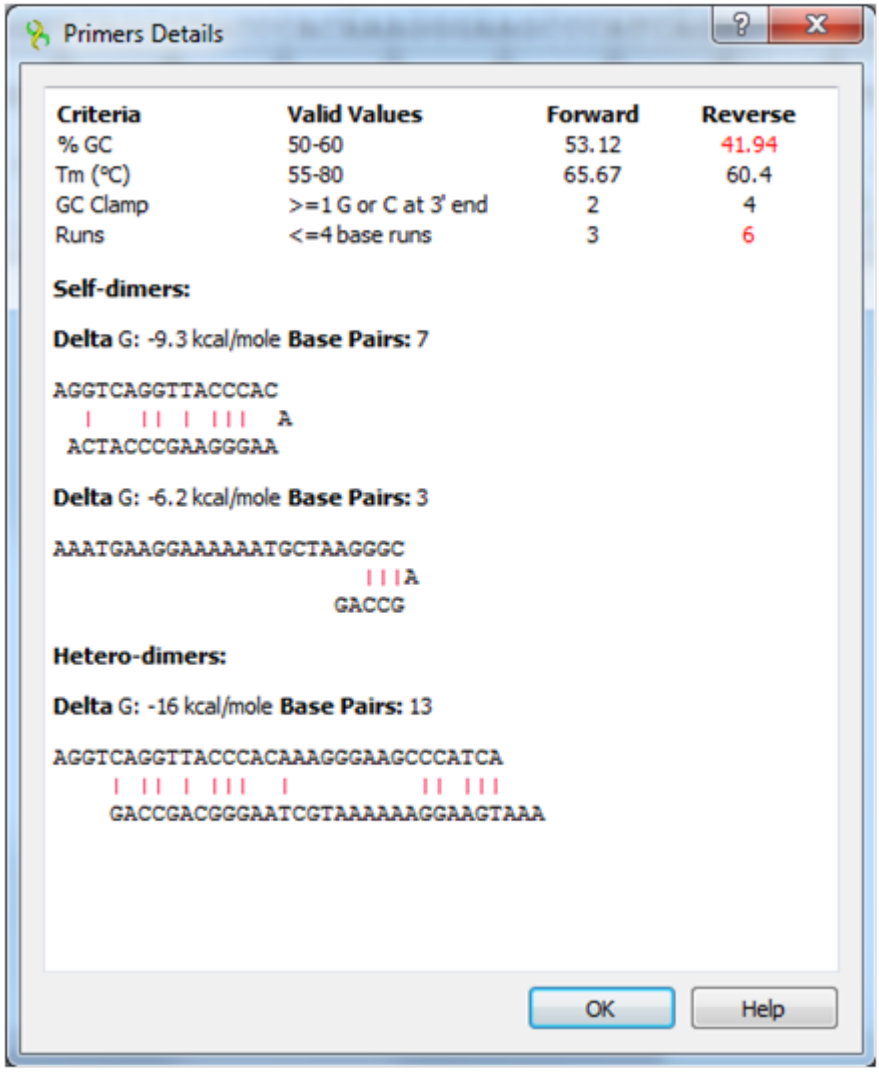
- Primers Details
- Primer Library

Primers Details

Click *Show primers details* for seeing statistic details about primers.



The following dialog will appear:

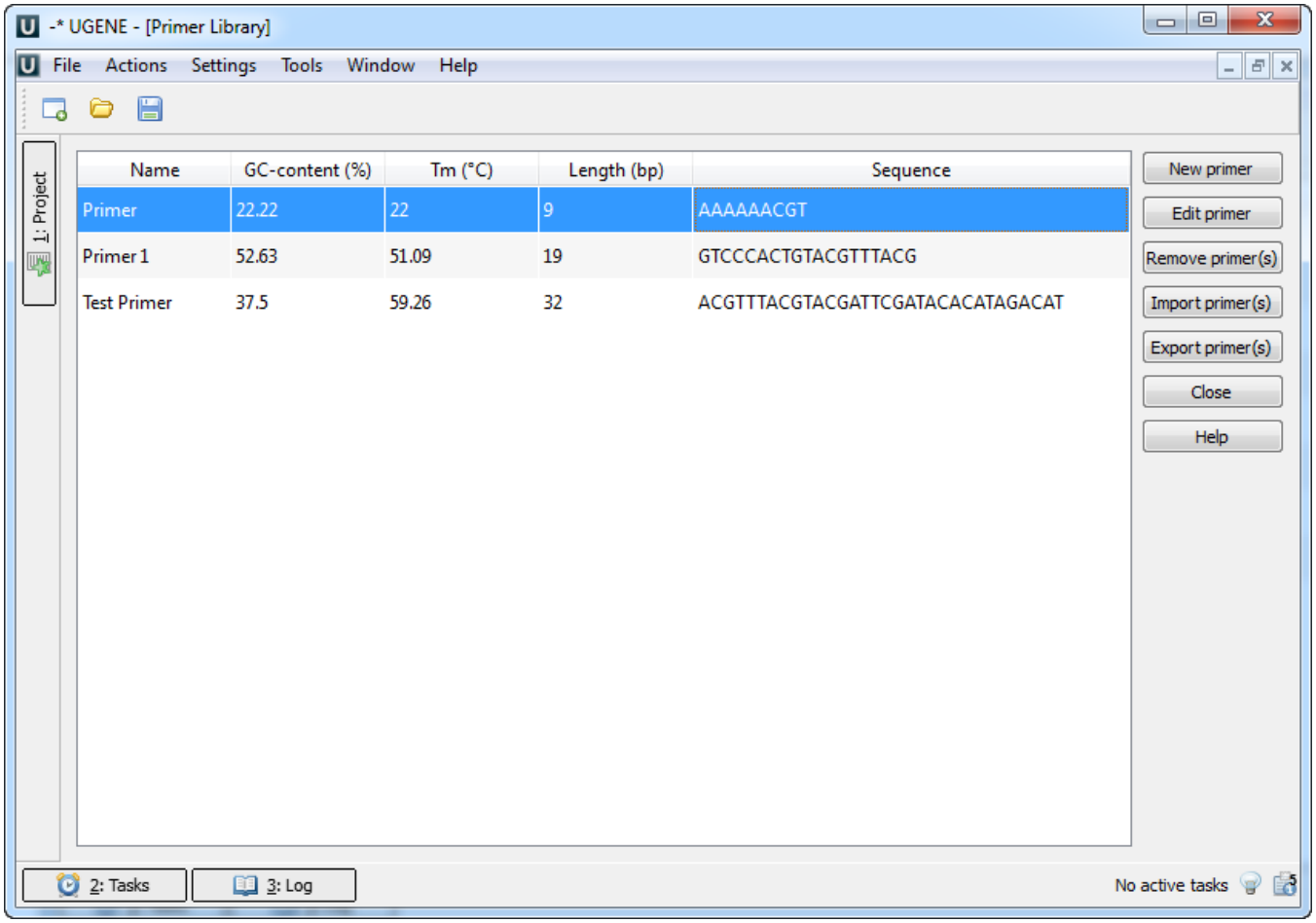


This is a dialog with statistic details about primers: melting temperature, GC content, dimers, self-dimers, etc. If a value is not correct for its criteria then it is colored in red.

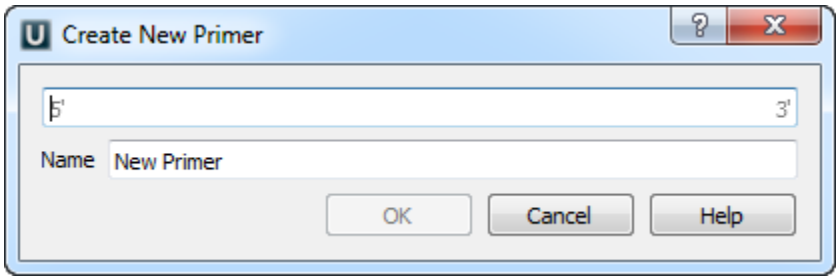
Primer Library

The primer library is a storage for keeping user primers. The added primers are stored between UGENE sessions.

Go to the *Tools->Primer->Primer library* context menu to configure the primer library. The following window will appear:



Click the *New primer* button to add a new primer. The following dialog will appear:

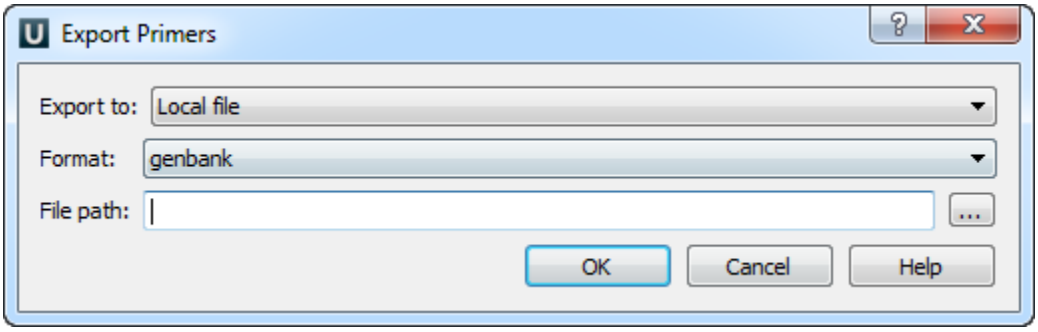


Input the primer sequence and primer name and click on the *OK* button.

Select the primer and click the *Edit primer* button to edit primer.

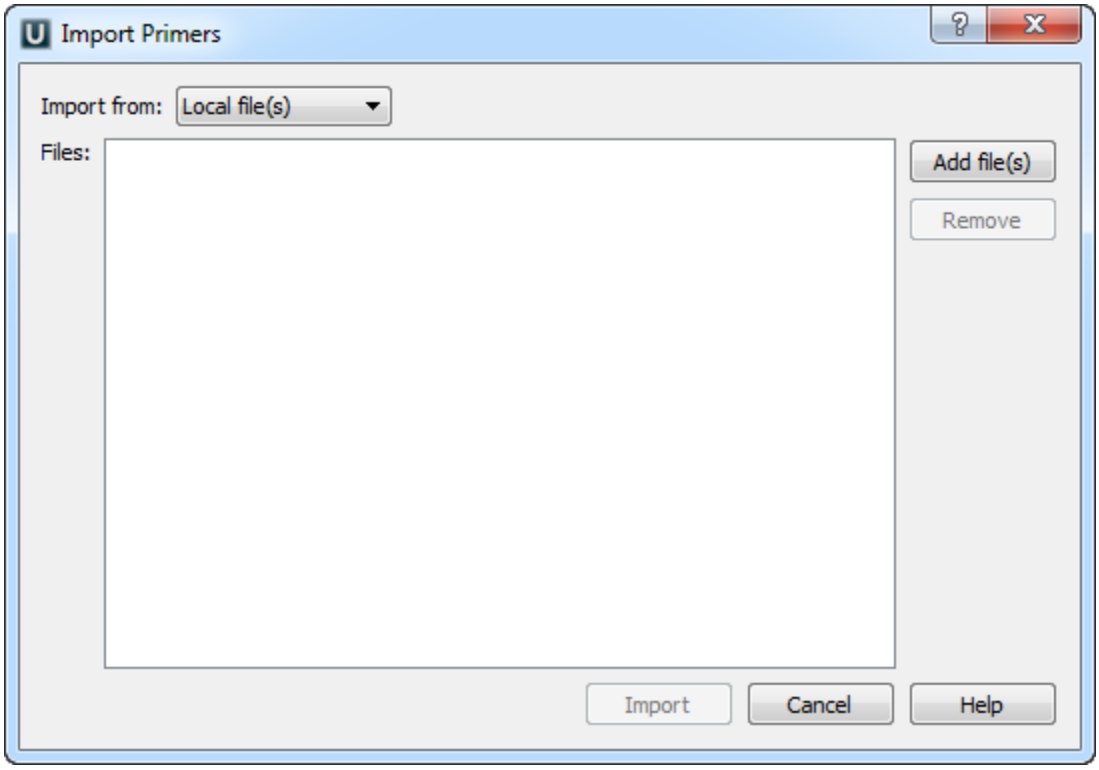
Select the primer in the table (you can use Ctrl and Shift) and click *Remove primer(s)* button to remove primer(s).

To export primer(s) select it and click the *Export primer(s)* button. The following dialog will appear:



Select file and file format and click on the *OK* button.

To import primer(s) click the *Import primer(s)* button. The following dialog will appear:



Add one or several files with primer sequences. Note that all sequence formats, supported by UGENE, can be imported, for example, FASTA, GenBank, etc. But the sequences must consist of ACGT characters only.

Click the *Import* button to import the added files into the primers library.

Secondary Structure Prediction

The *Secondary Structure Prediction* plugin provides a set of algorithms for the protein secondary structure (alpha-helix, beta-sheet) prediction from a raw sequence.

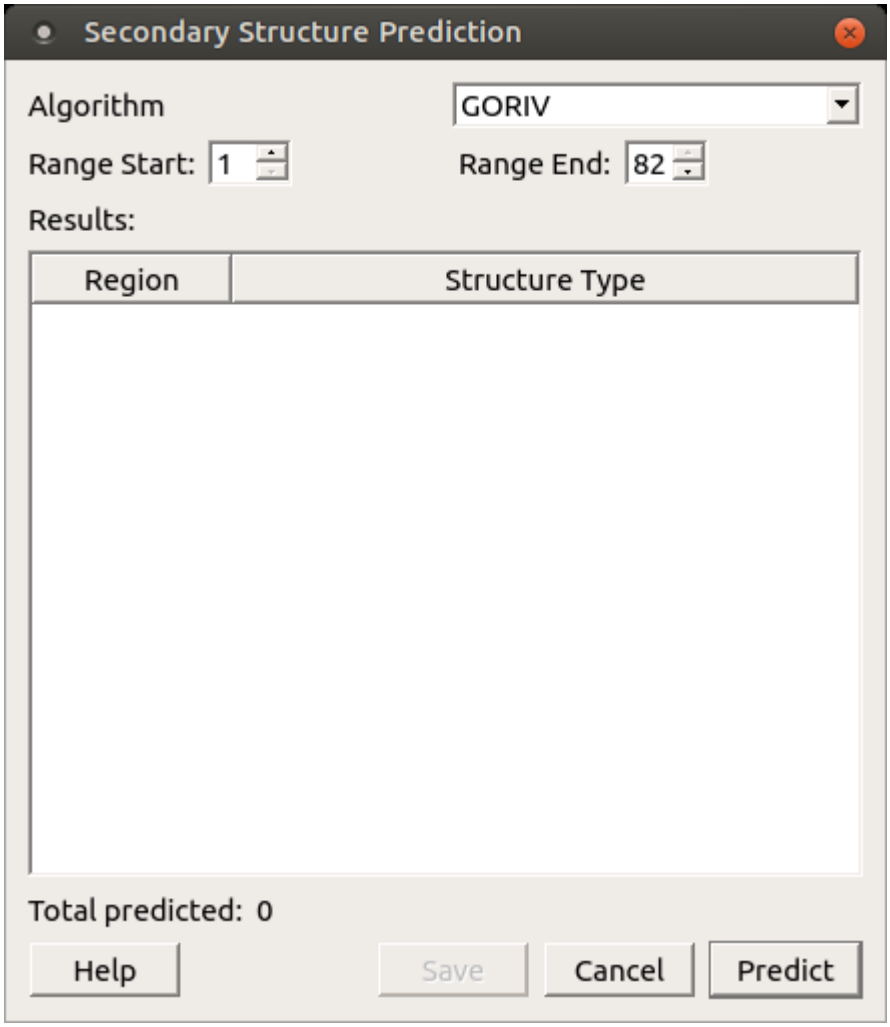
Currently, available algorithms are:

- **GORIV** Jean Garnier, Jean-Francois Gibrat, and Barry Robson,"GOR Method for Predicting Protein Secondary Structure from Amino Acid Sequence", in Methods in Enzymology, vol.266, pp. 540 - 553, (1996).

  The improved version of the GOR method in J. Garnier, D. Osguthorpe, and B. Robson, J. Mol. Biol., vol. 120, p. 97 (1978).
- **PsiPred** Bryson K, McGuffin LJ, Marsden RL, Ward JJ, Sodhi JS. & Jones DT. (2005) Protein structure prediction servers at University College London. Nucl. Acids Res. 33(Web Server issue): W36-38.

  Jones DT. (1999) Protein secondary structure prediction based on position-specific scoring matrices. J. Mol. Biol. 292: 195-202.

You can access these analysis capabilities for a protein sequence using the *Analyze Predict secondary structure...* context menu item. The dialog will appear:

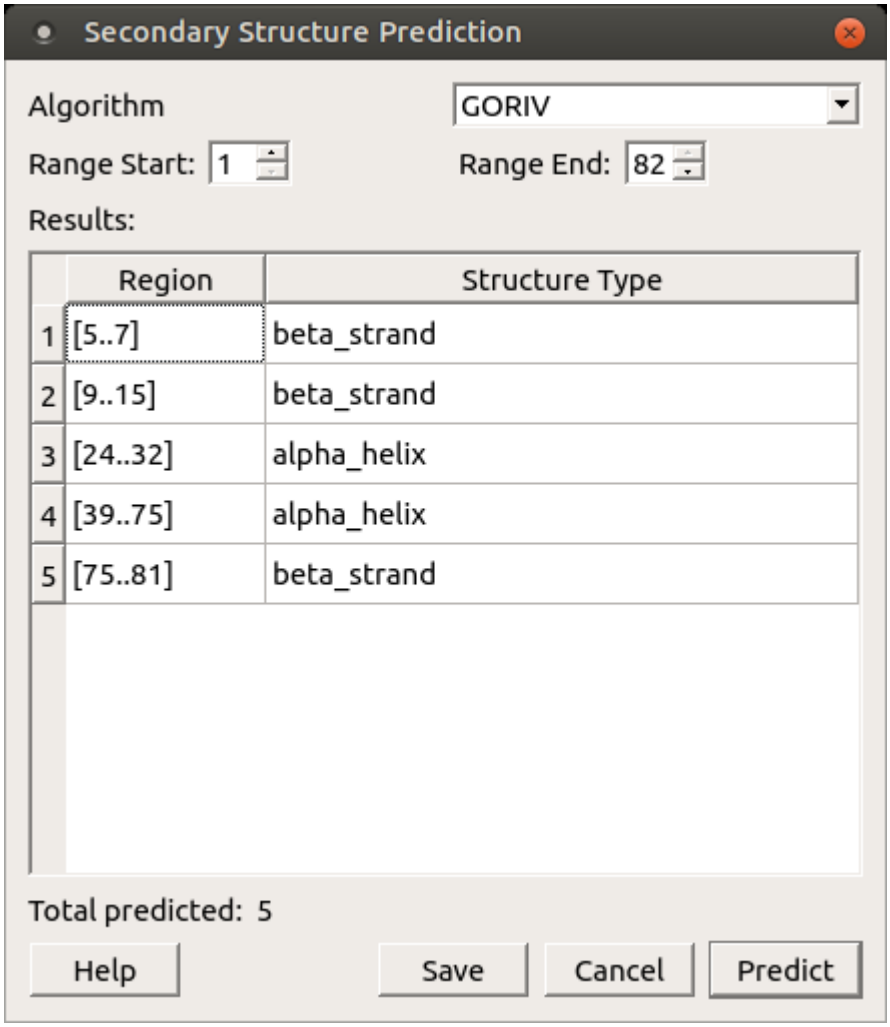


It supports the following options:

Algorithm — you can choose the preferred algorithm. Currently, “GORIV” and “PsiPred” algorithms are available.

Range start / *Range end* — select the sequence range for prediction.

Results — the visual representation of the prediction results, for example:



Save as annotation — select this button to save the results as annotations of the current protein sequence.

SITECON

SITECON — is a program package for recognition of potential transcription factor binding sites basing on the data about conservative conformational and physicochemical properties revealed on the basis of the binding sites sets analysis.

To cite SITECON use the following article:

“Oshchepkov D.Y., Vityaev E.E., Grigorovich D.A., Ignatieva E.V., Khlebodarova T.M.SITECON: a tool for detecting conservative conformational and physicochemical properties in transcription factor binding site alignments and for siterecognition. //Nucleic Acids Res. 2004 Jul 1;32(Web Server issue): W208-12.”

UGENE version of SITECON provides a tool for recognition of potential binding sites for over *90 types* of transcription factors. Also UGENE version of SITECON provides a tool for recognition of potential binding sites basing site alignment proposed by user. For the detailed method description see the original SITECON site.

Data about used context-dependent conformational and physicochemical properties are available in the PROPERTY Database.

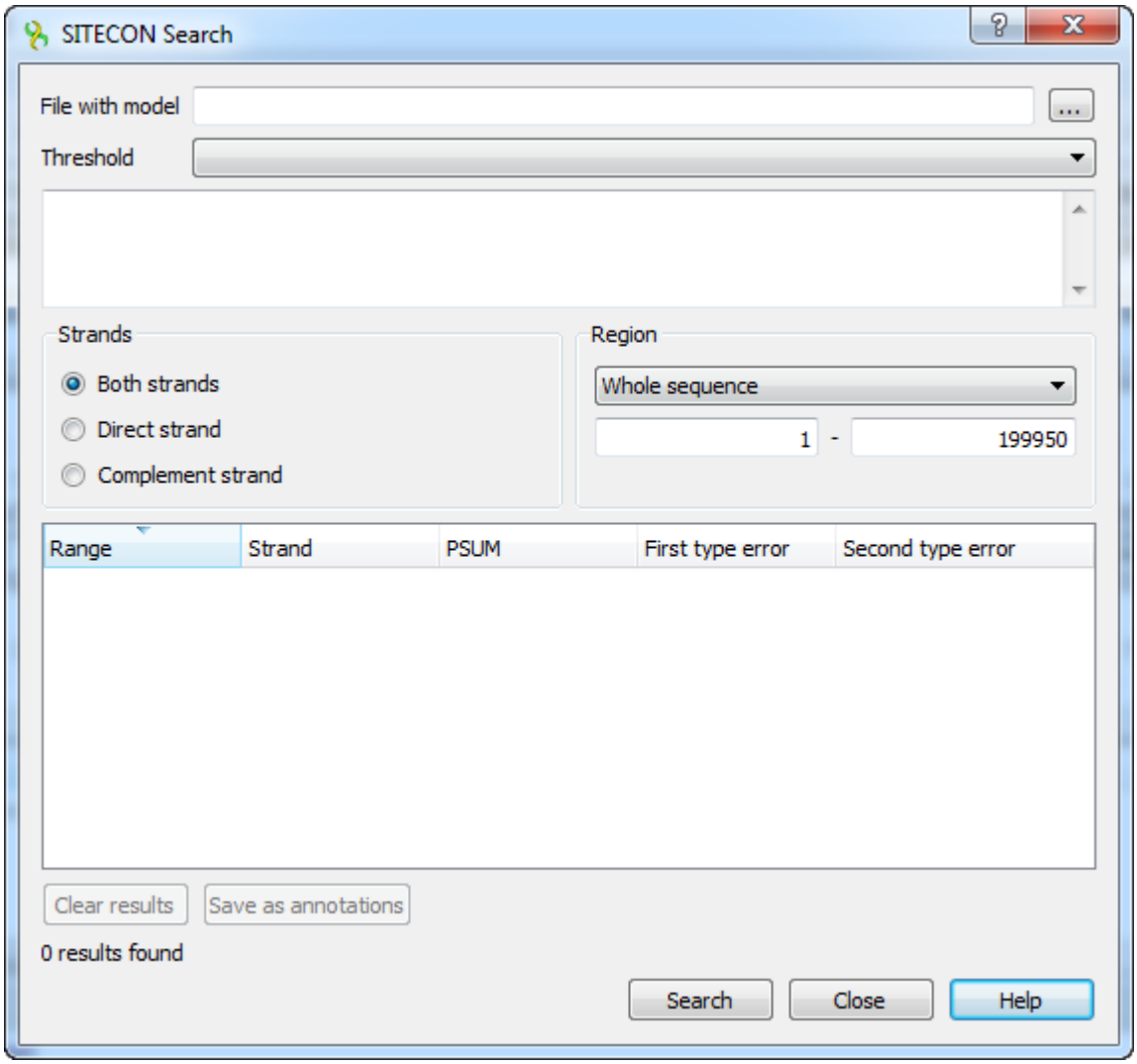
- SITECON Searching Transcription Factors Binding Sites
- Types of SITECON Models
  - Eukaryotic
  - Prokaryotic
- Building SITECON Model

SITECON Searching Transcription Factors Binding Sites

To search transcription factor binding sites in a DNA sequence select the *Analyze Search TFBS with SITECON...* context menu item.

In the appeared search dialog you must select a file with TFBS profile. The profiles supplied with UGENE are placed in the $UGENE/data/sitecon_models f older.

After the profile is loaded the threshold-filter is populated with values read from profile. You can use the filter to remove low-scoring regions from the result.



The regions found by SITECON algorithm can be saved as annotations to the DNA sequence in the Genbank format.

Every *SITECON* profile supplied with UGENE contains complete information about calibration settings provided to UGENE team by the author of *SITECON*.

The original TFBS alignments used to calculate profiles can be requested directly from the author of *SITECON*.

Types of SITECON Models

- Eukaryotic
- Prokaryotic

Eukaryotic

Name	Description
CEBP_a	CCAAT-enhancer-binding protein_alpha
CEBP_all	CCAAT-enhancer-binding proteins
CLOCK	Circadian Locomotor Output Cycles Kaput
cMyc_can	Myc (c-Myc) is a regulator gene that codes for a transcription factor. A mutated version of Myc is found in many cancers.

CRE	Cyclic AMP response element
E2F1	Transcription factor E2F1 is a protein that in humans is encoded by the E2F1 gene.
E2F1/DP1sel1	E2F factors bind to DNA as homodimers or heterodimers in association with dimerization partner DP1.
EGR1	Early growth response protein 1
EKLf	Erythroid Kruppel-like Factor
ER2	Estrogen receptor beta
GATA_all	GATA transcription factors are a family of transcription factors characterized by their ability to bind to the DNA sequence “GATA”
GATA-1	GATA-binding factor 1
GATA-2	GATA-binding protein 2
GATA-3	Trans-acting T-cell-specific transcription factor GATA-3
HMG-1	High-mobility group protein 1
HNF-1	Hepatocyte nuclear factor 1
HNF-3	Hepatocyte nuclear factor 3
HNF-4	Hepatocyte nuclear factor 4
IRF	Interferon regulatory factors
isre	Interferon stimulation response element
MyoD	MyoD belongs to a family of proteins known as myogenic regulatory factors (MRFs)
MyOGsel3	Myogenin
NF-1	Neurofibromin 1
NF-E2	Transcription factor NF-E2 45 kDa subunit is a protein that in humans is encoded by the NFE2 gene.
NFATp	Pre-existing component of the NFAT(Nuclear factor of activated T-cells) transcription complex.
NFkB_all	Nuclear factor kappa-light-chain-enhancer of activated B cells
NFkB_hetero	The p50 (NFKB1)/p65 (RELA) heterodimer is the most abundant form of NF-kB
NFkB_ homo	The c-Rel protein is a member of the NF-kB family of transcription factors and contains a Rel homology domain
Nfy	Nuclear transcription factor Y
Nrf2	Nuclear factor (erythroid-derived 2)-like 2
Oct-1	Octamer transcription factor 1
Oct_all	Octamer transcription factors
p53	Protein 53
PPRF	Paramedian pontine reticular formation
Pu1	Is a protein that in humans is encoded by the SPI1 gene
setCREB	cAMP response element-binding
setCREBzag	cAMP response element-binding
SRE_san	Serum response element
SRF	Serum response factor
STAT1	Signal Transducer and Activator of Transcription 1
STAT	Signal Transducer and Activator of Transcription
TTF1	Thyroid transcription factor 1
USF	Upstream stimulatory factors
yy1	Is a protein that in humans is encoded by the YY1 gene

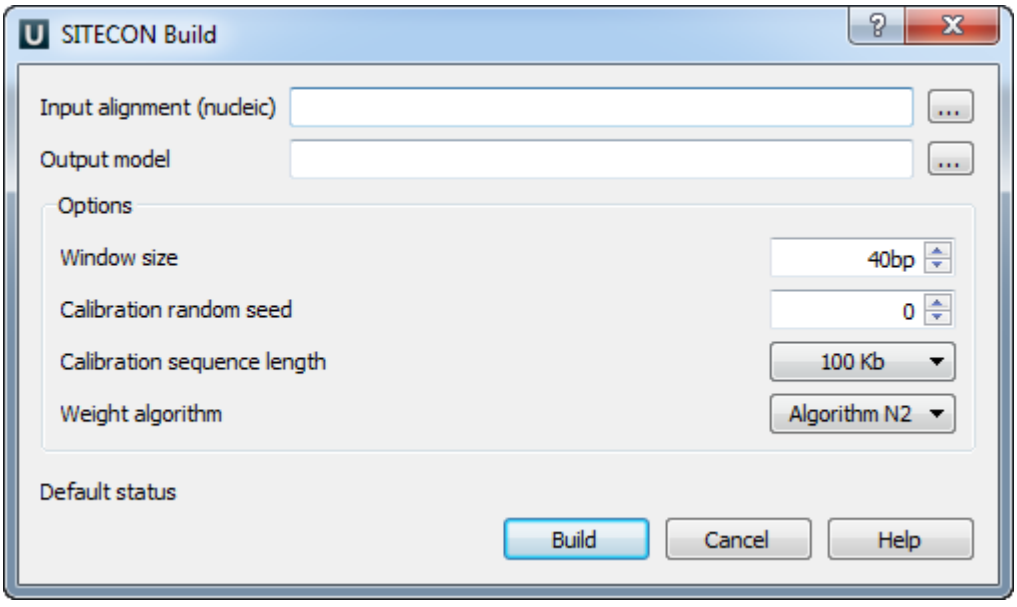
Prokaryotic

Name	Description
AgaR	N-acetylgalactosamine repressor, AgaR, negatively controls the expression of the aga gene cluster
AgaC	AgaC is the Enzyme IIC domain of a predicted N-acetylgalactosamine-transporting PEP-dependent phosphotransferase system
ArcA	ArcA transcriptional dual regulator
ArgR	ArgR complexed with L-arginine represses the transcription of several genes involved in biosynthesis and transport of arginine, transport of histidine, and its own synthesis and activates genes for arginine catabolism.
CpxR	DNA-binding response regulator in two-component regulatory system with CpxA
Crp	cAMP receptor protein
CysB	Cysteine B
CytR	Cytidine Regulator
DeoR	Deoxyribose Regulator
DnaA	DnaA is the linchpin element in the initiation of DNA replication in E. coli.
FadR	Fatty acid degradation Regulon
fis	Factor for inversion stimulation
FlhDC	Operon that encodes two transcriptional regulators
Fnr	FNR is the primary transcriptional regulator that mediates the transition from aerobic to anaerobic growth through the regulation of hundreds of genes.
Frur	Fructose repressor
FUR	Ferric Uptake Regulation
GALR	Galactose repressor
GALS	Galactose isorepressor
GLPR	sn-Glycerol-3-phosphate repressor
GNTP	Is a member of the GntP family transporters
HNS	Histone-like nucleoid structuring protein
ICLR	Isocitrate lyase Regulator
IHF	Integration host factor
ISCR1	Iron-sulfur cluster Regulator 1
ISCR3	Iron-sulfur cluster Regulator 3
LEXA	LexA represses the transcription of several genes involved in the cellular response to DNA damage or inhibition of DNA replication
Lrp	Leucine-responsive regulatory protein
MALT	Maltose regulator
MARA	Multiple antibiotic resistance
MELR	Melibiose regulator
MEtJ	MetJ represses the expression of genes involved in biosynthesis and transport of methionine
MetR1	MetR participates in controlling several genes involved in methionine biosynthesis [Weissbach91] and a gene involved in protection against nitric oxide
MLC	DgsA, better known as Mlc, “makes large colonies,” is a transcriptional dual regulator that controls the expression of a number of genes encoding enzymes of the Escherichia coli phosphotransferase (PTS) and phosphoenolpyruvate (PEP) systems
MODE	Molybdate-responsive transcription factor
NAC	Nitrogen assimilation control
NAGC_	N-acetylglucosamine

new2	
NANR	N-acetyl-neuraminic acid regulator
NARL2	Nitrate/nitrite response regulator NarL
NARL	Nitrate/nitrite response regulator NarL
NARP	Nitrate/nitrite response regulator NarP
NIRC	NirC is a nitrite transporter which is a member of the FNT family of formate and nitrite transporters
OmpC	OmpC is a member of the GMP family
OxyR	Oxidative stress regulator
PHOB	PhoB is a dual transcription regulator that activates expression of the Pho regulon in response to environmental Pi
PHOP	Member of the two-component regulatory system phoQ/phoP involved in adaptation to low Mg2+ environments and the control of acid resistance genes
PurR	PurR dimer controls several genes involved in purine nucleotide biosynthesis and its own synthesis
RcsB_1	Regulator capsule synthesis B
RcsB_2	Regulator capsule synthesis B
Rob2	Right origin-binding protein
ROB	Right origin-binding protein
soxS	SoxS is a dual transcriptional activator and participates in the removal of superoxide and nitric oxide and protection from organic solvents and antibiotics
TORR	TorR response regulator
TRPR	Tryptophan (trp) transcriptional repressor
TyrR	Tyrosine repressor

Building SITECON Model

To build a new SITECON model call the *Tools->SITECON->Build new SITECON model from alignment* main menu item. The following dialog will appear:

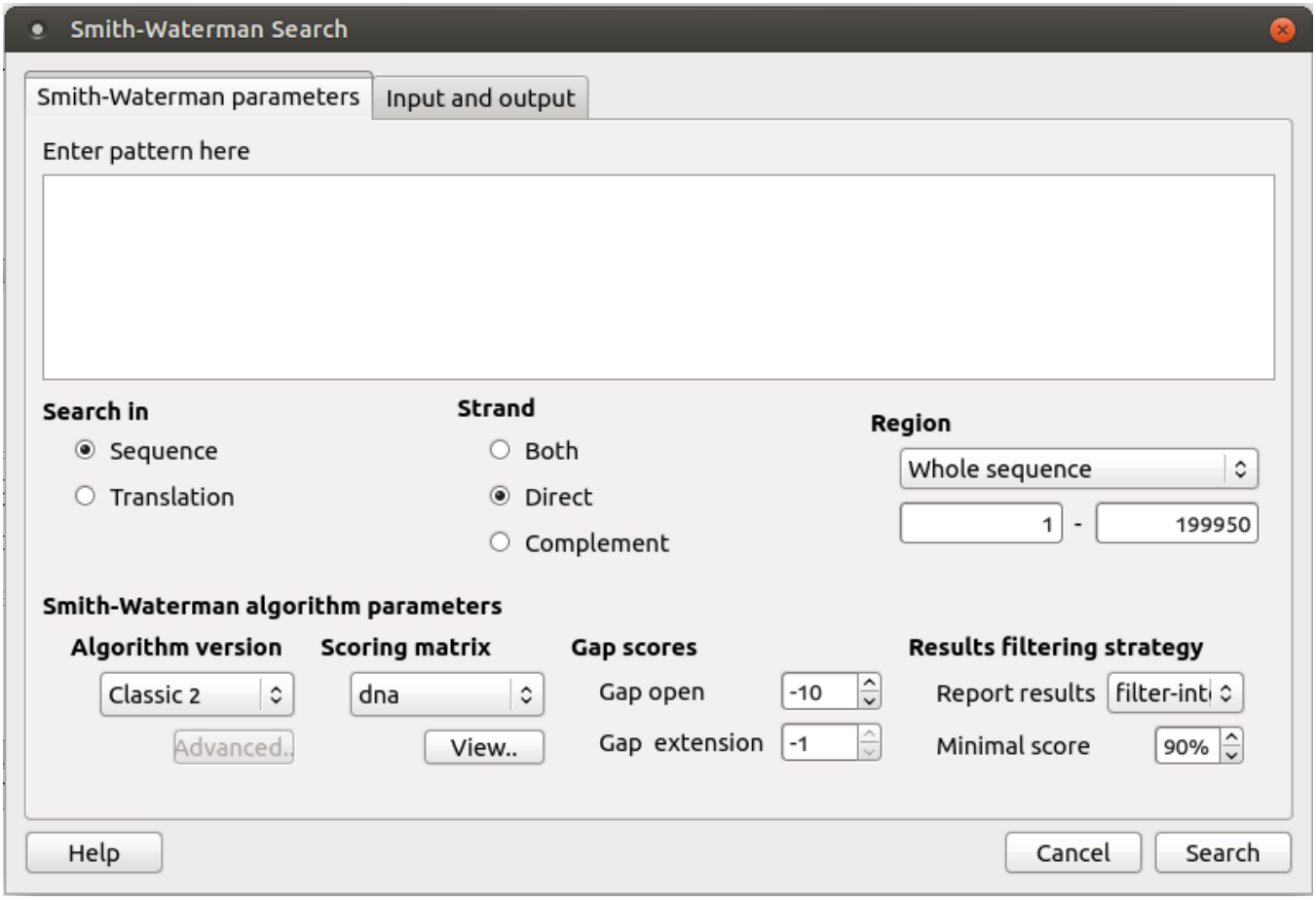


Here you need to select a nucleotide alignment and an output model. Optionally, you can change other parameters. After that click on the *Build* button.

Smith-Waterman Search

The *Smith-Waterman Search* plugin adds a complete implementation of the Smith-Waterman algorithm to UGENE.

To use the plugin open a nucleotide or protein sequence in the *Sequence View* and select the *Analyze Find pattern [Smith-Waterman]* item in the context menu. The *Smith-Waterman Search* dialog appears:



First of all you need to specify the pattern to search for. The rest parameters are optional:

Search in — select either to search in the sequence or in its translation.

Strand — select the strand to search in: direct, complementary or both strands.

Region — specifies the region of the sequence that will be used to search for the pattern. By default, if a subsequence has been selected when the dialog has been opened, then the selected subsequence is searched for the pattern. Otherwise, the whole sequence is used. You can also input a custom range.

Algorithm version — version of the algorithm implementation. Non-classic versions produce the same results as classic but much faster. To use these optimizations our system must support these capabilities.

- Classic 2
- SSE2
- CUDA
- OPENCL

Scoring matrix — can be chosen from a bunch of matrices supplied with UGENE. To view a matrix selected click the *View* button.

Gap open — penalty for opening a gap.

Gap extension — penalty for extending a gap

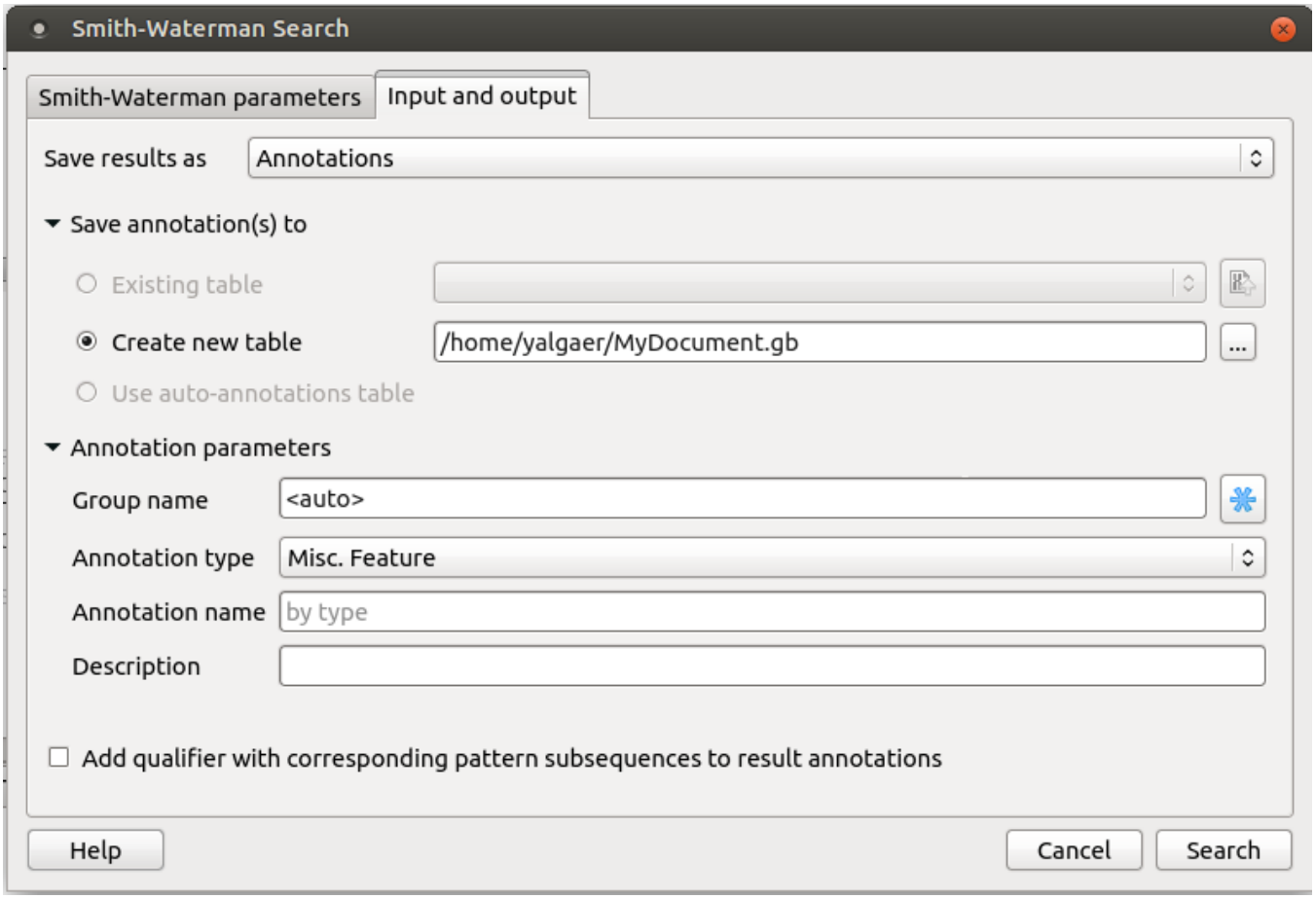
Report results — simple heuristic which allows to filter intersected hits. If it is set to *none*, the algorithm may report large set of almost identical results in the same region.

Minimal score — another simple heuristic which measures sequences similarity. It is more convenient than using some abstract scores. If set to 100%, the algorithm will search for exact substring match.

The results of the search are saved as annotations or as multiple alignment. To set the saving parameters go to the *Input and output* tab of the dialog.

If you want to save the results as annotations input *the annotations saving* parameters (*Annotation name*, *Group name*, *Annotation type, Description* and a file to save the annotation to). Also you can add qualifier with corresponding pattern subsequences to result annotations. Check the corresponding checkbox for it.

If you want to save the results as multiple alignment select the following parameters:

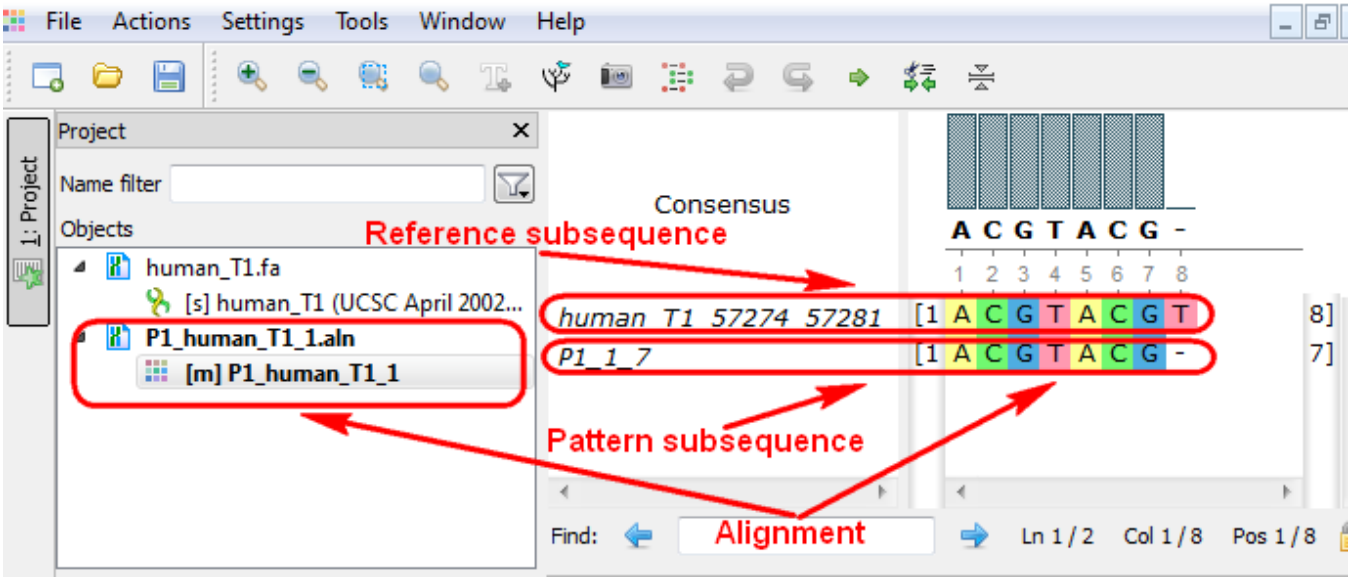


Here you can select a file to save the alignment to (*Alignment files directory path* parameter).

Using the *Set advanced options* checkbox you can select the saving options.

You can set the different templates for files names: create your own or create by using the following: [E] — adds a subsequence end position, [hms] — adds a time, [MDY] — adds a date, [S] — adds a subsequence start position, [L] — adds a subsequence length, [SN] — adds a reference sequence name prefix, [PN] — adds a pattern sequence name prefix, [C] — adds a counter.

You can create templates for alignment files names, reference subsequence names, pattern subsequence names and for pattern sequence name:



HMM2

The *HMM2* plugin is a toolkit based on the Sean Eddy's HMMER2 package.

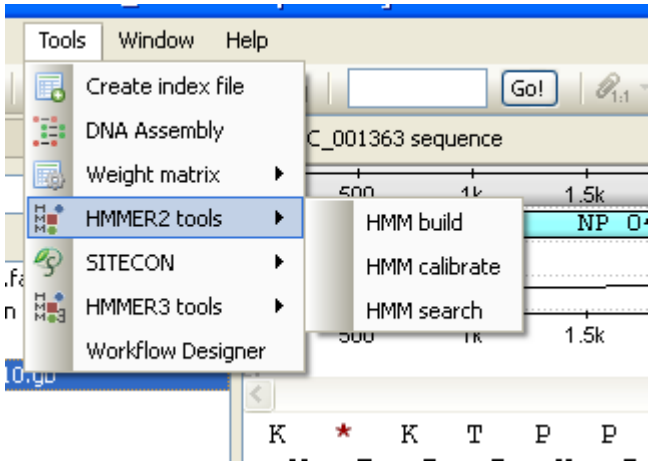
While working on this plugin we were guided by the following principles:

- Make the HMMER2 tools accessible to a wider user audience by providing graphical interface for all supported utilities for most of the platforms.
- Be compatible with the original HMMER2 package.
- Create the high-performance solution utilizing modern multi-core processors and SIMD instructions.

The current version of UGENE provides user interface for three HMM2 tools: *HMM build*, *HMM calibrate* and *HMM search*.

In the original program the corresponding commands are: "hmmbuild", "hmmcalibrate" and "hmmsearch".

To access these tools select the *Tools HMMER2 tools* submenu of the program main menu:



We highly recommend reading the original HMMER2 documentation to learn how to use utilities provided by the plugin.

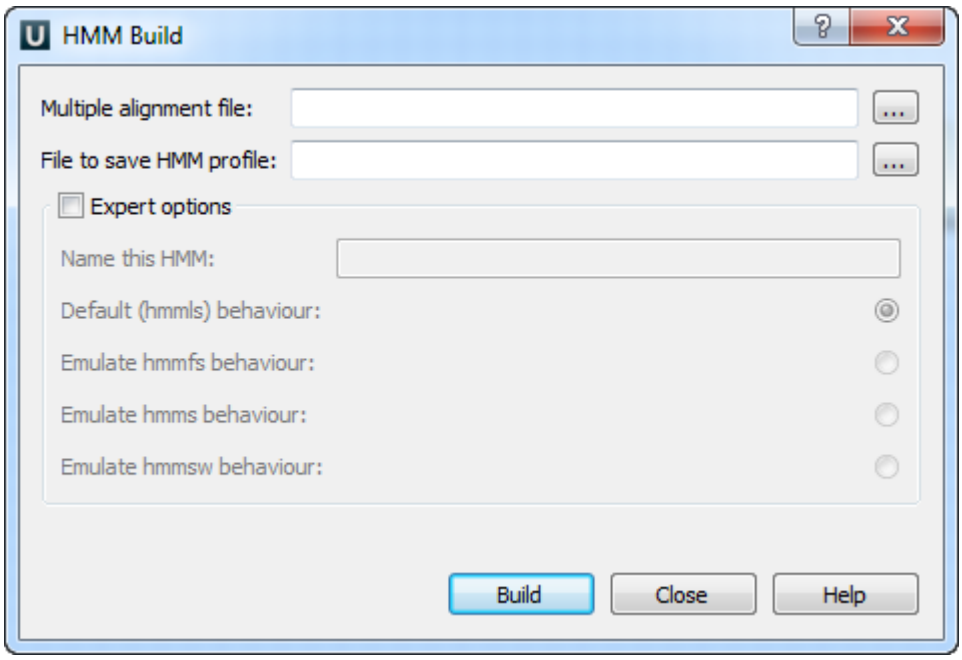
SSE2 algorithm is implemented by Leonid Konyaev, Novosibirsk State University. Use of the SSE2 optimized version of the *HMM search* algorithm with quad-core CPU gives >30x performance boost when compared with the original single-threaded algorithm (single sequence mode).

- Building HMM2 Model
- Calibrating HMM2 Model
- Searching Sequence Using HMM2 Profile

Building HMM2 Model

HMM build tool is used to build a new HMM profile from a multiple alignment.

You can use any alignment file formats supported by UGENE. The output HMM profile format is compatible with the HMMER2 package.

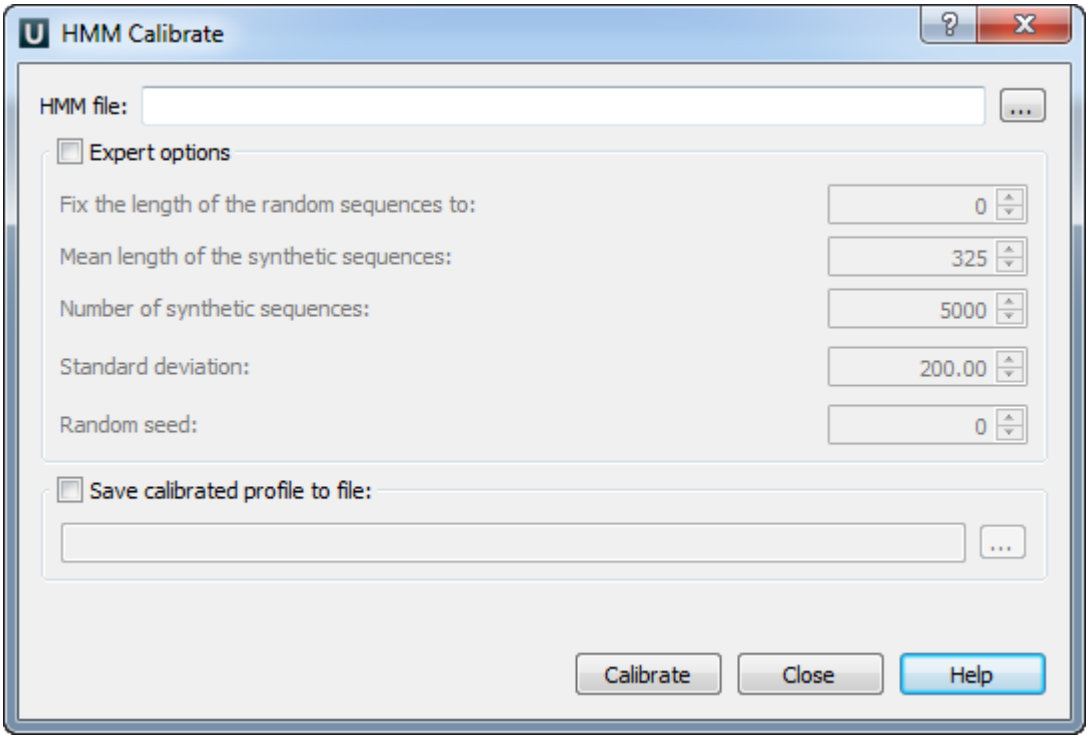


The *HMM build* tool does not automatically calibrate a profile. Use the *HMM calibrate* tool to calibrate the profile.

Calibrating HMM2 Model

The *HMM calibrate* tool reads a HMM profile file, scores a large number of synthesized random sequences with it, fits an extreme value distribution (EVD) to the histogram of those scores, and re-saves the hmm file including the EVD parameters.

To avoid modification of the original HMM file you can select a new location for the calibrated profile.



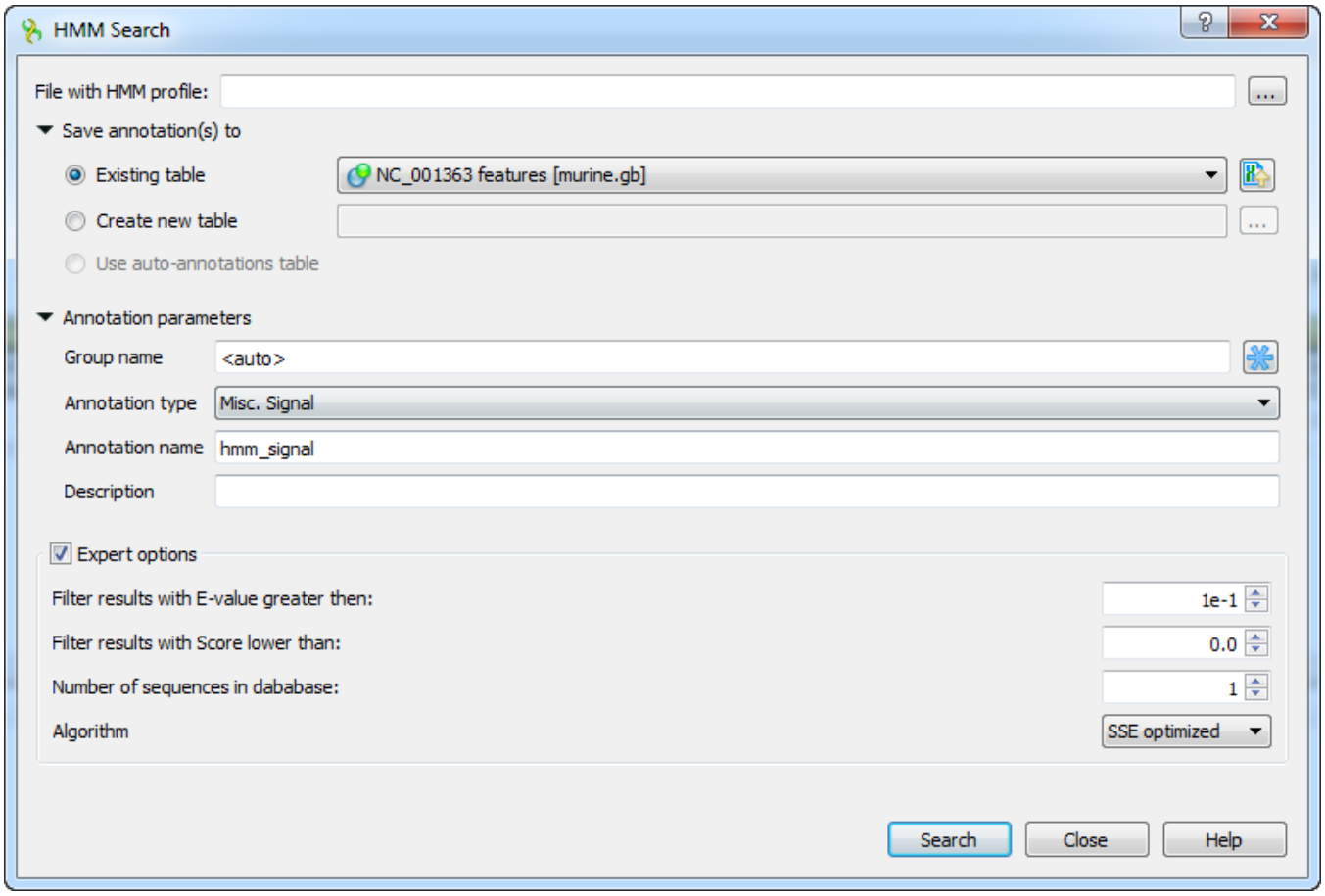
Searching Sequence Using HMM2 Profile

The *HMM search* tool reads a HMM profile from a file and searches the sequence for significantly similar sequence matches.

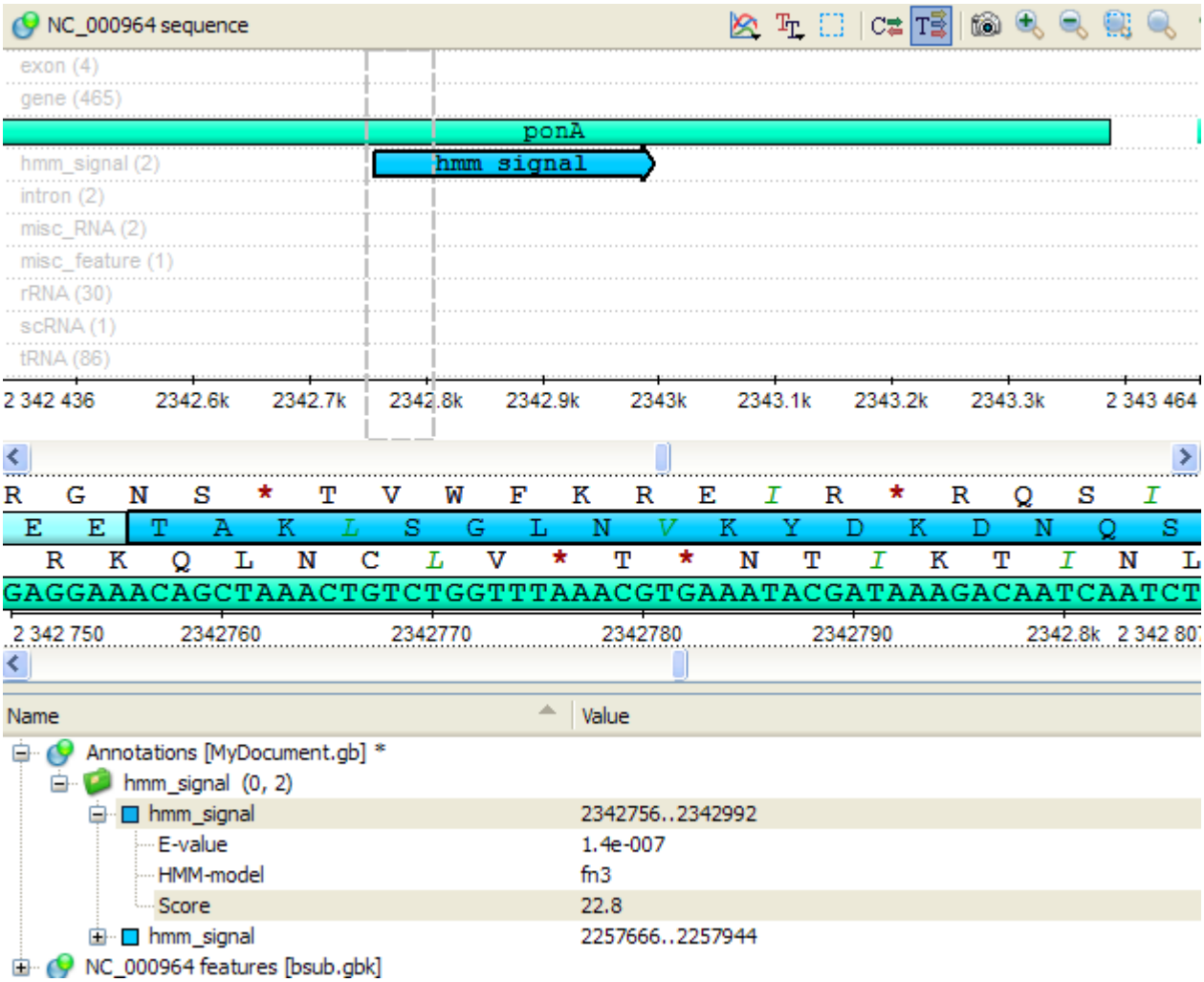
The sequence must be selected in the *Project View* or there must be an active *Sequence View* window opened.

If the selected sequence is nucleic and the HMM profile is built for amino alignment, the sequence is automatically translated and all 6 translations are used to search in.

If a HMM profile is built for nucleic alignment, the search is performed for both strands (direct and complement).



The search results are stored as sequence annotations in the Genbank file format.



All HMM2 UGENE tools work only with files that contain a single HMM model.

HMM3

The *HMM3* plugin is a toolkit based on the Sean Eddy's HMMER3 package.

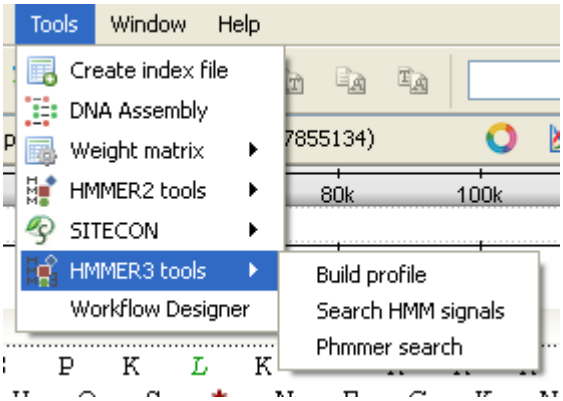
While working on this plugin we were guided by the following principles:

- Make the HMMER3 tools accessible to a wider user audience by providing graphical interface for all supported utilities for most of the platforms.
- Be compatible with the original HMMER3 package.
- Create the high-performance solution utilizing modern multi-core processors.

The current version of UGENE provides user interface for three HMM3 tools: *HMM3 build*, *HMM3 search* and *Phmmer search*.

In the original program the corresponding commands are: "hmmbuild", "hmmsearch" and "phmmer".

To access these tools select the *Tools HMMER3 tools* submenu of the program main menu:



We highly recommend reading the original HMMER3 documentation to learn how to use utilities provided by the plugin.

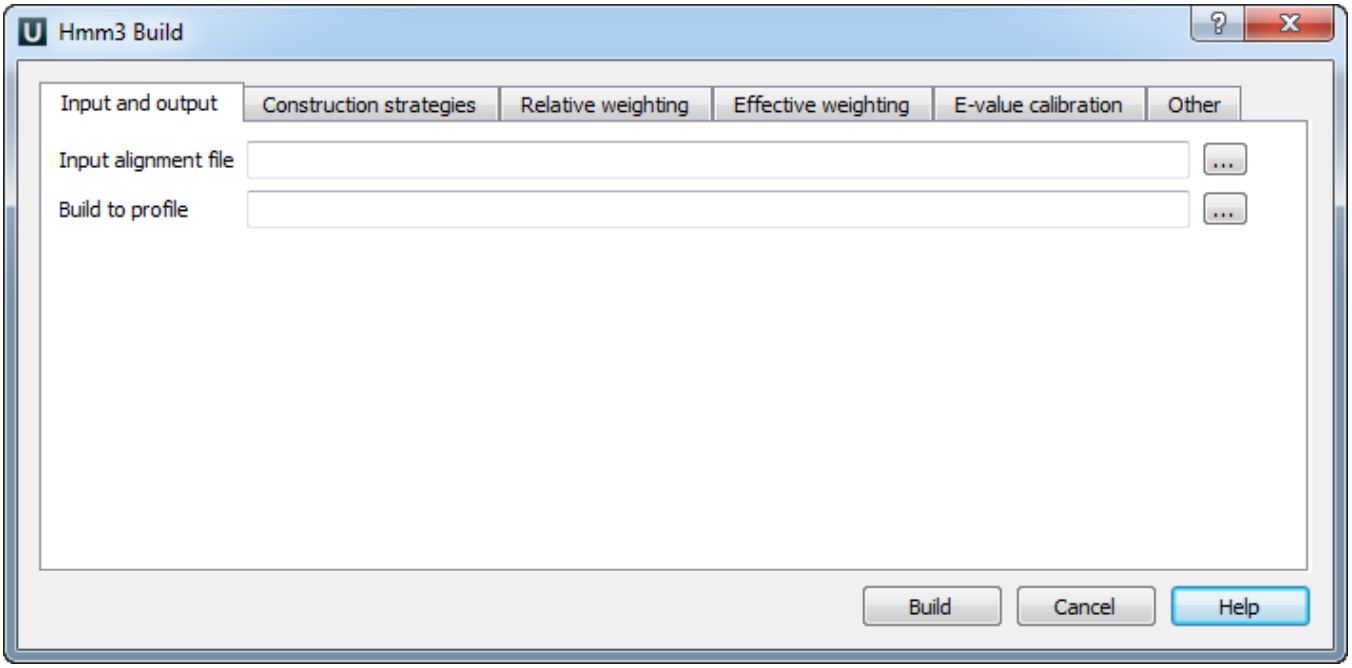
- Building HMM Model
- Searching Sequence Using HMM Profile
- Searching Sequence Against Sequence Database

Building HMM Model

The *HMM3 build* tool is used to build a new HMM profile from a multiple alignment. You can use any alignment file formats supported by UGENE.

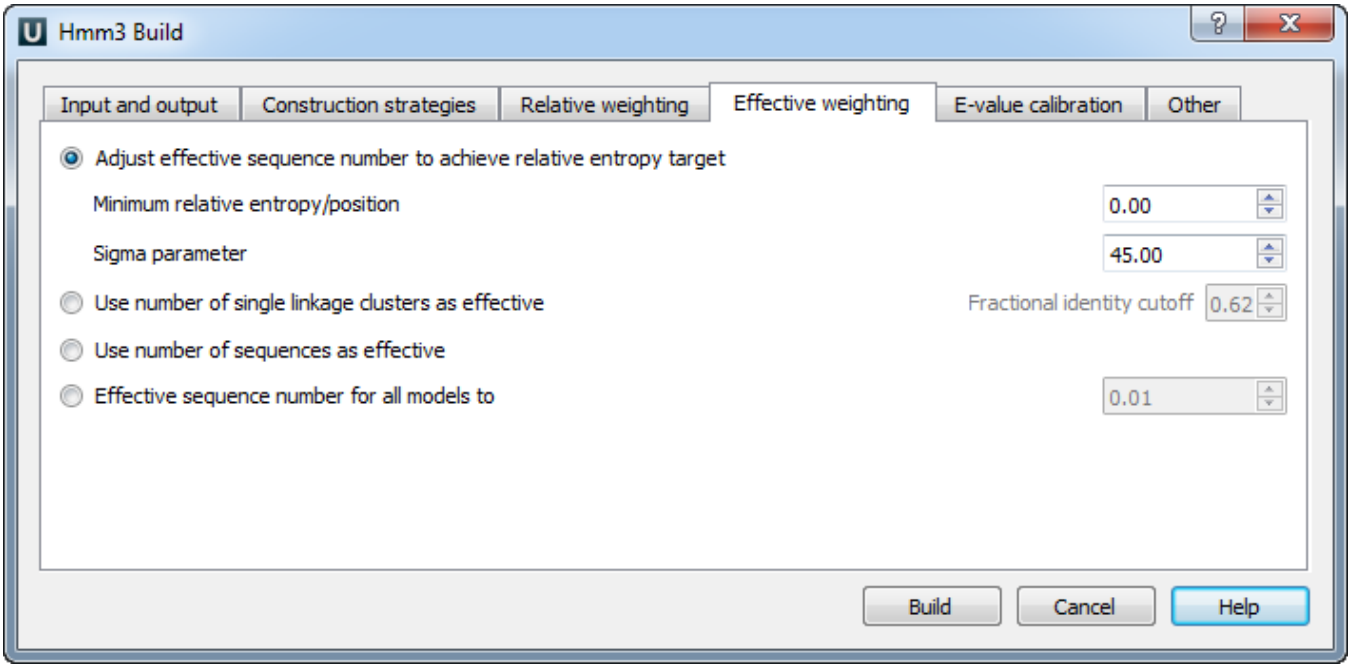
The output HMM profile format is compatible with the HMMER3 package, but it is not compatible with the HMMER2.

The *HMM3 build* automatically calibrates the target model.



The HMM3 configuration dialog provides an easy way to set appropriate search parameters.

Here you can see effective weighting strategies options:



Searching Sequence Using HMM Profile

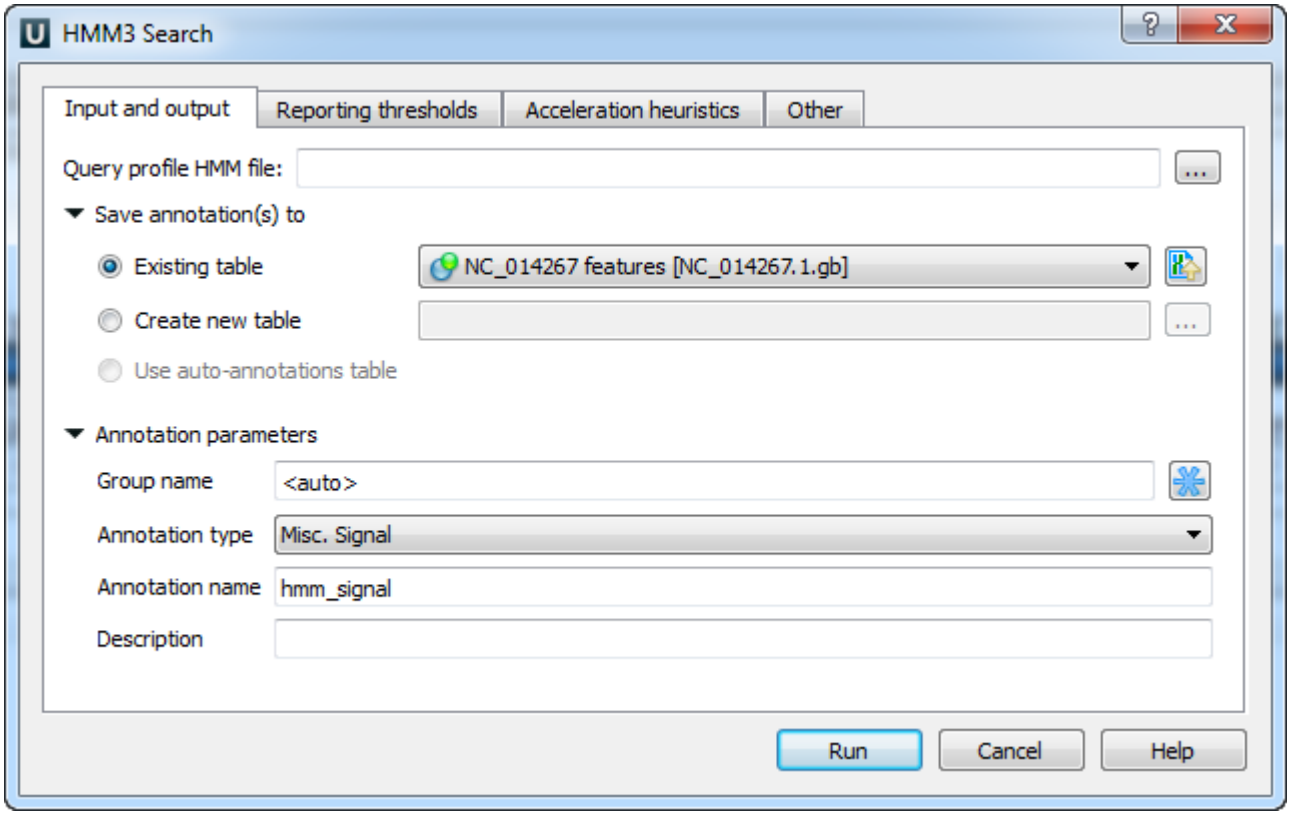
The *HMM3 search* tool reads a HMM profile from a file and searches a sequence for significantly similar sequence matches.

The sequence must be selected in the *Project View* or there must be an active *Sequence View* window opened.

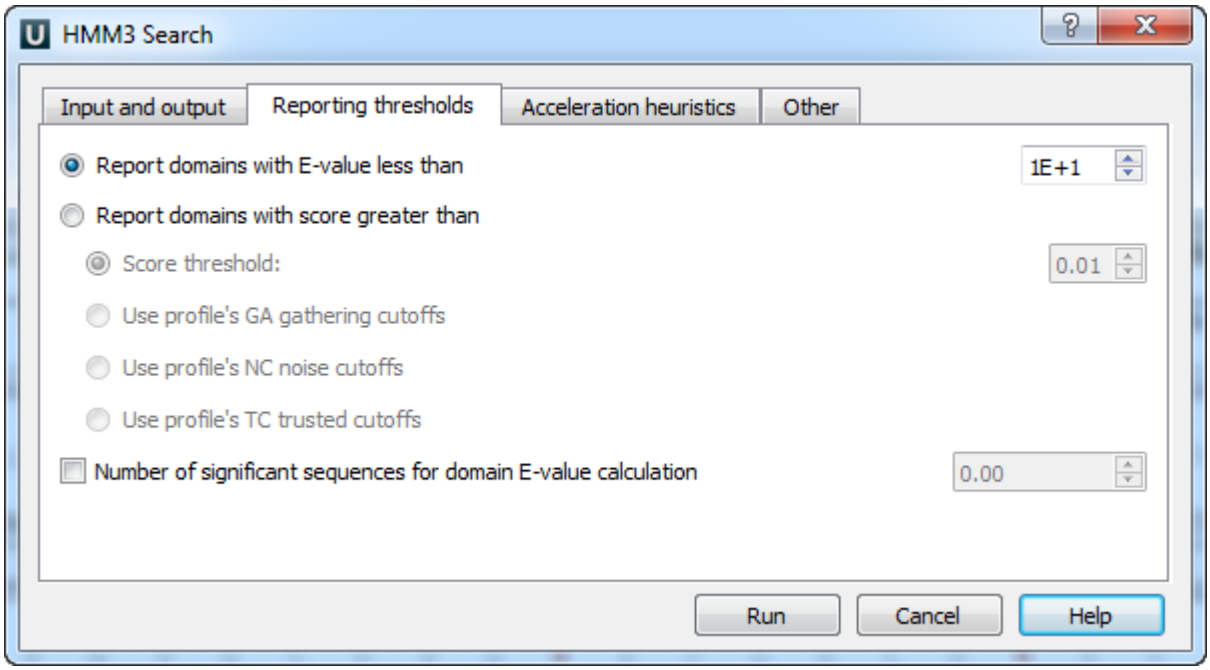
If the selected sequence is nucleic and profile HMM is built from amino alignment, the sequence will be automatically translated and searched in all possible frames (6 totally).

If a profile HMM is built for nucleic alignment, the search is performed for both strands (direct and complement).

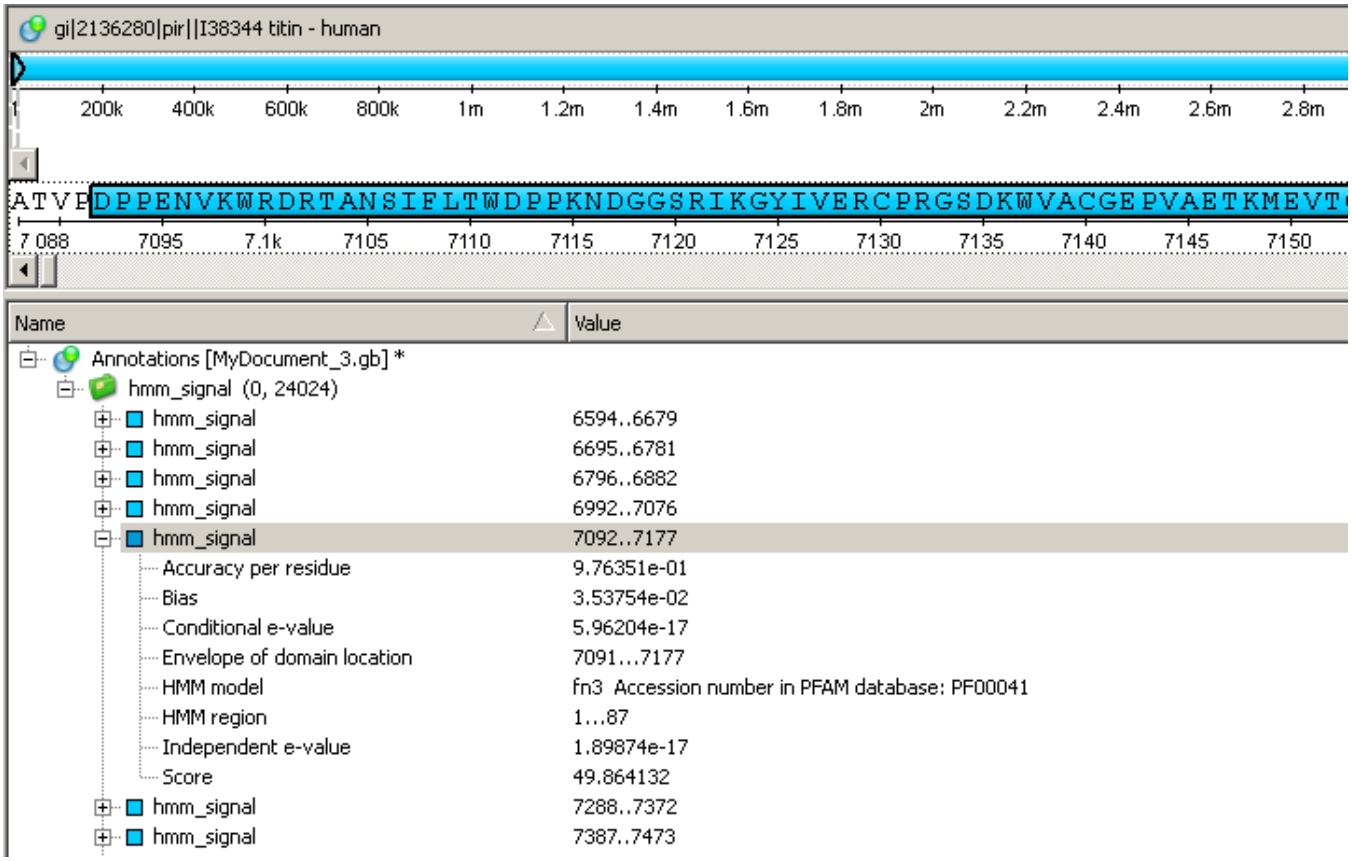
The *HMM3 search* accepts the HMMER2 HMM profiles (amino only) as a backward compatibility feature. An interesting post about using the HMMER2 models with the HMMER3 is available on the Sean Eddy's blog.



For example, reporting thresholds options can be configured using the dialog:



The search results are stored as sequence annotations in the Genbank file format.



The *HMM3 search* works only with files that contain a single HMM model.

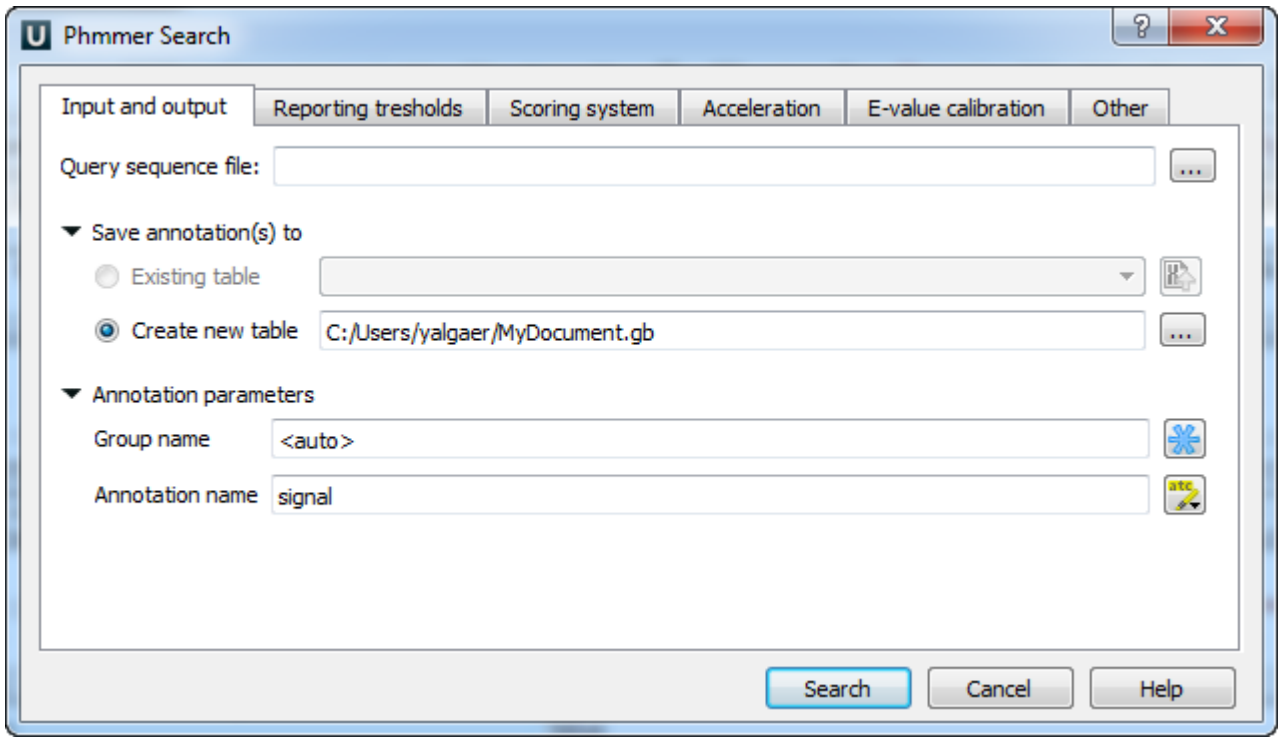
Searching Sequence Against Sequence Database

The *Phmmer search* tool searches for query sequence matches in sequence database, much as BLASTP or FASTA would do.

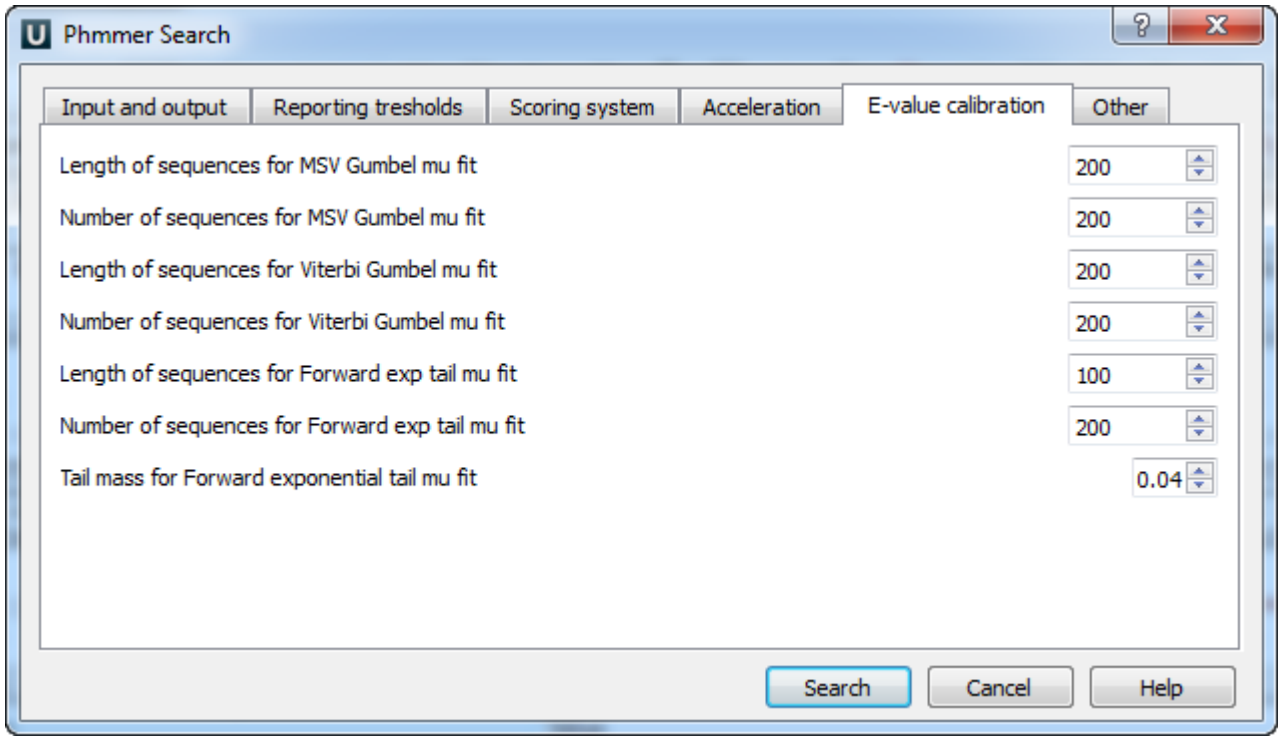
The *Phmmer search* works essentially like the *HMM3 search* does, except you provide a query sequence instead of a query profile HMM.

The database sequence must be selected in the *Project View* or there must be an active *Sequence View* window opened.

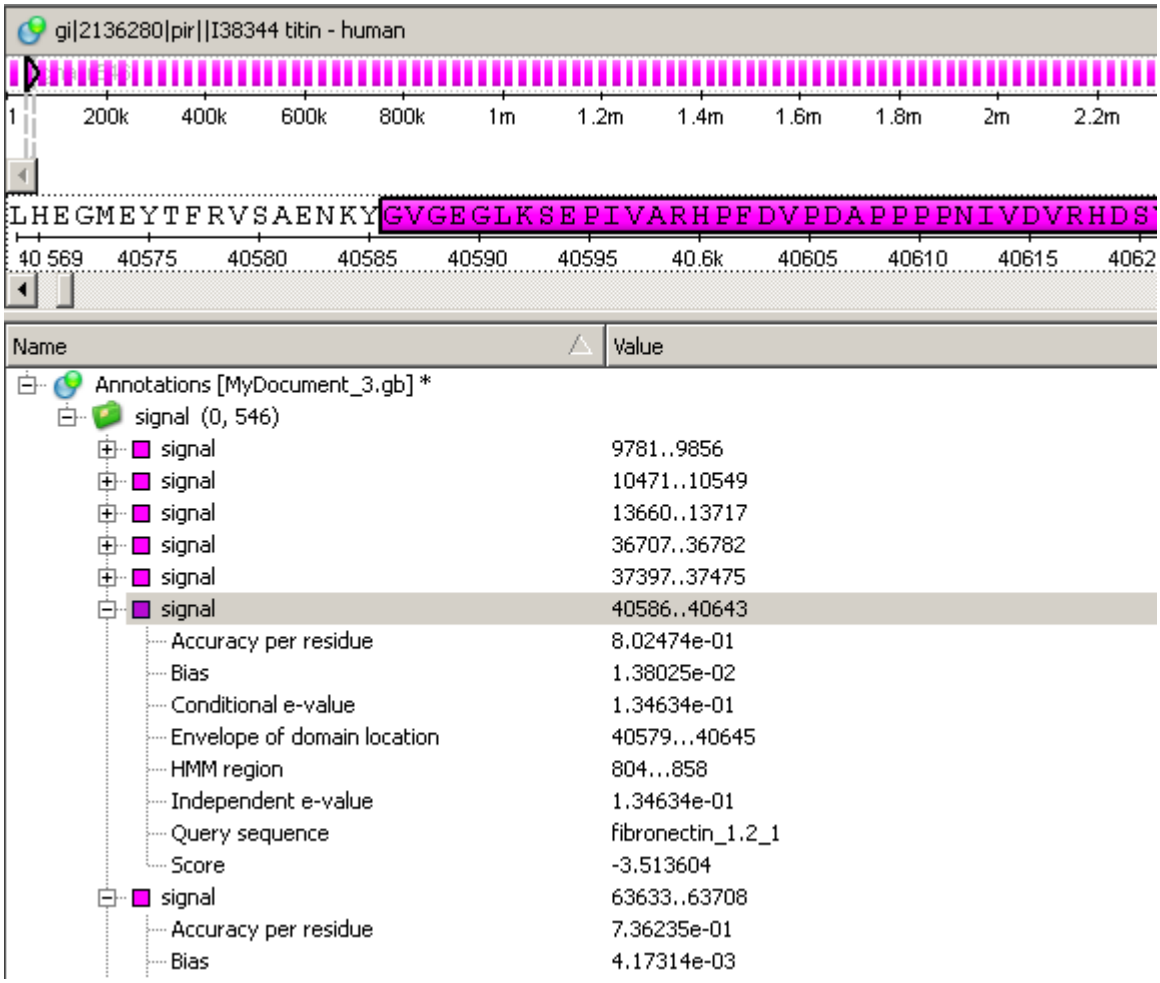
Select the query sequence in the *Phmmer search* dialog:



You can set options of the *Phmmer search* by choosing the needed dialog tab. Here you can see the e-value calibration options:



The results are stored as sequence annotations in the Genbank file format.



The *Phmmer search* works only with single-sequence databases.

uMUSCLE

UGENE contains graphical ports of the Robert C. Edgar's MUSCLE tool for multiple alignment.

MUSCLE4 is not supported since UGENE version 1.7.2.

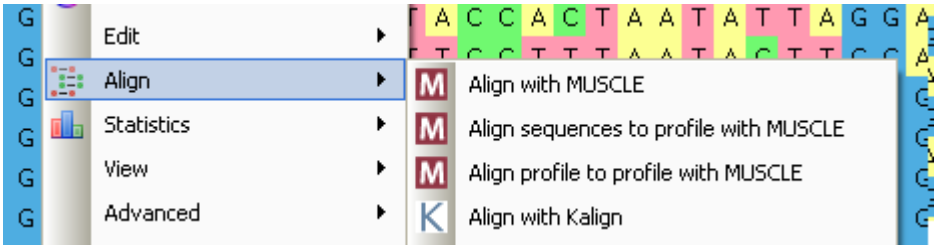
The package is integrated completely, so there is no need in extra files for using it. It is possible to run several multiple alignment tasks in parallel, check the progress and cancel the running tasks safely.

The k-mer clustering part of the MUSCLE algorithm was optimized for multicore systems by Timur Tleukenov, Novosibirsk State Technical University.

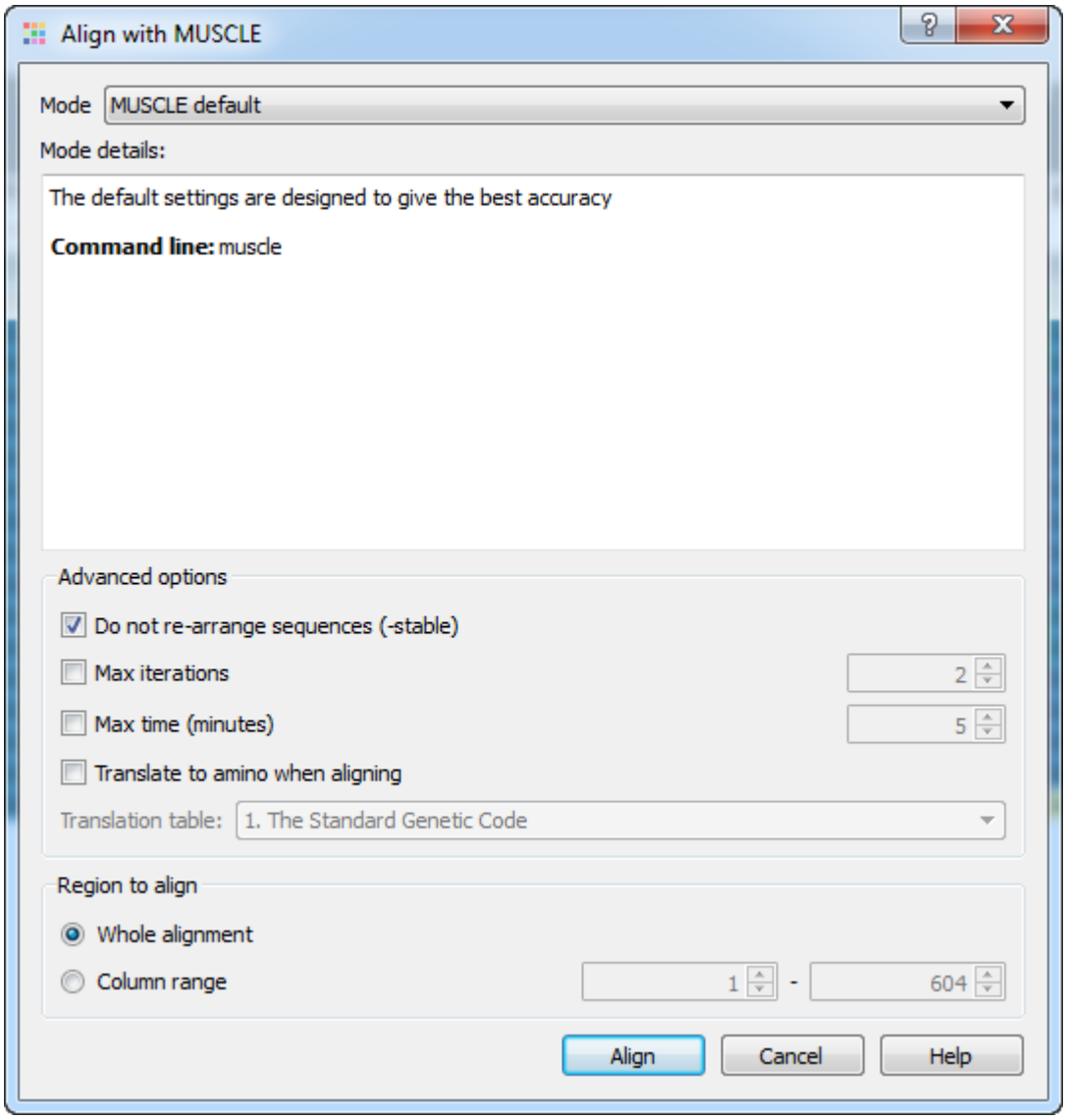
- MUSCLE Aligning
- Aligning Profile to Profile with MUSCLE
- Aligning Sequences to Profile with MUSCLE

MUSCLE Aligning

To run the classic MUSCLE use the *Align Align with MUSCLE* context menu item in the *Alignment Editor*.



The dialog contains the list of MUSCLE modes: *MUSCLE default*, *Large alignment*, *Refine only*.



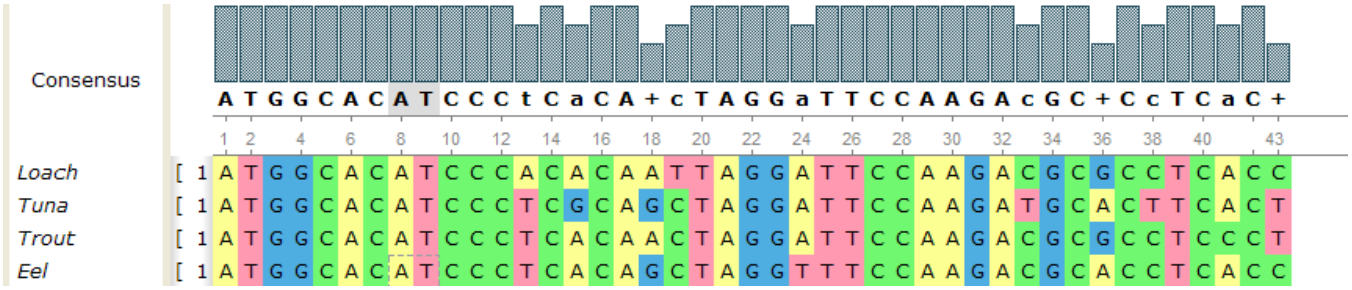
By default UGENE does not rearrange sequence order in an alignment, but the original MUSCLE package does. To enable sequence rearrangement uncheck the *Do not re-arrange sequences (-stable)* option in the dialog.

One of the improvements to the original MUSCLE package is the ability to align only a part of the model. When the *Column range* item is selected the region of the specified columns is only passed to the MUSCLE alignment engine. The resulted alignment is inserted into the original one with gaps added or removed on the region boundaries.

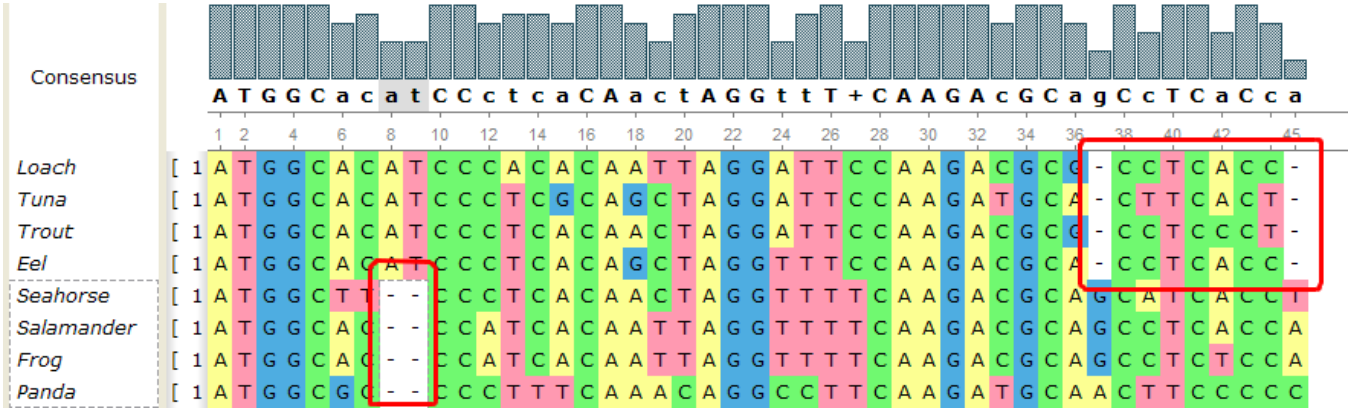
To visually select the column range to align, make a selection in the alignment editor first. Then invoke the MUSCLE plugin. Its column range boundary values will automatically match the given selection.

Aligning Profile to Profile with MUSCLE

The *Align Align profile to profile with MUSCLE* context menu item allows to align an existing profile to an active alignment. During this process the MUSCLE does not realign the profiles, but inserts columns with gaps characters only ('—' characters). For example, the alignment in the picture below could be used as a profile:



The same profile after profile-to-profile alignment:



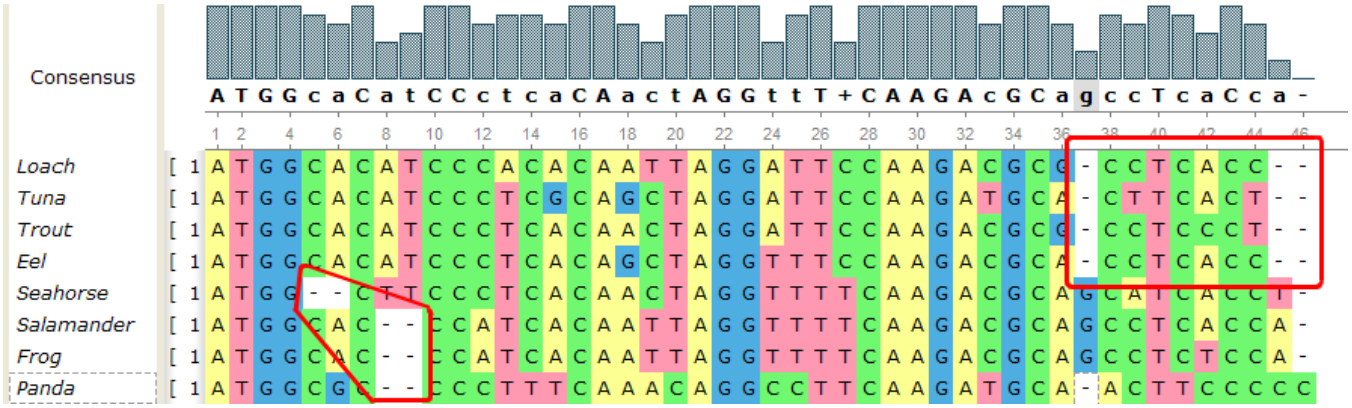
There are two gap columns inserted into the source profile, and two gap columns inserted into the added one. Therefore the profiles columns kept intact and the alignments haven't been changed.

Aligning a profile to the active alignment you will modify the original alignment file, since it will contain 2 profiles after the operation is completed.

Aligning Sequences to Profile with MUSCLE

Another feature provided by the plugin is aligning a set of unaligned sequences to an existing profile. To use this feature select the *Align Align sequences to profile with MUSCLE* context menu item.

This option is not available in the original MUSCLE package (v3.7) and is a new functionality for original MUSCLE users. In this mode each sequence from the input file is aligned to the active profile separately and is merged to the result alignment only after all sequences are processed. For example, the alignment in the picture above can be used as a profile again. And the added profile can be used as a set of sequences. The result of such sequences-to-profile alignment is presented on the picture below:



The original alignment is not modified, only columns with gap ('—') character can be inserted.

The second profile was considered as a set of sequences and therefore is modified.

Note that if a file with another alignment is used as a source of unaligned sequences, the gap characters are removed and each input sequence is processed independently.

This method is quite fast, for example an alignment of 3000 sequences (1000 bases each) to the existing profile takes about 5 minutes on the usual Core2Duo computer.

ClustalW

Clustal is a widely used multiple sequence alignment program. It is used for both nucleotide and protein sequences. *ClustalW* is a command-line version of the program.

Clustal home page: http://www.clustal.org

If you are using Windows OS, there are no additional configuration steps required, as *ClustalW* executable file is included to the UGENE distribution package. Otherwise:

- Install the *Clustal* program on your system.
- Set the path to the *ClustalW* executable on the *External tools* tab of UGENE *Application Settings* dialog.

Now you are able to use *Clustal* from UGENE.

Open a multiple sequence alignment file and select the *Align with ClustalW* item in the context menu or in the *Actions* main menu. The *Align with ClustalW* dialog appears (see below), where you can adjust the following parameters:

Gap opening penalty — cost of opening up a new gap in the alignment. Increasing this value will make gaps less frequent.

Gap extension penalty — cost of every item in a gap. Increasing this value will make gaps shorter.

Weight matrix — specifies a single weight matrix for nucleotide sequences or series of matrices for protein sequences.

For nucleotide sequences the weight matrix selected defines the scores assigned to matches and mismatches (including IUB ambiguity codes), it can take values:

- *IUB* — default scoring matrix used by BESTFIT for the comparison of nucleic acid sequences. X's and N's are treated as matches to any IUB ambiguity symbol. All matches score 1.9; all mismatches for IUB symbols score 0.
- *CLUSTALW* — previous system used by ClustalW, in which matches score 1.0 and mismatches score 0. All matches for IUB symbols also score 0.

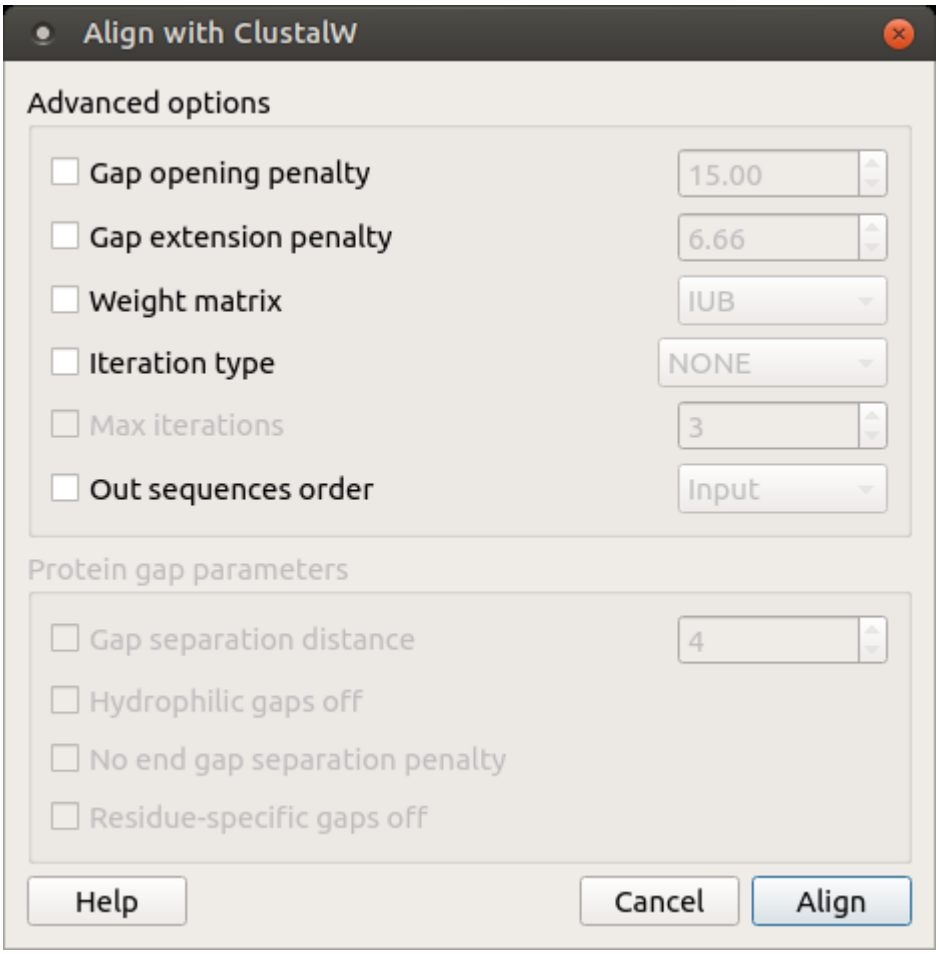
For protein sequences it describes the similarity of each amino acid to each other. The following values are available:

- *BLOSUM* — BLOcks of Amino Acid SUbstitution Matrices first introduced in a paper by Henikoff and Henikoff. These matrices appear to be the best available for carrying out data base similarity (homology searches).
- *PAM* — Point Accepted Mutation matrices introduced by Margaret Dayhoff. These have been extremely widely used since the late '70s.
- *GONNET* — these matrices were derived using almost the same procedure as the Dayhoff one (above) but are much more up to date and are based on a far larger data set. They appear to be more sensitive than the Dayhoff series.
- *ID* — identity matrix which gives a score of 1.0 to two identical amino acids and a score of zero otherwise.

Iteration type — specifies the iteration type to use. During the iteration step each sequence is removed in turn and realigned. It is kept if the resulting alignment is better than the one has been made before. This process is repeated until the score converges or until the maximum number of iterations is reached. Available values are:

- *NONE* — specifies not to use iterations.
- *TREE* — specifies to iterate at each step of the progressive alignment.
- *ALIGNMENT* — specifies to iterate on the final alignment.

Max iterations — maximum number of iterations.



The following parameters are only available for protein sequences:

Gap separation distance — tries to decrease the chances of gaps being too close to each other. Gaps that are less than this distance apart are penalized more than other gaps. This does not prevent close gaps; it makes them less frequent, promoting a block-like appearance of the alignment.

Hydrophilic gaps off — increases the chances of a gap within a run of hydrophilic amino acids.

No end gap separation penalty — treats end gaps just like internal gaps to avoid gaps that are too close.

Residue-specific gaps off — amino acid specific gap penalties that reduce or increase the gap opening penalties at each position in the alignment or sequence. For example, positions that are rich in glycine are more likely to have an adjacent gap than positions that are rich in valine.

MAFFT

Originally, MAFFT is a multiple sequence alignment program for UNIX-like operating systems. However, currently, it is available for macOS, Linux, and Windows. It is used for both nucleotide and protein sequences.

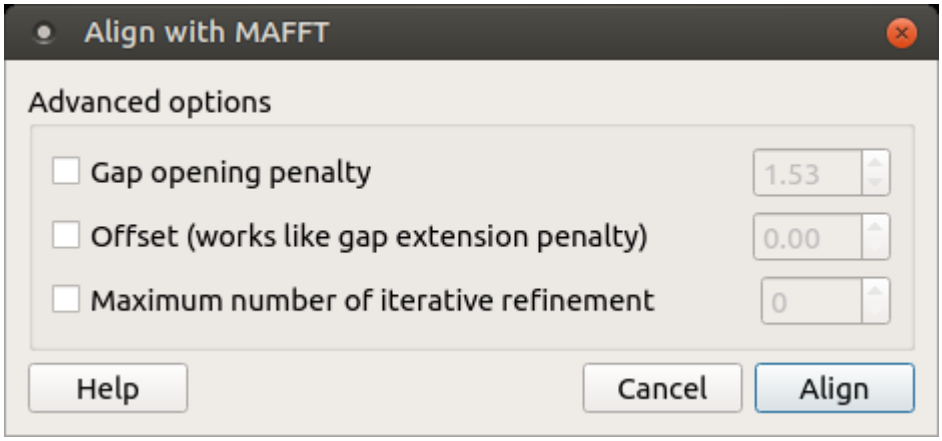
MAFFT homepage: http://mafft.cbrc.jp/alignment/software

To make *MAFFT* available from UGENE:

- Install the *MAFFT* program on your system.
- Set the path to the *MAFFT* executable on the *External tools* tab of UGENE *Application Settings* dialog.

  For example, on Windows, you need to specify the path to the mafft.bat file.

To use *MAFFT* open a multiple sequence alignment file and select the *Align with MAFFT* item in the context menu or in the *Actions* main menu. The following dialog appears:



The following parameters are available:

Gap opening penalty — Gap opening penalty at group-to-group alignment.

Offset (works like gap extension penalty) — offset value, which works like the gap extension penalty, for group-to-group alignment.

Maximum number of iterative refine — specifies the number of cycles of iterative refinement to perform.

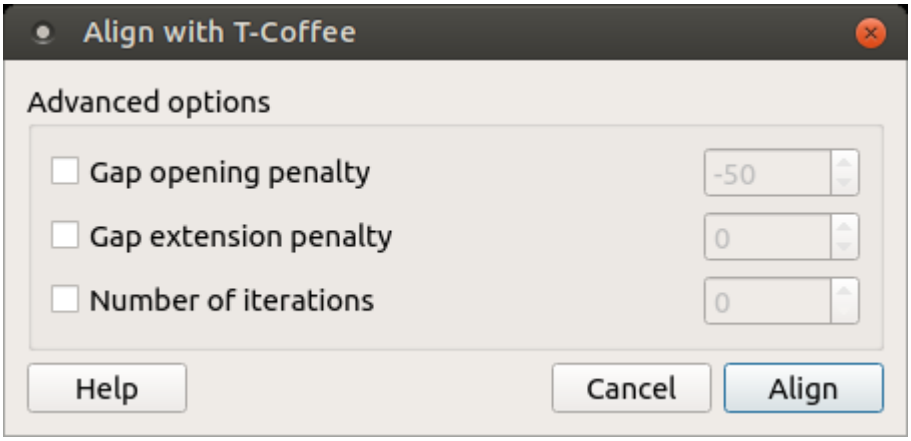
T-Coffee

T-Coffee is a multiple sequence alignment package.

T-Coffee home page: T-Coffee

To make *T-Coffee* available from UGENE see the *External Tools*.

To use *T-Coffee* open a multiple sequence alignment file and select the *Align with T-Coffee* item in the context menu or in the *Actions* main menu. The following dialog appears:



The following parameters are available:

Gap opening penalty — indicates the penalty applied for opening a gap. The penalty must be negative.

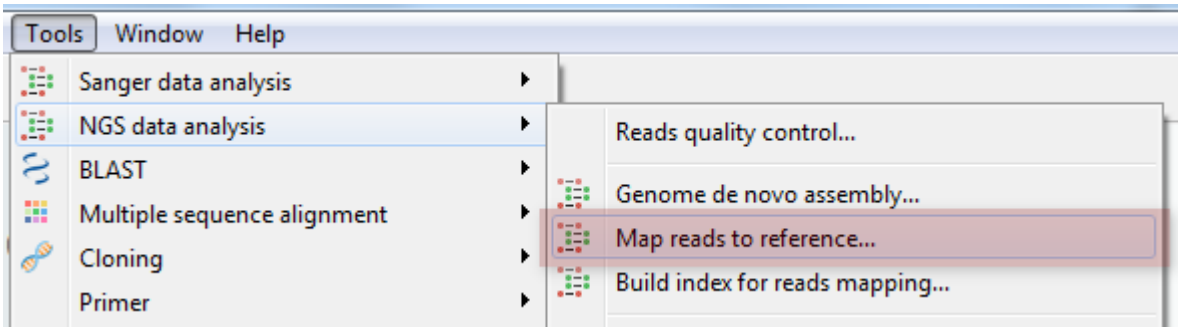
Gap extension penalty — indicates the penalty applied for extending a gap.

Number of iterations — specifies the number of iterations.

Bowtie

Bowtie is a popular short read aligner. Click this link to open *Bowtie* homepage. *Bowtie* is embedded as an *external tool* into UGENE.

Open *Tools DNA Assembly* submenu of the main menu.

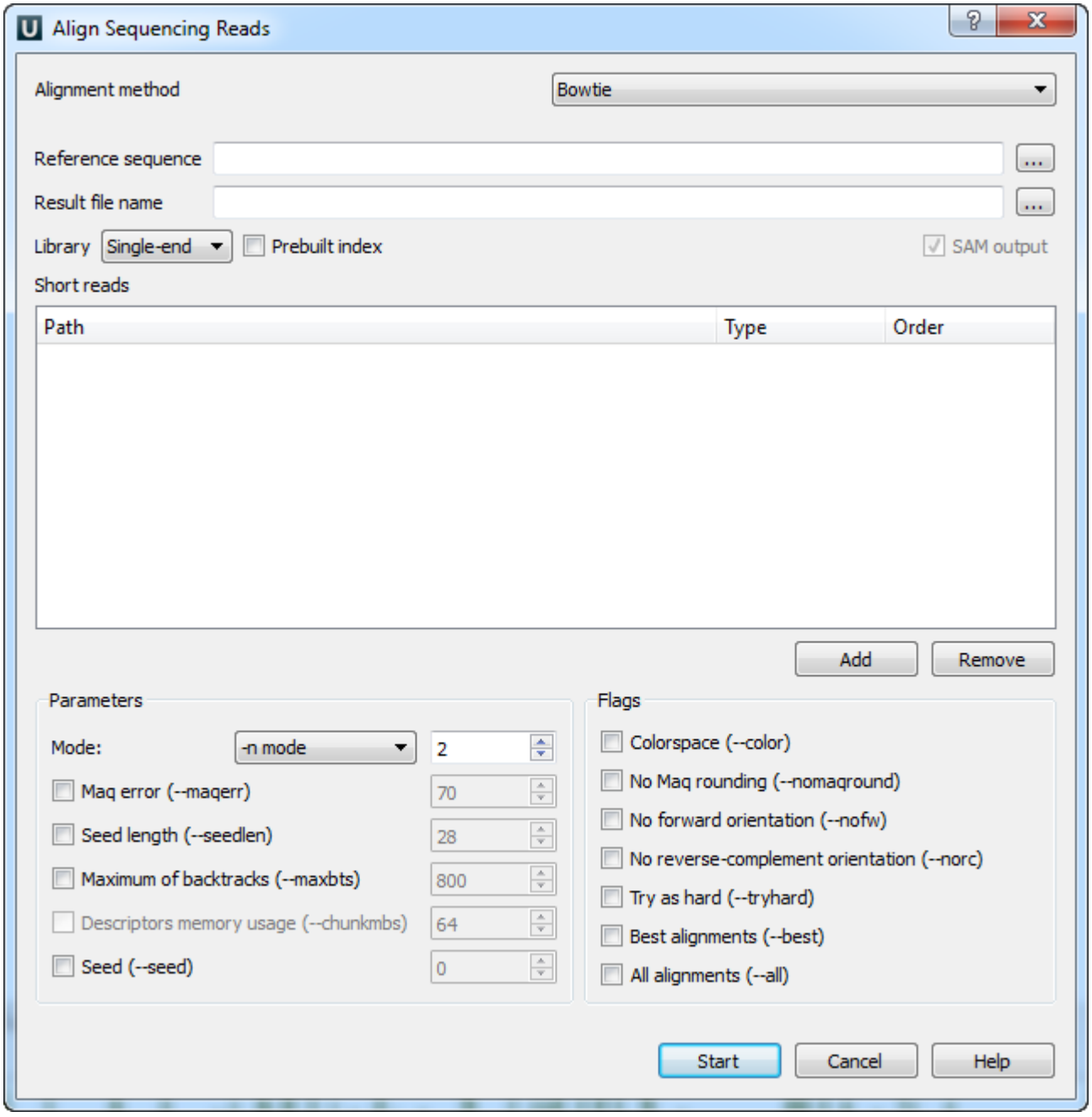


Select the *Align short reads* item to align short reads to a DNA sequence using *Bowtie*. Or select the *Build index* item to build an index for a DNA sequence which can be used to optimize aligning of the short reads to the sequence.

- Bowtie Aligning Short Reads
- Building Index for Bowtie

Bowtie Aligning Short Reads

When you select the *Tools DNA Assembly Align short reads* item in the main menu, the *Align Short Reads* dialog appears. Set value of the *Align short reads method* parameter to *Bowtie*. The dialog looks as follows:



There are the following parameters:

Reference sequence — DNA sequence to align short reads to. This parameter is required.

Result file name — file in SAM format to write the result of the alignment into. This parameter is required.

Library - single-end or paired-end reads.

Prebuilt index — check this box to use an index file instead of a source reference sequence. The index is a set of 6 files with suffixes .1.ebwt, .2.ebwt, .3.ebwt, .4.ebwt, .rev.1.ebwt, and .rev.2.ebwt. The index is created during the alignment. Also you can *build it manually*.

SAM output — always save the output file in the SAM format (the option is disabled for *Bowtie*).

Short reads — each added short read is a small DNA sequence file. At least one read should be added.

⚠ Short reads length for *Bowtie* can't be more than 1024.

You can also configure other parameters. They are the same as in the original *Bowtie* (you can read detailed description of the parameters on the Bowtie manual page).

Select one of the following alignment modes:

The *-n* alignment mode:

When the *-n mode* is selected, *Bowtie* determines which alignments are valid according to the following policy. Alignments may have no more than N mismatches (where N is a number 0-3) in the first L bases (where L is a number 5 or greater, set with *Seed length*) on the high-quality (left) end of the read. The sum of the Phred quality values at all mismatched positions (not just in the seed) may not exceed E (set with *Maq error*). Where qualities are unavailable (e.g. if the reads are from a FASTA file), the Phred quality defaults to 40.

The *-v* alignment mode:

In *-v mode*, alignments may have no more than V mismatches, where V may be a number from 0 through 3. Quality values are ignored. The *-v mode* is mutually exclusive with the *-n mode*.

The following parameters are available:

Maq error (–maqerr) — maximum permitted total of quality values at all mismatched read positions throughout the entire alignment, not just in the "seed". The default is 70. By default, *Bowtie* rounds quality values to the nearest 10 and saturates at 30. Note that the rounding can be disabled with *No Maq rounding*.

Seed Length (–seedlen) — the number of bases on the high-quality end of the read to which the *-n* applies. The lowest permitted setting is 5 and the default is 28.

Maximum of backtracks (-maxbts) — the maximum number of backtracks (default: 125 without *Best*, 800 with *Best*). A "backtrack" is the introduction of a speculative substitution into the alignment.

Descriptors memory usage (–chunkmbs) — the number of megabytes of memory a given thread is given to store path descriptors in the *Best* flag. Default: 64. This parameter is available if the *Best* flag is checked.

Seed (–seed) — pseudo-random number generator.

The following flags are available:

Colorspace (–color) — the input is read in colorspace, colors are encoded as characters A/C/G/T (A=blue, C=green, G=orange, T=red).

No Maq rounding (–nomaqround) — Maq (Mapping and Assembly with Quality) accepts quality values in the Phred quality scale, but internally rounds values to the nearest 10, with a maximum of 30. By default, *Bowtie* also rounds this way. *No Maq rounding* prevents this rounding in *Bowtie*.

No forward orientation (–nofw) — do not attempt to align against the forward reference strand.

No reverse-complement orientation (–norc) — do not attempt to align against the reverse-complement reference strand.

Try as hard (–tryhard) — try as hard as possible to find valid alignments when they exist, including paired-end alignments.

Best alignments (–best) — make *Bowtie* guarantee that reported singleton alignments are "best" in terms of stratum (i.e. number of mismatches, or mismatches in the seed for the case of *-n mode*) and in terms of the quality values at the mismatched position(s).

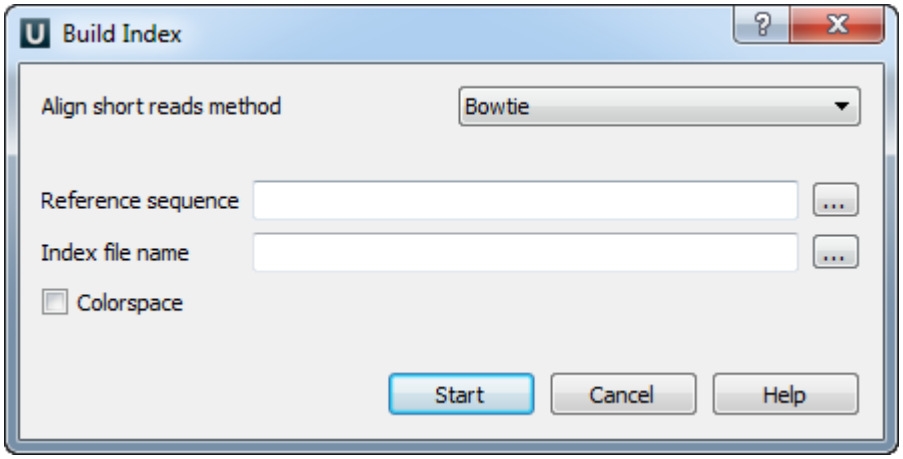
All alignments (–all) — report all valid alignments per read or pair. Validity of alignments is determined by the alignment policy (combined effects of *-n mode*, *-v mode*, *Seed length*, and *Maq error*).

Select the required parameters and press the *Start* button.

Building Index for Bowtie

To build *Bowtie* index select the *Tools NGS data analysis Build index for reads mapping* item in the main menu. The *Build Index* dialog appears. Set the *Align short reads method* parameter to *Bowtie*.

The dialog looks as follows:



There are the following parameters:

Reference sequence — DNA sequence to which short reads would be aligned to. This parameter is required.

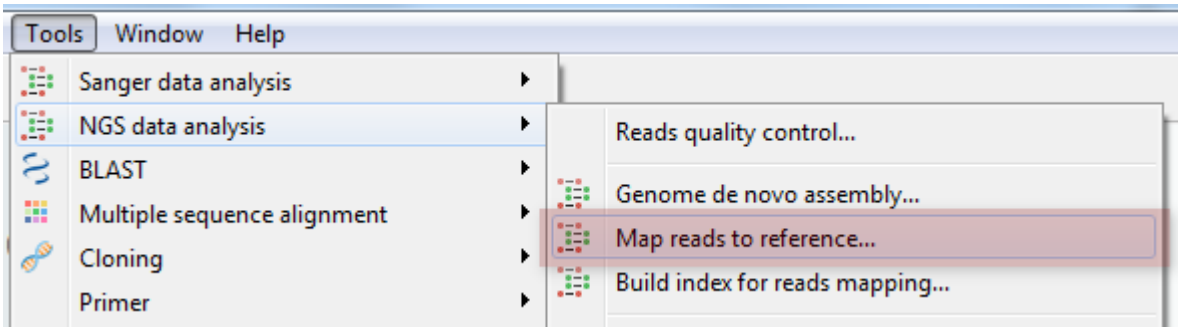
Index file name — a file to save the created index to. This parameter is required.

Colorspace (–color) — the input is read in colorspace, colors are encoded as characters A/C/G/T (A=blue, C=green, G=orange, T=red).

Bowtie 2

Bowtie 2 is a popular ultrafast and memory-efficient tool for aligning sequencing reads to long reference sequences. Click this link to open *Bowtie 2* homepage. *Bowtie 2* is embedded as an *external tool* into UGENE.

Open *Tools* *Align to reference* submenu of the main menu.



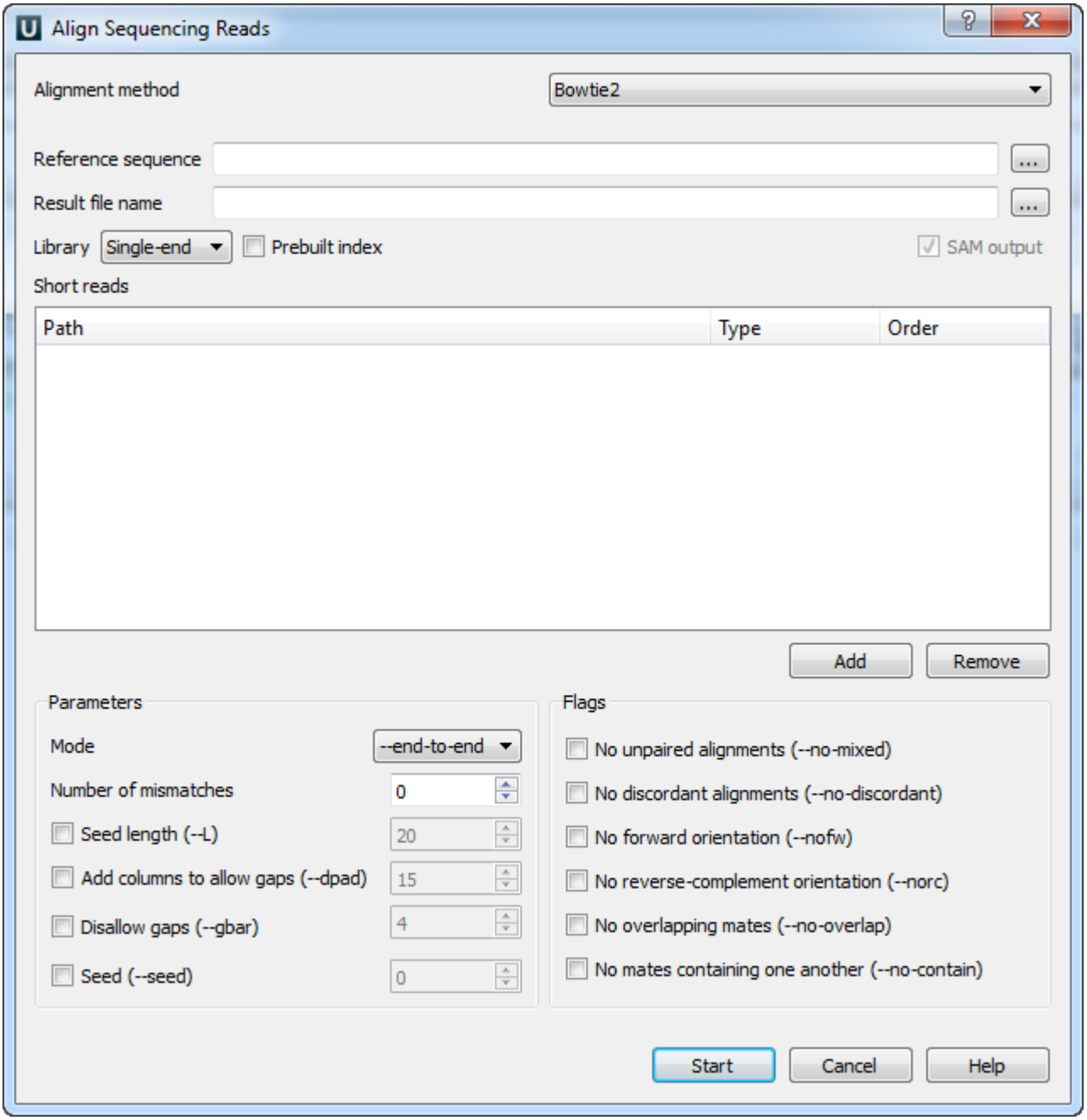
Select the *Align short reads* item to align short reads to a DNA sequence.

Or select the *Build index* item to build an index for a DNA sequence which can be used to optimize aligning of the short reads to the sequence:

- Bowtie 2 Aligning Short Reads
- Building Index for Bowtie 2

Bowtie 2 Aligning Short Reads

When you select the *Tools* *Align to reference* *Align short reads* item in the main menu, the *Align Sequencing Reads* dialog appears. Set value of the *Align short reads method* parameter to *Bowtie 2*. The dialog looks as follows:



There are the following parameters:

Reference sequence — DNA sequence to align short reads to. This parameter is required.

Result file name — file in SAM format to write the result of the alignment into. This parameter is required.

Library - single-end or paired-end reads.

Prebuilt index — check this box to use an index file instead of a source reference sequence. The index is a set of 6 files with suffixes .1.ebwt, .2.ebwt, .3.ebwt, .4.ebwt, .rev.1.ebwt, and .rev.2.ebwt. The index is created during the alignment. Also you can *build it manually*.

SAM output — always save the output file in the SAM format (the option is disabled for *Bowtie*).

Short reads — each added short read is a small DNA sequence file. At least one read should be added.

You can also configure other parameters. They are the same as in the original *Bowtie 2* (you can read detailed description of the parameters on the Bowtie 2 manual page).

Select one of the following alignment modes:

The --end-to-end alignment mode:

By default, Bowtie 2 performs end-to-end read alignment. That is, it searches for alignments involving all of the read characters. This is also called an "untrimmed" or "unclipped" alignment.

When the --local option is specified, Bowtie 2 performs local read alignment. In this mode, Bowtie 2 might "trim" or "clip" some read characters from one or both ends of the alignment if doing so maximizes the alignment score.

The following parameters are available:

Number of mismatches (--N) — sets the number of mismatches to allowed in a seed alignment during multiseed alignment. Can be set to 0 or 1. Setting this higher makes alignment slower (often much slower) but increases sensitivity.

Seed length (–L) — Sets the length of the seed substrings to align during multiseed alignment. Smaller values make alignment slower but more senstive.

Add columns to allow gaps (--dpad) — "Pads" dynamic programming problems by `<int>` columns on either side to allow gaps.

Disallow gaps (–gbar) — disallow gaps within `<int>` positions of the beginning or end of the read.

Seed (–seed) — use `<int>` as the seed for pseudo-random number generator.

The following flags are available:

No unpaired alignments (–no-mixed) — by default, when `bowtie2` cannot find a concordant or discordant alignment for a pair, it then tries to find alignments for the individual mates. This option disables that behavior.

No discordant alignments (–no-discordant) — by default, `bowtie2` looks for discordant alignments if it cannot find any concordant alignments. A discordant alignment is an alignment where both mates align uniquely, but that does not satisfy the paired-end constraints. This option disables that behavior.

No forward orientation (–nofw) — if `--nofw` is specified, `bowtie2` will not attempt to align unpaired reads to the forward (Watson) reference strand.

No reverse-complement orientation (–norc) — if `--norc` is specified, `bowtie2` will not attempt to align unpaired reads against the reverse-complement (Crick) reference strand.

No overlapping mates (–no-overlap) — if one mate alignment overlaps the other at all, consider that to be non-concordant.

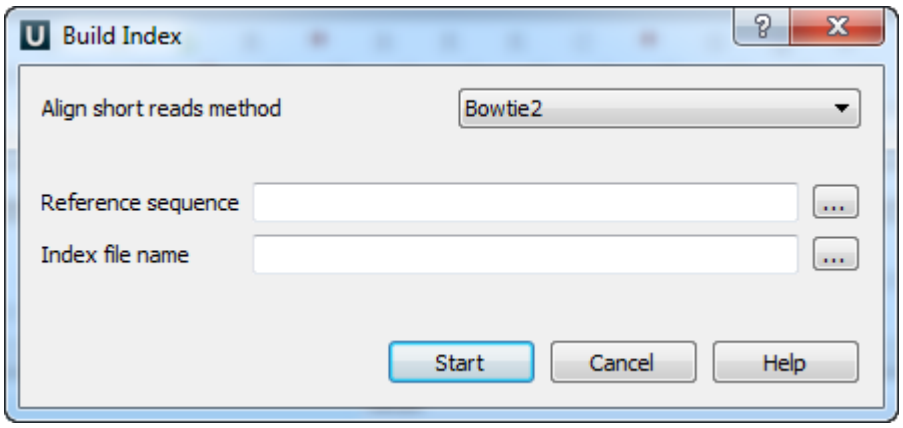
No mates containing one another (–no-contain) — if one mate alignment contains the other, consider that to be non-concordant.

Select the required parameters and press the *Start* button.

Building Index for Bowtie 2

To build *Bowtie 2* index select the *Tools NGS data analysis Build index for reads mapping* item in the main menu. The *Build Index* dialog appears. Set the *Align short reads method* parameter to *Bowtie 2*.

The dialog looks as follows:



There are the following parameters:

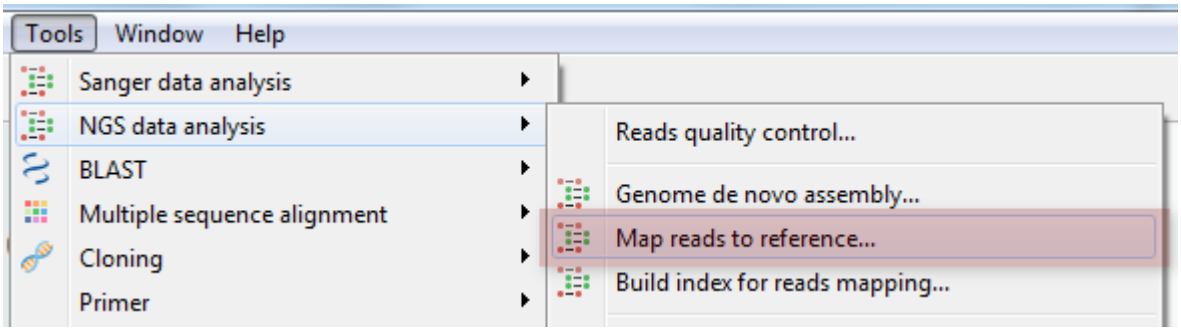
Reference sequence — DNA sequence to which short reads would be aligned to. This parameter is required.

Index file name — a file to save the created index to. This parameter is required.

BWA

BWA is a fast light-weighted tool that aligns relatively short reads to a reference sequence. Click this link to open *BWA* homepage. *BWA* is embedded as an *external tool* into UGENE.

Open *Tools DNA assembly* submenu of the main menu.

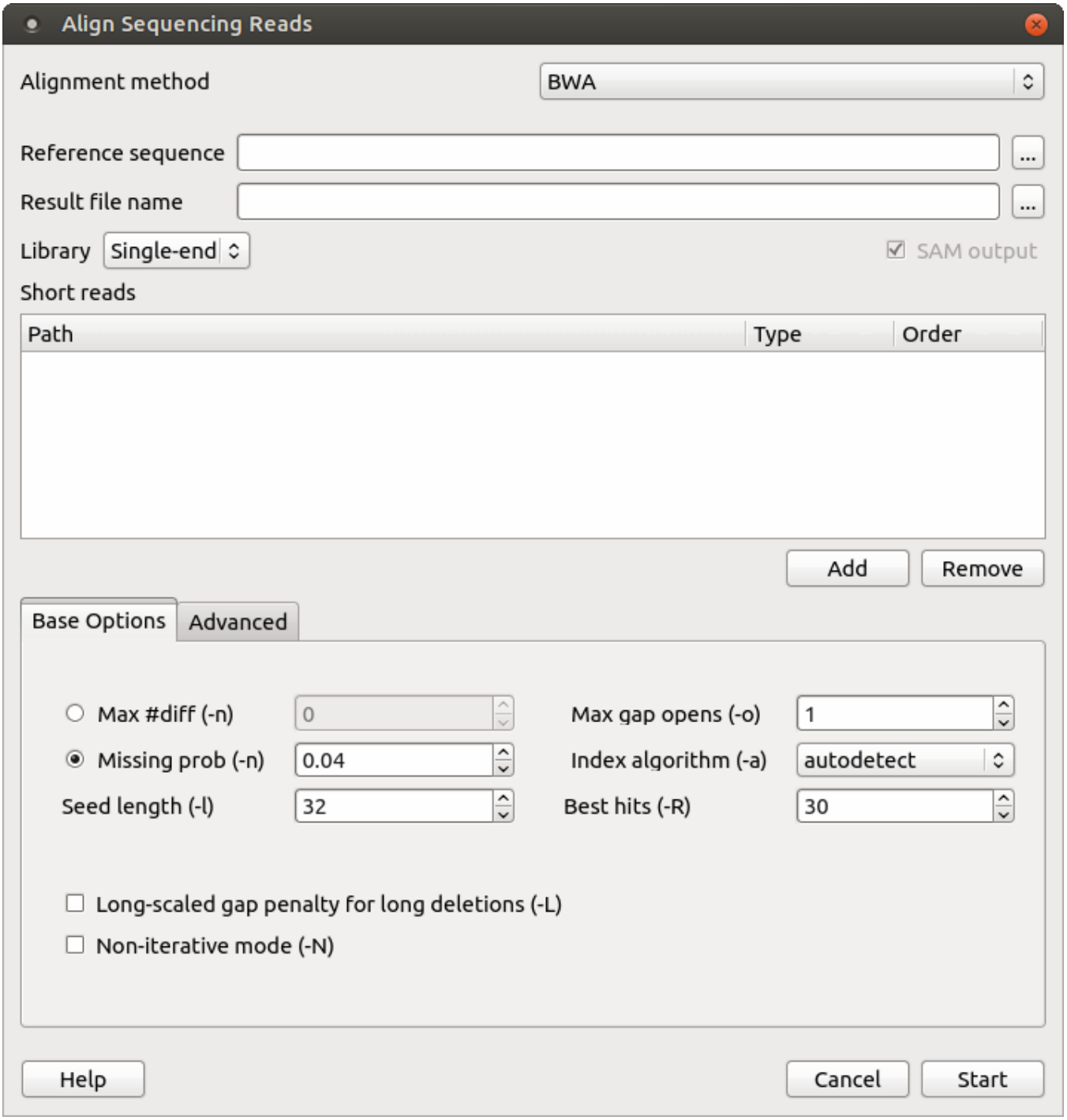


Select the *Align short reads* item to align short reads to a DNA sequence using *BWA*. Or select the *Build index* item to build an index for a DNA sequence which can be used to optimize aligning of short reads.

- Aligning Short Reads with BWA
- Building Index for BWA

Aligning Short Reads with BWA

When you select the *Tools DNA Assembly Align short reads* item in the main menu, the *Align Short Reads* dialog appears. Set value of the *Align short reads method* parameter to *BWA*. The dialog looks as follows:



There are the following parameters:

Reference sequence — DNA sequence to align short reads to. This parameter is required.

Result file name — file in SAM format to write the result of the alignment into. This parameter is required.

Library - single-end or paired-end reads.

Prebuilt index — check this box to use an index file instead of a source reference sequence. Also you can *build it manually*.

SAM output — always save the output file in the SAM format (the option is disabled for *BWA*).

Short reads — each added short read is a small DNA sequence file. At least one read should be added.

You can also configure other parameters. They are the same as in the original *BWA* (you can read detailed description of the parameters on the BWA manual page). Select one of the following parameters, that correspond to the *-n* option in the original *BWA*.

Max #diff (-n) — maximum edit distance. An integer value should be input.

Missing prob (-n) — the fraction of missing alignments given 2% uniform base error rate. A float value is used.

Seed length (-l) — take the subsequence of the specified length as seed. If the specified length is larger than the query sequence, seeding will be disabled. For long reads, this option is typically ranged from 25 to 35.

Max gap opens (-o) — maximum number of gap opens.

Index algorithm (-a) — algorithm for constructing BWA index.

It implements three different algorithms:

- *is* — designed for short reads up to ~200bp with low error rate (<3%). It does gapped global alignment w.r.t. reads, supports paired-end reads, and is one of the fastest short read alignment algorithms to date while also visiting suboptimal hits.
- *bwtsw* — is designed for long reads with more errors. It performs heuristic Smith-Waterman-like alignment to find high-scoring local hits. Algorithm implemented in BWA-SW. On low-error short queries, *BWA-SW*. is slower and less accurate than the *is* algorithm, but on long reads, it is better.
- *div* — does not work for long genomes.

Best hits (-R) — proceed with suboptimal alignments if there are no more than specified number of equally best hits. This option only affects paired-end mapping. Increasing this threshold helps to improve the pairing accuracy at the cost of speed, especially for short reads (~32bp).

Long-scaled gap penalty for long deletion (-L) — long-scaled gap penalty for long deletion.

Non-iterative mode (-N) — disable iterative search. All hits with no more than *Max #diff* differences will be found. This mode is much slower than the default.

You can also configure the following advanced parameters:

Enable long gaps — checking this box allows one to set the *Max gap extentions* parameter.

Max gap extensions (-e) — maximum number of gap extensions.

Indel offset (-i) — disallow insertions and deletions within the specified number of base pairs towards the ends.

Max long deletion extensions (-d) — disallow a long deletions within the specified number of base pairs towards the 3`-end.

Max queue entries (-m) — maximum queue entries.

Barcode length (-B) — length of barcode starting from the 5`-end. When the specified length is positive, the barcode of each read will be trimmed before mapping and will be written at the BC SAM tag. For paired-end reads, the barcode from both ends are concatenated.

Threads (-t) — number of threads.

Max seed differences (-k) — maximum edit distance in the seed.

Mismatch penalty (-M) — *BWA* will not search for suboptimal hits with a score lower than the specified value.

Gap open penalty (-O) — gap open penalty.

Gap extension penalty (-E) — gap extension penalty.

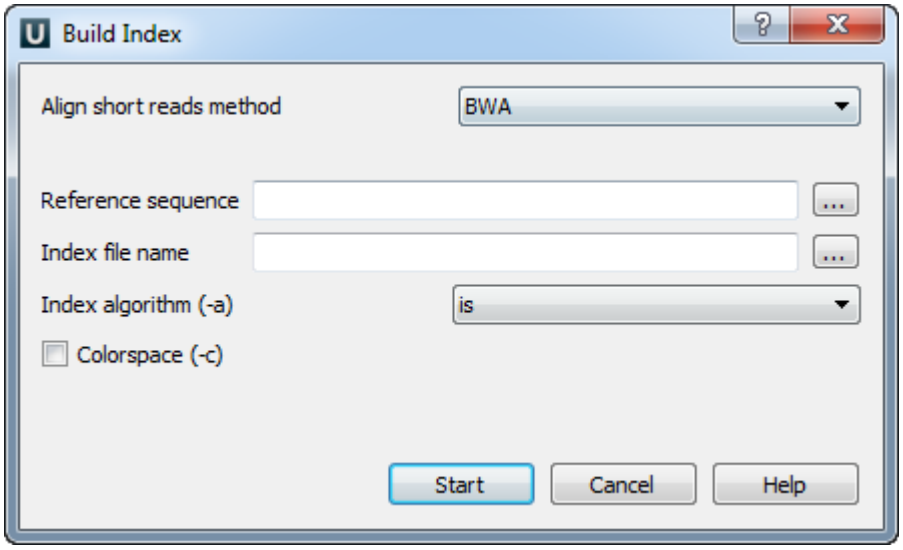
Quality threshold (-q) — parameter for read trimming.

Select the required parameters and press the *Start* button.

Building Index for BWA

To build *BWA* index select the *Tools NGS data analysis Build index for reads mapping* item in the main menu. The *Build Index* dialog appears. Set the *Ali gn short reads method* parameter to *BWA*.

The dialog looks as follows:



There are the following parameters:

Reference sequence — DNA sequence to which short reads would be aligned to. This parameter is required.

Index file name — file to save index to. This parameter is required.

Index algorithm (-a) — Algorithm for constructing BWA index. Available options are:

It implements three different algorithms

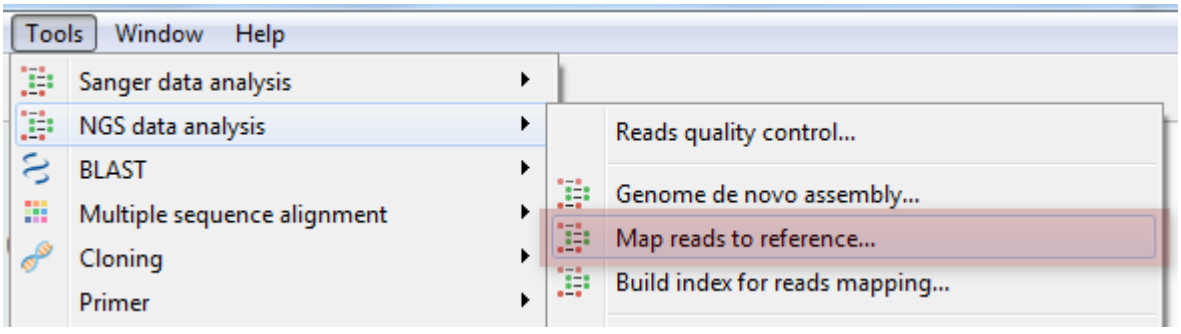
- *is* — designed for short reads up to ~200bp with low error rate (<3%). It does gapped global alignment w.r.t. reads, supports paired-end reads, and is one of the fastest short read alignment algorithms to date while also visiting suboptimal hits.
- *bwtsw* — is designed for long reads with more errors. It performs heuristic Smith-Waterman-like alignment to find high-scoring local hits. Algorithm implemented in BWA-SW. On low-error short queries, *BWA-SW*. is slower and less accurate than the *is* algorithm, but on long reads, it is better.
- *div* — does not work for long genomes.

Colorspace (–color) — the input is read in colorspace, colors are encoded as characters A/C/G/T (A=blue, C=green, G=orange, T=red).

BWA-SW

BWA is a fast light-weighted tool that aligns relatively short reads to a reference sequence. Click this link to open *BWA* homepage. BWA-SW share similar features such as long-read support and split alignment. *BWA-SW* is embedded as an *external tool* into UGENE.

Open *Tools* *Align to reference* submenu of the main menu.

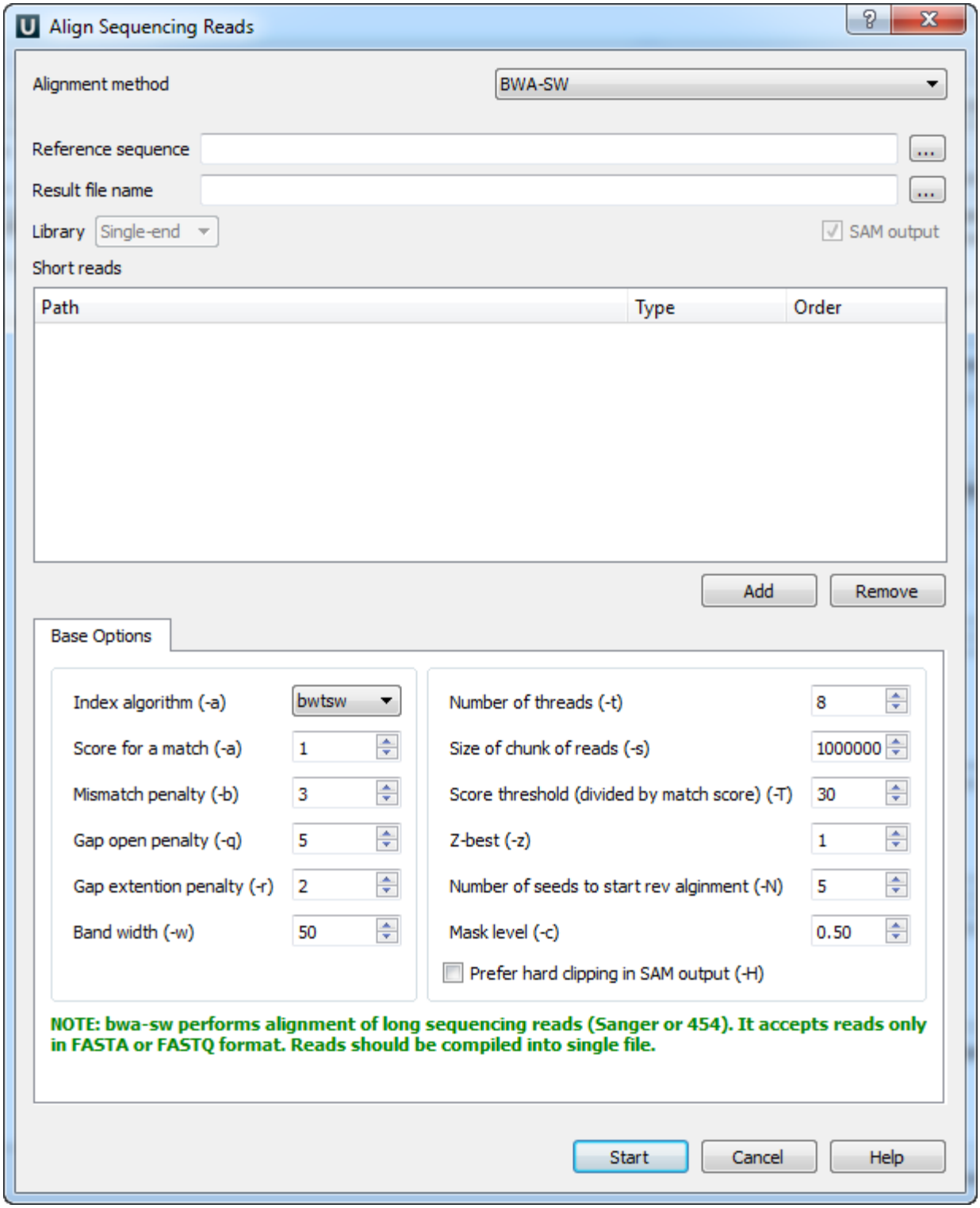


Select the *Align short reads* item to align short reads to a DNA sequence using *BWA-SW*. Or select the *Build index* item to build an index for a DNA sequence which can be used to optimize aligning of short reads.

- Aligning Short Reads with BWA-SW
- Building Index for BWA-SW

Aligning Short Reads with BWA-SW

When you select the *Tools Align to reference Align short reads* item in the main menu, the *Align Sequencing Reads* dialog appears. Set value of the *Align short reads method* parameter to *BWA-SW*. The dialog looks as follows:



There are the following parameters:

Reference sequence — DNA sequence to align short reads to. This parameter is required.

Result file name — file in SAM format to write the result of the alignment into. This parameter is required.

SAM output — always save the output file in the SAM format (the option is disabled for *BWA*).

Short reads — each added short read is a small DNA sequence file. At least one read should be added.

You can also configure other parameters.

Index algorithm (-a) — algorithm for constructing BWA-SW index.

It implements three different algorithms:

- *is* — designed for short reads up to ~200bp with low error rate (<3%). It does gapped global alignment w.r.t. reads, supports paired-end reads, and is one of the fastest short read alignment algorithms to date while also visiting suboptimal hits.
- *bwtsw* — is designed for long reads with more errors. It performs heuristic Smith-Waterman-like alignment to find high-scoring local hits. Algorithm implemented in BWA-SW. On low-error short queries, *BWA-SW*. is slower and less accurate than the *is* algorithm, but on long reads, it is better.
- *div* — does not work for long genomes.

Score for a match (-a) — score of a match.

Mismatch penalty (-b) — mismatch penalty.

Gap open penalty (-q) — gap open penalty.

Gap extention penalty (-r) — Gap extension penalty. The penalty for a contiguous gap of size k is q+k*r.

Band width (-w) - Band width in the banded alignment.

Number of threads (-t) - Number of threads in the multi-threading mode.

Size of chunk of reads (-s) - Maximum SA interval size for initiating a seed. Higher -s increases accuracy at the cost of speed.

Score threshold (divided by much score) (-T) - minimum score threshold.

Z-best (-z) - Z-best heuristics. Higher -z increases accuracy at the cost of speed.

Number of seeds to start rev alignment (-N) - Minimum number of seeds supporting the resultant alignment to skip reverse alignment.

Mask level (-c) - Coefficient for threshold adjustment according to query length. Given an l-long query, the threshold for a hit to be retained is a*max {T,c*log(l)}.

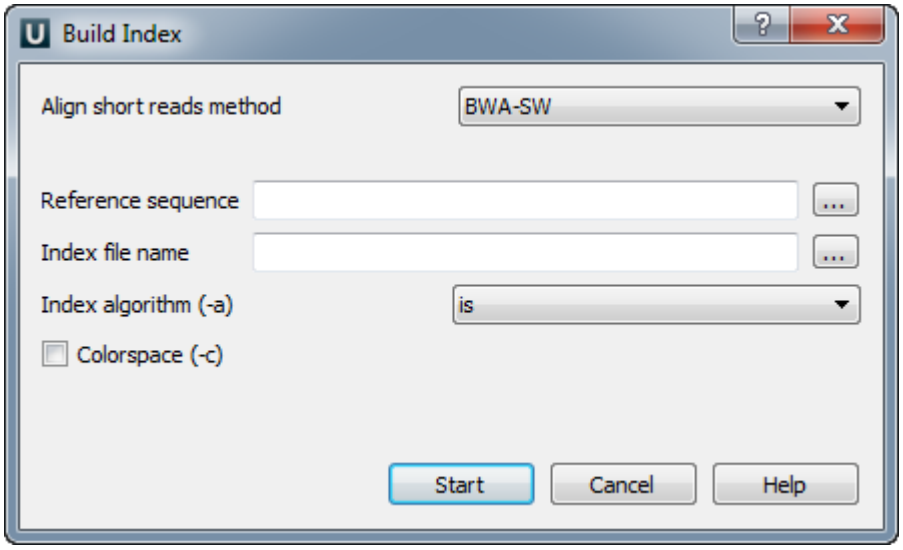
Prefer hard clipping in SAM output (-H) - use hard clipping in the SAM output. This option may dramatically reduce the redundancy of output when mapping long contig or BAC sequences.

Select the required parameters and press the *Start* button.

Building Index for BWA-SW

To build *BWA-SW* index select the *Tools NGS data analysis Build index for reads mapping* item in the main menu. The *Build Index* dialog will appears. Set the *Align short reads method* parameter to *BWA-SW*.

The dialog looks as follows:



There are the following parameters:

Reference sequence — DNA sequence to which short reads would be aligned to. This parameter is required.

Index file name — file to save index to. This parameter is required.

Index algorithm (-a) — Algorithm for constructing BWA index. Available options are:

It implements three different algorithms

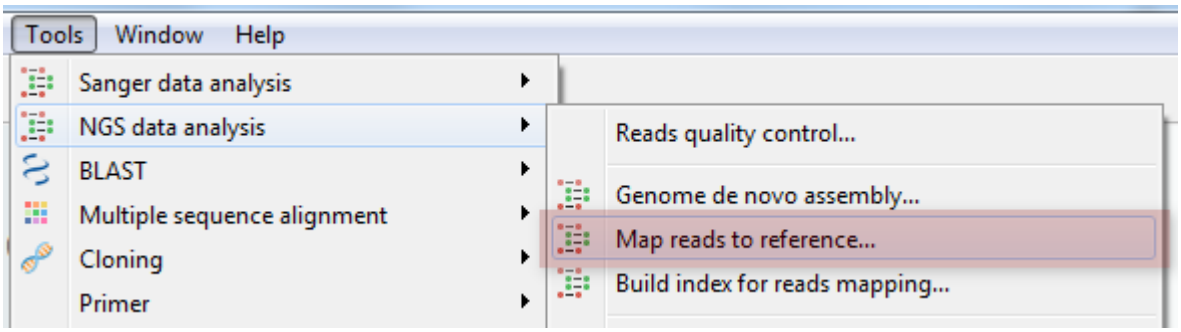
- *is* — designed for short reads up to ~200bp with low error rate (<3%). It does gapped global alignment w.r.t. reads, supports paired-end reads, and is one of the fastest short read alignment algorithms to date while also visiting suboptimal hits.
- *bwtsw* — is designed for long reads with more errors. It performs heuristic Smith-Waterman-like alignment to find high-scoring local hits. Algorithm implemented in BWA-SW. On low-error short queries, *BWA-SW*. is slower and less accurate than the *is* algorithm, but on long reads, it is better.
- *div* — does not work for long genomes.

Colorspace (–color) — the input is read in colorspace, colors are encoded as characters A/C/G/T (A=blue, C=green, G=orange, T=red).

BWA-MEM

BWA is a fast light-weighted tool that aligns relatively short reads to a reference sequence. Click this link to open *BWA* homepage. BWA-MEM is generally recommended for high-quality queries as it is faster and more accurate. BWA-MEM also has better performance than BWA-backtrack for 70-100bp Illumina reads.

Open *Tools Align to reference* submenu of the main menu.

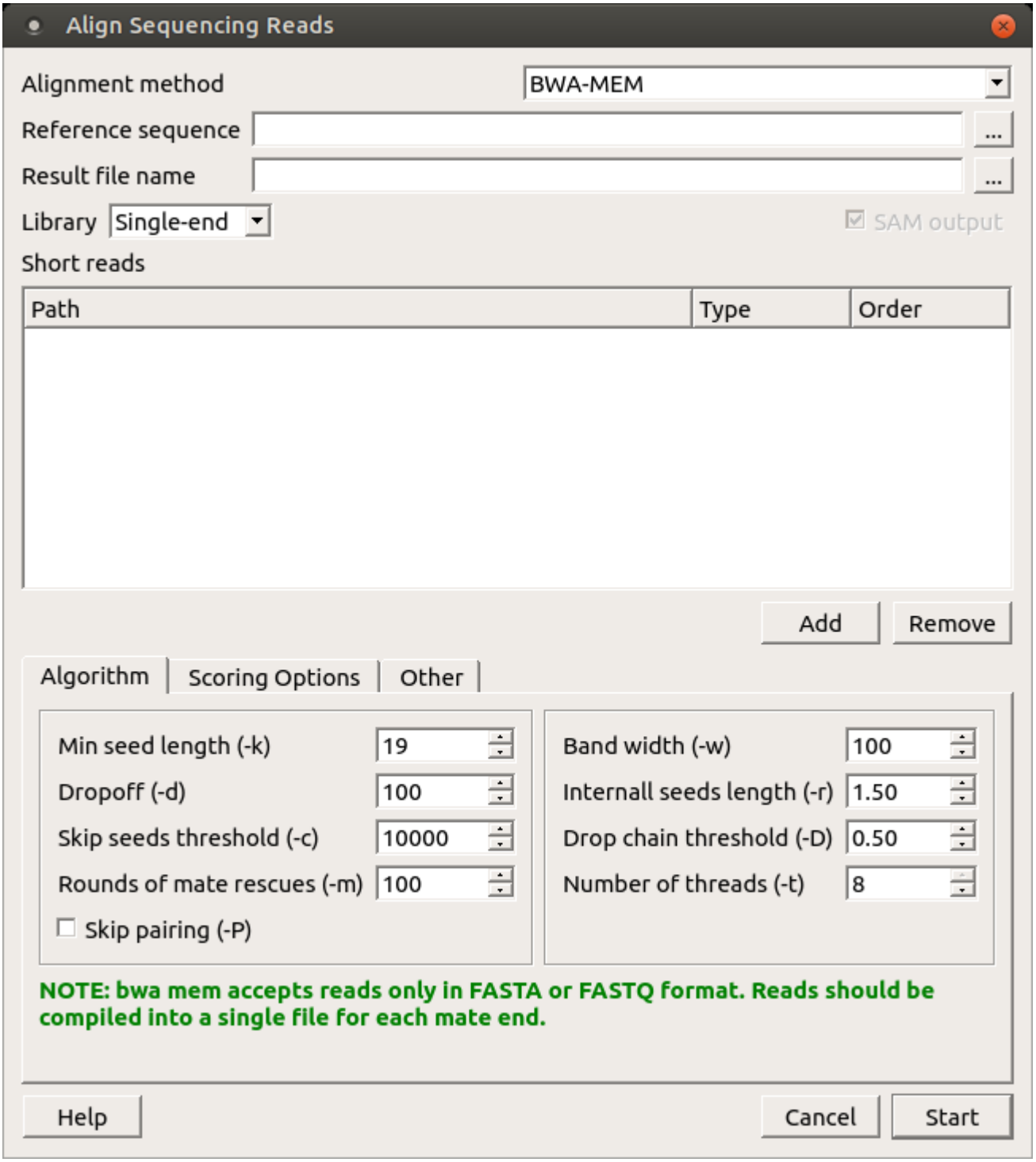


Select the *Align short reads* item to align short reads to a DNA sequence using *BWA-MEM*. Or select the *Build index* item to build an index for a DNA sequence which can be used to optimize aligning of short reads.

- Aligning Short Reads with BWA-MEM
- Building Index for BWA-MEM

Aligning Short Reads with BWA-MEM

When you select the *Tools Align to reference Align short reads* item in the main menu, the *Align Sequencing Reads* dialog appears. Set value of the *Align short reads method* parameter to *BWA-MEM*. The dialog looks as follows:



There are the following parameters:

Reference sequence — DNA sequence to align short reads to. This parameter is required.

Result file name — file in SAM format to write the result of the alignment into. This parameter is required.

Prebuilt index — check this box to use an index file instead of a source reference sequence. Also you can *build it manually*.

SAM output — always save the output file in the SAM format (the option is disabled for *BWA*).

Short reads — each added short read is a small DNA sequence file. At least one read should be added.

You can also configure other parameters.

Index algorithm (-a) — algorithm for constructing BWA index.

It implements three different algorithms:

- *is* — designed for short reads up to ~200bp with low error rate (<3%). It does gapped global alignment w.r.t. reads, supports paired-end reads, and is one of the fastest short read alignment algorithms to date while also visiting suboptimal hits.
- *bwtsw* — is designed for long reads with more errors. It performs heuristic Smith-Waterman-like alignment to find high-scoring local hits. Algorithm implemented in BWA-SW. On low-error short queries, *BWA-SW*. is slower and less accurate than the *is* algorithm, but on long reads, it is better.
- *div* — does not work for long genomes.

Number of threads (-t) — number of threads.

Min seed length (-k) — minimum seed length. Matches shorter than *INT* will be missed. The alignment speed is usually insensitive to this value unless it significantly deviates 20.

Band width (-w) — band width. Essentially, gaps longer than *INT* will not be found. Note that the maximum gap length is also affected by the scoring matrix and the hit length, not solely determined by this option.

Dropoff (-d) — off-diagonal X-dropoff (Z-dropoff). Stop extension when the difference between the best and the current extension score is above $|i-j|*A+INT$, where *i* and *j* are the current positions of the query and reference, respectively, and *A* is the matching score. Z-dropoff is similar to BLAST's X-dropoff except that it doesn't penalize gaps in one of the sequences in the alignment. Z-dropoff not only avoids unnecessary extension, but also reduces poor alignments inside a long good alignment.

Internall seeds length (-r) - trigger re-seeding for a MEM longer than *minSeedLen*FLOAT*. This is a key heuristic parameter for tuning the performance. Larger value yields fewer seeds, which leads to faster alignment speed but lower accuracy.

Skip seeds threshold (-c) - discard a MEM if it has more than *INT* occurence in the genome. This is an insensitive parameter.

Drop chain threshold (-D) - drop chains shorter than FLOAT fraction of the longest overlapping chain.

Rounds of mate rescues (-m) - perform at most INT rounds of mate rescues for each read.

Skip mate rescue (-S) - skip mate rescue.

Skip pairing (-P) - in the paired-end mode, perform SW to rescue missing hits only but do not try to find hits that fit a proper pair.

Score for a match (-A) - matching score.

Mismatch penalty (-B) - mismatch penalty. The sequence error rate is approximately: {.75 * exp[-log(4) * B/A]}.

Gap open penalty (-O) - gap open penalty.

Gap extention penalty (-E) - gap extension penalty. A gap of length k costs O + k*E (i.e. *Gap open penalty* is for opening a zero-length gap).

Penalty for clipping (-L) - clipping penalty. When performing SW extension, BWA-MEM keeps track of the best score reaching the end of query. If this score is larger than the best SW score minus the clipping penalty, clipping will not be applied. Note that in this case, the SAM AS tag reports the best SW score; clipping penalty is not deducted.

Penalty unpaired (-U) - penalty for an unpaired read pair. BWA-MEM scores an unpaired read pair as scoreRead1+scoreRead2-*INT* and scores a paired as scoreRead1+scoreRead2-insertPenalty. It compares these two scores to determine whether we should force pairing.

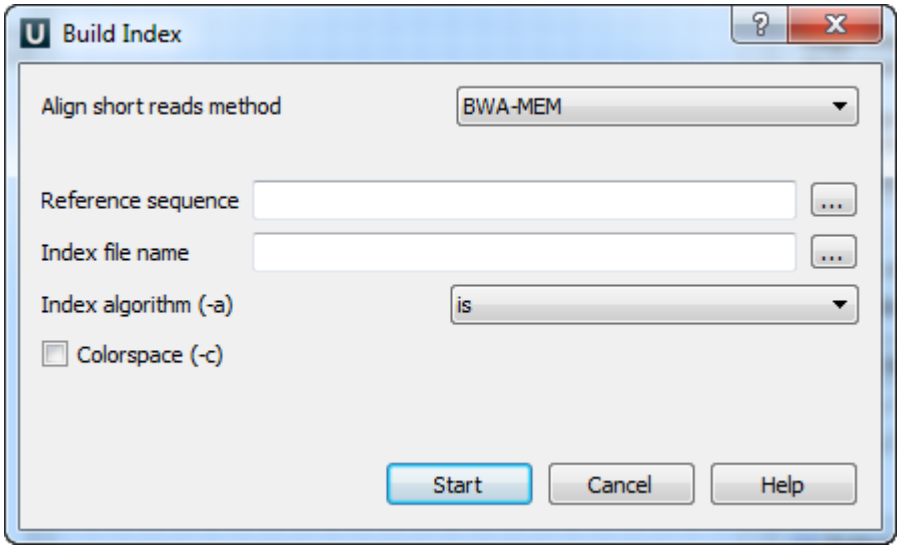
Score threshold (-T) - don't output alignment with score lower than score threshold. This option only affects output.

Select the required parameters and press the *Start* button.

Building Index for BWA-MEM

To build *BWA-SW* index select the *Tools NGS data analysis Build index for reads mapping* item in the main menu. The *Build Index* dialog will appears. Set the *Align short reads method* parameter to *BWA-MEM*.

The dialog looks as follows:



There are the following parameters:

Reference sequence — DNA sequence to which short reads would be aligned to. This parameter is required.

Index file name — file to save index to. This parameter is required.

Index algorithm (-a) — Algorithm for constructing BWA index. Available options are:

It implements three different algorithms

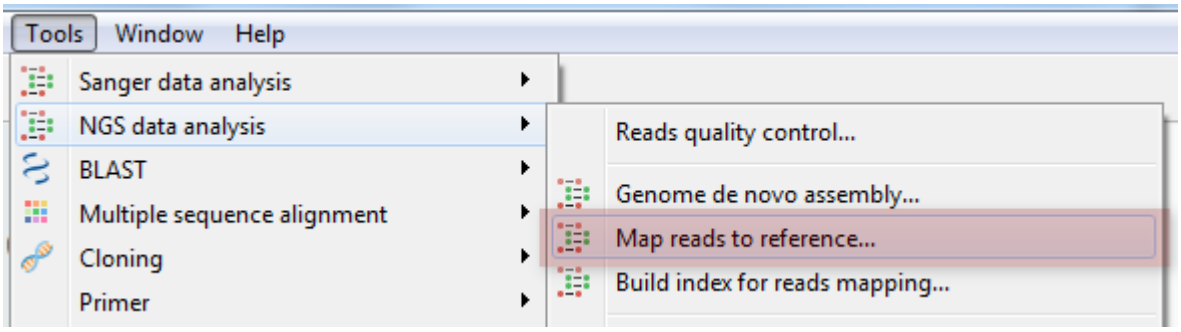
- *is* — designed for short reads up to ~200bp with low error rate (<3%). It does gapped global alignment w.r.t. reads, supports paired-end reads, and is one of the fastest short read alignment algorithms to date while also visiting suboptimal hits.
- *bwtsw* — is designed for long reads with more errors. It performs heuristic Smith-Waterman-like alignment to find high-scoring local hits. Algorithm implemented in BWA-SW. On low-error short queries, *BWA-SW*. is slower and less accurate than the *is* algorithm, but on long reads, it is better.
- *div* — does not work for long genomes.

Colorspace (–c) — the input is read in colorspace, colors are encoded as characters A/C/G/T (A=blue, C=green, G=orange, T=red).

UGENE Genome Aligner

The UGENE Genome Aligner is a fast short read aligner. It aligns DNA sequences of various lengths to the reference genome with configurable mismatch rate.

It is available from the *Tools DNA assembly* submenu of the main menu.

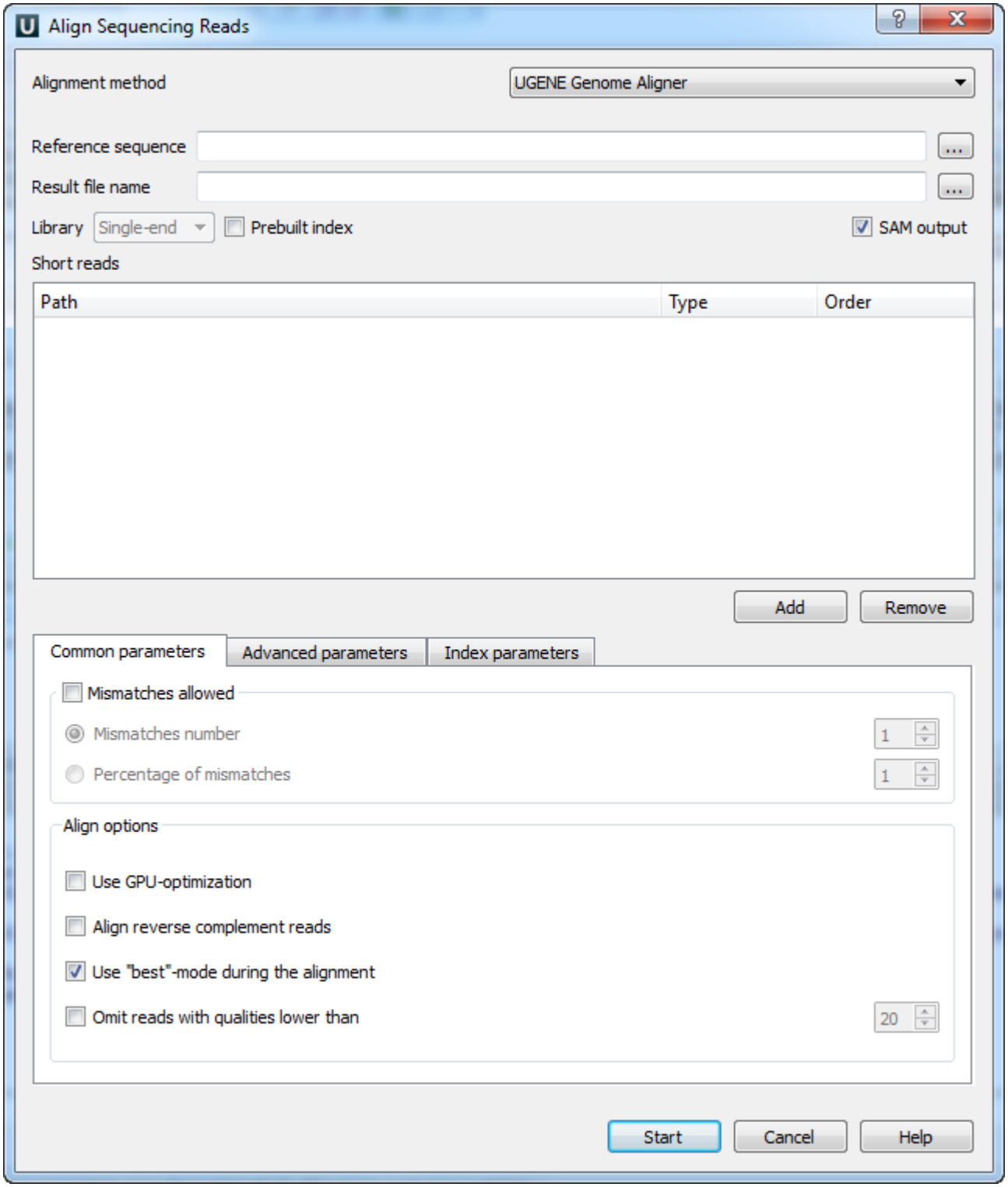


Select the *Align short reads* item to align short reads to a DNA sequence or *Build index* item to build an index for a DNA sequence which can be used to optimize aligning short reads to the sequence.

- Aligning Short Reads with UGENE Genome Aligner
- Building Index for UGENE Genome Aligner
- Converting UGENE Assembly Database to SAM Format

Aligning Short Reads with UGENE Genome Aligner

When you select the *Tools DNA Assembly Align short reads* item in the main menu, the *Align Short Reads* dialog appears. Set the *Align short reads method* parameter to *UGENE Genome Aligner*. The dialog looks as follows:



The following parameters are available:

Reference sequence — DNA sequence to align short reads to. This parameter is required.

Result file name — file in UGENE database format or SAM format (if the box *SAM output* check), to write the result of the alignment into. This parameter is required.

Prebuilt index — check this box to use an index file instead of a reference sequence. Also you can *build it manually*.

SAM output — checking this box allows one to save output files in the SAM format. The default format of output files is the UGENE database format (ugenedb).

Short reads — each added short read is a small DNA sequence file. At least one read should be added.

The *Aligning Short Reads with UGENE Genome Aligner* has no limitation on short reads length.

Common parameters:

Mismatches allowed — check this box to allow mismatches between the reference sequence and a short read. Select one of the following:

- *Mismatches number* to set the number of mismatched nucleotides allowed. This parameter can take values: 1, 2 and 3.
- *Percentage of mismatches* to set the number of mismatches in percents. Note, that in this case the absolute number of mismatches can vary for different reads. This parameter can take values: 1 - 10 %.

Align options:

- *Use GPU-optimization* — use an openCL-enabled GPU during the alignment (the corresponding hardware should be available on your computer).
- *Align reverse complement reads* — use both: a read and its reverse complement during the alignment.
- *Use "best"-mode during the alignment* — report only about best alignments (in terms of mismatches).
- *Omit reads with qualities lower than* — omit all reads with qualities lower than the specified value. Reads that have no qualities are not omited.

Advanced parameters:

Maximum memory for short reads — maximum memory usage for short reads. This parameter allows one to decrease the load on the computer on one side and to increase the computer speed of the task on the other side.

- *Total memory usage* — shows the total memory usage.
- *System memory size* — shows the total system memory size.

Index parameters:

Reference fragmentation — this parameter influences the number of parts the reference will be divided. It is better to make it bigger, but it influences the amount of memory used during the alignment.

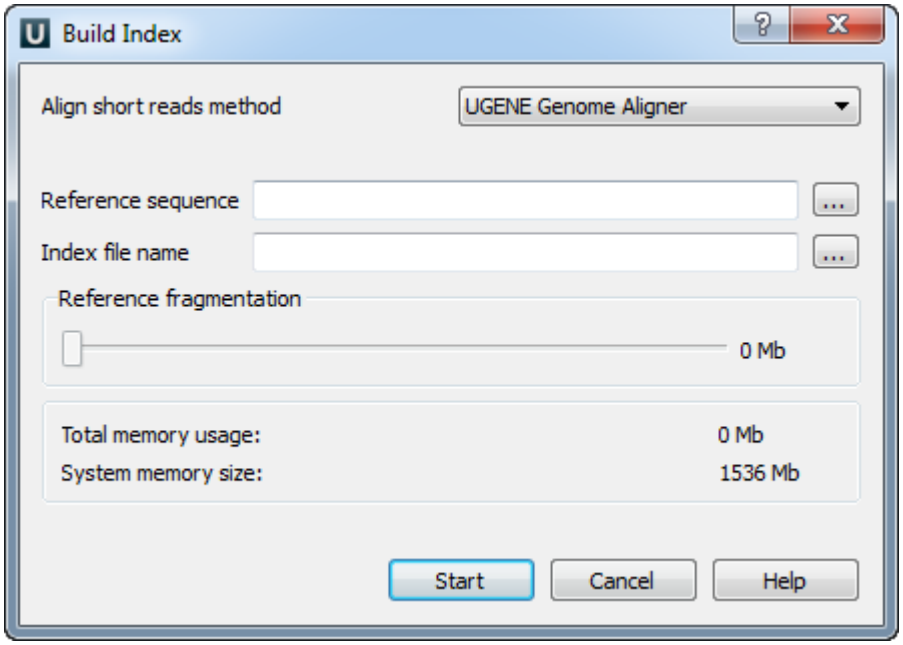
- *Index memory usage size* — shows the index memory usage.
- *Directory for index files* — temporary directory for saving index files.

You can choose a temporary directory for saving index files for the reference that will be built during the alignment. If you need to run this algorithm one more time with the same reference and with the same reference fragmentation parameter, you can use this prebuilt index that will be located in the temporary directory.

Building Index for UGENE Genome Aligner

You can build an index to optimize short reads alignment using *UGENE Genome Aligner*. To open the *Build Index* dialog, select the *Tools NGS data analysis Build index for reads mapping* item in the main menu. Set value of the *Align short reads method* parameter to *UGENE Genome Aligner*.

The dialog looks as follows:



The parameters are the following:

Reference sequence — DNA sequence to which short reads would be aligned to. This parameter is required.

Index file name — file to save index to. This parameter is required.

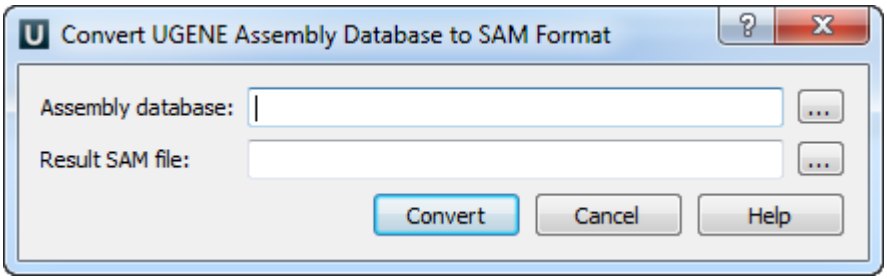
Reference fragmentation — this parameter influences the amount of parts the reference will be devided. It is better to make it bigger, but it influences the amount of memory used during the alignment.

Total memory usage — shows the total memory usage.

System memory size — shows the total system memory size.

Converting UGENE Assembly Database to SAM Format

To convert UGENE data base to SAM format click on the *Tools->DNA Assembly->Convert UGENE assembly database to SAM format* context main menu item. The following dialog will appear:



Select assembly and result files and click on the *Convert* button.

CAP3

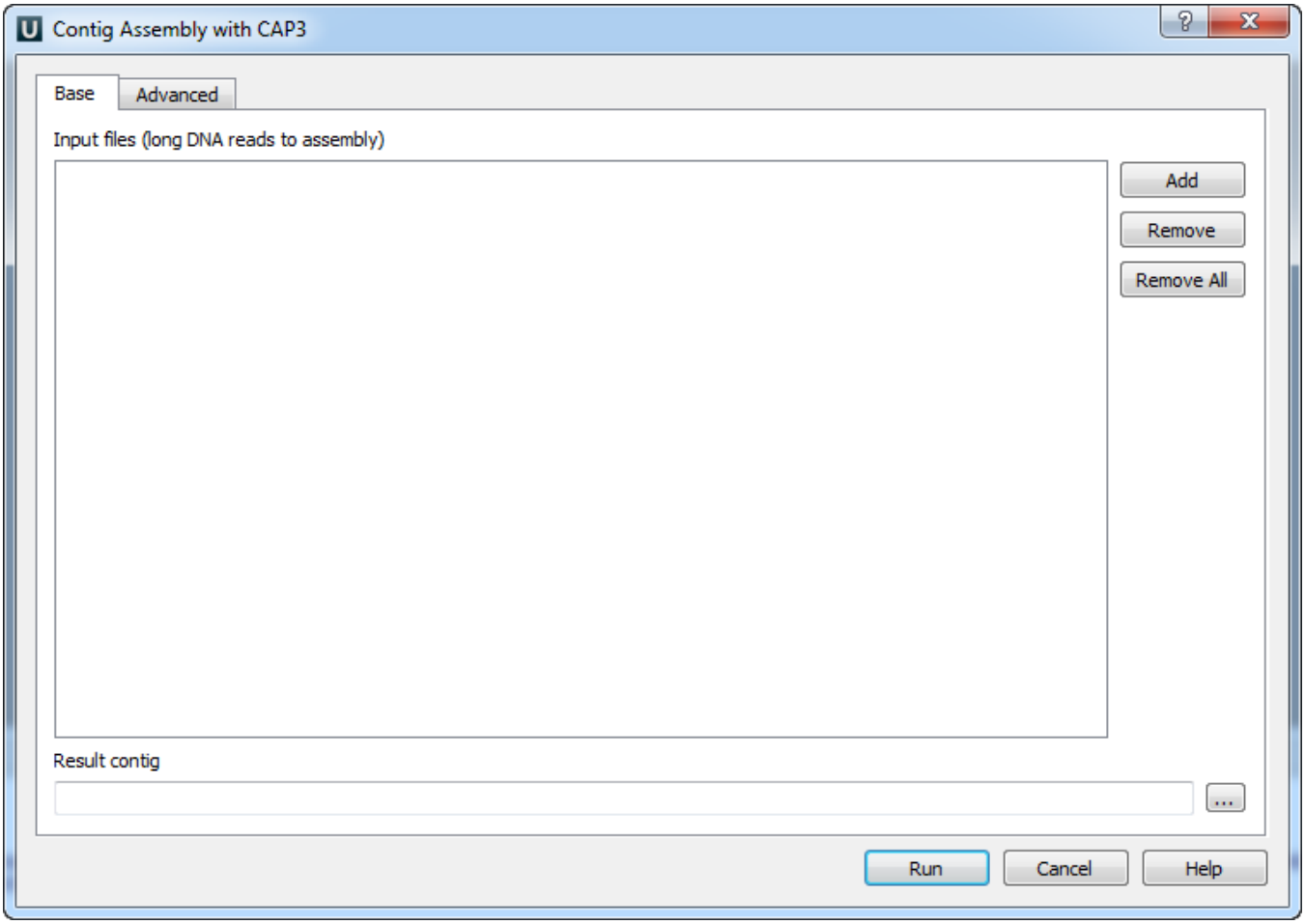
CAP3 (CONTIG ASSEMBLY PROGRAM Version 3) is a sequence assembly program for small-scale assembly with or without quality values. Click this link to open *CAP3* homepage. *CAP3* is embedded as an *external tool* into UGENE.

Open *Tools Sanger data analysis* submenu of the main menu.



Select the *Reads de novo assembly (with CAP3)* item to use the *CAP3*.

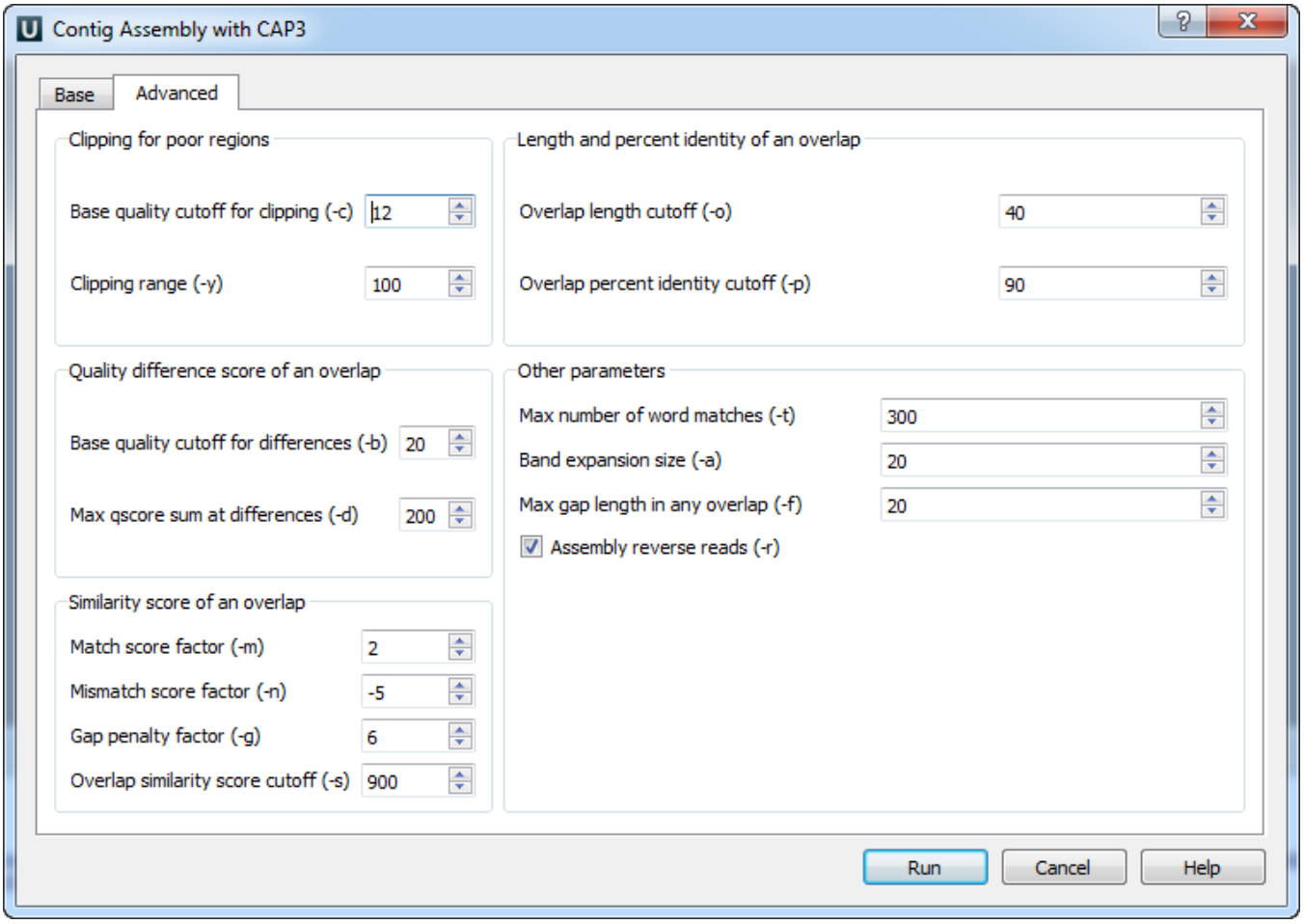
The *Contig Assembly With CAP3* dialog appears.



You can add or remove input files using *Add* and *Remove* buttons. To remove all files click the *Remove all* button. *Input files* are files with a long DNA reads in FASTA, FASTQ, SCF or ABI formats. At least one input file should be added. Input a *Result contig* name and press the *Run* button. *CAP3* produce s assembly results in the ACE file format (".ace"). The file contains one or several contigs assembled from the input reads.

The quality scores for FASTA sequences can be provided in an additional file. The file must be located in the same folder as the original sequences and have the same name as FASTA file, but another extension: ***.qual***.

Also you can change the following advanced parameters:



Clipping for poor regions parameters:

Clipping of a poor end region of a read is controlled by parameters *Base quality cutoff for clipping (-c)* (the specified value should be more than 5), and *Clipping range (-y)* (the specified value should be more than 5).

Quality difference score of an overlap parameters:

Base quality cutoff for differences (-b) — if an overlap contains a difference at bases of quality values q1 and q2, then the score at the difference is max(0, min(q1, q2) - b), where b is the specified value. The specified value should be more than 15. The difference score of an overlap is the sum of scores at each difference.

Max qscore sum at differences (-d) — remove an overlap if its difference score is greater than the specified value. The specified value should be more than 20.

Similarity score of an overlap parameters:

The following parameters are used to calculate the similarity score of an overlapping alignment:

Match score factor (-m) — a match at bases of quality values q1 and q2 is given a score of m * min(q1, q2), where m is the specified value. The specified value should be more than 0.

Mismatch score factor (-n) — a mismatch at bases of quality values q1 and q2 is given a score of n * min(q1, q2), where n is the specified value. The specified value should be less than 0.

Gap penalty factor (-g) — a base of quality value q1 in a gap is given a score -g * min(q1, q2), where g is the specified value; q2 is the quality value of the base in the other sequence right before the gap. The specified value should be more than 0.

The similarity score is caclulated as the sum of scores of each match, each mismatch and each gap. Based on this value and the following value some overlaps are removed:

Overlap similarity score cutoff (-s) — remove overlaps with similarity scores less than the specified value. The specified value should be more than 250.

Length and percent identity of an overlap parameters:

Overlap length cutoff (-o) — minimum length of an overlap (in base pairs). The specified value should be more than 15 base pairs.

Overlap percent identity cutoff (-p) — minimum percent identity of an overlap. The specified value should be more than 65%.

Other parameters:

Maximum number of word matches (-t) — an upper limit of word matches between a read and other reads. Increasing the value would result in more accuracy, however this could slow down the program. The specified value should be more than 0.

Band expansion size (-a) — a number of bases to expand a band of diagonals for an overlapping alignment between two sequence reads. The specified value should be more than 10.

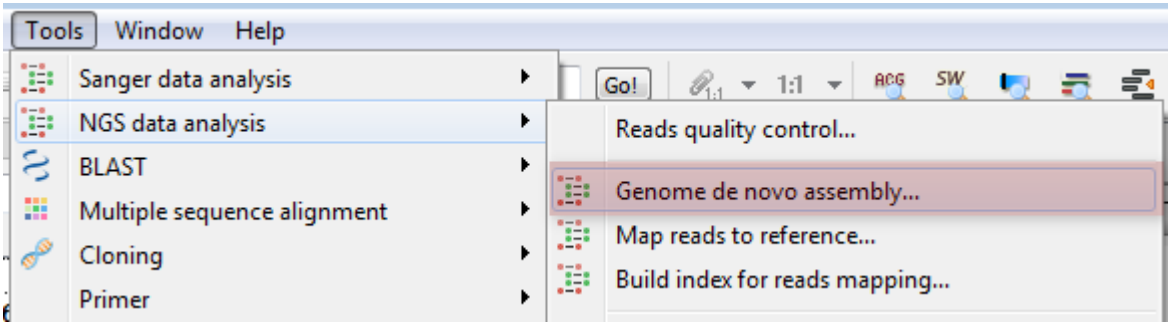
Max gap length in any overlap (-f) — reject overlaps with a gap longer than the specified value. A small value may cause the program to remove true overlaps and to produce incorrect results. This option may be used by the user to split reads from alternative splicing forms into separate contigs. The specified value should be more than 1.

Assembly reverse reads (-r) — consider reads in reverse orientation for assembly. The default value is “checked”.

SPAdes

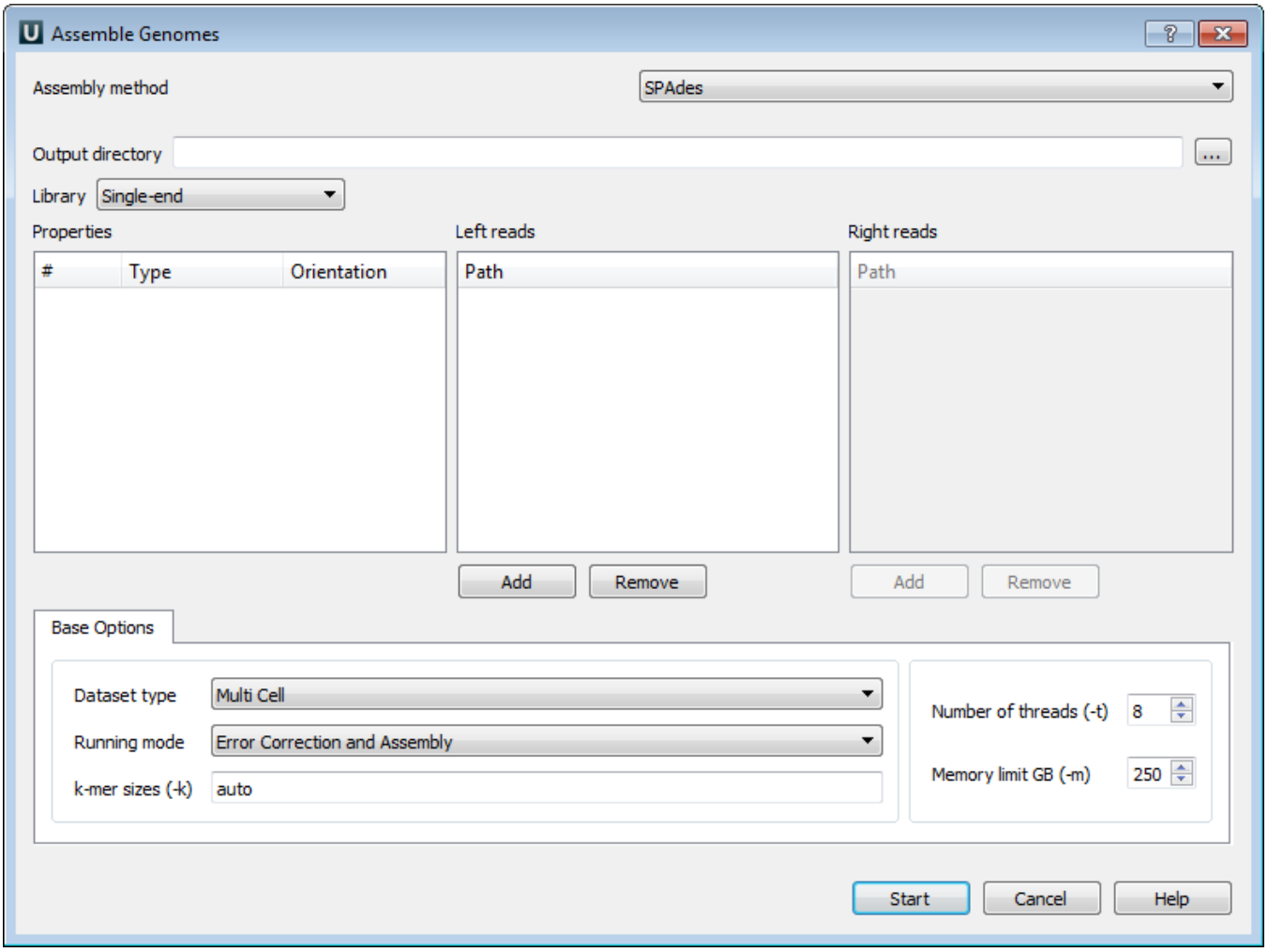
SPAdes – St. Petersburg genome assembler. Click this link to open SPAdes homepage. SPAdes is embedded as an *external tool* into UGENE.

Open *Tools NGS data analysis*.



Select the *Genome de novo assembly* item to use the *SPAdes*.

The *Assemble Genomes* dialog will appear.



The following parameters are available:

Output directory - SPAdes stores all output files in output directory, which is set by the user.

Library - to run SPAdes choose one of the following libraries:

- Single-end
- Paired-end
- Paired-end (Interplaced)
- Paired-end (Unpaired files)
- Sanger
- PacBio

Left reads - file(s) with left reads.

Right reads - file(s) with right reads.

For each dataset in the paired-end libraries you can change type and orientation.

Datasest type - dataset type.

Running mode - running mode.

k-mer sizes (-k) - k-mer sizes.

Number of threads (-t) - number of threads.

Memory limit GB (-m) - memory limit.

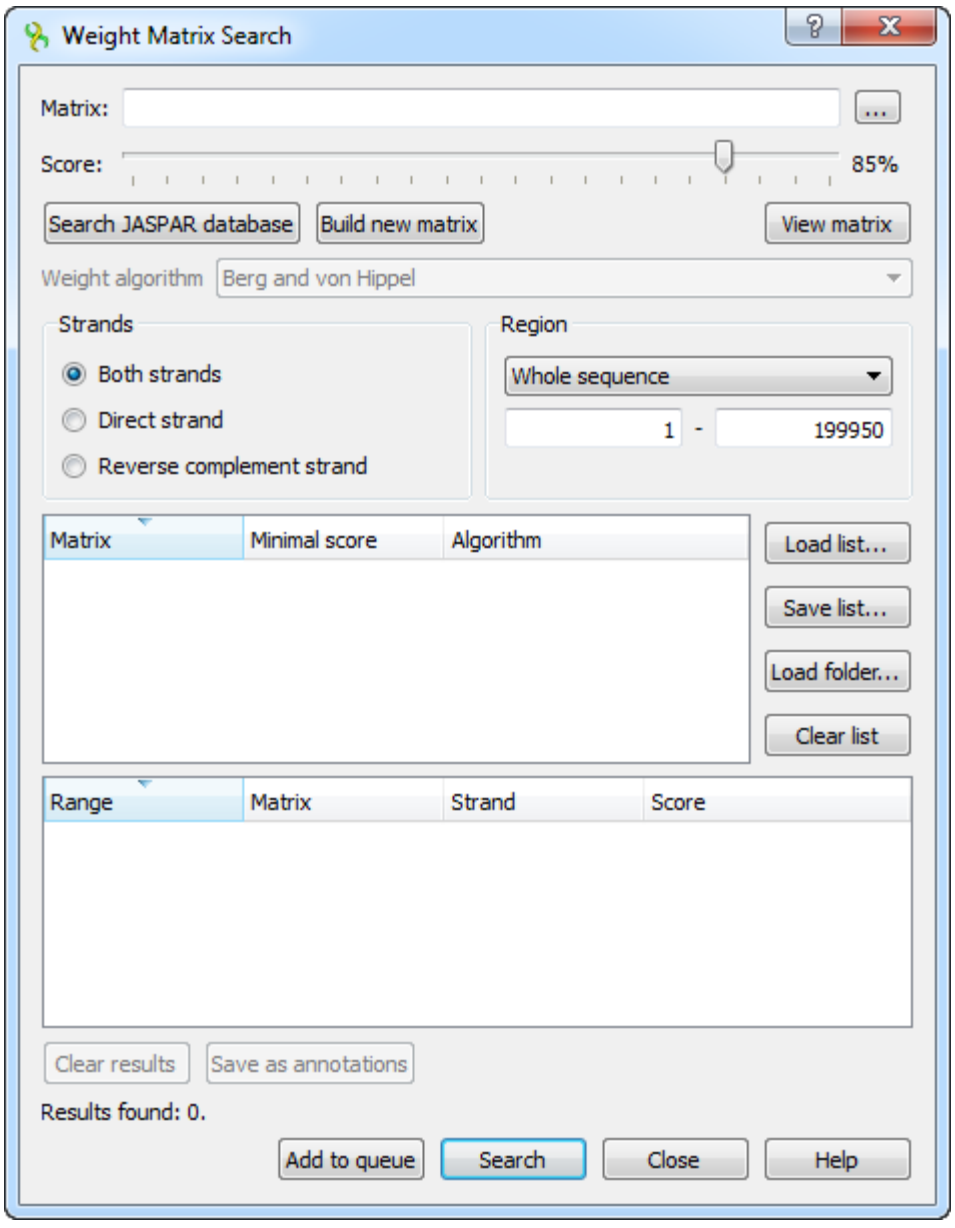
Weight Matrix

The *Weight Matrix* plugin is a tool for solving the problem of a sequence annotating. As well as for the *SITECON*, the main use case of the plugin is recognition of potential transcription factor binding sites on basis of the data about conservative conformational and physicochemical properties revealed with the binding sites sets analysis.

The *Weight Matrix* contains a lot of *position frequency matrices* (PFM 's) and *position weight matrices* (PWM 's, also known as *position specific score matrices* — PSSM 's). The matrices came from two wide-known open archives: JASPAR, which contains frequency matrices, and UniPROBE containing weight matrices.

Also the *Weight Matrix* plugin provides a tool for creating specific position frequency and weight matrices from an existing alignment or from a file with several sequences. The created matrix can be used as a profile for the search as well as the JASPAR and UNIPROBE ones.

To search for transcription factor binding sites in a DNA sequence select the *Analyze Search TFBS with matrices* context menu item. The *Weight matrix search* dialog will appear:



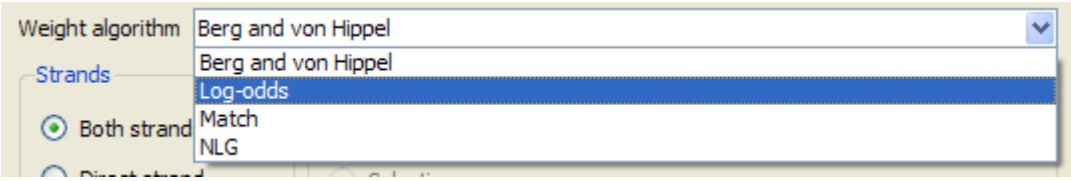
In the search dialog you must specify a file with PWM or PFM. You can do so by pressing the browse button and selecting the file.

Also you can use the *special interface to choose a JASPAR matrix* by pressing the *Search JASPAR database* button.

Alternative way to specify the position weight/frequency matrix is to create a specific one from an alignment or a file with several sequences with the *build a new matrix* tool.

After the profile (the matrix) is loaded, you can adjust the threshold value. The threshold sets the minimal identity score for a result to pass. The more the result score is, the more it is homologically related to the aligned region. By changing the threshold you can filter low- scoring results.

If the loaded matrix is a position frequency matrix, you must also specify the algorithm to build the corresponding position weight matrix which will represent the transcription factor. There are four algorithms available.

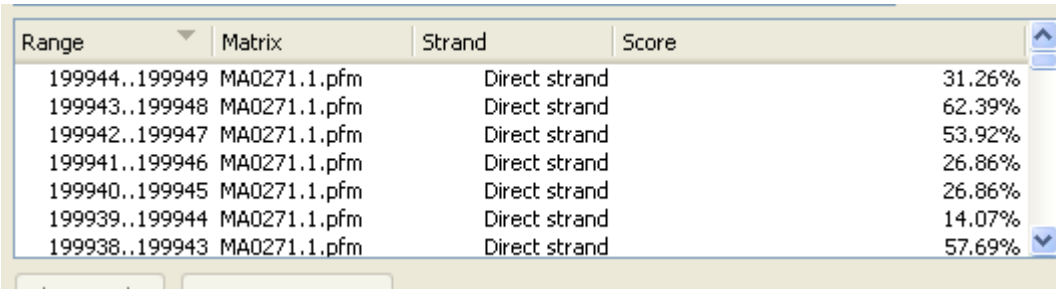


Also you can add a selected matrix with the specified *Minimal score* and the *Algorithm* to the matrices list. To do it, select the matrix and other options and press the *Add to queue* button. The plugin will search with all matrices specified in the list.

You can use the *Save list* button to export the list of matrices to a *.csv file. Later the list can be loaded from the file using the *Load list* button.

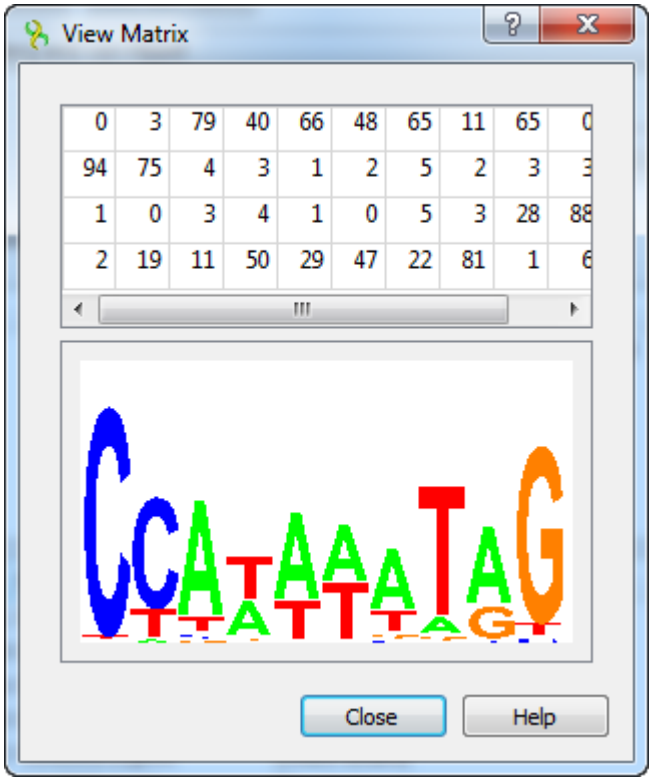
The rest options are standard sequence search options: the strand and the sequence region where to search for matches.

After specifying the necessary options press the *Search* button. The found results will appear in the dialog table. The corresponding results identity scores are in the *Score* column.



Range	Matrix	Strand	Score
199944..199949	MA0271.1.pfm	Direct strand	31.26%
199943..199948	MA0271.1.pfm	Direct strand	62.39%
199942..199947	MA0271.1.pfm	Direct strand	53.92%
199941..199946	MA0271.1.pfm	Direct strand	26.86%
199940..199945	MA0271.1.pfm	Direct strand	26.86%
199939..199944	MA0271.1.pfm	Direct strand	14.07%
199938..199943	MA0271.1.pfm	Direct strand	57.69%

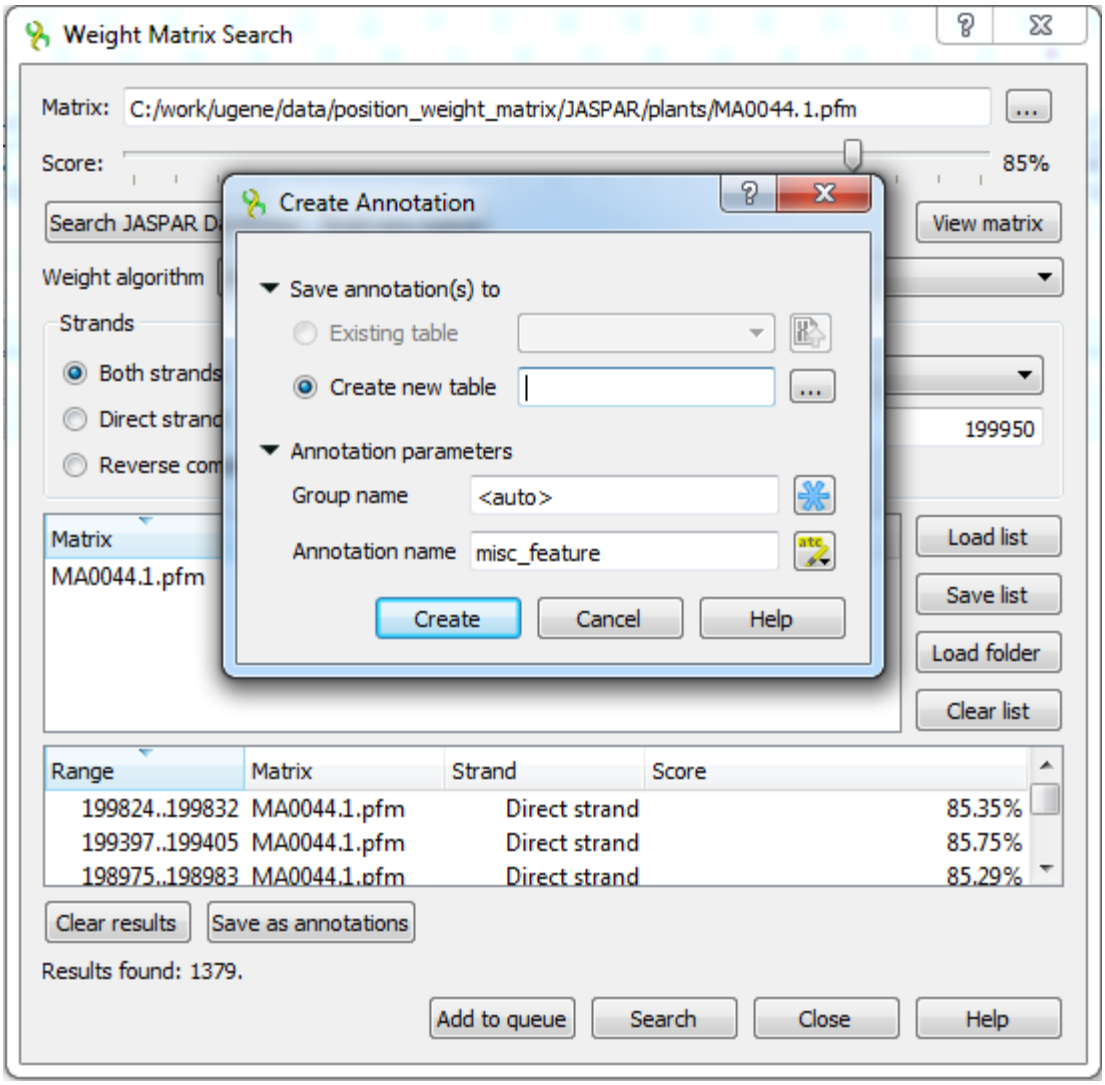
Also you can see the matrix by using the *View matrix* button:



The regions found by the weight matrix algorithm can be saved as annotations to the DNA sequence in the Genbank format by pressing the *Save as annotations* button.

After saving, the file with resulting annotations will be automatically added to the current project, and the annotations will be added to the original sequence.

Note that in case of selecting JASPAR or UNIPROBE matrix, the resulting annotations will contain the given matrix properties.

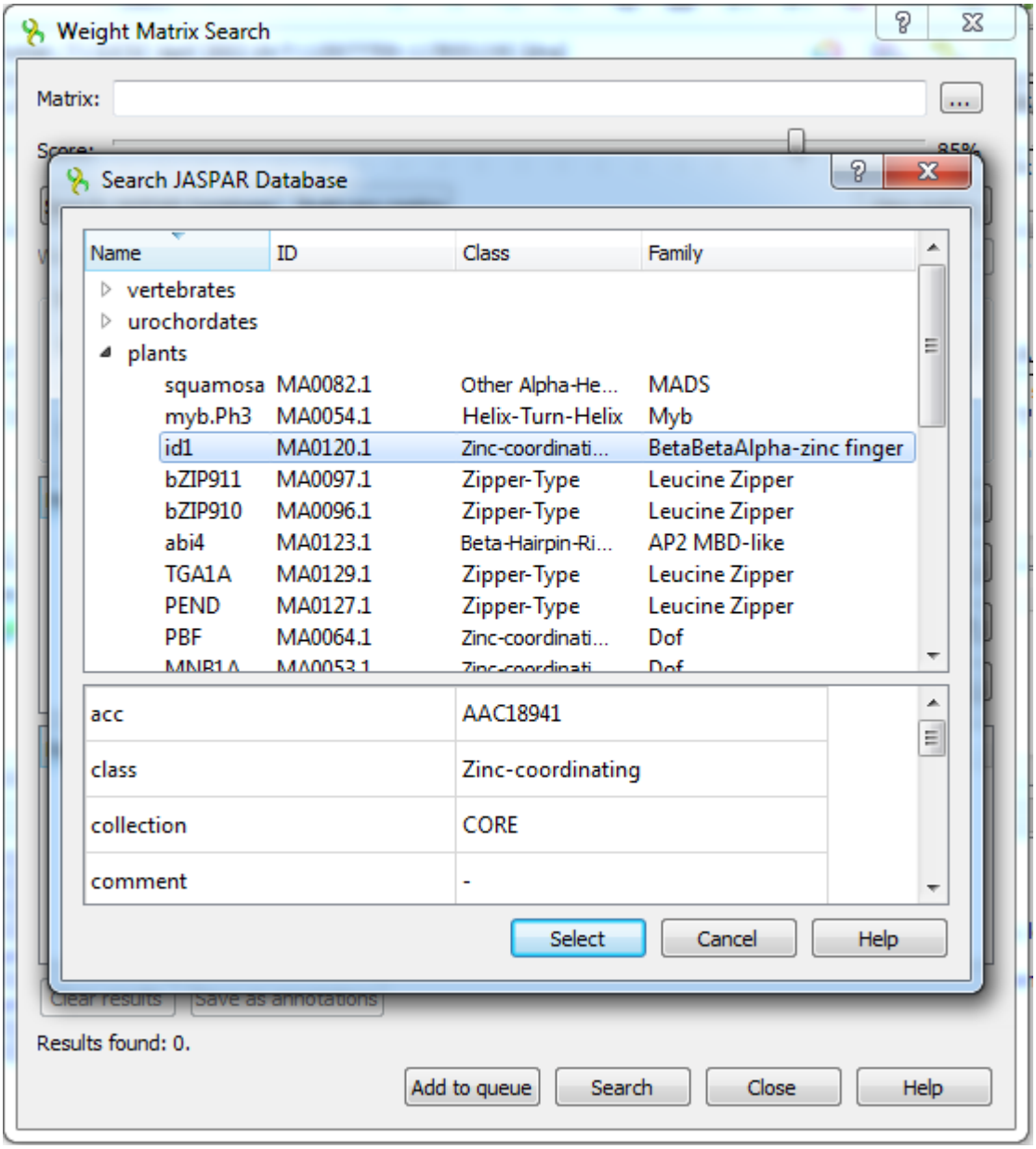


See also:

- Searching JASPAR Database
- Building New Matrix

Searching JASPAR Database

Press the *Search JASPAR database* button in the *Weight matrix search* dialog. The following dialog will appear:

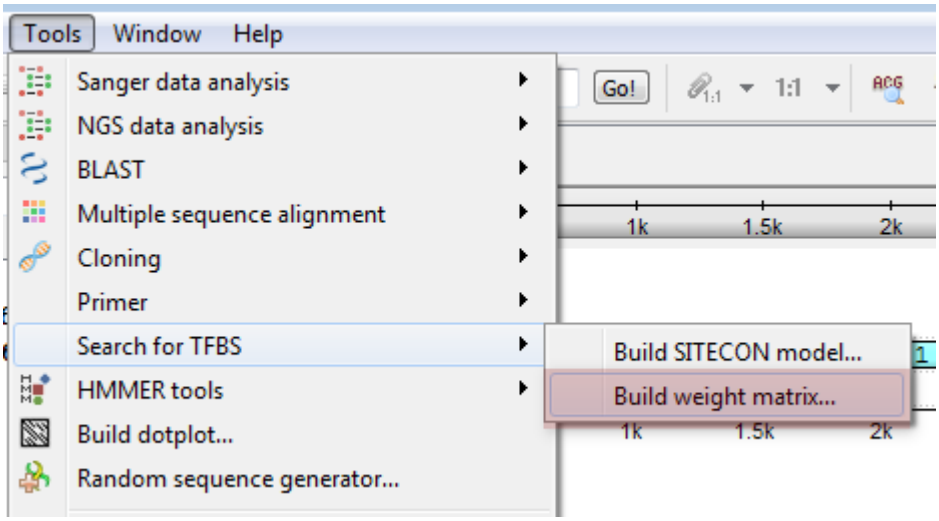


Here the matrices are divided into categories and you can read detailed information of a matrix which is represented by its properties. It could help you to choose the matrix properly.

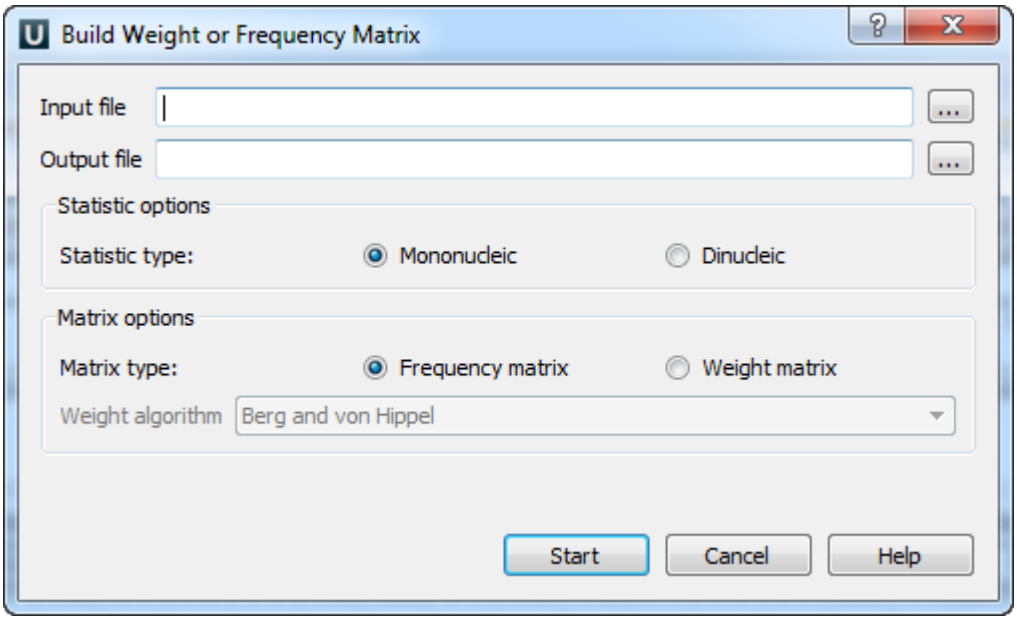
The matrices provided with UGENE are located in the $UGENE/data/position_weight_matrix folder.

Building New Matrix

To create a position weight or frequency matrix from an alignment or a file with several sequences, press the *Build new matrix* button in the *Weight matrix search* dialog, or select the *Tools Search for TFBS Build weight matrix* main menu item:



The *Build weight or frequency matrix* dialog will appear:



The following parameters are available:

Input file — an alignment or a file with several sequences to build the matrix from. The parameter is mandatory.

Output file — the resulting matrix will be saved in this file. The parameter is mandatory.

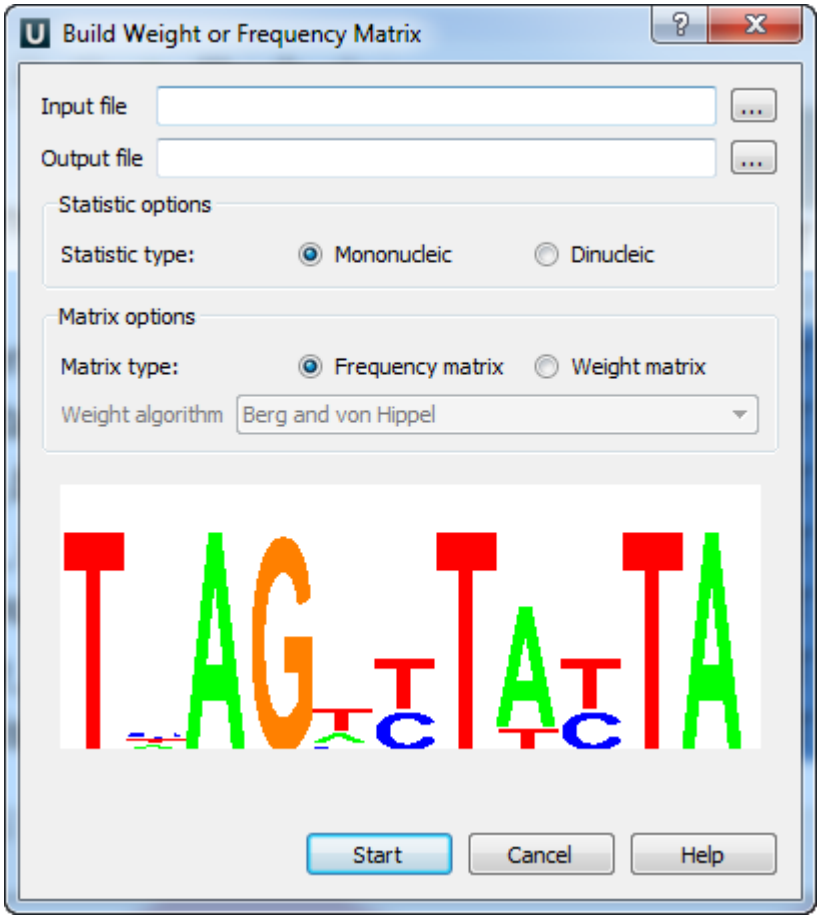
Statistic type — defines the way in which the statistics will be collected. The *Mononucleic* option is basically good for small alignments, and the *Dinu cleic* option must give more appropriate results for big alignments.

Matrix type — defines the type of the resulting matrix.

If the *Frequency matrix* option is selected then the frequency matrix will be created and saved into the resulting file.

If the *Weight matrix* option is selected then the intermediate frequency matrix will be created and then transformed into a weight matrix on basis of the selected *Weight algorithm*. Then the weight matrix will be saved into the resulting file.

For some input files the colored “Alignment Logo” appears at the bottom of the dialog. It gives the representation of the selected alignment.



The “Alignment logo” appears when:

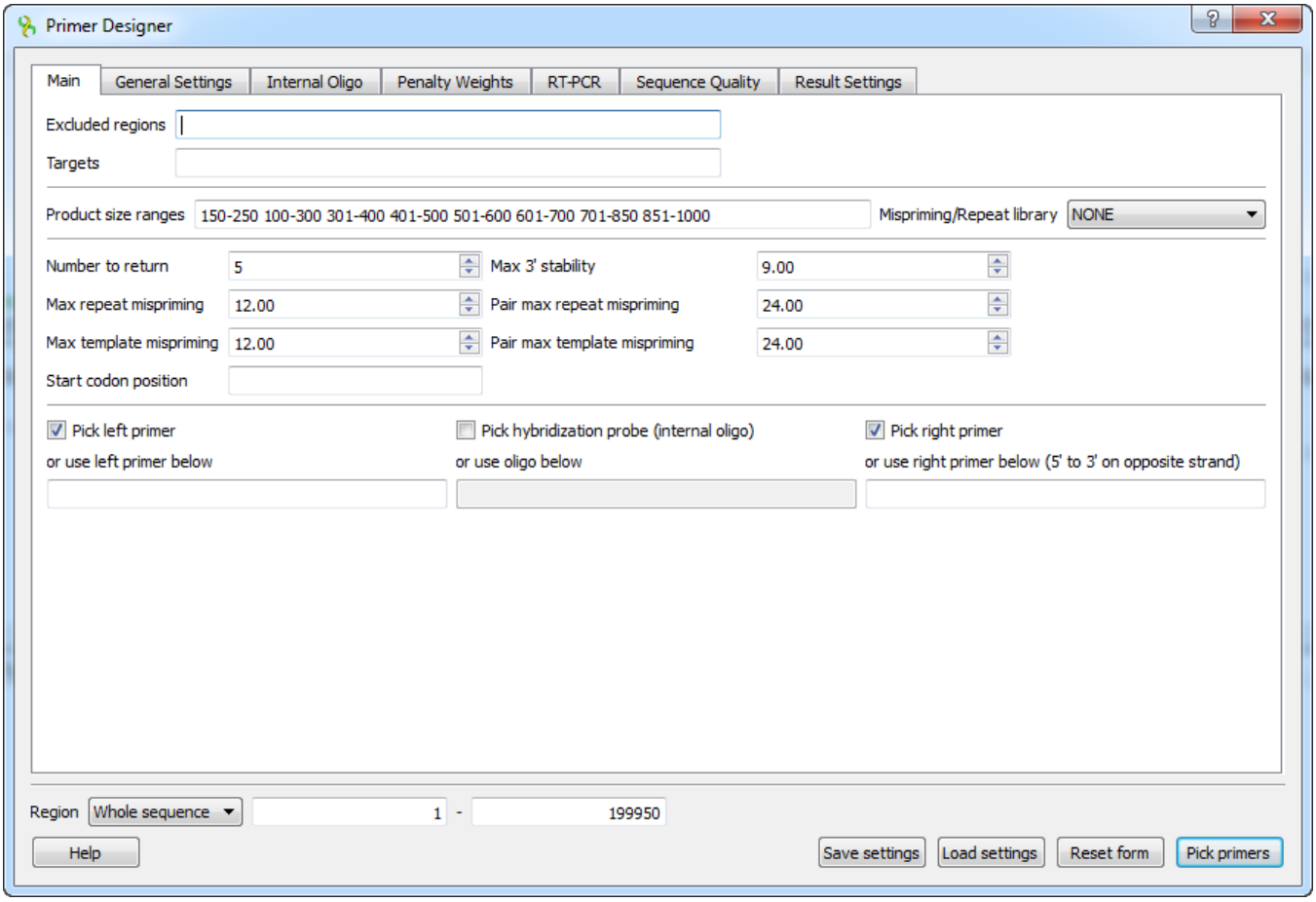
- The input file format is *.pfm, *.aln or it is a file with several sequences;
- The size of the input file is small enough.

To start the operation, press the *Start* button. The matrix will be created and saved. If the *Build weight or frequency matrix* dialog was invoked from the *Weight matrix search* dialog, then the matrix also will be chosen as the current profile.

Primer3

The *Primer3* plugin is a port of the Primer3 tool. It is intended to pick primers from a DNA sequence.

To use the *Primer3*, open a DNA sequence and select the *Analyze Primer3* context menu item. The dialog will appear:



All available parameters are the same as in the original Primer3.

However there is one additional feature available which is not originally a part of Primer3 tool. It allows user design primers for RT-PCR experiments by choosing which exons/introns to span with the primer product. This feature is described in detailed below. When you select the parameters you can save and load settings with a help of the corresponding buttons in the right corner of the dialog.

- RTPCR Primer Design

RTPCR Primer Design

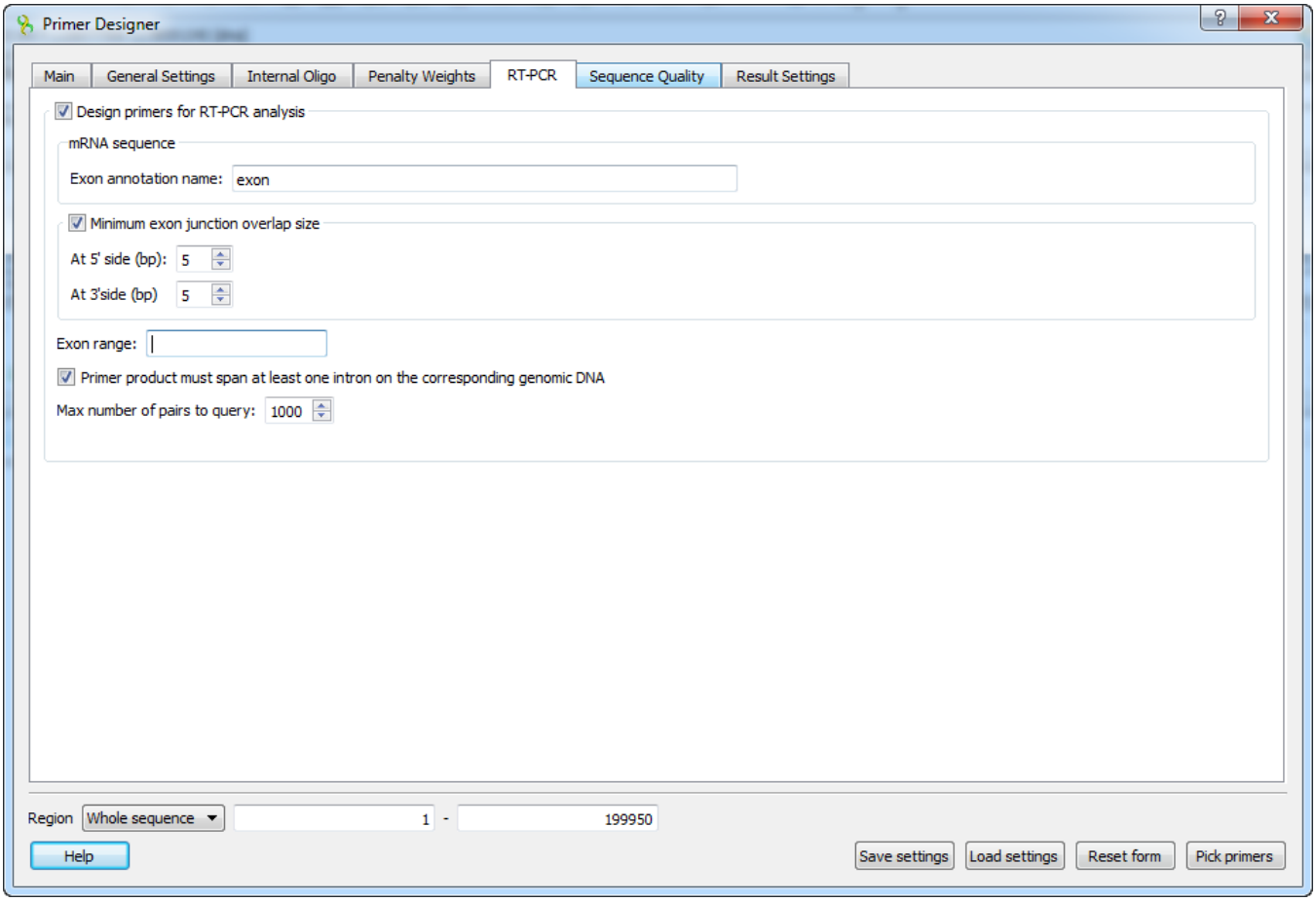
This feature allows to search for primer pairs that span introns on the genomic sequence or exon junctions on the mRNA sequence.

Note that RT-PCR design is only available for mRNA/cDNA sequences with annotated exons. There are several ways to obtain the cDNA for a corresponding DNA sequence.

- From NCBI or ENSEMBL database.
  For example, one can download the ***TMPRSS2 transcript variant 1*** from NCBI Genbank using identifier NM_001135099.1.
  This can be also done from UGENE using option Access remote database or Search NCBI Genbank.

- Align the genomic and cDNA sequences using spliced aligner.
  For this option one must have both genomic and cDNA sequences.
  In UGENE the spliced alignment can be performed using the Spidey tool.
  To run the alignment open the genomic sequence and select action *Align Align to mRNA sequence*.
  The generated exon annotations can be then exported using action *Export Export sequence of selected annotations*

To design primers for your mRNA sequence and go to the RT-PCR tab of the *Primer Designer* dilaog:



The following parameters are available:

Exon annotation name

To detect exon boundaries UGENE searches for exonic annotations. This option allows to set custom name for annotations denoting exons. Default value is "exon"

Minimum exon junction overlap size

If checked, then only the pairs with at least one of the primers overlapping exon junction in the mRNA sequence will be selected.

At 5' side (bp)

Minimum overlap size on the 5' side of the exon junction. Default is 5 bp.

At 3' side (bp)

Minimum overlap size on the 3' side of the exon junction. Default is 5 bp.

Exon range

This option allows to limit the sequence region, where the primers are searched for. For example, setting value "3-5" will limit the search to a sequence region consisting of exons 3,4,5 of the transcript, as defined by the order in the sequence. Default value is an empty string, which means that there are no limitations.

Span at least one intron

This option makes sure that primer product should span an intron on the genomic sequence i.e the forward and reverse primers must be located in different exons. The option is enabled by default.

Max numbers of pairs to query

The algorithm applied in RT-PCR primer design first searches for all available primers in a given sequence. Then it filters the detected pairs to make sure that they satisfy the selected configuration. This option allows to set the maximum number of pairs for the initial search query. Larger number will result in increased sensitivity, but also in a longer running time. Default value is 1000**.**

Important: using the **RT-PCR** primer design tab will reset the values set in the *Exlcuded regions* and *Targets* of the **Main** configuration tab. Additionally if the *Exon range* option is set, the defined sequence region will be ignored.

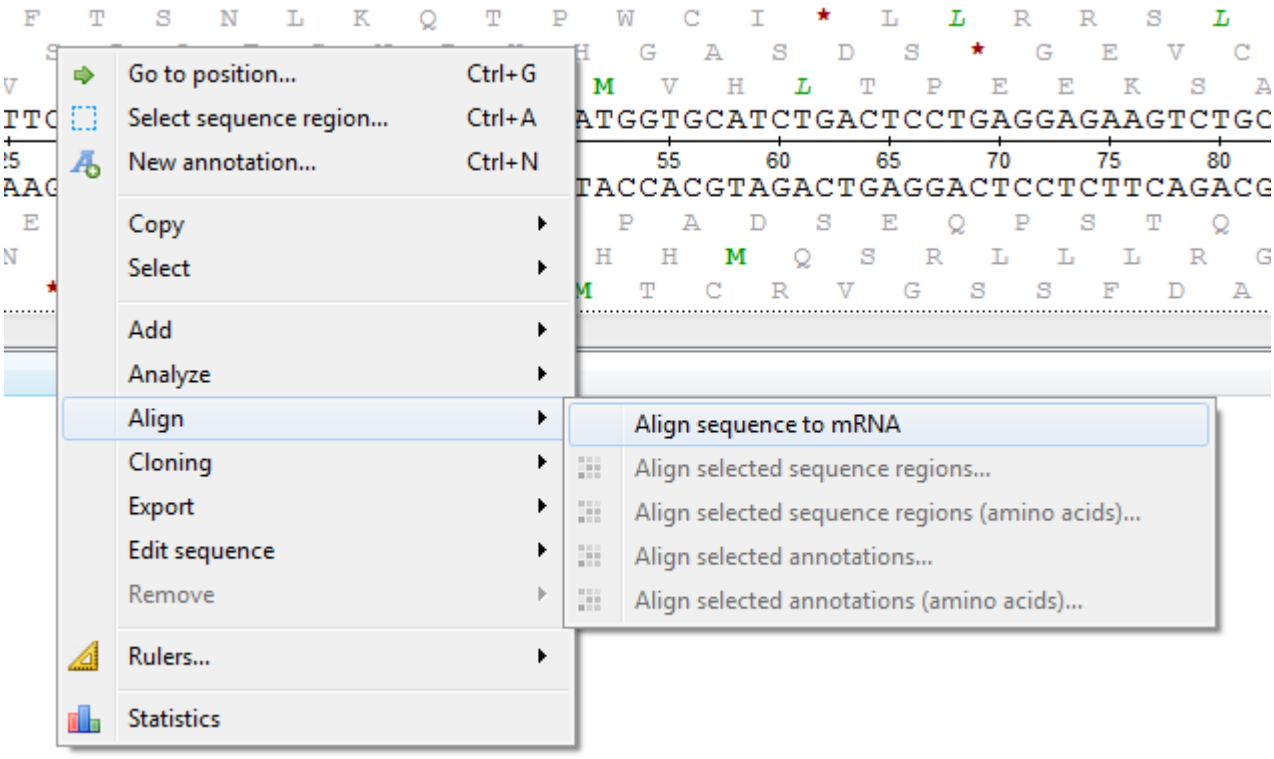
Spliced Alignment mRNA and cDNA

UGENE allows to align spliced mRNA/cDNA sequence to genomic sequences.

The default underlying algorithm which is used for the alignment is an external tool called Spidey.

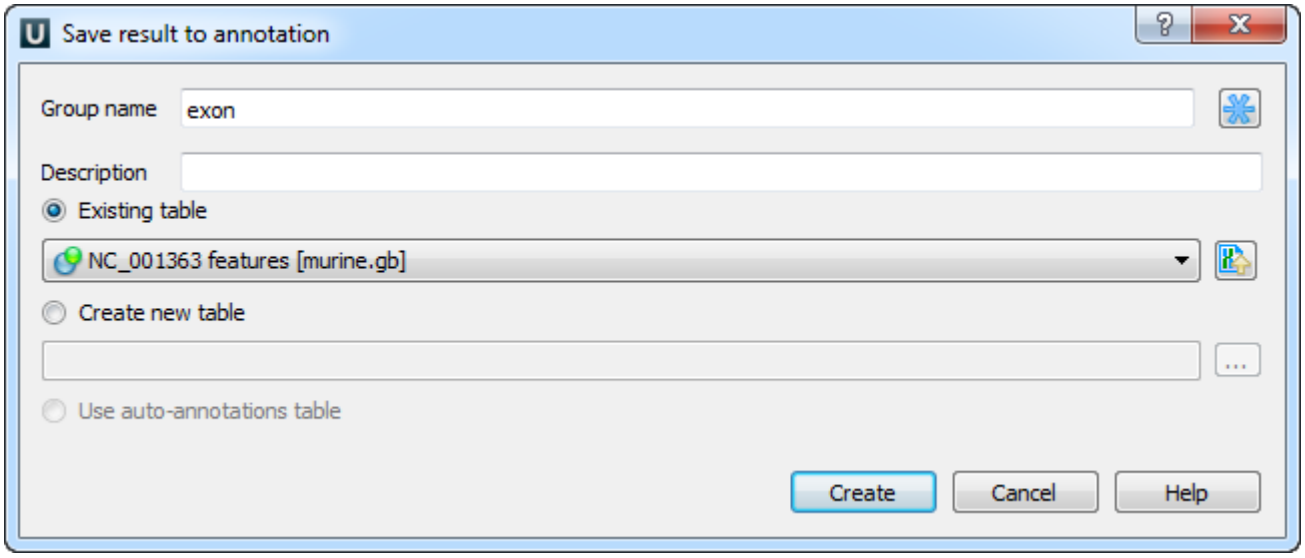
Before running the alignment make sure that Spidey is available and validated in the list of *External Tools*.

To perform the alignment of a mRNA sequence to a genomic sequence open the the genomic sequence in the *Sequence View*. Next activate context menu item *Align -> Align to sequence to mRNA*.



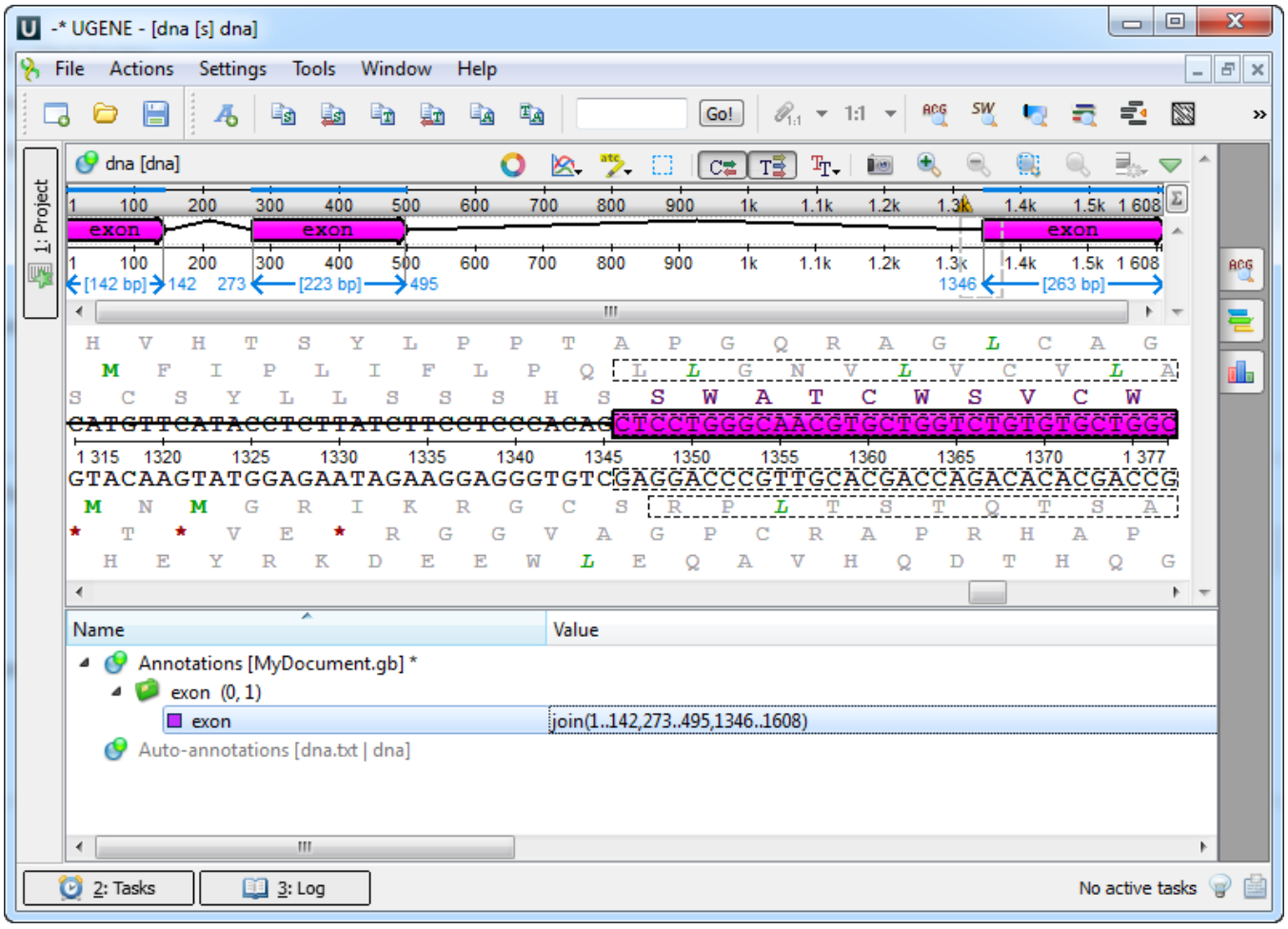
In the list of sequences select the corresponding mRNA sequence and click OK.

The following dialog will appear:



Here you can set up a file to store annotations. It could be either an existing annotation table object or a new annotation table or auto-annotations table (if it is possible). Also you can modify the group name parameter and add a description.

The resulting alignment will be saved as an annotation with the corresponding name:



External Tools Plugin

The *External Tools* plugin allows one to launch an external tool from UGENE.

To use an external tool from UGENE, the tool needs to be installed on the system and the path to it should be properly configured. However, there is no need in the additional configuration, if you've installed the UGENE Full Package, as it already contains all the tools by default.

Otherwise, if you've installed the UGENE Standard Package, you would need to configure an external tool in order to use it. Note that in this case you can download the package with all the external tools from this page.

To learn how to configure an external tool, read below.

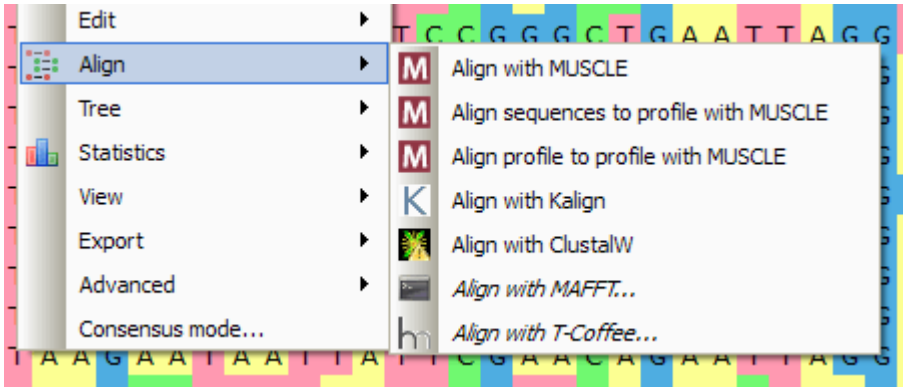
- Configuring External Tool

Configuring External Tool

To configure an external tool:

1. Make sure the tool is installed on your system.
2. Set a path to the tool executable file in UGENE. It can be set on the *External Tools* tab of the *Application Settings* dialog.

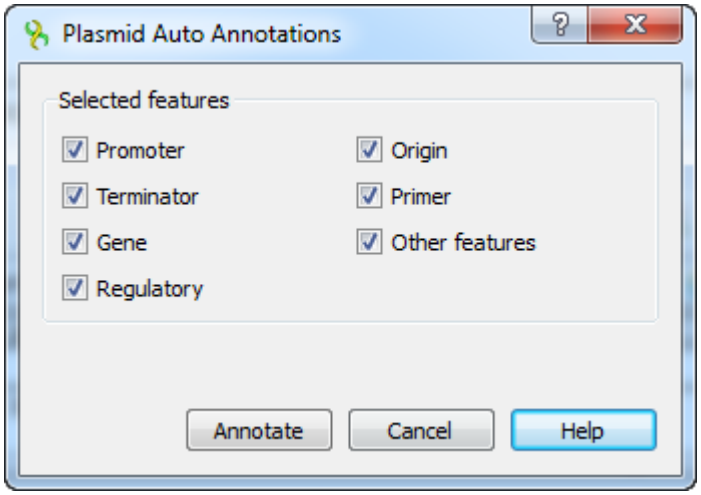
If the path hasn't been set for a tool, UGENE menu items that launch the tool are displayed in italic. For example, on the image below a path for the ClustalW external tool has been set, and paths for MAFFT and T-Coffee has not:



Plasmid Auto Annotation

Plasmid Auto Annotation feature allows to automatically annotate possible functional elements of the given sequence such as promoters, terminators, origin of replication, known genes, common primers and other features. Conceptually this functionality is similar to the one offered by PlasMapper software. The database for plasmid auto-annotation is based on the following resource.

To activate Plasmid Auto Annotation upon your sequence use the menu item *Analyze Annotate plasmid and custom features.* In the appeared dialog one can selected the features to search in sequence.



The detected plasmid features are stored as automatic annotations and can be controlled through corresponding menu. Refer *Automatic Annotations Highlighting* to learn more.

The database containing features and their sequences is located in a subfolder of UGENE data folder: ***data/custom_annotations/plasmid_features.txt***.

ClustalO

Clustal is a widely used multiple sequence alignment program. It is used for both nucleotide and protein sequences.Clustal Omega is the latest addition to the Clustal family. It offers a significant increase in scalability over previous versions, allowing hundreds of thousands of sequences to be aligned in only a few hours. It will also make use of multiple processors, where present.

Clustal home page: http://www.clustal.org

If you are using Windows OS, there are no additional configuration steps required, as *ClustalO* executable file is included to the UGENE distribution package. Otherwise:

- Install the *Clustal* program on your system.
- Set the path to the *ClustalW* executable on the *External tools* tab of UGENE *Application Settings* dialog.

Now you are able to use *ClustaOl* from UGENE.

Open a multiple sequence alignment file and select the *Align with ClustalO* item in the context menu or in the *Actions* main menu. The *Align with ClustalO* dialog will appear (see below), where you can adjust the following parameters:

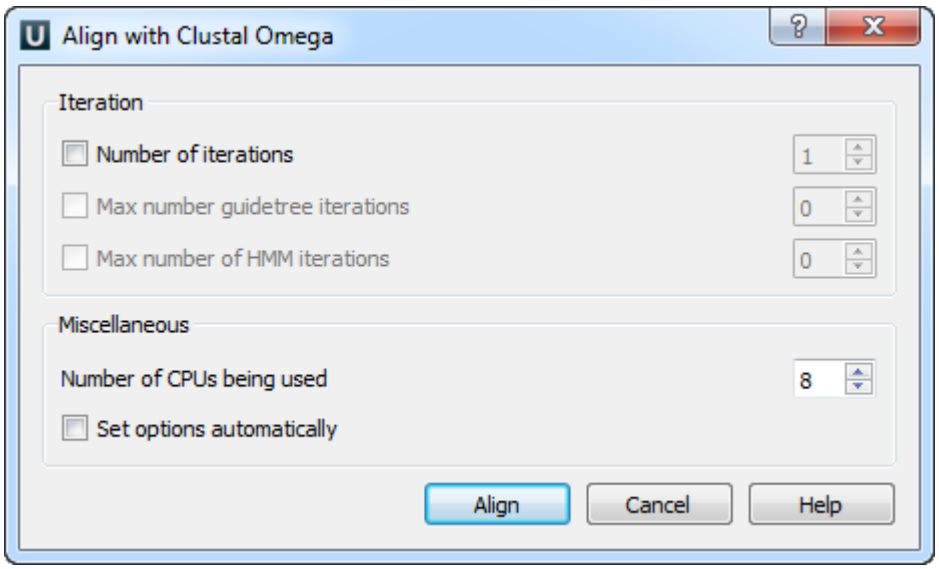
Number of iterations — number of (combined guide tree/HMM) iterations.

Max number guidetree iterations — maximum guide tree iterations.

Max number of HMM iterations —maximum number of HMM iterations.

Number of CPUs being used - number of processors to use.

Set options automatically - set options automatically (might overwrite some of your options).

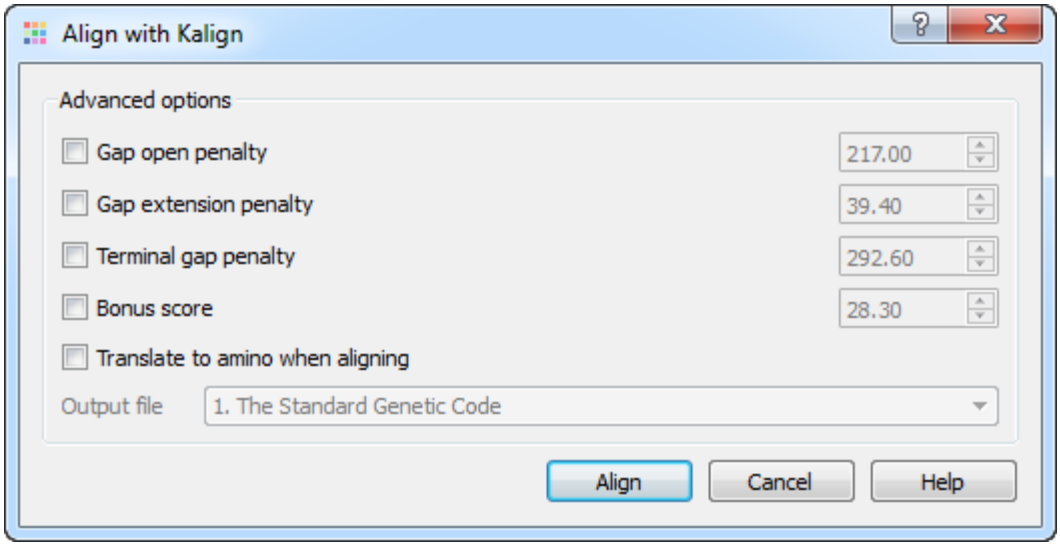


Kalign Aligning

Kalign is a fast and accurate multiple sequence package designed to align large numbers of protein sequences.

Kalign home page: KAlign

To use *Kalign* open a multiple sequence alignment file and select the *Align with Kalign* item in the context menu or in the *Actions* main menu. The following dialog appears:



The following parameters are available:

Gap opening penalty — indicates the penalty applied for opening a gap. The penalty must be negative.

Gap extension penalty — indicates the penalty applied for extending a gap.

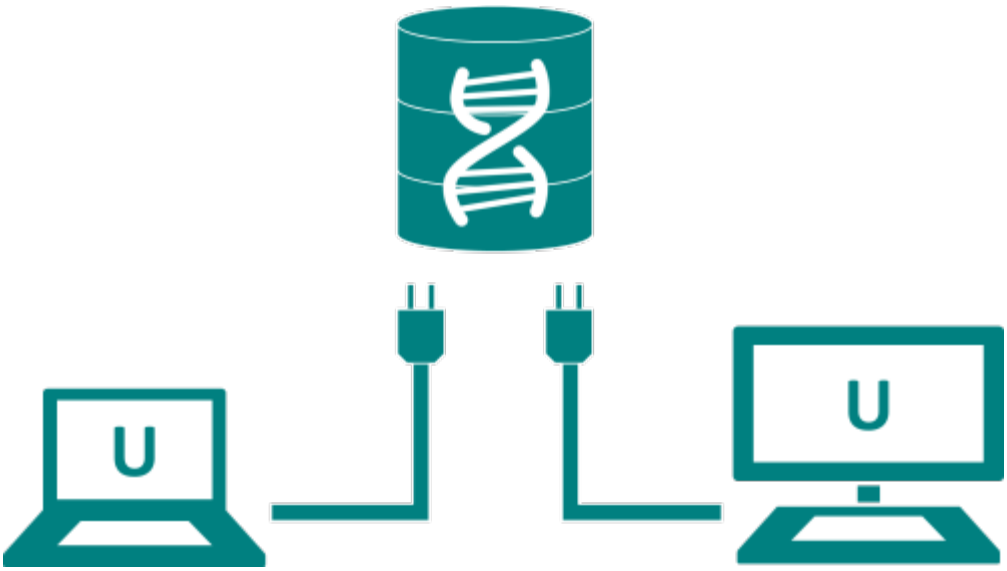
Terminal gap penalty — the penalty to extend gaps from the N/C terminal of protein or 5'/3' terminal of nucleotide sequences.

Bonus score - a bonus score that is added to each pair of aligned residues.

Translate to amino when aligning - translates an alignment to amino when aligning.

Shared Database

The rational storage of biological data is an ever-present issue. It is not only about large data sizes, but also about the requirement of simultaneous access to them by several scientists. For instance, a few researchers from a lab may need to work on the same data, like a set of primers or data produced by sequencing. That information has to be updated and synchronized between different users and kept in a common storage. That is what UGENE Shared Database is intended for.



To start sharing data via UGENE you need to deploy a public database server. MySQL servers are currently supported. See this paragraph for details about the required server configuration.

After that any UGENE user (who knows the correct login/password, however) can connect to the database. The connected database is shown in the Project View as a document exactly the same way as if the data were located on the local computer.

As described in this paragraph the users can have a read-only access to the database or be able to modify its content. A user with a read-only access can:

- Browse the data in the database
- Open the data in the UGENE views
- Export the data to the local computer

Users with write access, in addition, can:

- Add new objects to the database
- Create new folders to order the data in the database
- Modify the folders hierarchy inside the database (using drag'n'drop)
- Rename objects and folders
- Delete existed objects
- Delete folders

All UGENE instances connected to a database constantly monitors the state of the database and shows changes, made by other users.

UGENE accesses large remote data, such as NGS assemblies, so that only a viewed part of them is loaded to a client computer. So, if you store the assembly data on a server, the data can be browsed in the UGENE Assembly Browser on a local computer almost instantly, without the need to copy the data on the computer or use the hard disk space.

For details see the documentation below:

- Configuring Database
- Connecting to a Shared Database
- Adding Data to the Database
- Database in the Project
- Deleting Data
- Drag'n'drop in the Database
- Exporting Objects from the Database

Configuring Database

To make use of a shared database follow the steps below:

1. *Deploy a MySQL database server*

   We recommend you to download MySQL binaries from the official site. Note that UGENE supports MySQL versions from 5.5 to 5.7. Here you can also find instructions on how to install and launch a MySQL server instance for each platform.

2. *Create an empty database*

   Log in to the MySQL server as a user with administrative privileges (you must be able to create databases and users, and to grant privileges to the created users). In the MySQL console or in your favorite SQL browser execute the following command:
   > CREATE DATABASE `your_database_name`;

3. *Create database users*

   You may probably want to limit possible influence on the shared database by the UGENE users who will use it. In this case create a distinct MySQL user for each UGENE user (or a group of users). In order to do this, execute the following commands:
   > CREATE USER `user_nickname` IDENTIFIED BY 'user_password';
   Decide whether the created user is allowed to modify the database content or only to view it. In the first case execute the command below:
   > GRANT ALL PRIVILEGES ON your_database_name.* TO `user_nickname`@'%' IDENTIFIED BY 'user_password';
   and in the second case execute:
   > GRANT SELECT ON your_database_name.* TO `user_nickname`@'%' IDENTIFIED BY 'user_password';

4. *Use the database from a UGENE instance*

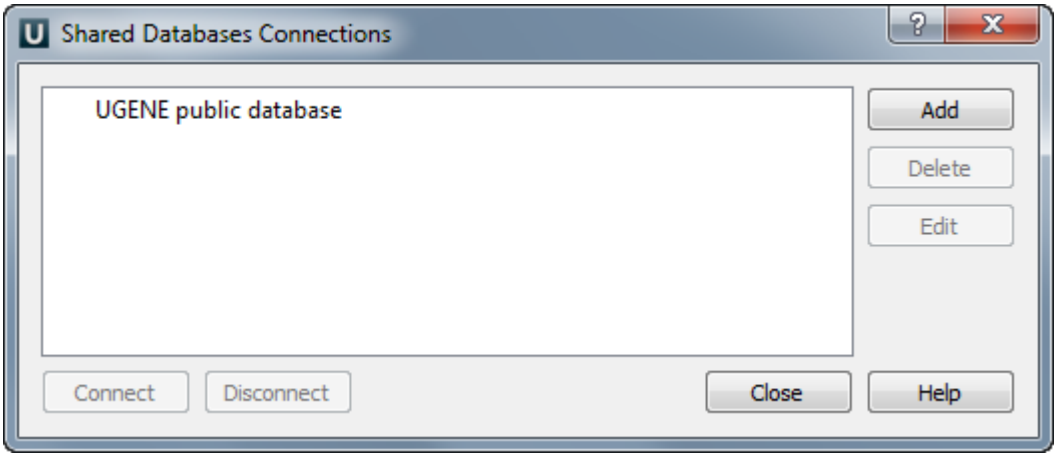
   The database with "your_database_name" is now available from a UGENE instance (version 1.14 or higher). It's time to try it out and fill it with some initial data. To do it, open UGENE and connect to the database. As we need to add the data to the database, use "user_nickname" and "user_password" of a user with privileges to modify the database. As soon as connection is established, add the required data to the database.

   From now on the data will be available for all users from this and other UGENE instances who connected to the same database.

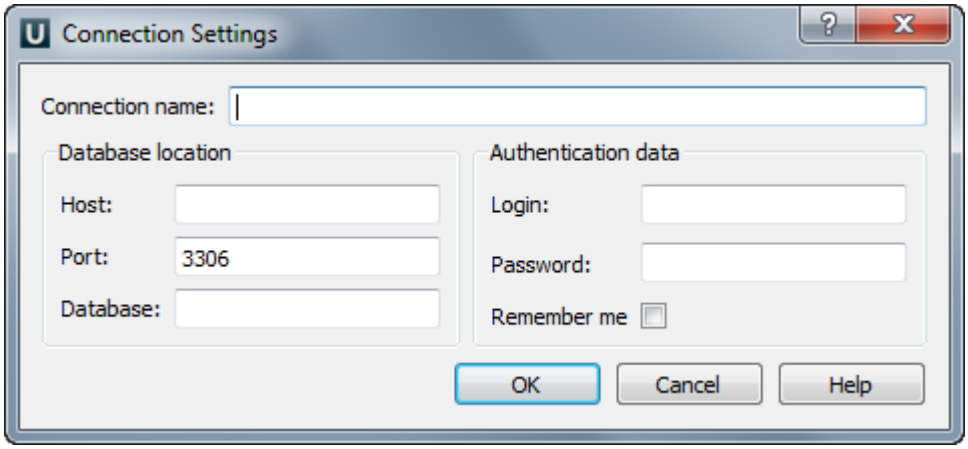
Connecting to a Shared Database

To start using the shared database you need to have a running public MySQL database server. Usually the system administrator of your department does it. You should ask him or her to give you the access to a MySQL database. Particularly, you need a few parameters to connect to the database: the IP-address of the server (the computer where a MySQL server is running), a user name for the MySQL database and a password. You can also install a MySQL server by yourself on any public computer you have an access to (even on your workstation), following the steps described in the Configuring Database section.

To connect to the database use the *File->Connect to shared database* main menu item. The following dialog appears:



To add new connection click on the *Add* button. The following dialog appears:



Here you need to specify *Host* (IP-address of the server), *Port* (number of the port used by the MySQL server) and *Database* (name of the database). You may also fill *Login* and *Password* fields. Otherwise, you are asked to input them every time you are establishing this connection until you check the *Remember me* box. Click on the *OK* button, then the connection is created and the appropriate item appears in the previous *Shared Database Connections* dialog.

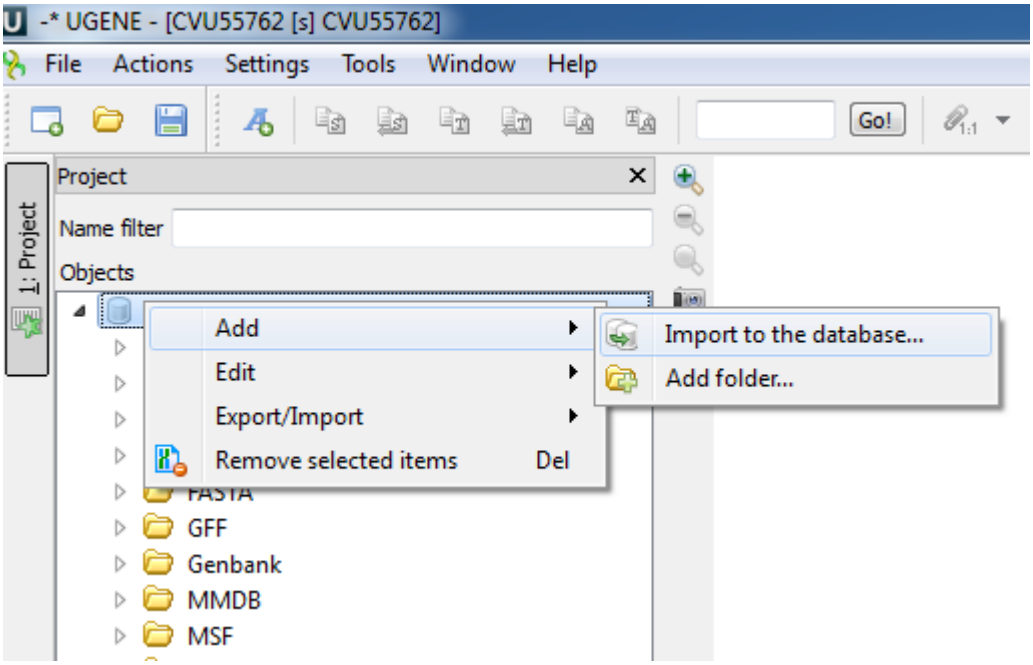
If you want to use already existing connection choose the appropriate item in the *Shared Database Connections* dialog and press the *Connect* button. This can also be done by double clicking the item. If the specified database is empty, UGENE has to initialize it. This routine is done only once. In this case you get an appropriate message box, asking whether to initialize the database or not. If you choose *Yes* the database is populated with UGENE data structures, if *No* it remains empty and UGENE does not connect to it.

If you want to delete some connection select it in the *Shared Database Connections* dialog and click on the *Delete* button. You may also edit connection parameters using the *Edit* button.

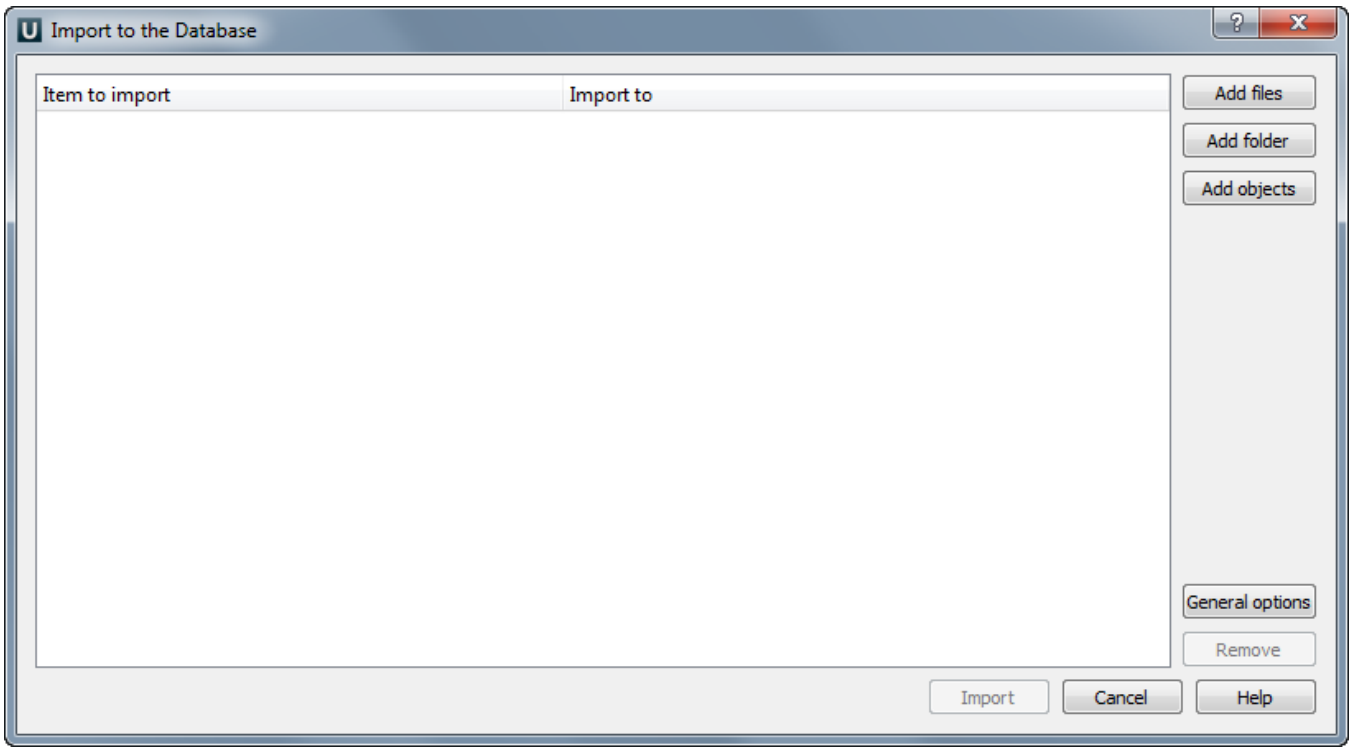
An established connection can be terminated by pressing the *Delete* button. The same effect is produced by removing the database document item from Project View.

Adding Data to the Database

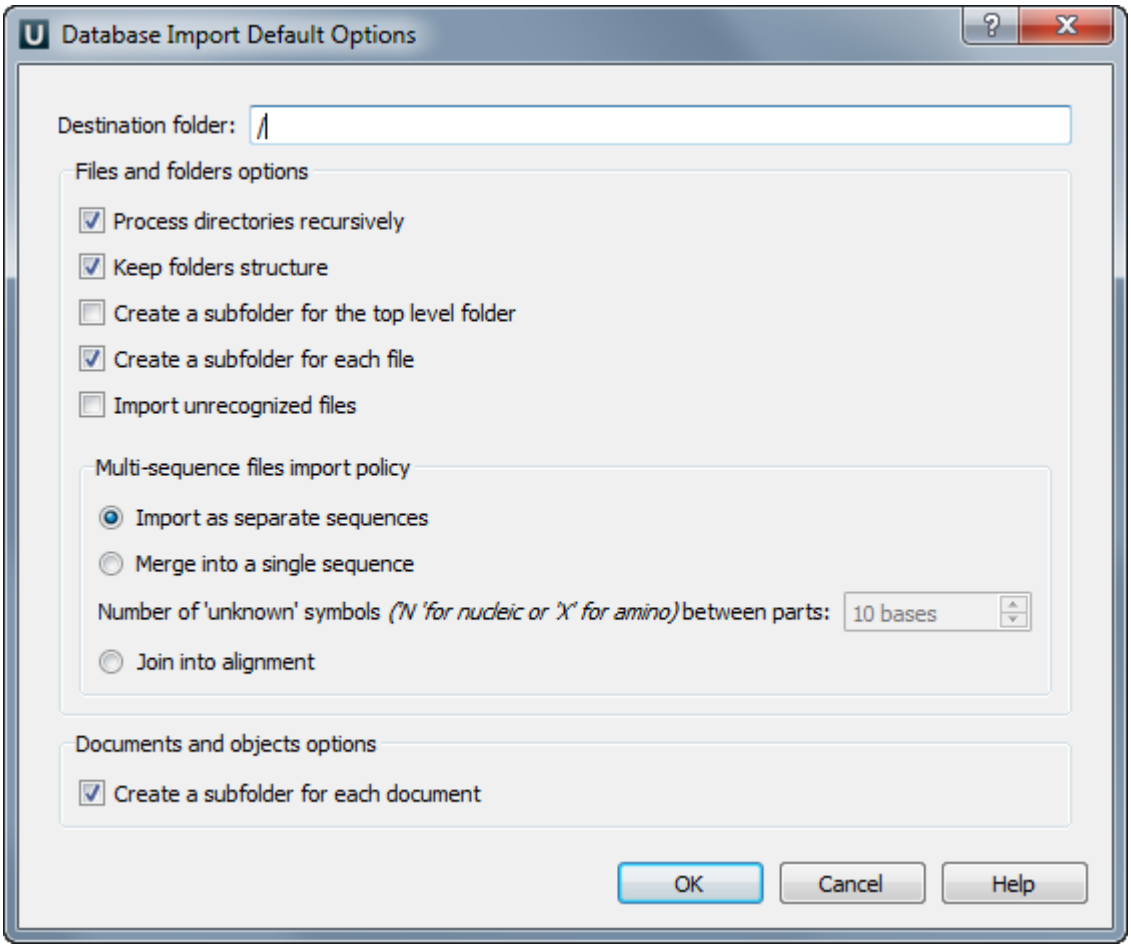
To add data to the database use the *Add->Import to the database* context menu item of the database in the project tree view. Also you can drag'n'drop it to a shared database folder.



The following dialog will appear:



Here you can add to the database files, folders or other objects from the current Project View. To do this use corresponding buttons. After specifying your data click on the *Import* button. The data will be imported and appear in the database data tree. Also you can change import settings. To do this click on the *General options* button. The following dialog will appear:



Available parameters are described below:

Process directories recursively - if this option is checked, the import procedure recreates the hierarchy of the imported directories and all their sub-directories in the database. Otherwise, only the content of the directories, specified for import, is uploaded to the *Destination folder* without taking into account any sub-directories.

Create a subfolder for each file - if this option is checked, for each file uploaded to the database a new folder is created, having the same name as the file, and the file content is placed in the folder. Otherwise, the file data are imported into the *Destination folder*.

Import as separate sequences - if this option is selected and an uploaded file contains several sequences, they are represented by distinct sequence objects in the database after the import is done.

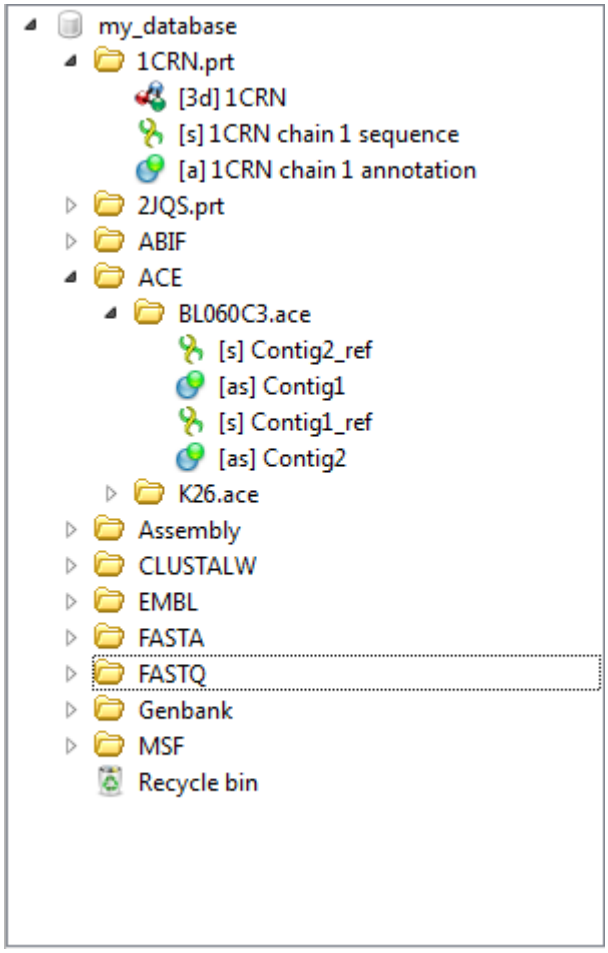
Merge into a single sequence - if this option is selected and an uploaded file contains several sequences, they are merged into a single sequence object in the database after the import is done.

Join into alignment - if this option is selected and an uploaded file contains several sequences, they are joined into a multiple sequence alignment object in the database after the import is done.

Create a subfolder for each document - if this option is checked, for each document or object uploaded to the database a new folder is created, having the same name as the file, and the data are placed in the folder. Otherwise, the data are imported into the *Destination folder*.

Database in the Project

The database in UGENE Project View looks like as a tree with folders and objects:



You can add a new folder to the database tree. To do that use the *Add->Add folder* database context menu item. To add a subfolder to some existing folder use the *Add->Add folder* folder context menu item. To delete an object or a folder press the *Delete* button or drag'n'drop it to the *Recycle bin*.

In this version of UGENE objects in the database are read-only. Nevertheless, there is a workaround to edit them. First, you need export the objects to files on your computer using the *Export/Import* object context menu. Then you can change that files locally, upload them to database and, finally, delete the originals.

If new data are added to the database by another user or removed from it, UGENE detects this and shows updates automatically in Project View.

Deleting Data

To remove an object or a folder select it and press the *Delete* button or drag it to the *Recycle bin* folder.

All removed items are located in the *Recycle bin* folder.

To delete all files from *Recycle bin* click on the *Empty recycle bin* context menu item of the *Recycle bin*.

To restore objects from the *Recycle bin* select them and call the *Restore selected items* context menu item.

When the database is updated outside, UGENE shows these changes on your computer automatically.



You cannot delete any object from *Recycle bin* if it is opened on the other computer. This situation can appear if the object was being viewed by another user when you moved it to *Recycle Bin*.

Drag'n'drop in the Database

In the database tree you can drag'n'drop objects between folders, folders between folders. Also you can drag'n'drop other objects and documents from project to the database.

Exporting Objects from the Database

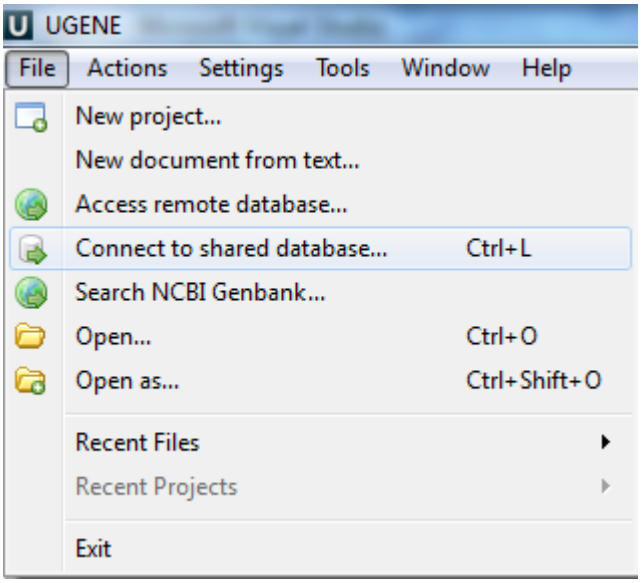
The objects in the database can not be altered though they can be deleted. To edit the objects you need to export them to the project, then make you modifications locally and replace existing originals. More detailed information about exporting you can find here.

UGENE Public Storage

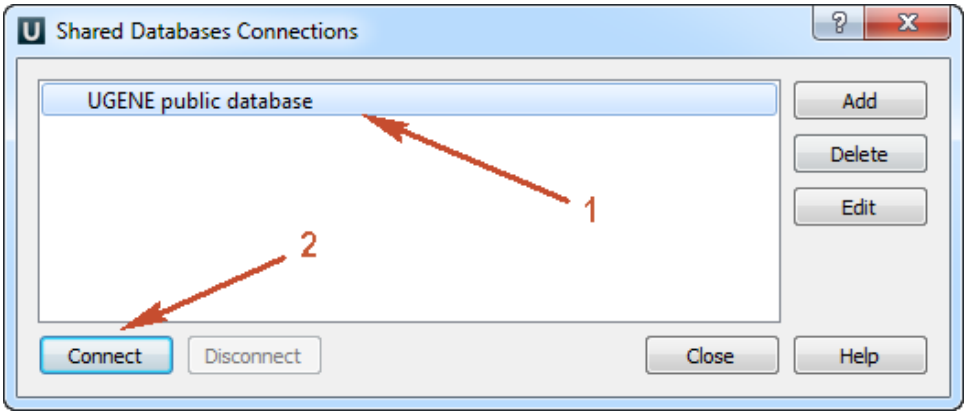
UGENE provides the free-to-use public bioinformatics data storage. This storage keeps DNA sequences of several popular genomes such as human, mouse, drosophila melanogaster, etc. and hundreds of plasmid sequences.

Follow the instructions for accessing the storage:

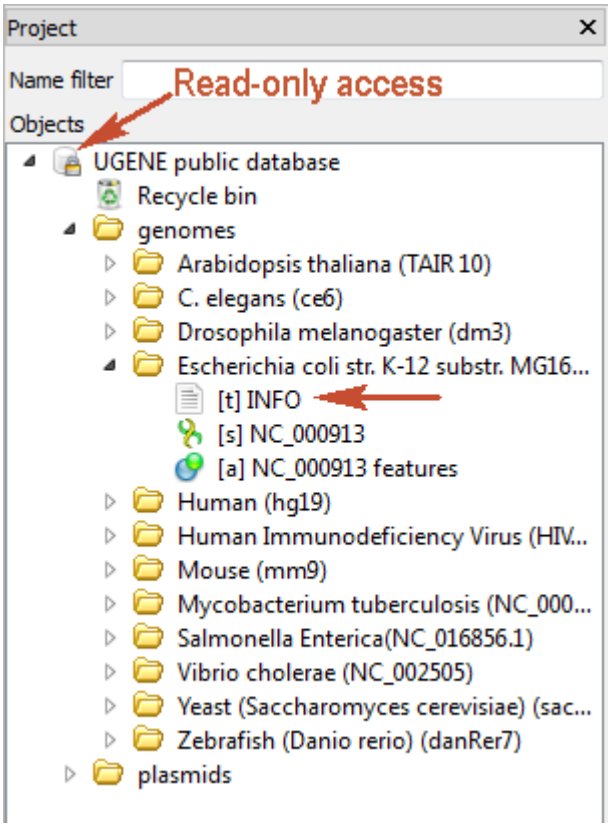
1. Use the menu *File -> Connect to shared database* (or press the *Ctrl+L* shortcut).



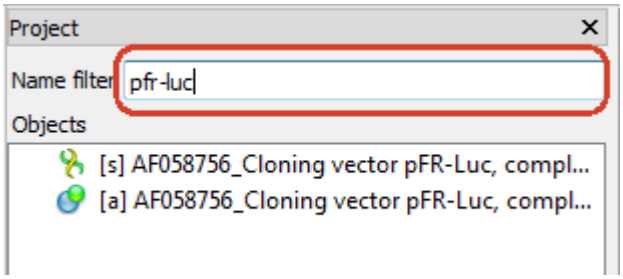
2. Choose the predefined "UGENE public database" item and click the *Connect* button.



3. Browse the storage content.



- The storage document is marked with the lock icon . It means that the storage provides the read-only access. Any data modifications are unavailable for such type of access (importing, removing or replacing of data).
- Each genome folder contains the *INFO* text object. It is the information about the genome or its source.
- You can export the data to your computer for working with the data locally.
- There are hundreds of plasmids in the storage. Use the name filter for fast navigating and searching an interesting plasmid:



The list of available genomes:

- Human (hg19)
- Mouse (mm9)
- Arabidopsis thaliana (TAIR 10)
- C. elegans (ce6)
- Drosophila melanogaster (dm3)
- Escherichia coli str. K-12 substr. MG1655 K12(NC_000913.3)
- Human Immunodeficiency Virus (HIV-2)
- Mycobacterium tuberculosis (NC_000962.3)
- Salmonella Enterica(NC_016856.1)
- Vibrio cholerae (NC_002505)
- Yeast (Saccharomyces cerevisiae)(sacCer3)
- Zebrafish (Danio rerio) (danRer7)

UGENE Command Line Interface

UGENE command line interface (CLI) was developed keeping in mind the following principles:

- To make it as easy as popular shell commands.
- To include all significant UGENE features.
- To allow users to add their own commands.

To use UGENE CLI make sure to add the path to the UGENE executable to your **%PATH%** environment variable.

The general syntax is the following:

```
ugene [[--task=]task_name] [--task_parameter=value ...] [-task_parameter value ...] [--option[=value]] [-option
[ value]]
```

Here:

task_name — task to execute, it can be one of the *predefined tasks* or *a task you have created*.

task_parameter — parameter of the specified task. Some parameters of a task are required, like *in* and *out* parameters of some tasks.

option — one of the *CLI options*.

See the example below:

```
ugene align --in=COI.aln -out result.aln -log-level-details
```


- CLI Options
- CLI Predefined Tasks
  - Format Converting Sequences
  - Converting MSA
  - Extracting Sequence
  - Finding ORFs
  - Finding Repeats
  - Finding Pattern Using Smith-Waterman Algorithm
  - Adding Phred Quality Scores to Sequence
  - Local BLAST+ Search
  - Remote NCBI BLAST and CDD Requests
  - Annotating Sequence with UQL Schema
  - Building Profile HMM Using HMMER2
  - Searching HMM Signals Using HMMER2
  - Aligning with MUSCLE
  - Aligning with ClustalW
  - Aligning with ClustalO
  - Aligning with Kalign
  - Aligning with MAFFT
  - Aligning with T-Coffee
  - Building PFM
  - Searching for TFBS with PFM
  - Building PWM
  - Searching for TFBS with Weight Matrices
  - Building Statistical Profile for SITECON
  - Searching for TFBS with SITECON
  - Fetching Sequence from Remote Database
  - Gene-by-Gene Report
  - Reverse-Complement Converting Sequences
  - Variants Calling
  - Generating DNA Sequence

- Creating Custom CLI Tasks

CLI Options

--help | -h [<option_name> | <task_name>]

Shows help information. For example:

```
ugene --help                    ## Shows general UGENE CLI help.
ugene -h

ugene --help=<option_name>      ## Shows help for the <option_name> option.
ugene -h <option_name>

ugene --help=<task_name>        ## Shows help for the <task_name> task.
ugene -h <task_name>
```

--task=<task_name> [<task_parameter>=value ...]

Specifies the task to run. A user-defined UGENE workflow schema can be used as a task name. For example:

```
ugene --task=align --in=COI.aln -out result.aln

ugene --task=C:\myschema.uwl --in=COI.aln --out=res.aln
```

--log-no-task-progress

A task progress is shown by default when a task is running. This option specifies not to show the progress.

--log-level="[<category1>=]<level1> [, ...]"

Sets the log level per category. If a category is not specified, the log level is applied to all categories.

The following categories are available:

- “Algorithms”
- “Console”
- “Core Services”
- “Input/Output”
- “Performance”
- “Remote Service”
- “Scripts”
- “Tasks”.

The following log levels are available: TRACE, DETAILS, INFO, ERROR or NONE.

By default, loglevel=ERROR.

For example:

```
ugene --log-level=NONE

ugene --log-level="Tasks=DETAILS, Console=DETAILS"
```

--log-format="<format_string>"

Specifies the format of a log line.

Use the following notations: L - level, C - category, YYYY or YY - year, MM - month, dd - day, hh - hour, mm - minutes, ss - seconds, zzz - milliseconds.

By default, logformat=”[L][hh:mm]”.

--license

Shows license information.

--lang=language_code

Specifies the language to use (e.g. for the log output). The following values are available:

- *CS* (Czech)
- *EN* (English)
- *RU* (Russian)

--log-color-output

If log output is enabled, this option make it colored: *ERROR* messages are displayed in red, *DETAILS* messages are displayed in green, *TRACE* me ssages are displayed in blue.

--session-db

Session database is stored in the temporary file that is created for every UGENE run. But it can supplied with the command line argument. If the supplied file does not exest it will be created. The session database file is removed after closing of UGENE.

For example:

```
ugene --session-db=D:/session.ugenedb
```

--version

Shows version information.

--tmp-dir=<path_to_file>

Path to teporary folder.

--ini-file=<path_to_file>

Loads configuration from the specified .ini file. By default the UGENE.ini file is used.

--genome-aligner

UGENE Genome Aligner is an efficient and fast tool for short read alignment. It has 2 work modes: build index and align short reads (default mode). If there is no index available for reference sequence it will be built on the fly.

Usage: *ugene --genome-aligner { --option[=argument] }*

The following options are available:

--build-index Use this flag to only build index for reference sequence.

--reference Path to reference genome sequence

--short-reads Path to short-reads data in FASTA or FASTQ format

--index Path to prebuilt index (base file name or with .idx extension). If not set, index is searched in system temporary directory. If --build-index option is applied, index will be saved to specified
path.

--result Path to output alignment in UGENEDB or SAM format (see --sam)

--memsize Memory size (in Mbs) reserved for short-reads. The bigger value the faster algorithm works. Default value depends on available system memory.

--ref-size Index fragmentation size (in Mbs). Small fragments better fit into RAM, allowing to load more short reads. Default value is 10.

--n-mis Absolute amount of allowed mismatches per every short-read (mutually exclusive with --pt-mis). Default value is 0.

--pt-mis Percentage amount of allowed mismatches per every short-read (mutually exclusive with --n-mis). Default value is 0.

--rev-comp Use both the read and its reverse complement during the aligning.

--best Report only about best alignments (in terms of mismatches).

--omit-size Omit reads with qualities lower than the specified value. Reads which have no qualities are not omitted. Default value is 0.

--sam Output aligned reads in SAM format. Default value is false.

For example:

```
Build index for reference sequence:
ugene --genome-aligner --build-index --reference=/path/to/ref


Align short reads using existing index:
ugene --genome-aligner --reference=/path/to/ref --short-reads=/path/to/reads --result=/path/to/result
```

CLI Predefined Tasks

Using current version of UGENE you can perform the following tasks by running a simple command:

- Format Converting Sequences
- Converting MSA
- Extracting Sequence
- Finding ORFs
- Finding Repeats
- Finding Pattern Using Smith-Waterman Algorithm
- Adding Phred Quality Scores to Sequence
- Local BLAST+ Search
- Remote NCBI BLAST and CDD Requests
- Annotating Sequence with UQL Schema
- Building Profile HMM Using HMMER2
- Searching HMM Signals Using HMMER2
- Aligning with MUSCLE
- Aligning with ClustalW
- Aligning with ClustalO
- Aligning with Kalign
- Aligning with MAFFT
- Aligning with T-Coffee
- Building PFM
- Searching for TFBS with PFM
- Building PWM
- Searching for TFBS with Weight Matrices
- Building Statistical Profile for SITECON
- Searching for TFBS with SITECON
- Fetching Sequence from Remote Database
- Gene-by-Gene Report
- Reverse-Complement Converting Sequences
- Variants Calling
- Generating DNA Sequence

Format Converting Sequences

Task Name: convert-seq

Converts a sequence from one format to another.

Parameters:

in — input sequence file. [String, Required]

out — name of the output file. [String, Required]

format — format of the output file. [String, Optional]

The following values are available:

- fasta
- fastq
- genbank
- gff
- raw

Example:

```
ugene convert-seq --in=human_T1.fa --out=human_T1.gbk --format=genbank
```

Converting MSA

Task Name: convert-msa

Converts a multiple sequence alignment file from one format to another.

Parameters:

in — input multiple sequence alignment file. [String, Required]

out — name of the output file. [String, Required]

format — format of the output file. [String, Optional]

The following values are available:

- clustal (default)
- fasta
- mega
- msf
- nexus
- phylip-interleaved
- phylip-sequential
- stockholm

Example:

```
ugene convert-msa --in=CBS.sto --out=CBS --format=msf
```

Extracting Sequence

Task Name: extract-sequence

Extracts annotated regions from an input sequence.

Parameters:

in — semicolon-separated list of input files. [String, Required]

out — output file. [String, Required]

annotation-names — list of annotations names which will be accepted or filtered. [String, Optional]

annotation-names-file - file with annotation names, separated with whitespaces which will be accepted or filtered. [String, Optional]

accumulate - accumulate all incoming data in one file or create separate files for each input. In the latter case, an incremental numerical suffix is added to the file name (using 'True' by default). [Boolean]

accept-or-filter — if set to *true*, accepts only the specified annotations, if set to *false*, accepts all annotations except the specified ones. [Boolean, Optional]

complement — complements the annotated regions if the corresponding annotation is located on the complement strand. [Boolean, Optional]

extend-left — extends the resulting regions to the left for the specified number of base symbols. [Number, Optional]

extend-right — extends the resulting regions to the right for the specified number of base symbols. [Number, Optional]

gap-length — inserts a gap of the specified length between the merged annotations.

transl - translates the annotated regions. [Boolean, Optional]

Example:

```
ugene extract-sequence --in=sars.gb --out=res.fa --annotation-names=gene
```

Finding ORFs

Task Name: find-orfs

Searches for Open Reading Frames (ORFs) in nucleotide sequences and saves the regions found as annotations.

Parameters:

in — semicolon-separated list of input files. [String, Required]

out — output file with the annotations. [String, Required]

name — name of the annotated regions. [String, Optional, Default: “ORF”]

min-length — ignores ORFs shorter than the specified length. [String, Optional, Default: 100]

require-stop-codon — ignores boundary ORFs that last beyond the search region (i.e. have no stop codon within the range). [Boolean, Optional, Default: false]

require-init-codon — allows ORFs starting with any codon other than terminator. [Boolean, Optional, Default: true]

allow-alternative-codons — allows ORFs starting with alternative initiation codons, accordingly to the current translation table. [Boolean, Optional, Default: false]

Example:

```
ugene find-orfs --in=human_T1.fa --out=result.gb --require-init-codon=false
```

Finding Repeats

Task Name: find-repeats

Searches for repeats in sequences and saves the regions found as annotations.

Parameters:

in — semicolon-separated list of input files. [String, Required]

out — output file with the annotations. [String, Required]

name — name of the annotated regions. [String, Optional, Default: “repeat_unit”]

min-length — minimum length of the repeats. [Number, Optional, Default: 5]

identity — percent identity between repeats. [Number, Optional, Default: 100]

min-distance — minimum distance between the repeats. [Number, Optional, Default: 0]

max-distance — maximum distance between the repeats. [Number, Optional, Default: 5000]

inverted — if *true*, searches for the inverted repeats. [Boolean, Optional, Default: false]

Example:

```
ugene find-repeats --in=murine.gb --out=murine_repeats.gb --identity=99
```

Finding Pattern Using Smith-Waterman Algorithm

Task Name: find-sw

Searches for a pattern in a nucleotide or protein sequence using the Smith-Waterman algorithm and saves the regions found as annotations.

Parameters:

in — input sequence file. [String, Required]

out — output file with the annotations. [String, Required]

name — name of the annotated regions. [String, Optional, Default: “misc_feature”]

ptrn — subsequence pattern to search for (e.g. *AGGCCT*). [String, Required]

score — percent identity between the pattern and a subsequence. [Number, Optional, Default: 90]

matrix — scoring matrix. [String, Optional, Default: “Auto”]

Among others the following values are available:

- blosum62
- dna
- rna
- dayhoff
- gonnet
- pam250
- etc.

The matrices available are stored in the $UGENE\data\weight_matrix directory.

filter — results filtering strategy. [String, Optional, Default: “filter-intersections”]

The following values are available:

- filter-intersections
- none

Example:

```
ugene find-sw --in=human_T1.fa --out=sw.gb --ptrn=TGCT --filter=none
```

Adding Phred Quality Scores to Sequence

Task Name: join-quality

Adds Phread quality scores to a sequence and saves the result to the output FASTQ file.

Parameters:

in — input sequence file. [String, Required]

quality — input Phred quality scores file. [String, Required]

out — output FASTQ file. [String, Required]

Example:

```
ugene join-quality --in=e_coli.fa --quality=e_coli.qual --out=res.fastq
```

Local BLAST+ Search

Task Name: local-blast+

Performs a search on a local BLAST database using BLAST+.

BLAST+ is used as an *external tool* and must be installed on your system.

Parameters:

toolpath — path to an appropriate BLAST executable (e.g. blastn, blastp, etc.). By default, the path specified in the *Application Settings* is applied. [String, Optional, Default: “default”]

tmpdir — directory for temporary files. By default, the path specified in the *Application Settings* is applied. [String, Optional, Default: “default”]

in — semicolon-separated list of input sequence files. [String, Required]

dbpath — path to the BLAST database files. [String, Required]

dbname — base name of the BLAST+ database files. [String, Required]

out — output Genbank file, the results of the search are stored as annotations. [String, Required]

name — name of the annotations. [String, Optional, Default: “blast result”]

p — type of the BLAST search. [String, Optional, Default: “blastn”]

The following values are available:

- blastn
- blastp
- blastx
- tblastn
- tblastx

e — expectation value threshold. [Number, Optional, Default: 10]

Example:

```
ugene local-blast+ --in=input.fa --dbpath=. --dbname=mydb --out=output.gb
```

Remote NCBI BLAST and CDD Requests

Task Name: remote-request

Performs remote requests to the NCBI. Saves the results as annotations.

Parameters:

in — semicolon-separated list of input files. A file can be of any format containing sequences or alignments. [String, Required]

db — database to search in. [String, Optional, Default: “ncbi-blastn”]

The following databases are available:

- “ncbi-blastn” for nucleotide sequences
- “ncbi-cdd” for amino acid sequences
- “ncbi-blastp” for amino acid sequences

out — output Genbank file. [String, Required]

eval — specifies the statistical significance threshold for reporting matches against database sequences. [Number, Optional, Default: 10]

hits — maximum number of hits, that will be shown. [Number, Optional, Default: 10]

name — name of the result annotations. If not set, name will be specified with the “cdd” result or the “blast” result. [String, Optional, Default: “cdd” or “blast”]

short — optimizes search for short sequences. [Boolean, Optional, Default: false]

blast-output — path to the file with the NCBI-BLAST output (only for the “ncbi-blastp” and “ncbi-blastn” databases). [Boolean, Optional, Default: the file is not saved]

Example:

```
ugene remote-request --in=seq.fa --db=ncbi-blastp --out=res.gb
```

Annotating Sequence with UQL Schema

Task Name: query

Annotates a sequence in compliance with a UGENE Query Language (UQL) schema. This allows to analyze a sequence using different algorithms at the same time imposing constraints on the positional relationship of the results.

To learn more about the UQL schemas read the Query Designer Manual.

Parameters:

in — semicolon-separated list of input sequence files. [String, Required]

out — output Genbank file with the annotations. [String, Required]

schema — UQL schema. [String, Required]

merge — if true, merges regions of each result into a single annotation. [Boolean, Optional, Default: false]

offset — if *merge* is set to true, specified left and right offsets for merged annotations. [Number, Optional, Default: 0]

Example:

```
ugene query --in=input.fa --out=result.gb --schema=RepeatsWithORF.uql
```

Building Profile HMM Using HMMER2

Task Name: hmm2-build

Builds a profile HMM using the HMMER2 tools.

Parameters:

in — semicolon-separated list of input multiple sequence alignment files. [String, Required]

out — output HMM file. [String, Required]

name — name of the profile HMM. [String, Optional, Default: “hmm_profile”]

calibrate — enables/disables calibration. [Boolean, Optional, Default: true]

seed — random seed, a non-negative integer. [Number, Optional, Default: 0]

Example:

```
ugene hmm2-build --in=CBS.sto --out=CBS.hmm
```

Searching HMM Signals Using HMMER2

Task Name: hmm2-search

Searches each input sequence for the significantly similar sequence that matches to all specified profile HMM using the HMMER2 tool.

Parameters:

seq — semicolon-separated list of the input sequence files. [String, Required]

hmm — semicolon-separated list of the input HMM files. [String, Required]

out — output file with annotations. [String, Required]

name — name of the result annotations. [String, Optional, Default: “hmm_signal”]

e-val — e-value that can be used to exclude low-probability hits from the result. [Number, Optional, Default: 1e-1]

score — score based filtering which is an alternative to e-value filtering to exclude low-probability hits from the result. [Number, Optional, Default: -1000000000]

Example:

```
ugene hmm2-search --seq=CBS_seq.fa --hmm=CBS.hmm --out=CBS_hmm.gb
```

Aligning with MUSCLE

Task Name: align

Performs multiple sequence alignment with MUSCLE algorithm and saves the resulting alignment to file. Source data can be of any format containing sequences or alignments.

Parameters:

in - Input alignment [Url datasets]
max-iterations - Maximum number of iterations (using '2' by default) [Number]
mode - Selector of preset configurations, that give you the choice of optimizing accuracy, speed,or some compromise between the two. The default favors accuracy (using 'MUSCLE default' by default) [Number]
range - Whole alignment or column range e.g. 1..100 (using 'Whole alignment' by default) [String]
stable - Do not rearrange aligned sequences (using 'True' by default) [Boolean]
format - Document format of output alignment (using 'clustal' by default) [String]
out - Output alignment [String]

Example:

```
ugene align  --in=test.aln --out=test_out.aln --format=clustal
```

Aligning with ClustalW

Task Name: align-clustalw

Multiple sequence alignment with ClustalW.

ClustalW is used as an *external tool* and must be installed on your system.

Parameters:

toolpath — path to the ClustalW executable. By default, the path specified in the *Application Settings* is applied. [String, Optional, Default: “default”]

tmpdir — directory for temporary files. [String, Optional]

in — semicolon-separated list of input files. [String, Required]

out — output file. [String, Required]

format — format of the output file. [String, Optional]

Example:

```
ugene align-clustalw  --in=COI.aln --out=COI.sto --format=stockholm
```

Aligning with ClustalO

Task Name: align-clustalo

Create alignment with ClustalO. ClustalO is a general purpose multiple sequence alignment program for proteins.

ClustalO is used as an *external tool* and must be installed on your system.

Parameters:

in - Input alignment [Url datasets]

format - Document format of output alignment (using 'clustal' by default) [String]

out - Output alignment [String]

max-guidetree-iterations - Maximum number guidetree iterations (using '0' by default) [Number]

max-hmm-iterations - Maximum number of HMM iterations (using '0' by default) [Number]

iter - Number of (combined guide-tree/HMM) iterations (using '1' by default) [Number]

toolpath - ClustalO location (using the path specified in UGENE by default) [String]

auto - Set options automatically (might overwrite some of your options) (using 'False' by default) [Boolean]

tmpdir - Directory to store temporary files (using UGENE temporary directory by default) [String]

Example:

```
ugene align-clustalw  --in=test.aln --out=test_out.aln --format=clustal
```

Aligning with Kalign

Task Name: align-kalign

Multiple sequence alignment with Kalign.

Parameters:

in — semicolon-separated list of input files. [String, Required]

out — output file in the ClustalW format. [String, Required]

Example:

```
ugene align-kalign --in=COI.aln --out=COI_aligned.aln
```

Aligning with MAFFT

Task Name: align-mafft

Multiple sequence alignment with MAFFT.

MAFFT is used as an *external tool* and must be installed on your system.

Parameters:

toolpath — path to the MAFFT executable. By default, the path specified in the *Application Settings* is applied. [String, Optional, Default: “default”]

tmpdir — directory for temporary files. [String, Optional]

in — semicolon-separated list of input files. [String, Required]

out — output file. [String, Required]

format — format of the output file. [String, Required]

op — penalty for opening a gap. [Number, Optional]

ep — penalty for extending a gap. [Number, Optional]

maxiterate — maximum number of cycles of iterative refinement. [Number, Optional]

Example:

```
ugene align-mafft --in=COI.aln --out=COI_aligned.aln
```

Aligning with T-Coffee

Task Name: align-tcoffee

Create alignment with T-Coffee. T-Coffee is a collection of tools for computing, evaluating and manipulating multiple alignments of DNA, RNA, Protein Sequences.

T-Coffee is used as an *external tool* and must be installed on your system.

Parameters:

gap-ext-penalty - Gap Extension Penalty. Positive values give rewards to gaps and prevent the alignment of unrelated segments (using '0' by default) [Number]

gap-open-penalty - Gap Open Penalty. Must be negative, best matches get a score of 1000 (using '-50' by default) [Number]

iter-max - Number of iteration on the progressive alignment: 0 - no iteration (default), -1 - Nseq iterations (using '0' by default) [Number]

toolpath - T-Coffee location (using the path specified in UGENE by default) [String]

tmpdir - Directory to store temporary files (using UGENE temporary directory by default) [String]

in - Input alignment [Url datasets]

format - Document format of output alignment (using 'clustal' by default) [String]

out - Output alignment [String]

Example:

```
ugene align-tcoffee  --in=test.aln --out=test_out.aln --format=clustal
```

Building PFM

Task Name: pfm-build

Builds a position frequency matrix from a multiple sequence alignment file.

Parameters:

in — semicolon-separated list of input MSA files. [String, Required]

out — output file. [String, Required]

type — type of the matrix. [Boolean, Optional, Default: false]

The following values are available:

- true (dinucleic type)
- false (mononucleic type)

Dinucleic matrices are more detailed, while mononucleic ones are more useful for small input data sets.

Example:

```
ugene pfm-build --in=COI.aln --out=result.pfm
```

Searching for TFBS with PFM

Task Name: pfm-search

Searches for transcription factor binding sites (TFBS) with position weight matrices (PWM) converted from input position frequency matrices (PFM) and saves the regions found as annotations.

Parameters:

seq — semicolon-separated list of input sequence files to search TFBS in. [String, Required]

matrix — semicolon-separated list of the input PFM. [String, Required]

out — output Genbank file.

name — name of the annotated regions. [String, Optional, Default: “misc_feature”]

type — type of the matrix. [Boolean, Optional, Default: false]

The following values are available:

- true (dinucleic type)
- false (mononucleic type)

Dinucleic matrices are more detailed, while mononucleic ones are more useful for small input data sets.

algo — algorithm used to convert a PFM to a PWM. [String, Optional, Default: “Berg and von Hippel”]

The following values are available:

- Berg and von Hippel
- Log-odds
- Match
- NLG

score — minimum percentage score to detect TFBS. [Number, Optional, Default: 85]

strand — strands to search in. [Number, Optional, Default: 0]

The following values are available:

- 0 (both strands)
- 1 (direct strand)
- -1 (complement strand)

Example:

```
ugene pfm-search --seq=in.fa --matrix=MA0265.1.pfm;MA0266.1.pfm --out=res.gb
```

Building PWM

Task Name: pwm-build

Builds a position weight matrix from a multiple sequence alignment file.

Parameters:

in — semicolon-separated list of input MSA files. [String, Required]

out — output file. [String, Required]

type — type of the matrix. [Boolean, Optional, Default: false]

The following values are available:

- true (dinucleic type)
- false (mononucleic type)

Dinucleic matrices are more detailed, while mononucleic ones are more useful for small input data sets.

algo — algorithm used to build the matrix. [String, Optional, Default: “Berg and von Hippel”]

The following values are available:

- Berg and von Hippel
- Log-odds
- Match
- NLG

Example:

```
ugene pwm-build --in=COI.aln --out=result.pwm
```

Searching for TFBS with Weight Matrices

Task Name: pwm-search

Searches for transcription factor binding sites (TFBS) with position weight matrices (PWM) and saves the regions found as annotations.

Parameters:

seq — semicolon-separated list of input sequence files to search TFBS in. [String, Required]

matrix — semicolon-separated list of the input PWM. [String, Required]

out — output Genbank file.

name — name of the annotated regions. [String, Optional, Default: “misc_feature”]

min-score — minimum percentage score to detect TFBS. [Number, Optional, Default: 85]

strand — strands to search in. [Number, Optional, Default: 0]

The following values are available:

- 0 (both strands)
- 1 (direct strand)
- -1 (complement strand)

Example:

```
ugene pwm-search --seq=input.fa --matrix=Aro80.pwm;Aft1.pwm --out=res.gb
```

Building Statistical Profile for SITECON

Task Name: sitecon-build

Builds a statistical profile for SITECON. It can be later used to search for TFBS.

Parameters:

in — semicolon-separated list of input DNA multiple sequence alignment files. An input file must not contain gaps. [String, Required]

out — output file. If several input files have been supplied, then a sitecon profile is built for each input file, i.e. several output files (with different indexes) are generated. [String, Required]

wsize — window size. The window is a region of the alignment used to build the profile. It is picked up from the center of the alignment and occupies the specified length. The edges of the alignment beyond the window are not taken into account. The recommended length is a bit less than the alignment length, but not more than 50 bp. [Number, Optional, Default: 40]

clength — length of a random synthetic sequence used to calibrate the profile. [Number, Optional, Default: 1000000]

rseed — random seed used to calibrate the profile, e.g. to generate the random synthetic sequence. Use the same value to get the same calibration results twice on the same data. By default, new random seed is generated each time a calibration occurs. [Number, Optional, Default: 0]

walg — specifies to use the *Algorithm 2* weight algorithm. In most cases it is not required, but in some cases it can increase the recognition quality. [Boolean, Optional, Default: false]

Example:

```
ugene sitecon-build --in=COI.aln --out=result.sitecon
```

Searching for TFBS with SITECON

Task Name: sitecon-search

Searches for transcription factor binding sites (TFBS) with SITECON and saves the regions found as annotations.

Parameters:

in — semicolon-separated list of input sequence files to search TFBS in. [String, Required]

inmodel — input SITECON profile(s). If several profiles have been supplied, searches with all profiles one by one and outputs merged set of annotations for each input sequence. [String, Required]

out — output Genbank file. [String, Required]

annotation-name — name of the annotated regions. [String, Optional, Default: “misc_feature”]

min-score — recognition quality threshold. The value must be between 60 and 100. Choosing too low threshold will lead to recognition of too many TFBS recognised with too low trustworthiness. Choosing too high threshold may result in no TFBS recognised. [Number, Optional, Default: 85]

min-err1 — setting for filtering results, minimal value of Error type I. [Number, Optional, Default: 0]

max-err2 — setting for filtering results, maximum value of Error type II. [Number, Optional, Default: 0.001]

strand — strands to search in. [Number, Optional, Default: 0]

The following values are available:

- 0 (both strands)
- 1 (direct strand)
- -1 (complement strand)

Example:

```
ugene sitecon-search --in=input.fa --inmodel=profile.sitecon --out=res.gb
```

Fetching Sequence from Remote Database

Task Name: fetch-sequence

Fetches a sequence from a remote database. The supported databases are accesed via alias.

Database	Alias
NCBI Genbank (DNA)	genbank
NCBI Genbank (protein)	genbank-protein
Protein Data Bank	pdb
SwissProt	swissprot
Uniprot	uniprot

Parameters:

db — database alias to read from. [String, Required]

id — semicolon-separated list of resource IDs in the database. [String, Required]

save-dir — directory to store sequence files loaded from the database. [String, Optional]

Example:

```
ugene fetch-sequence --db=PDB --id=3INS;1CRN
```

Gene-by-Gene Report

Task Name: gene-by-gene

Suppose you have genomes and you want to characterize them. One of the ways to do that is to build a table of what genes are in each genome and what are not there.

1. Create a local BLAST db of your genome sequence/contigs. One db per one genome.
2. Create a file with sequences of genes you what to explore. This file will be the input file for the scheme
3. Setup location and name of BLAST db you created for the first genome.
4. Setup output files: report location and output file with annotated (with BLAST) sequence. You might want to delete the "Write Sequence" element if you do not need output sequences.
5. Run the scheme
5*. Run the scheme on the same input and output files changing BLAST db for each genome that you have.

As the result you will get the report file. With "Yes" and "No" field. "Yes" answer means that the gene is in the genome. "No" answer MIGHT mean that there is no gene in the genome. It is a good idea to analyze al
l the "No" sequences using annotated files. Just open a file and find a sequence with a name of a gene that has "No" result.

Parameters:

in - Input sequence file [Url datasets]

final-name - Annotation name used to compare genes and reference genomes (using 'blast_result' by dafault) [String]

exist-file - If a target report already exists you should specify how to handle that. Merge two table in one. Overwrite or Rename existing file (using 'Merge' by default) [String]

ident - Identity between gene sequence length and annotation length in per cent. BLAST identity (if specified) is checked after (using '90.0' percents by default) [Number]

out - Output report file [String]

blast-out - Location of BLAST output file [String]

search-type - Type of BLAST searches (using 'blastn' by default) [String]

db-name - Name of BLAST DB [String]

blast-path - Path to BLAST DB [String]

expected-value - This setting specifies the statistical significance threshold for reporting matches against database sequences (using '10.0' by default) [Number]

gapped-aln - Perform gapped alignment (using 'use' by default) [Boolean]

blast-name - Name for annotations (using 'blast_result' by default) [String]

tmpdir - Directory for temporary files (using UGENE temporary directory by default) [String]

toolpath - External tool path (using the path specified in UGENE by default) [String]

out-type - Type of BLAST output file (using 'XML (-m 7)' by default) [String]

Example:

```
ugene gene-by-gene --in=human_T1.fa --out=human_T1_report
```

Reverse-Complement Converting Sequences

Task Name: revcompl

Convert input sequence into its reverse, complement or reverse-complement counterpart and write result sequence to file

Parameters:

type - Type of operation. Available are 'Reverse Complement', 'Complement' and 'Reverse' (using 'Reverse Complement' by default) [String]

in - Input file [Url datasets]

accumulate - Accumulate all incoming data in one file or create separate files for each input. In the latter case, an incremental numerical suffix is added to the file name (using 'True' by default) [Boolean]

format - Output file format (using 'fasta' by default) [String]

split - Split each incoming sequence on several parts (using '1' by default) [Number]

out - Output file [String]

Example:

```
ugene revcompl --in=human_T1.fa --out=human_T1_result.fa --format=fasta --type=reverse
```

Variants Calling

Task Name: snp

Call variants for an input assembly and a reference sequence using SAMtools mpileup and bcftool

Parameters:

bam - Input sorted BAM file(s) [Url datasets]

ref - Input reference sequence [Url datasets]

wout - Out file with variations [String]

bN - A/C/G/T only [Boolean]

bI - List of sites [String]

ml - BED or position list file [String]

bg - Per-sample genotypes [Boolean]

mC - Mapping quality downgrading coefficient [Number]

bT - Pair/trio calling [String]

mB - Disable BAQ computation [Boolean]

me - Gap extension error [Number]

mE - Extended BAQ computation [Boolean]

bF - Indicate PL [Boolean]

vw - Gap size [Number]

m6 - Illumina-1.3+ encoding [Boolean]

bi - INDEL-to-SNP Ratio [Number]

bA - Retain all possible alternate [Boolean]

vD - Max number of reads per input BAM [Number]

md - Max number of reads per input BAM [Number]

mL - Max INDEL depth [Number]

va - Alternate bases [Number]

v2 - BaseQ bias [String]

vd - Minimum read depth [Number]

v4 - End distance bias [Number]

v3 - MapQ bias [Number]

Q - Minimum RMS quality [Number]

v1 - Strand bias [Number]

mQ - Minimum base quality [Number]

mq - Minimum mapping quality [Number]

bd - Min samples fraction [Number]

b1 - N group-1 samples [Number]

bU - N permutations [Number]

bG - No genotype information [Boolean]

mI - No INDELs [Boolean]

mo - Gap open error [Number]

mP - List of platforms for indels [String]

vp - Log filtered [Boolean]

bP - Prior allele frequency spectrum. [String]

bQ - QCALL likelihood [Boolean]

mr - Pileup region [String]

bs - List of samples [String]

mh - Homopolymer errors coefficient [Number]

bt - Mutation rate [Number]

mA - Count anomalous read pairs [Boolean]

vW - A/C/G/T only [Number]

Example:

```
ugene snp --bam=test.bam --ref=test_ref.fa --wout=test_out.vcf
```

Generating DNA Sequence

Task Name: generate-dna

Generates a random DNA sequence with specified nucleotide content

Parameters:

algo - Algorithm for generating (using 'GC Content' by default) [String]

content - Specifies if the nucleotide content of generated sequence(s) will be taken from reference or specified manually (A, G, C, T parameters) (using 'manual' by default) [String]

count - Number of sequences to generate (using '1' by default) [Number]

length - Length of the resulted sequence(s) (using '1000' bp by default) [Number]

a - Adenine content (using '25' percents by default) [Number]

c - Cytosine content (using '25' percents by default) [Number]

g - Guanine content (using '25' percents by default) [Number]

t - Thymine content (using '25' percents by default) [Number]

ref - Path to the reference file (could be a sequence or an alignment) [String]

seed - Value to initialize the random generator. By default (seed = -1) the generator is initialized with the system time (using '-1' by default) [Number]

wnd-size - Size of window where set content (using '1000' by default) [Number]

accumulate - Accumulate all incoming data in one file or create separate files for each input. In the latter case, an incremental numerical suffix is added to the file name (using 'True' by default) [Boolean]

format - Output file format (using 'fasta' by default) [String]

split - Split each incoming sequence on several parts (using '1' by default) [Number]

out - Output file [String]

Example:

```
ugene generate-dna --length=2000 --a=45 --out=test.fa
```

Creating Custom CLI Tasks

The *predefined tasks* are actually the *Workflow Designer* schemas stored in the $UGENE/data/cmdline directory.

Follow the instructions in the Workflow Designer Manual on how to create a schema and to run it from the command line.

You may also find useful the following video tutorial devoted to the creating of a custom console command:

- Creating custom console command (MUSCLE alignment with various output format)

APPENDIXES

- Appendix A. Supported File Formats
  - Specific File Formats
  - UGENE Native File Formats
  - Other File Formats

Appendix A. Supported File Formats

UGENE is able to read and write files compressed with Unix/Linux *gzip* utility. You don't have to unpack the files.

- Specific File Formats
- UGENE Native File Formats
- Other File Formats

Specific File Formats

File format	File extension	Read	Write	Comment
ABIF	*.ab1, *.abi, *.abif	+	-	A chromatogram file format. **See also:** *Chromatogram Viewer*
ACE	*.ace	+	-	A file format for storing data about genomic contigs. **See also:** *Alignment Editor*
APR	*.apr	+	-	Vector NTI / AlignX Alignment format. **See also:** *Alignment Editor*
Bairoch	*.bairoch	+	+	A file format to store enzymes. **See also:** *Restriction Analysis*
BAM	*.bam	+	-	Binary compressed SAM format. **See also:** *Assembly Browser*
ClustalW	*.aln	+	+	A multiple sequence alignments (MSA) file format. **See also:** *Alignment Editor*
EBWT	*.ebwt	+	+	A Bowtie prebuilt index file.

				See also: *Bowtie*
EMBL	*.em, *.emb, *.embl	+	-	A rich format for storing sequences and their annotations. **See also:** *Sequence View*
FASTA	*.fa, *.mpfa, *.fna, *.fsa, *.fas, *.fasta, *.sef *.seqs	+	+	One of the oldest and simplest sequence file format. **See also:** *Sequence View*
FASTQ	*.fastq	+	+	A file format used to store a sequence and its corresponding quality scores. It was originally developed at the “Wellcome Trust Sanger Institute”. **See also:** *Sequence View*
Genbank	*.gb, *.gbk, *.gen, *.genbank	+	+	A rich format for storing sequences and associated annotations. **See also:** *Sequence View*
GFF	*.gff	+	+	The Gene Finding Format (GFF) format is used to store features and annotations. **See also:** *Sequence View*
HMM	*.hmm	+	+	A file format to store HMM profiles. **See also:** *HMM2*, *HMM3*
MMDB	*.prt	+	-	ASN.1 format used by the Molecular Modeling Database (MMDB). **See also:** *3D Structure Viewer*
MSF	*.msf	+	+	A multiple sequence alignments file format. **See also:** *Alignment Editor*
Mega	*.meg, *.meg.gz	+		A multiple sequence alignments file format. **See also:** *Alignment Editor*
Newick	*.nwk, *.newick	+	+	A tree file format. **See also:** *Building Phylogenetic Tree*, *Phylogenetic Tree Viewer*
Nexus	*.nex *.nxs	+	+	A multiple alignment and phylogenetic trees file format. **See also:** *Alignment Editor*, *Building Phylogenetic Tree*, *Phylogenetic Tree Viewer*
PDB	*.pdb	+	-	The Protein Data Bank (PDB) format allows to view the 3D structure of the sequence. **See also:** *3D Structure Viewer*
pDRAW32	*.pdw	+	-	A sequence file format used by pDRAW32 software. **See also:** *Sequence View*
PFM	*.pfm	+	+	A file format for a position frequency matrix. **See also:** *Weight Matrix*
Phylip	*.phy	+	+	A multiple alignment file format. **See also:** *Alignment Editor*
PWM	*.pwm	+	+	A file format for a position weight matrix. **See also:** *Weight Matrix*
Raw	*.seq	+	+	A raw sequence format. **See also:** *Sequence View*
SAM	*.sam	+	+	The Sequence Alignment/Map (SAM) format is a generic alignment format for storing read alignments against reference sequences. **See also:** *Assembly Browser*, *Bowtie*, *UGENE Genome Aligner*
SCF	*.scf	+	-	It is a Standard Chromatogram Format. **See also:** *Chromatogram Viewer*
SITECON	*.sitecon	+	-	A file format to store TFBS profile. **See also:** *SITECON*
Stockholm	*.sto	+	+	A multiple sequence alignments file format. **See also:** *Alignment Editor*
Swiss-Prot	*.txt *.sw	+	-	An annotated protein sequence in format of the UniProtKB/Swiss-Prot database.

				See also: *Sequence View*
Vector NTI Sequence	*.gb *.gp	+	+	A rich format for storing sequences and associated annotations, produced by Vector NTI software. **See also:** *Sequence View*
VCF	*.vcf	+	+	The VCF specifies the format of a text file used for storing gene sequence variations. **See also:** *Assembly Browser*

UGENE Native File Formats

File format	File extension	Read	Write	Comment
Dotplot	*.dpt	+	+	Stores a dotplot of a sequence. **See also:** *Dotplot*
UGENE database file	*.ugenedb	+	+	UGENE database files stores information for imported BAM or SAM files and can be used for converting this information into a SAM file. **See also:** *Import BAM/SAM File*
Short Reads FASTA	*.srfa, *.srfasta	+	+	A multiple sequence alignments file format. **See also:** *Alignment Editor*
UGENE Workflow Language	*.uwl	+	+	Human-readable format to store workflows, created in UGENE *Workflow Designer*. **See also:** *Workflow Designer*
UGENE Query Language	*.uql	+	+	Human-readable format to store schemas, created in UGENE *Query Designer*. **See also:** *Query Designer*
Workflow element for command line tool	*.etc	+	+	Format for storing workflow elements that can launch an external command line tool. **See also:** *Workflow Designer*

Other File Formats

File format / extension	Comment
***.csv**	**Example of usage:** annotations can be exported to this format; the *Weight Matrix* matrices list can also be saved to this format.
***.html**	For example it is used to store reports.
image formats: *.bmp, *.jpg, *.png, *.tiff, *.svg, etc.; *.pdf	These formats are used throughout the program to save screenshots, etc.
***.txt**	It is possible to view and modify plain text files in UGENE.