

Unipro UGENE Workflow Designer Manual

Version 1.27

August 21, 2017



Workflow Designer Manual

- 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 Command Line 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
    - File List Element
    - Read Alignment Element
    - Read Annotations Element
    - Read Assembly Element
    - Read from Remote Database Element
    - Read Plain Text Element
    - Read Sequence Element
    - Read Variations Element
  - Data Writers
    - Write Alignment Element
    - Write Annotations Element
    - Write Assembly Element
    - Write FASTA Element
    - Write Plain Text Element
    - Write Sequence Element
    - Write Variations 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
  - Local BLAST+ Search Element
  - Merge Annotations Element
  - ORF Marker Element
  - Remote BLAST 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 to Reference 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
  - Split Alignment into Sequences Element
- NGS: Align Short Reads
  - Align Reads with Bowtie Element
  - Align Reads with Bowtie2 Element
  - Align Reads with BWA Element
  - Align Reads with BWA-MEM Element
  - Align Reads with UGENE Genome Aligner Element
- NGS: Basic Functions
  - Assemble Genomes with SPAdes Element
  - 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
  - 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: RNA-Seq Analysis
  - Assembly Transcripts with Cufflinks Element
  - Extract Transcript Sequences with gffread Element
  - Find Splice Junction with TopHat Element
  - Merge Assemblies with Cuffmerge 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
  - Create VCF consensus
  - 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 Sequences by Annotation Number
    - Marking Sequences 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
    - De novo Assembly with Spades
    - Call Variants with SAMtools
    - ChIP-Seq Coverage
    - ChIP-seq Analysis with Cistrome Tools
    - Extract Consensus from Assembly
    - Extract Coverage from Assembly
    - Extract Transcript Sequences
    - Quality Control by FastQC
    - Raw ChIP-Seq Data Processing
    - Raw DNA-Seq Data Processing
    - Raw RNA-Seq Data Processing
    - RNA-seq Analysis with Tuxedo Tools
    - Get Unmappet Reads
    - Variation Annotation with SnpEff
  - 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
- Merge Sequences and Annotations
- In Silico PCR
- Remote BLASTing
- Get Amino Translations of a Sequence

- Transcriptomics
  - Search for Transcription Factor Binding Sites (TFBS) in Genomic Sequences

About the Workflow Designer

UGENE Workflow Designer is a part of UGENE genome analysis suite that allows a molecular biologist to create and run complex computational workflows even if he or she is not familiar with any programming language.

The workflows comprise reproducible, reusable and self-documented research routines, with a simple and unambiguous visual representation suitable for publications.

The workflows can be run both locally and remotely, either using graphical interface or launched from the command line.

The elements that a workflow consists of corresponds to the bulk of algorithms integrated into UGENE. Additionally you can create custom workflow elements.

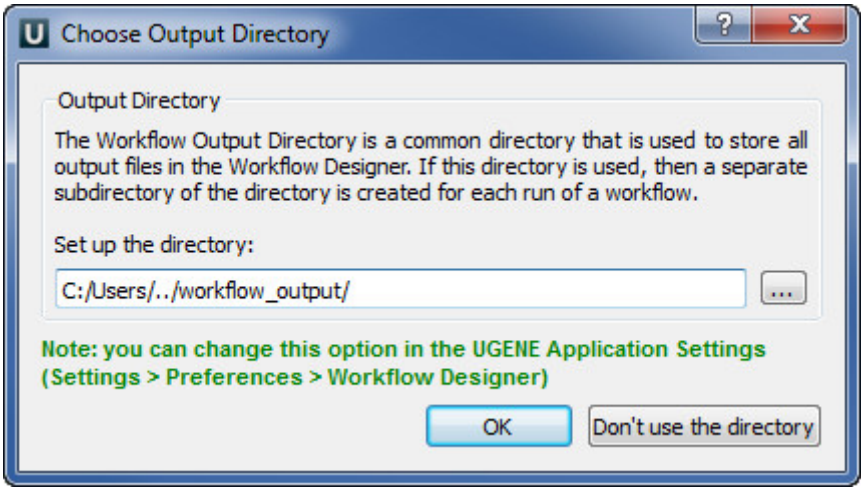
Introduction

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

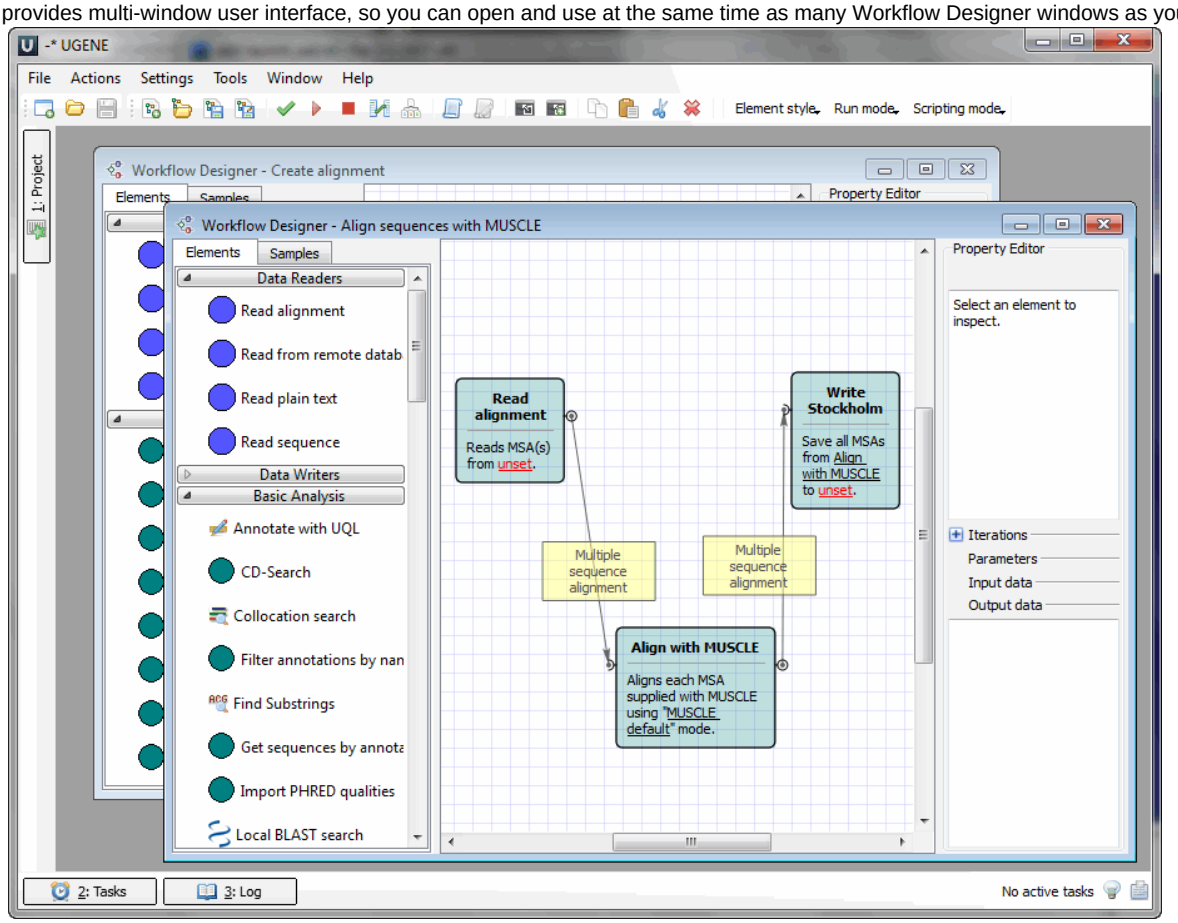
Launching Workflow Designer

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*.

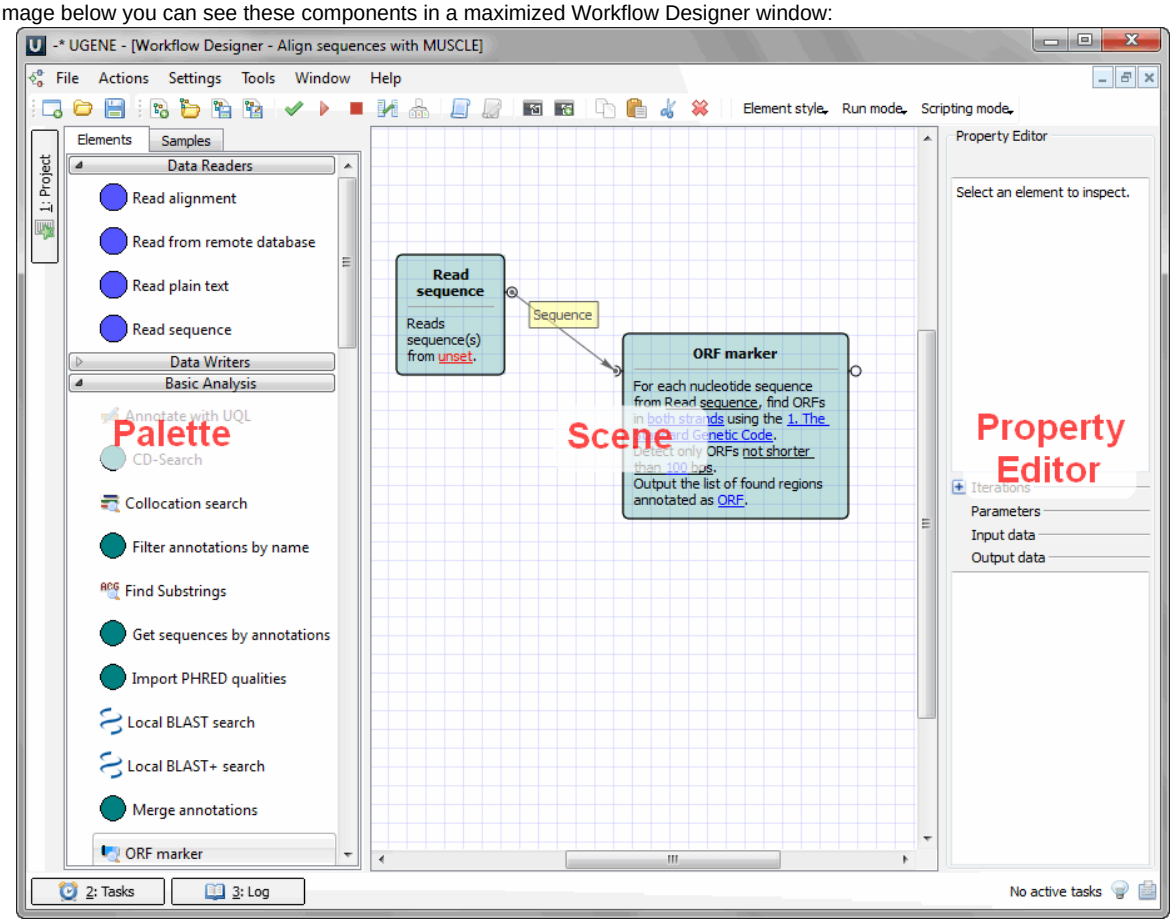
Scene

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.

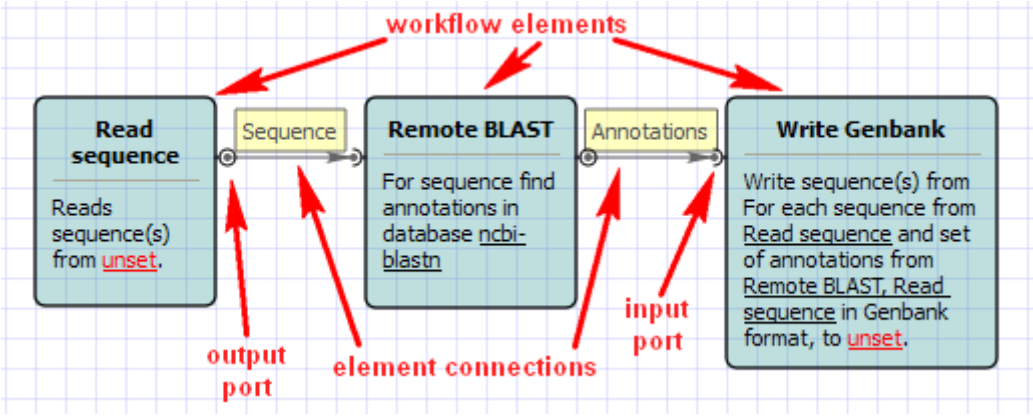
In a workflow, an element usually have access to slots of the connected elements, located in the workflow before it.

message

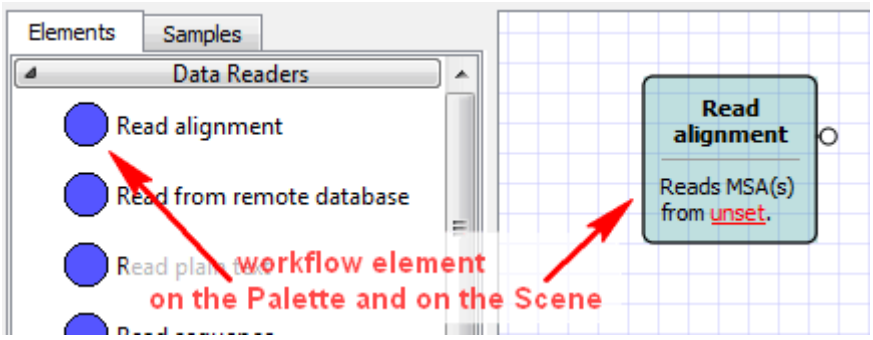
A message is a single data chunk, transferred from an output slot of one element to an input slot of another element. The slots must have the same type to make the transfer possible.

The Scene is initially empty and you start with creating a workflow on it:

See an example of a workflow on the image below:

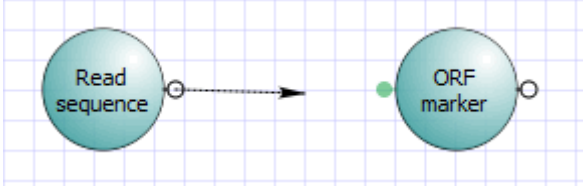


Your first step is to *add* necessary workflow elements, for example, by dragging them from the *Palette* to the Scene:



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.

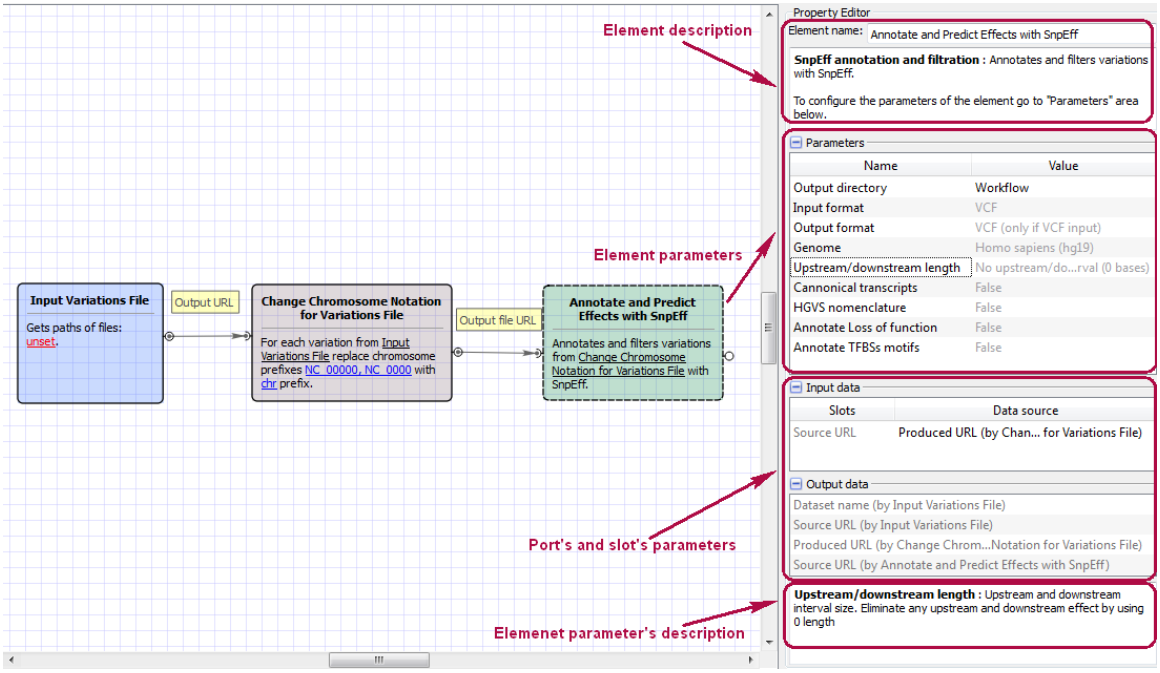
Once created, a connection will follow movements of the linked elements; you cannot redirect or reshape the connection arrow but only

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* ports, 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 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

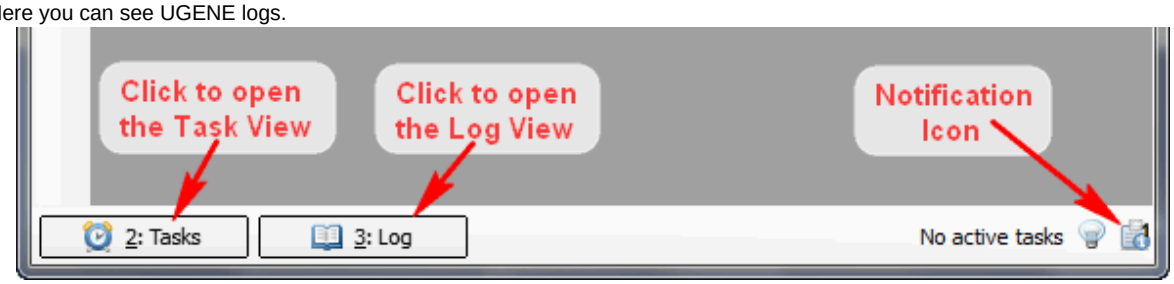
Here you can see the tasks currently executed in UGENE.

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 *Notificat ion Icon*.

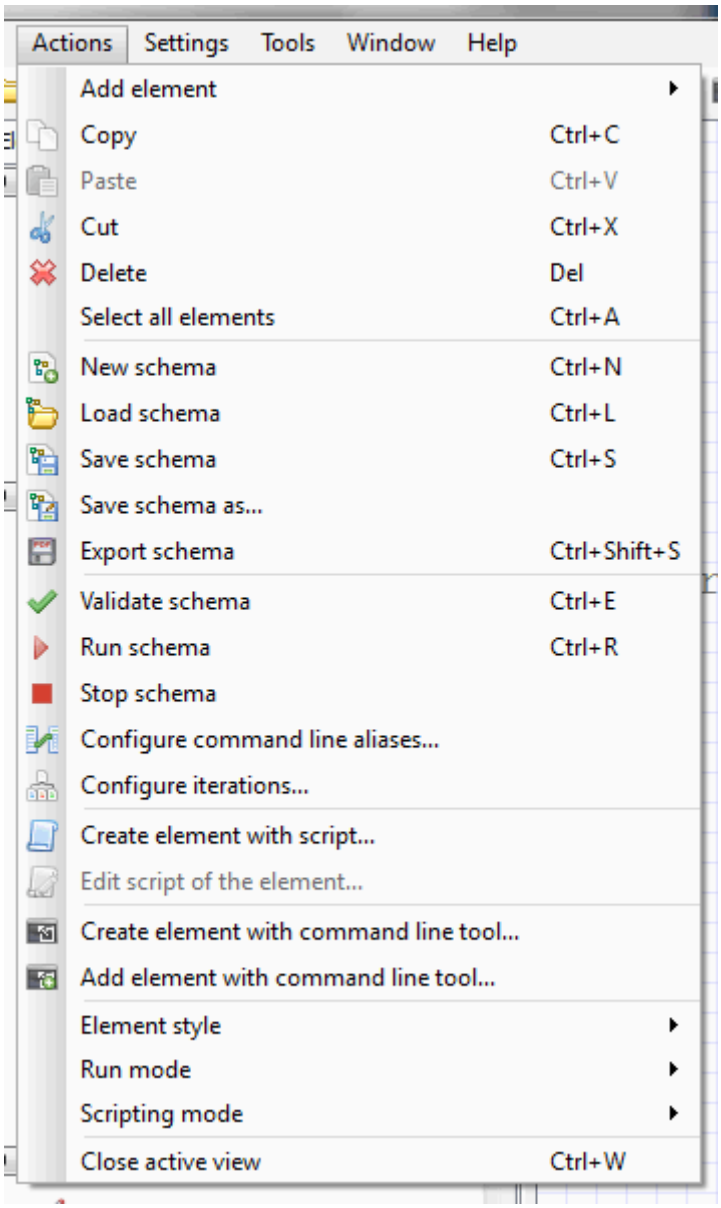
Log View

Here you can see UGENE logs.



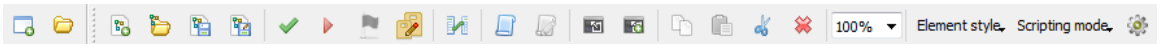
Actions Menu

When a Workflow Designer window is active, all standard actions to work with workflow are available from the *Action* main menu:



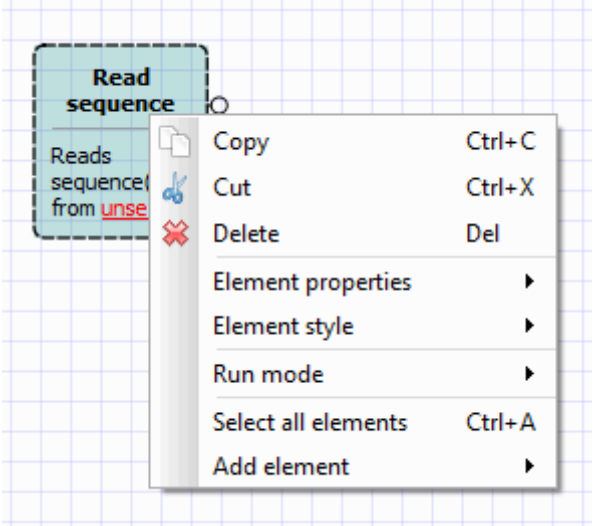
Toolbar

Most common actions are available on the main toolbar:



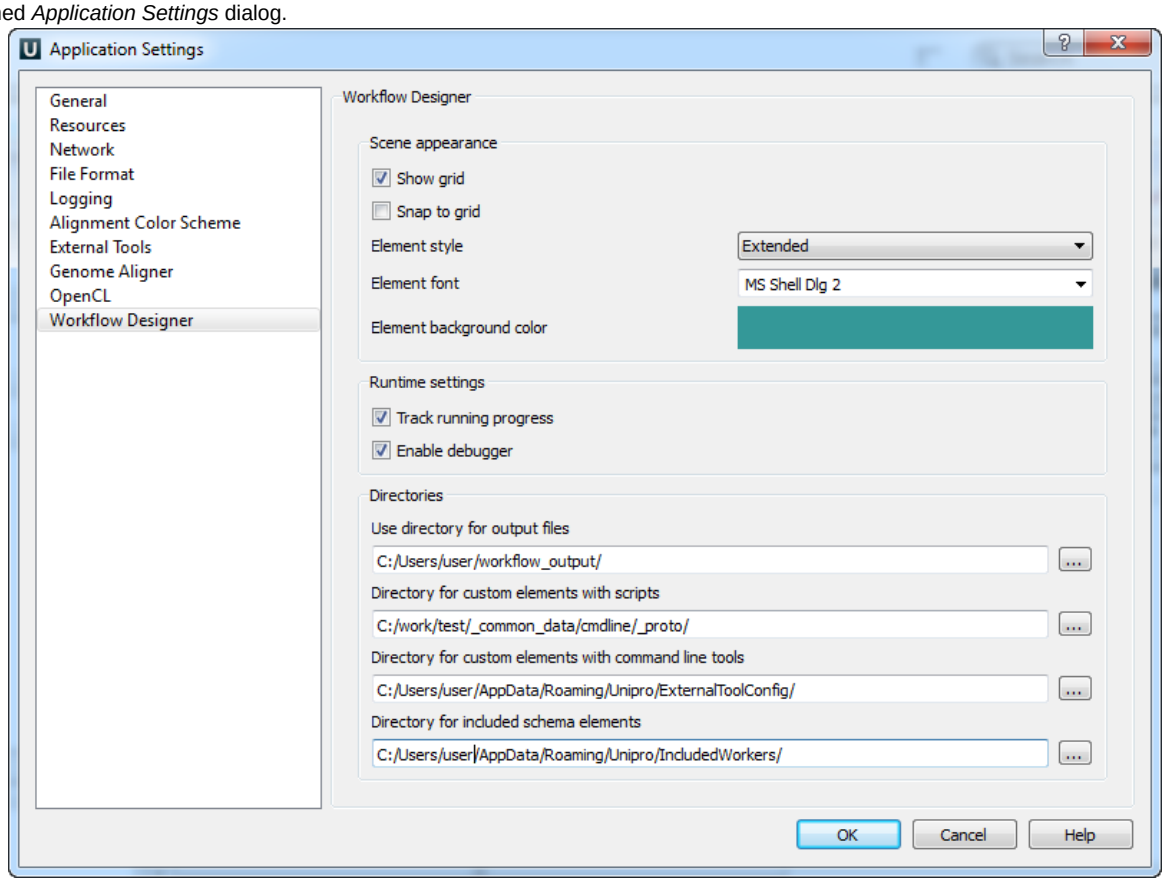
Context Menus

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.



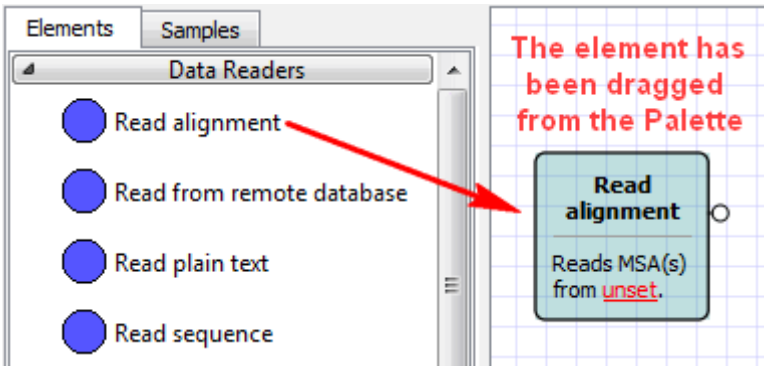
How to Create and Run Workflow

- Select *Tools –> Workflow Designer* or *File->New worflow* items in the main menu.

  Result: The Workflow Designer window appears.

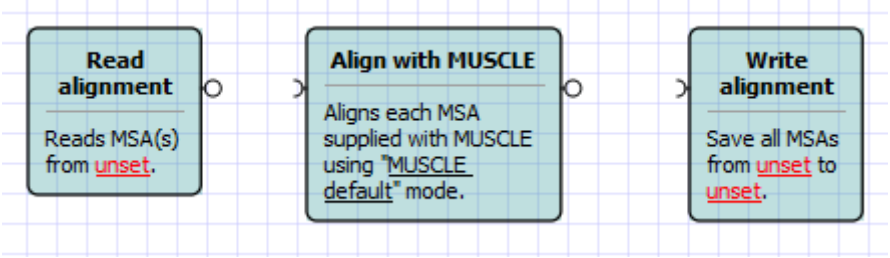
- On the *Elements* tab of the *Palette* find the *Read alignment* element. It is located in the *Data sources* group and drag it to the *Scene*.

  Result: The element is shown on the Scene.



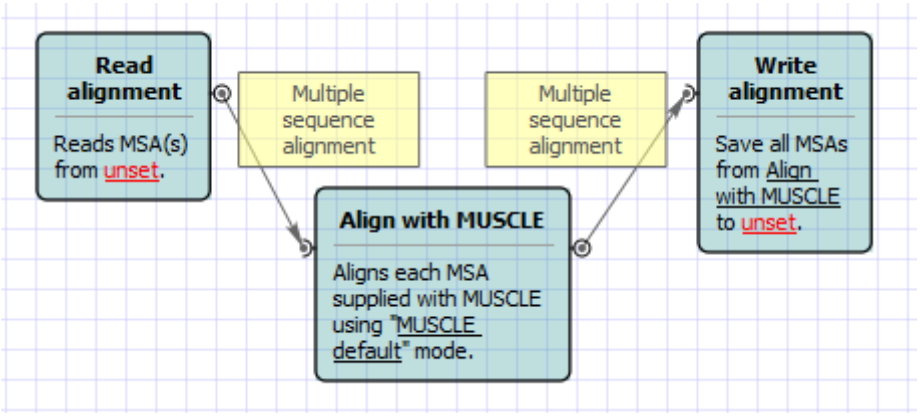
- Repeat the previous step for the *Write Alignment* element from the *Data sinks* group and for the *Align with MUSCLE* element from the *Multiple sequence alignment* group.

  Result: All three elements are on the Scene.

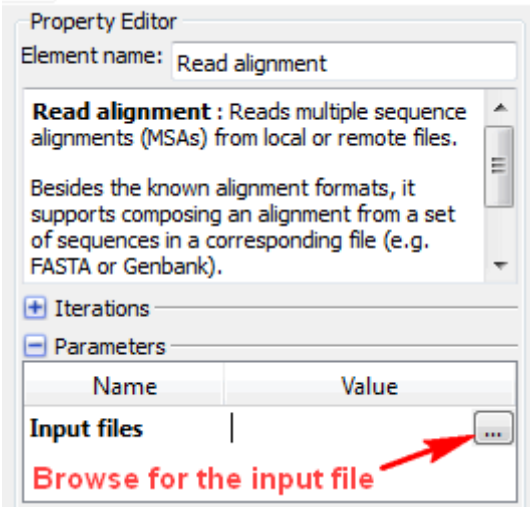


- Connect the elements:
  - Drag an arrow from the *output port* of the *Read alignment* element to the *Align with MUSCLE* element.
  - Drag an arrow from the output port of the *Align with MUSCLE* element to the *Write alignment* element.

  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* param eter:



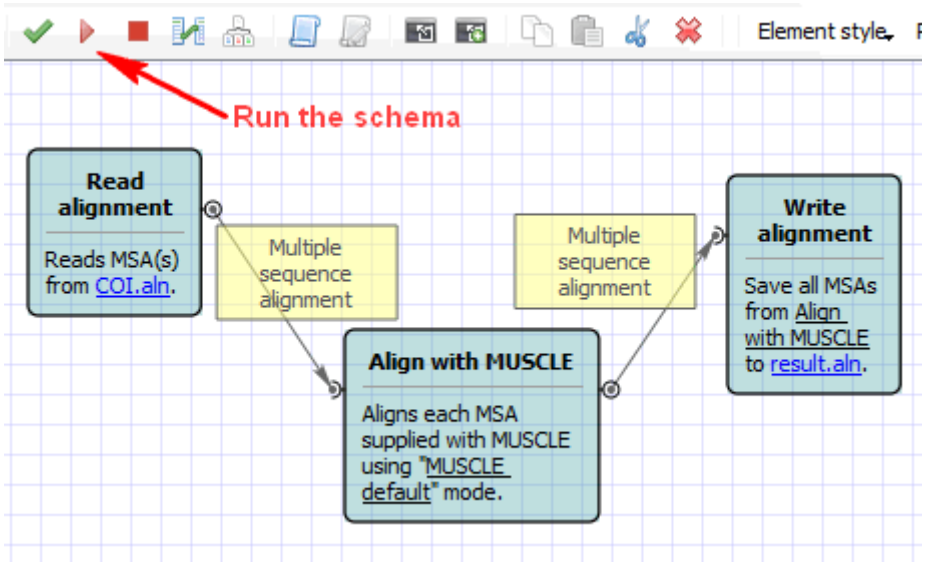
- And browse for an input file, e.g.Select the $UGENE\data\samples\CLUSTALW\COI.aln file.

  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

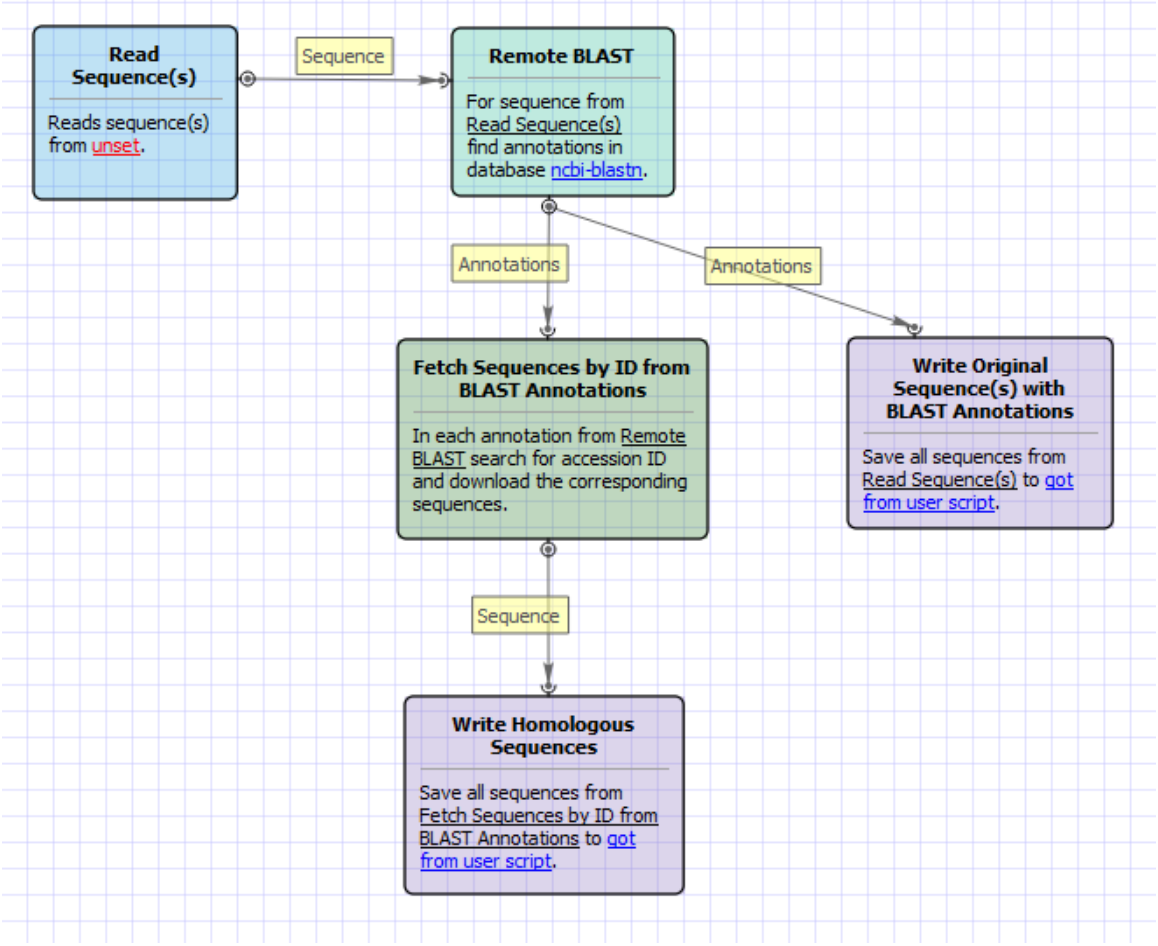
UGENE Workflow Designer contains a set of sample workflows that help a biologist to solve certain tasks for multiple input files or datasets at the same time. The list of samples can be found in the Workflow Samples section of the documentation.

To use a sample:

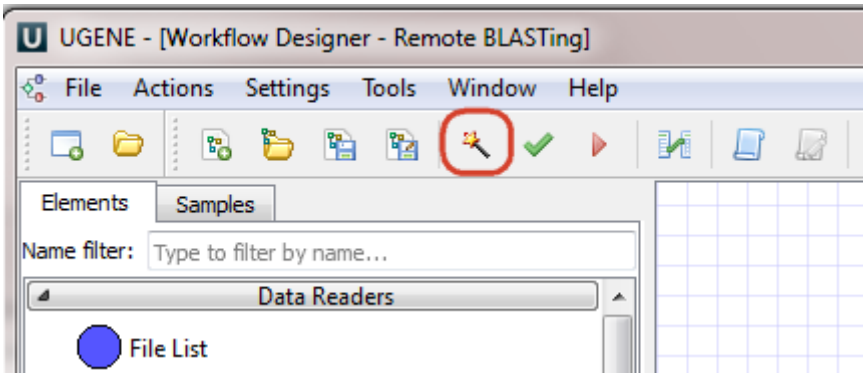
1. Start the Workflow Designer by selecting "Tools > Workflow Designer" in the main menu of the UGENE window.
   See also: the paragraph about launching the Workflow Designer.

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.

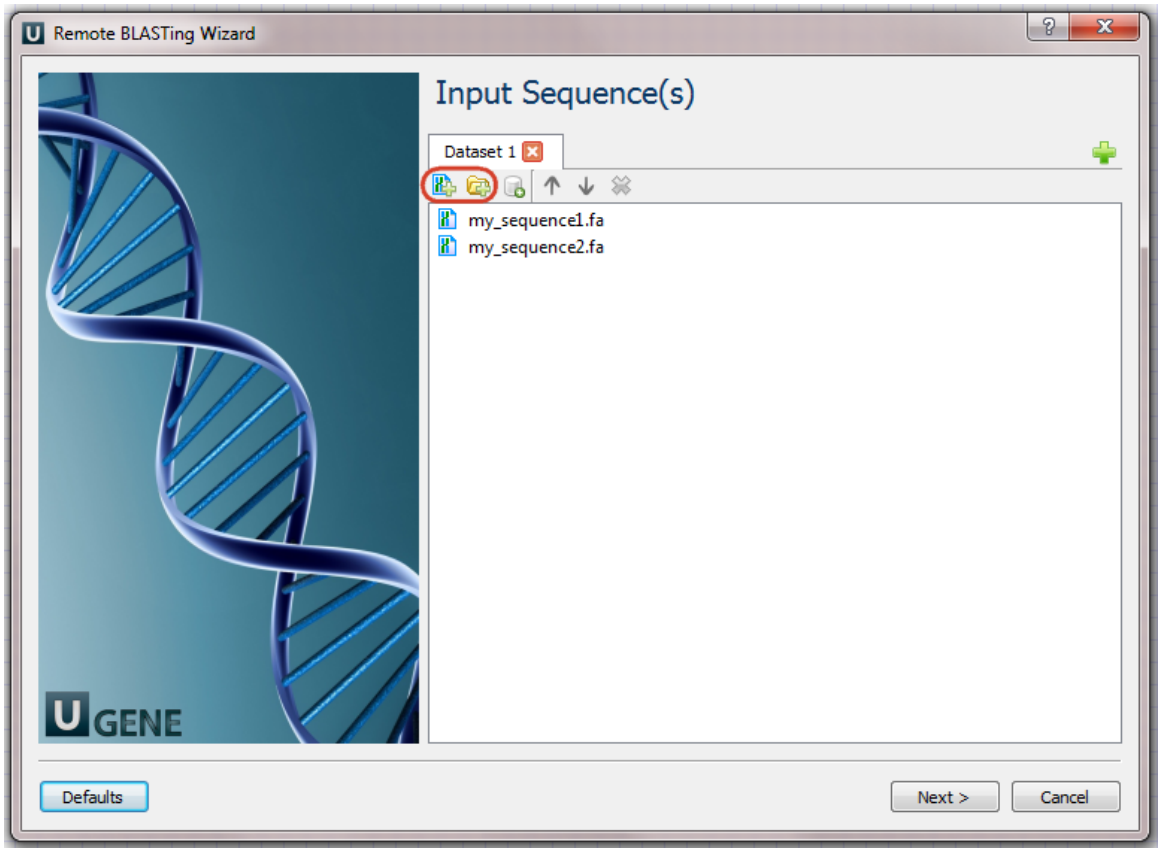
3. Double-click on the required sample.
   The workflow will be opened and shown on the Workflow Designer scene, i.e. the center area of the window.
   For example, a workflow for doing BLAST and getting the results from the NCBI server is shown below.



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.

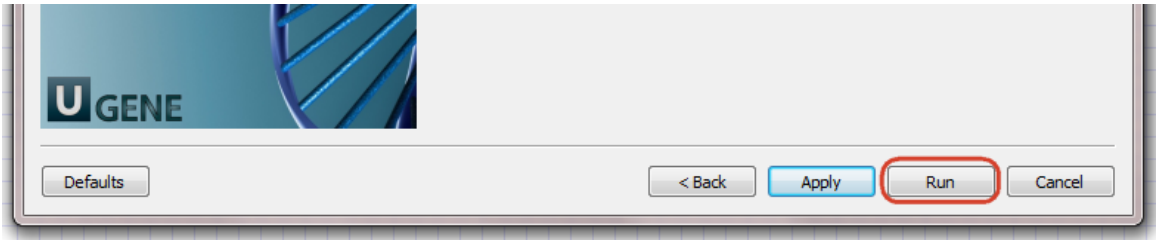


5. Input the required data. The input varies very much on the workflow that has been selected on step 3 (see above).
   For example, in case of the remote BLAST workflow, at least one sequence is expected to be input. On the image below two sequences were input for the workflow. Buttons that can be used to add different files or even folders with files are also marked on the image.

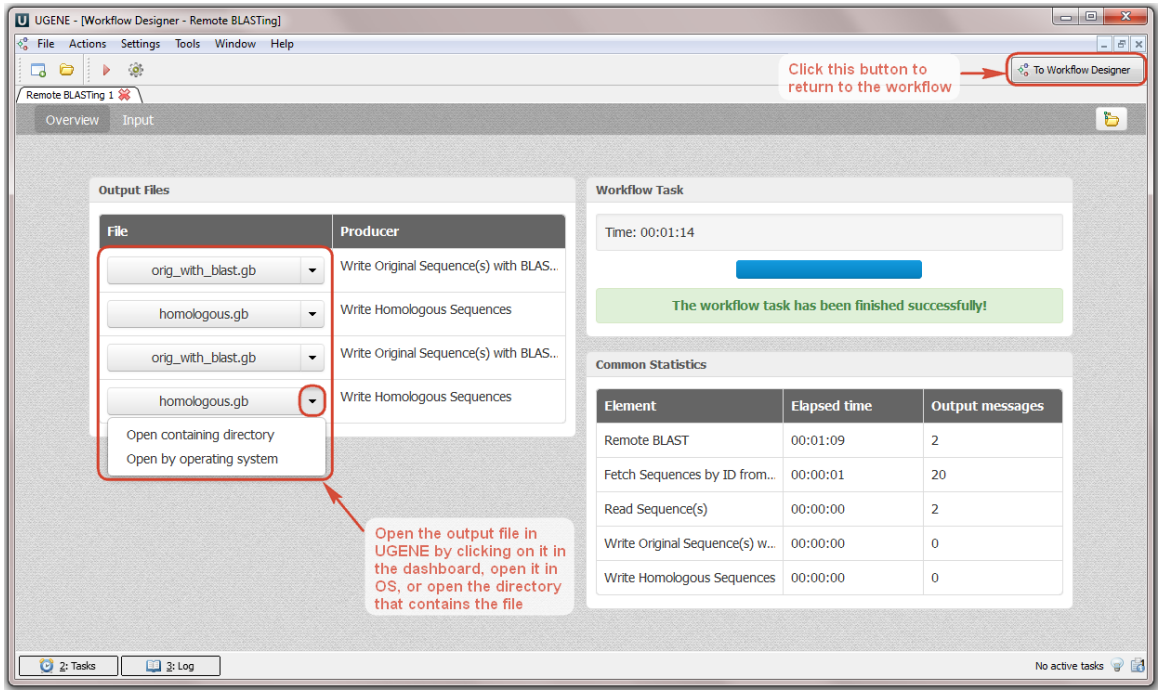


6. Optionally, modify the workflow parameters on other pages of the wizard.

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:


UGENE - [Workflow Designer - Remote BLASTing]
File Actions Settings Tools Window Help
Remote BLASTing 1
Overview Input
Click this button to return to the workflow
To Workflow Designer
Output Files
File
Producer
orig_with_blast.gb
Write Original Sequence(s) with BLAS...
homologous.gb
Write Homologous Sequences
orig_with_blast.gb
Write Original Sequence(s) with BLAS...
homologous.gb
Write Homologous Sequences
Open containing directory
Open by operating system
Open the output file in UGENE by clicking on it in the dashboard, open it in OS, or open the directory that contains the file
Workflow Task
Time: 00:01:14
The workflow task has been finished successfully!
Common Statistics
Element
Elapsed time
Output messages
Remote BLAST
00:01:09
2
Fetch Sequences by ID from...
00:00:01
20
Read Sequence(s)
00:00:00
2
Write Original Sequence(s) w...
00:00:00
0
Write Homologous Sequences
00:00:00
0
2: Tasks
3: Log
No active tasks

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.

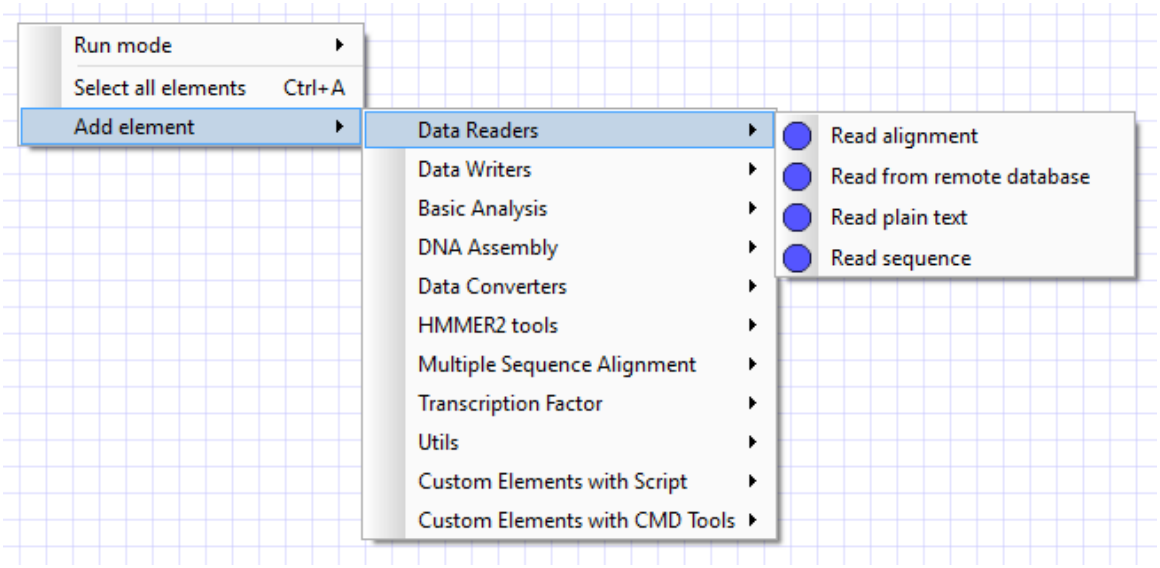
- Adding Element
- Copying Element
- Pasting Element
- Cutting Element
- Deleting Element
- Selecting All Elements on Scene

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.

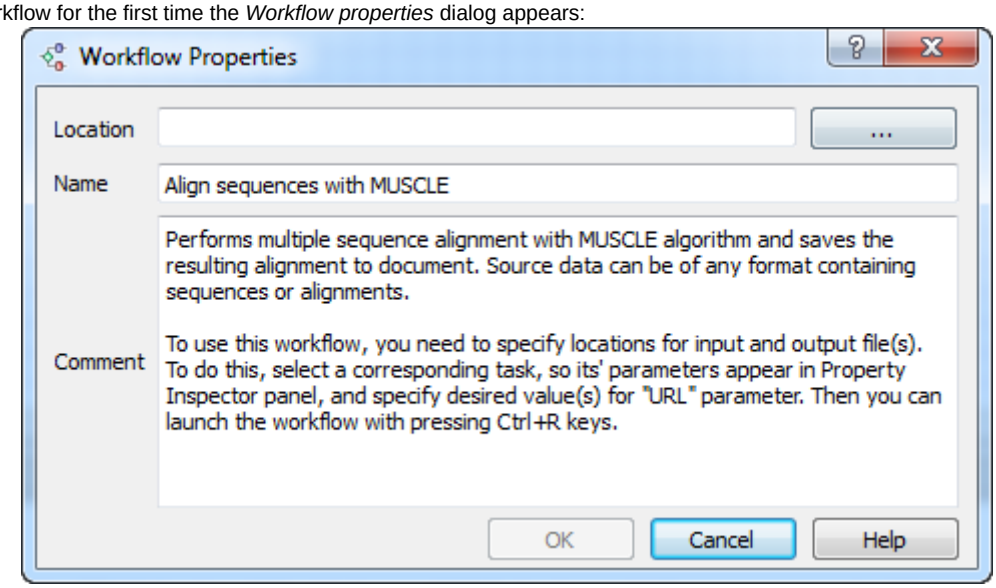
Saving Workflow

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:

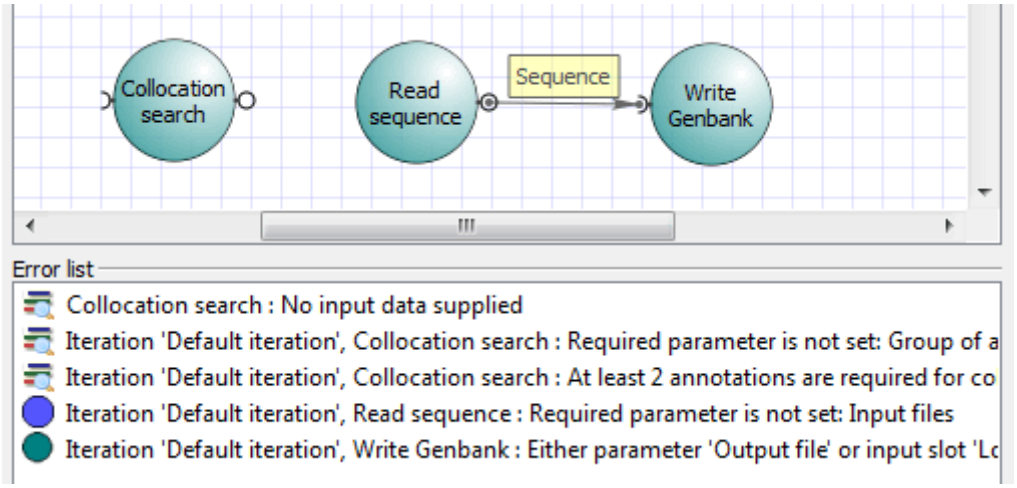
- Raster image (*.png, *.bmp, *.jpg, *.jpeg, *.ppm, *.xbm, *.xpm)
- Vector image (*.svg)
- Portable document (*.pdf, *.ps)

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.

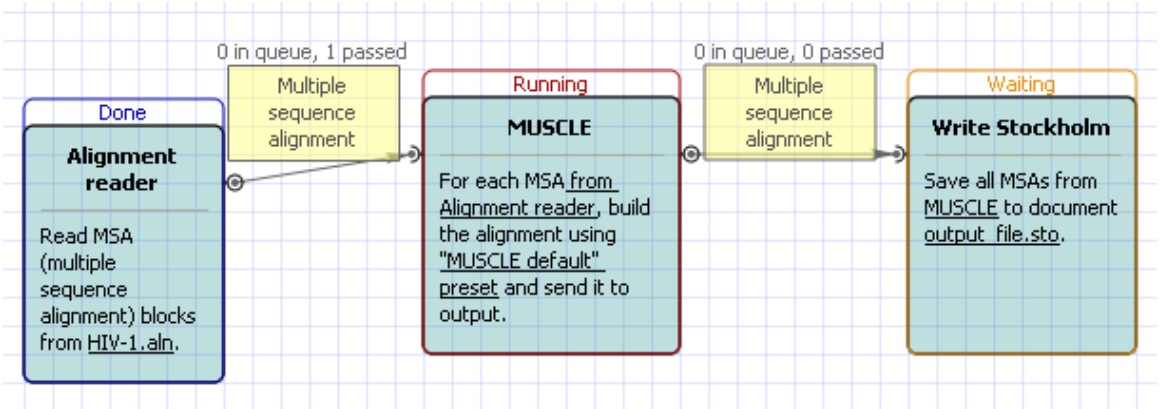


Double-clicking on items in the list selects the faulty element/iteration.

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 *Actions 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:

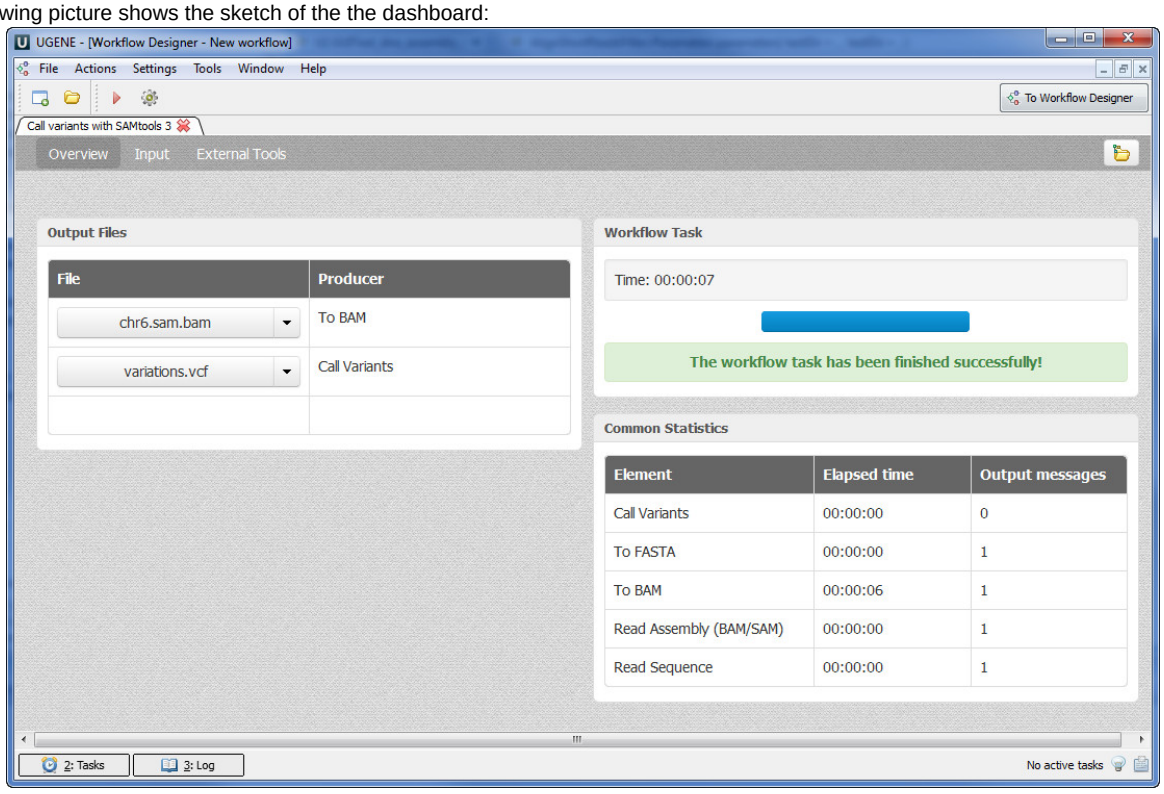
- Overview tab

- 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

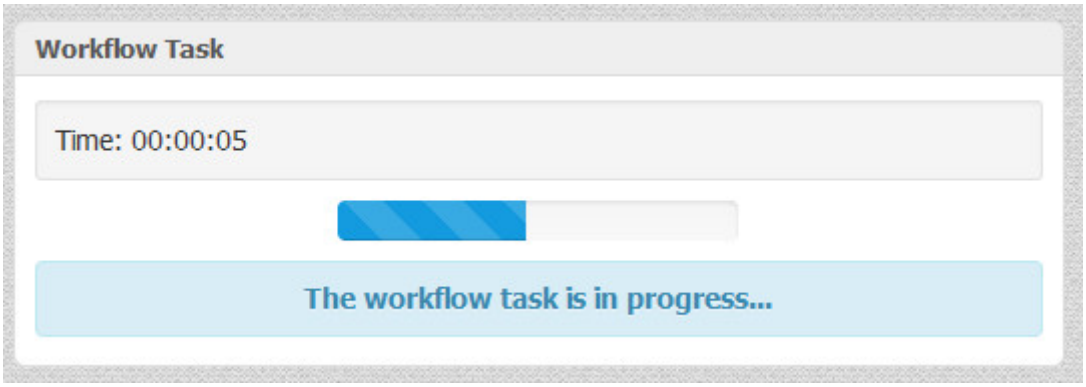
Dashboard Window Components

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.;

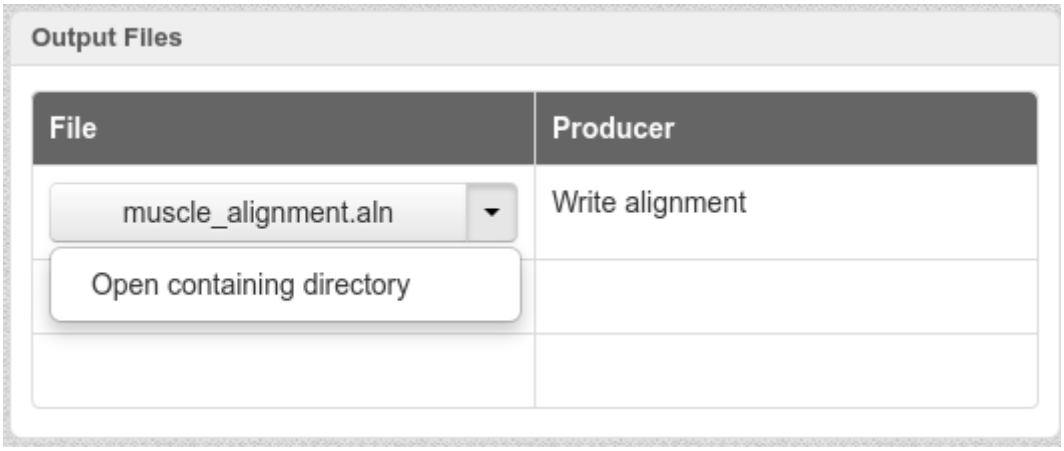


"Output Files" widget

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

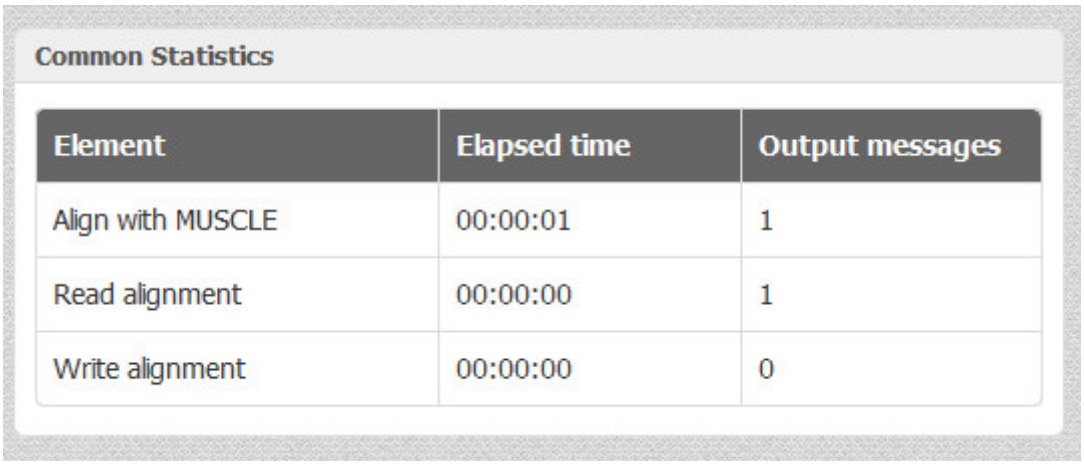
file by operating system);
- the name of the workflow element that has produced the file;



"Common Statistics" widget

It contains a table with common statistic for each workflow element in the workflow. The table columns are:

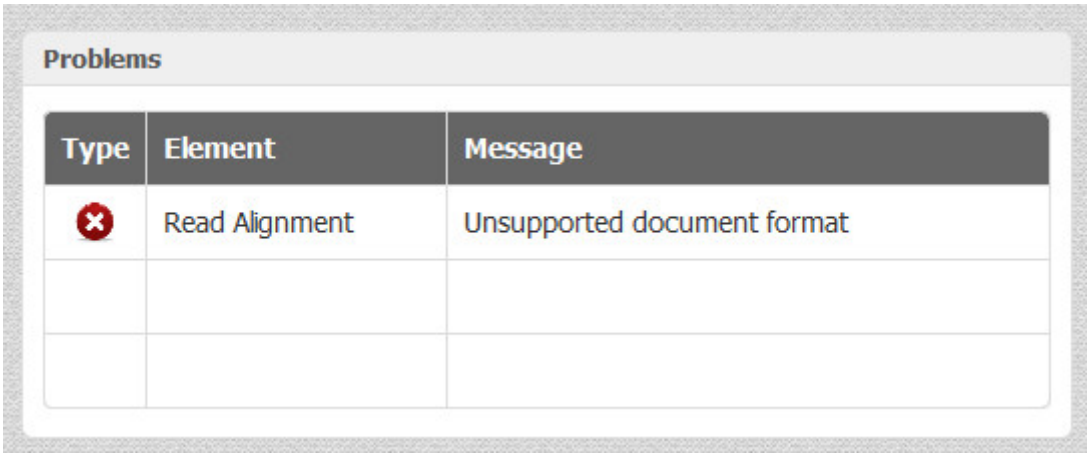
- name of the workflow element;
- time of the workflow element execution;
- the number of messages that has been retrieved;



"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

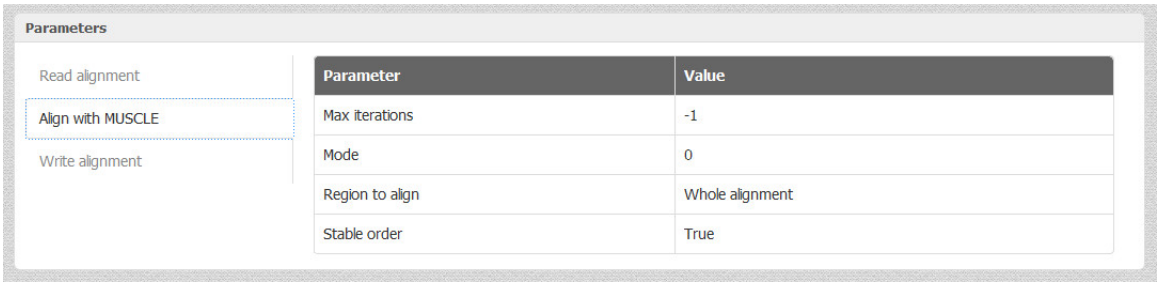


Input tab

"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);

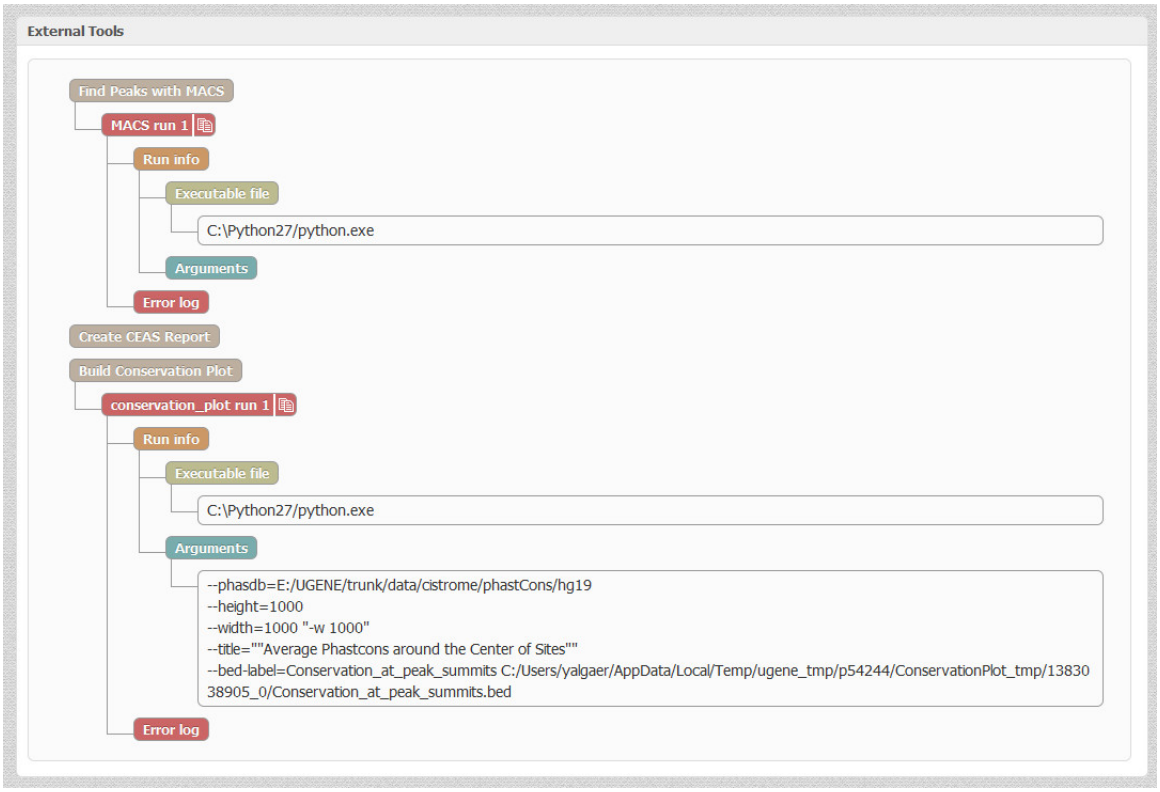


External Tools tab

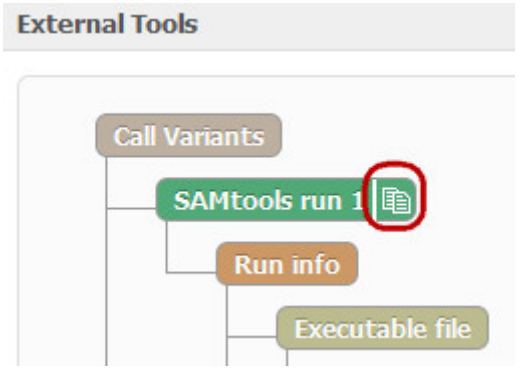
"External Tools" widget

It contains information about external tools. There are:

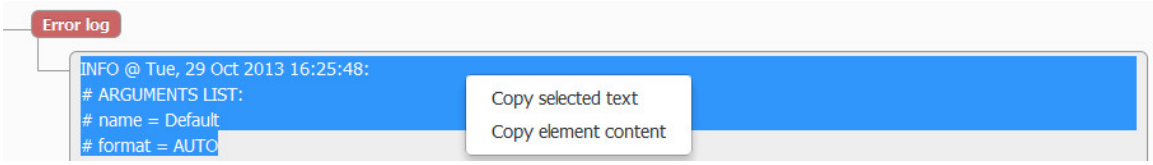
- names of the external tools;
- executable file of the external tool;
- arguments of the external tool;



To copy external tool run string click the following button:

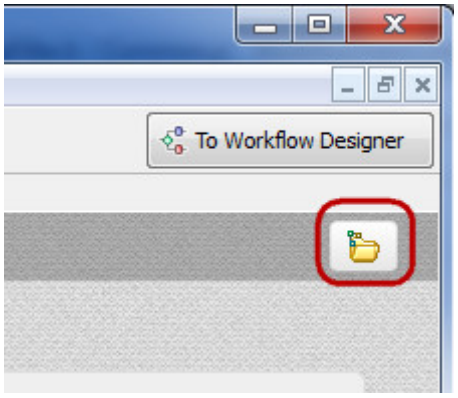


With a help of the context menu of this widget you can copy selected text from the dashboard or copy all text of the active element:

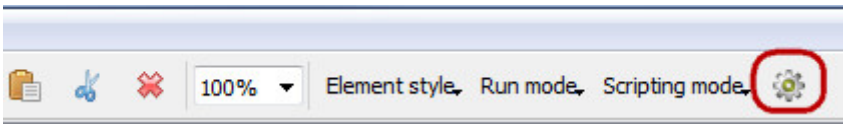


Using Dashboard

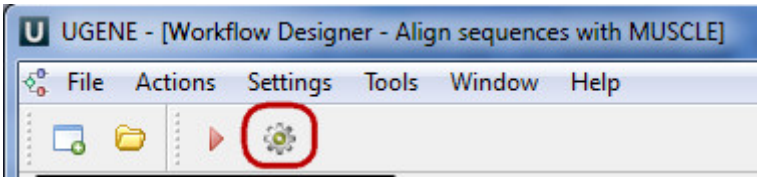
For each workflow which has been runned new dashboard will be opened. This dashboards will be saved in the *selected directory*. Also you will see this dashboard after UGENE will be runned again. Furthermore you can open the original workflow for your results by clicking on this button:



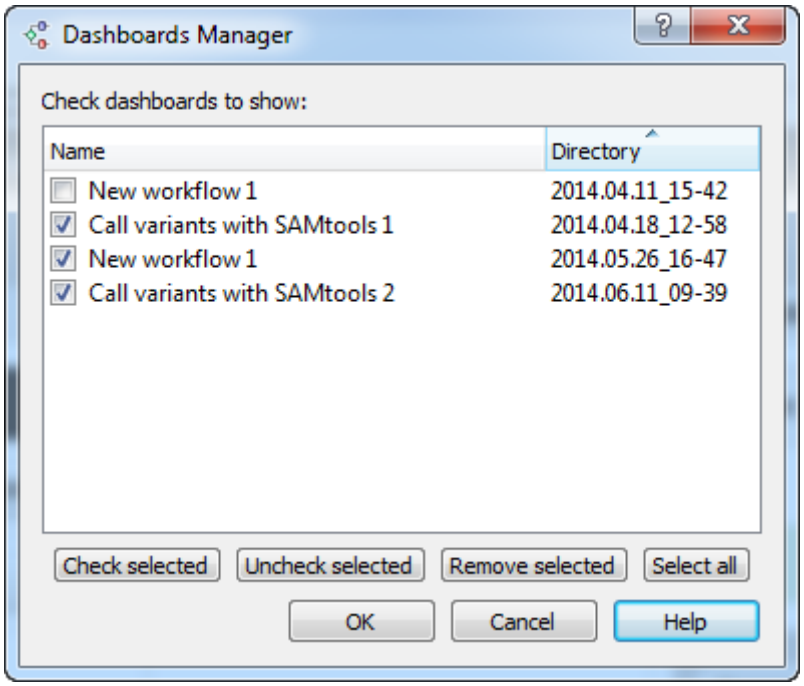
To remove or to load a dashboard click to the *Dashboards manager* button on the *Workflow Designer* main toolbar:



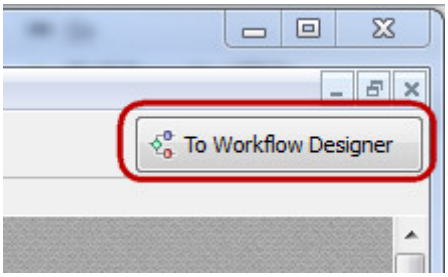
or on the *Dashboard* toolbar:



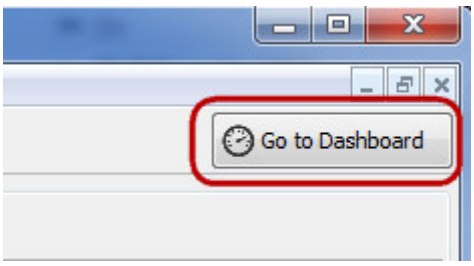
The following dialog appears:



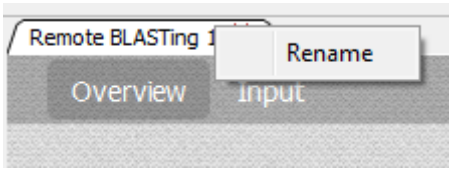
To see a dashboard select it and check it's checkbox. To remove a dashboard select it and click the *Remove selected* button. Click OK button. The selected and checked dashboards appears in the *Dashboard* main window. You can go back to the *Workflow Designer* main window from *Dashboard* window by clicking on this button:



And go back to the *Dashboard* main window from *Workflow Designer* main window by clicking on this button:

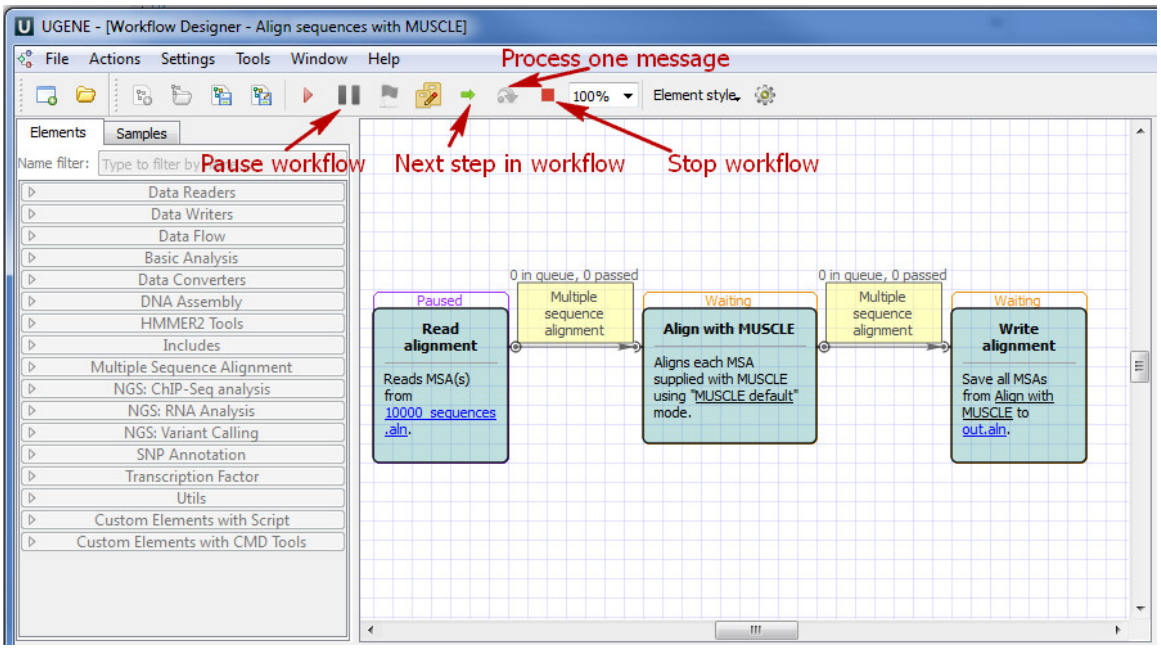


To rename a dashboard tab use the following context menu:



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.

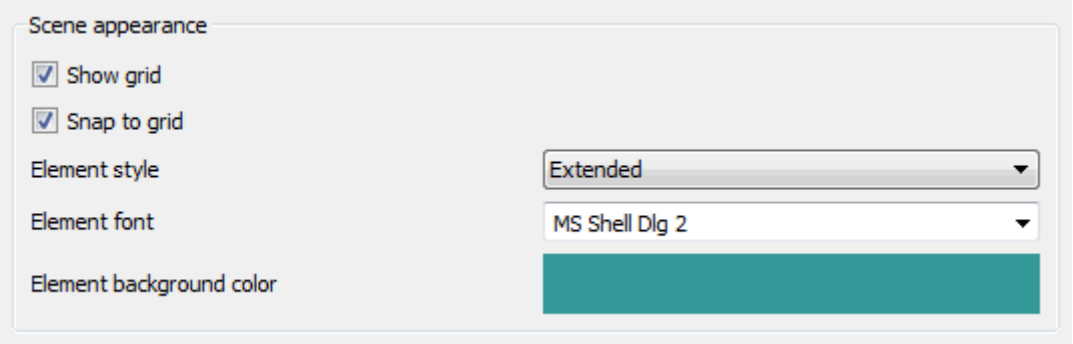
Next step in workflow - do the next step in workflow.

Process one message - do the first queue message step of the selected element in workflow. It is active if an element selected.

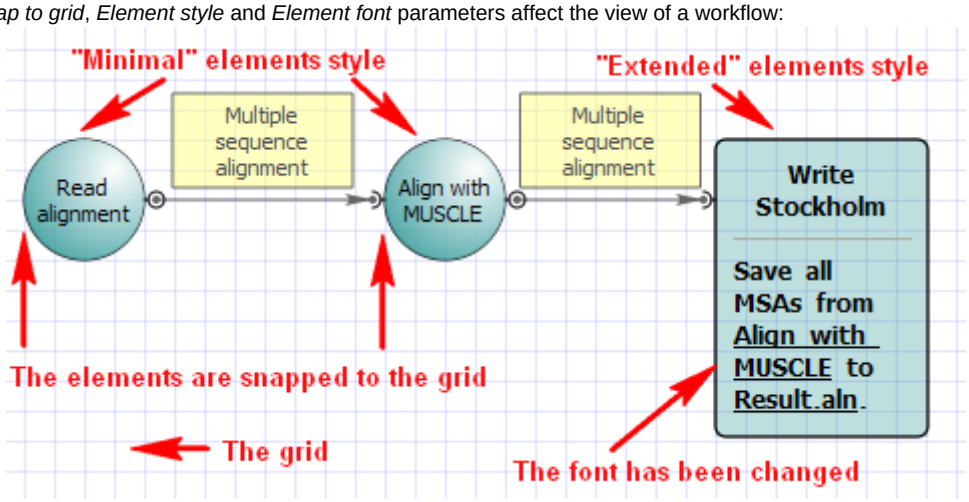
Stop workflow - cancel workflow process.

Changing Appearance

Default setting that influence the Workflow Designer appearance can be set in the *Application Settings* dialog. The parameters are shown on the image below:

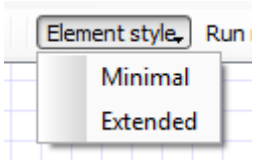


The *Show grid*, *Snap to grid*, *Element style* and *Element font* parameters affect the view of a workflow:



To change an appearance of a particular element use it's context menu submenus *Item properties* and *Item style*.

Another way to change an element style is to use the *Item style* submenu in the toolbar.



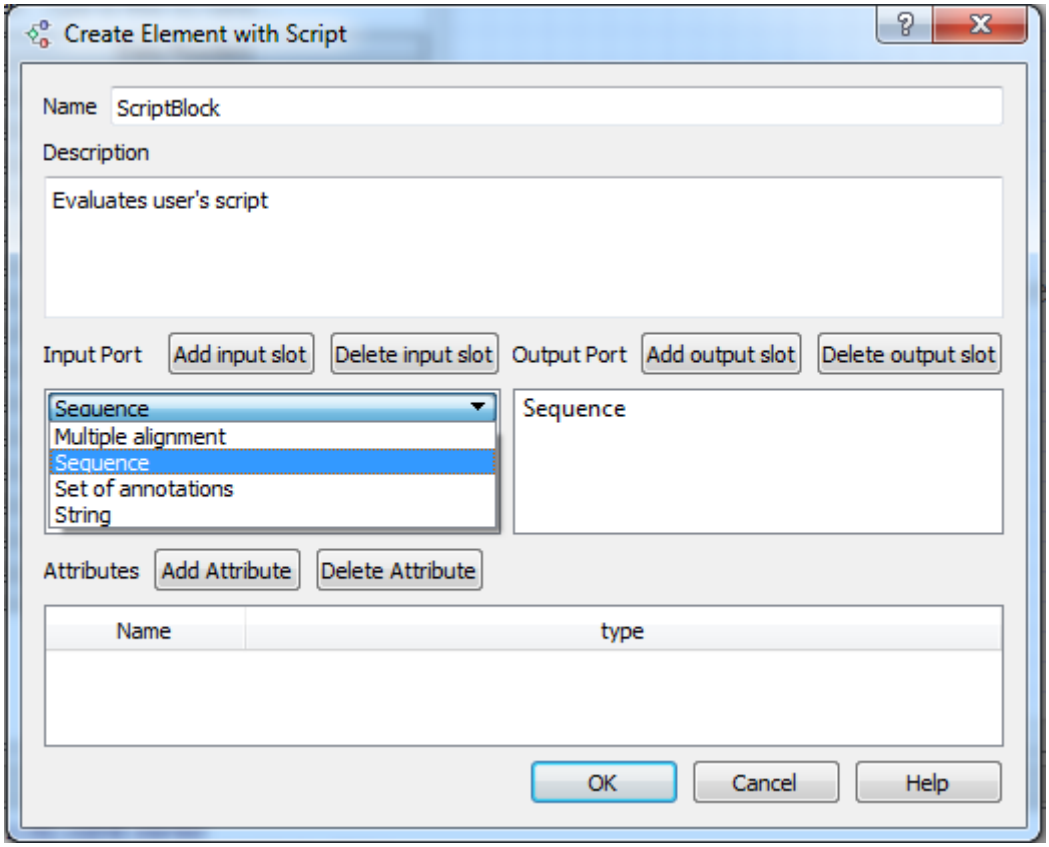
Custom Elements with Scripts

It is possible to create custom algorithmic blocks using scripts in the Workflow Designer.

To create an element either select *Actions Create Script Object* in the main menu, select *Create element with script* in the context menu or click on the following button on the toolbar:



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 alignment
- 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.

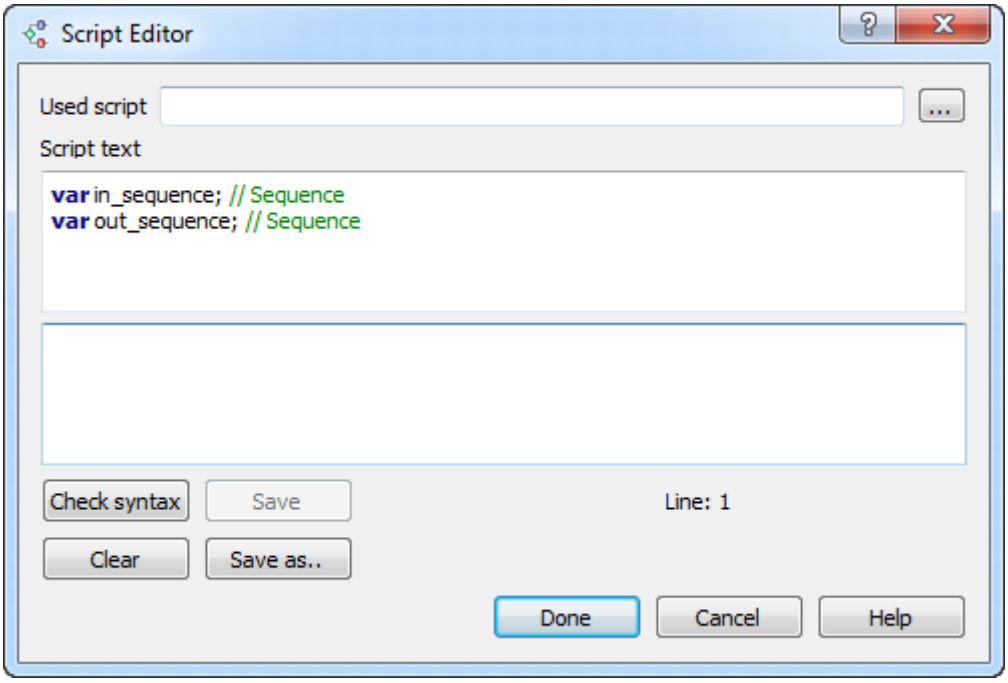
The element also becomes available in the *Custom Elements with Scripts* group on the *Palette*.

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, use the *Edit script of the element* item in the context menu or click on the following button on the toolbar:



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 Command Line Tools

In UGENE you can create a custom workflow *element* that would launch any command line tool.

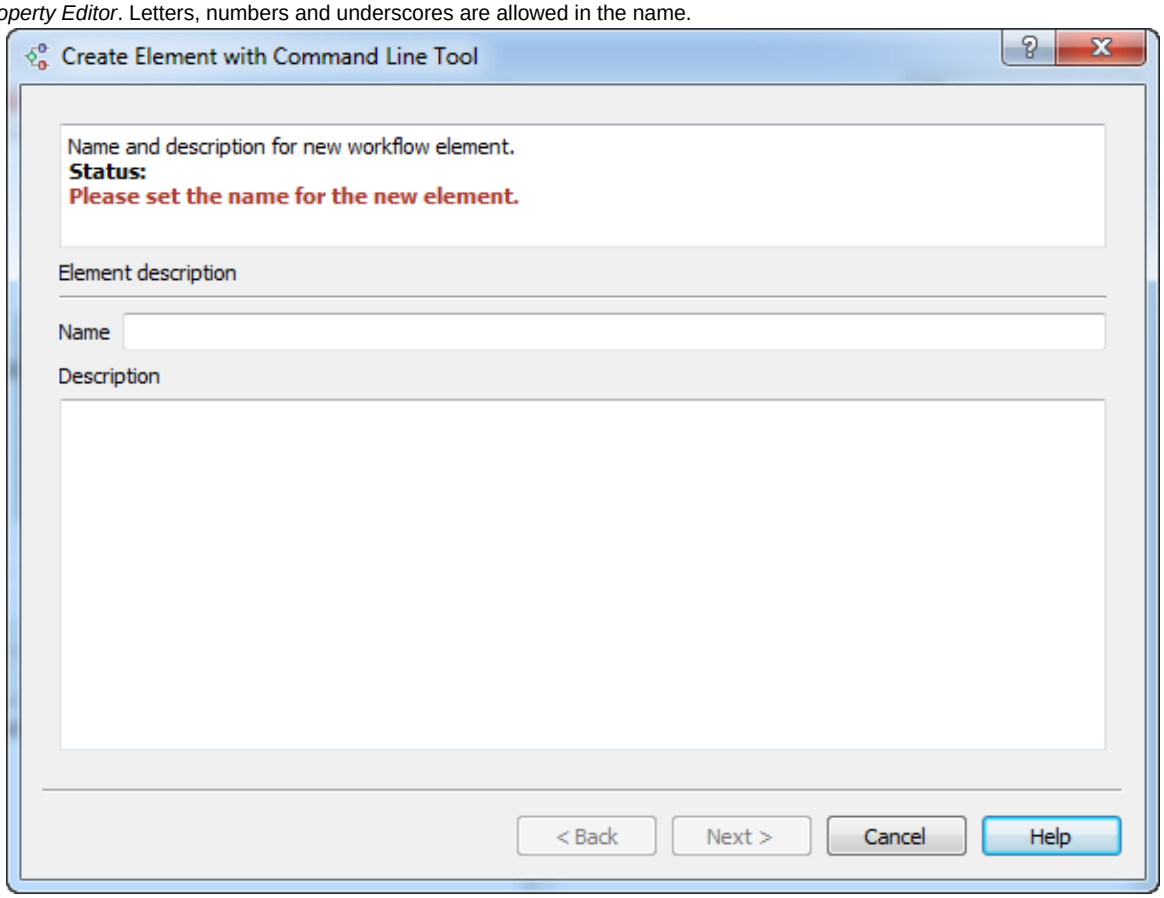
- Creating Element
- Editing Element
- Adding Existent Element
- Removing Element

Creating Element

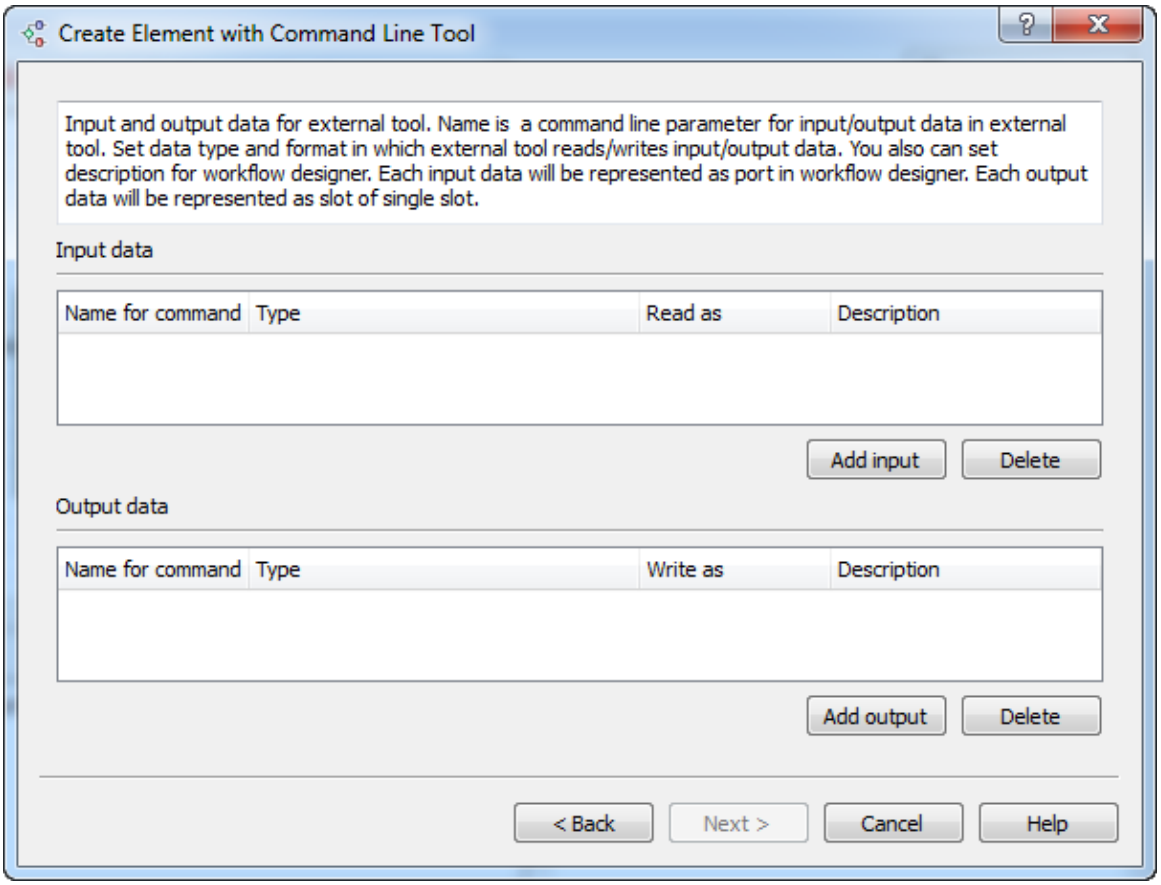
To create an element for a command line tool select either *Actions* *Create element with command line tool* in the main menu or the following icon on the toolbar:



The *Create Element with Command Line Tool* wizard appears. On the first page of the wizard input a name and a description of the element in the *Property Editor*. Letters, numbers and underscores are allowed in the name.



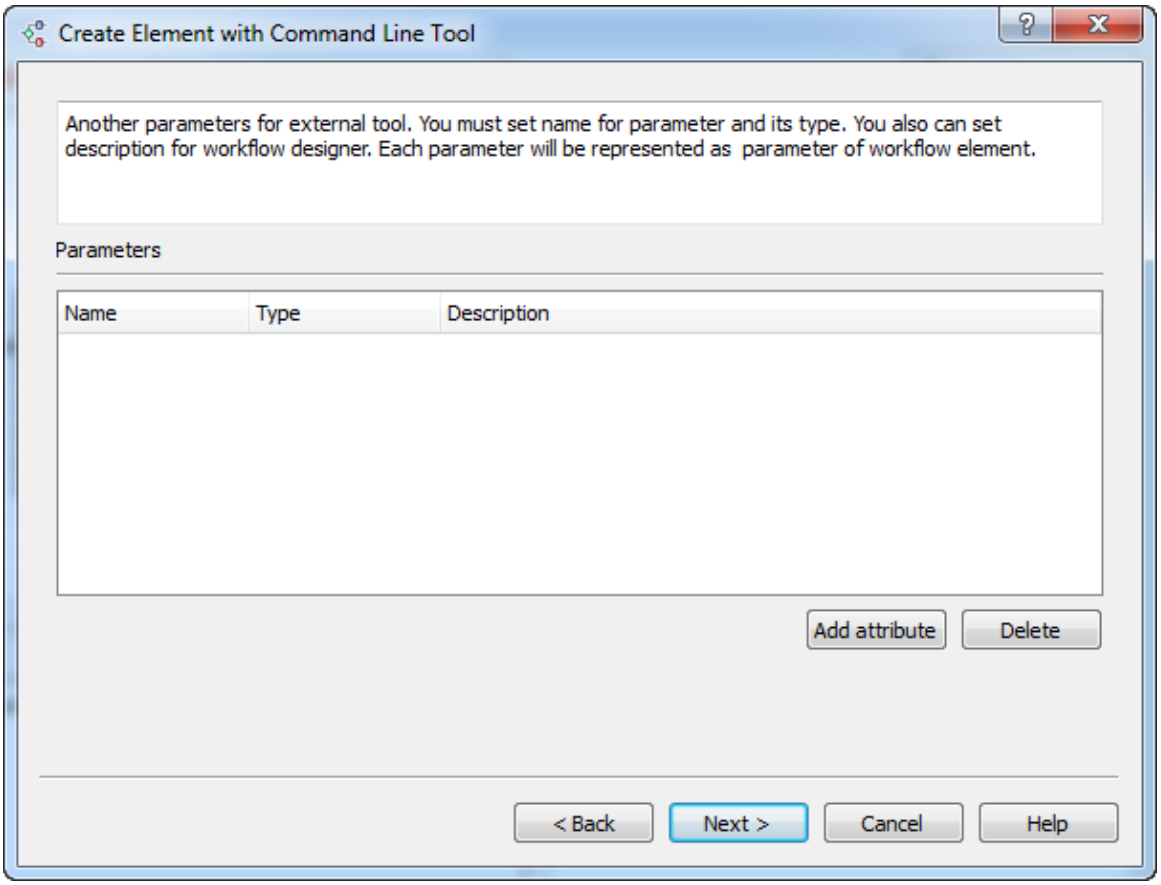
On the second page add the required input and output data:



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.
- Specify how the input or output should be handled (for example, you can specify that a value of the input parameter should be handled as a FASTA file).
- Optionally input a description.

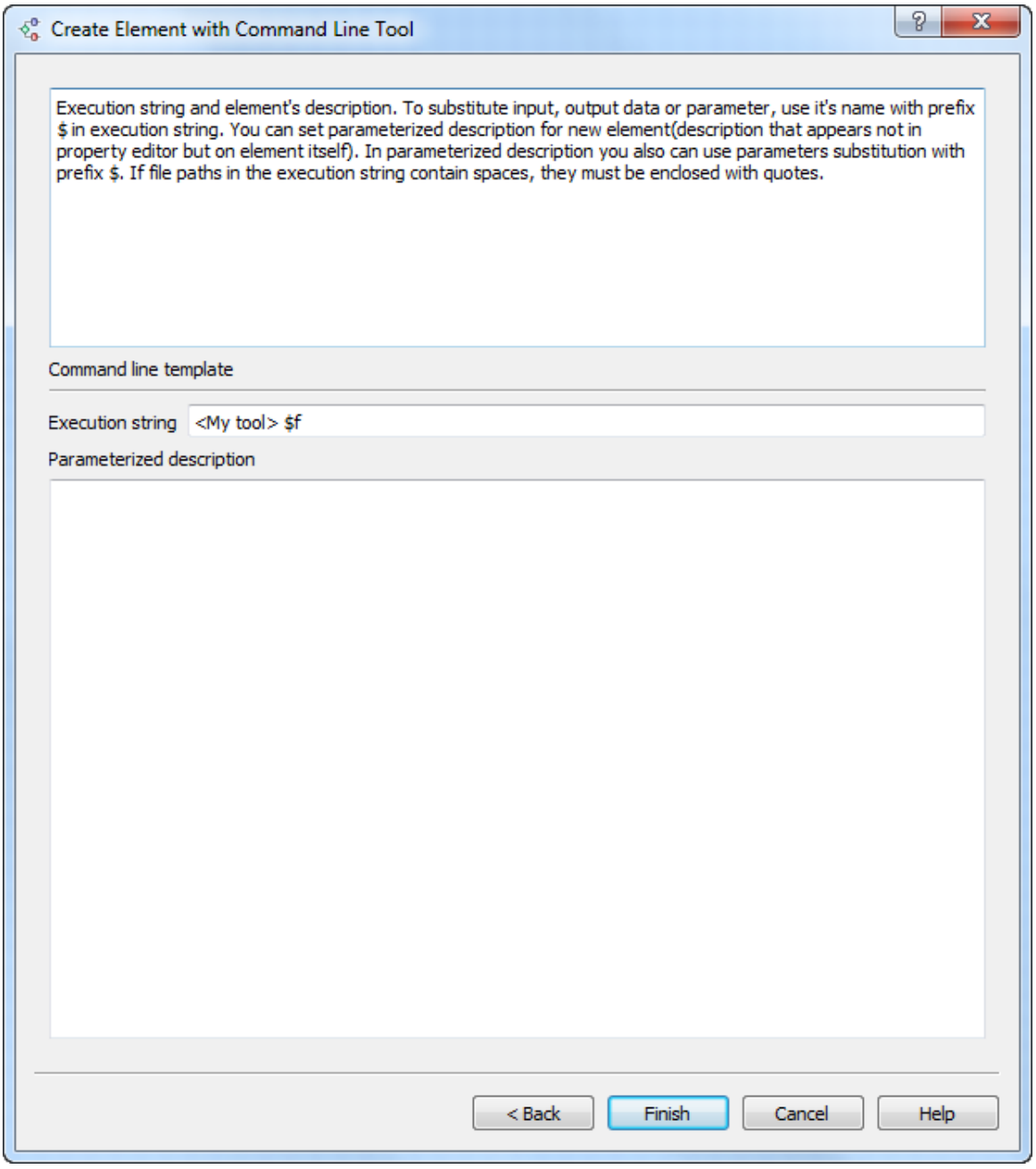
On the third page of the wizard you can add attributes for the command line tool. Later you would be able to set values for the attributes in the Property Editor, i.e. the attributes are actually the parameters of the new element.



For each attribute added you should:

- Input a name (letters, numbers and underscores are allowed in the name).
- Select it’s type: boolean, number, string or URL.
- Optionally input the description.

On the last 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 parameterized description for new element (description that appears not in property editor but on element itself). In 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.

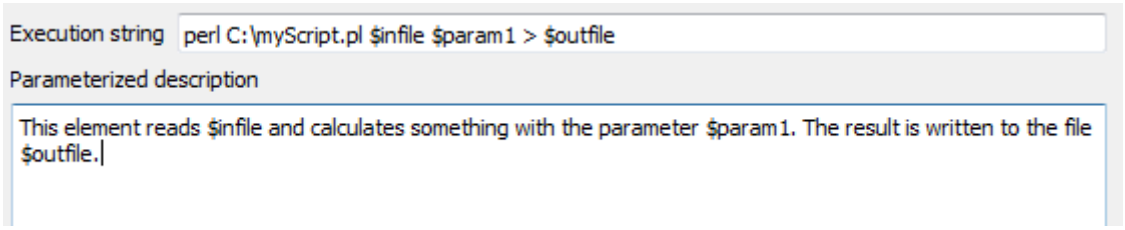
For example? let there be some perl script “myScript.pl” that accepts an input file as the first attribute and accepts the second attribute denoted as “param1”. The command may look as follows:

```
perl [path_to_script]myScript.pl $infile $param1 > $outfile
```

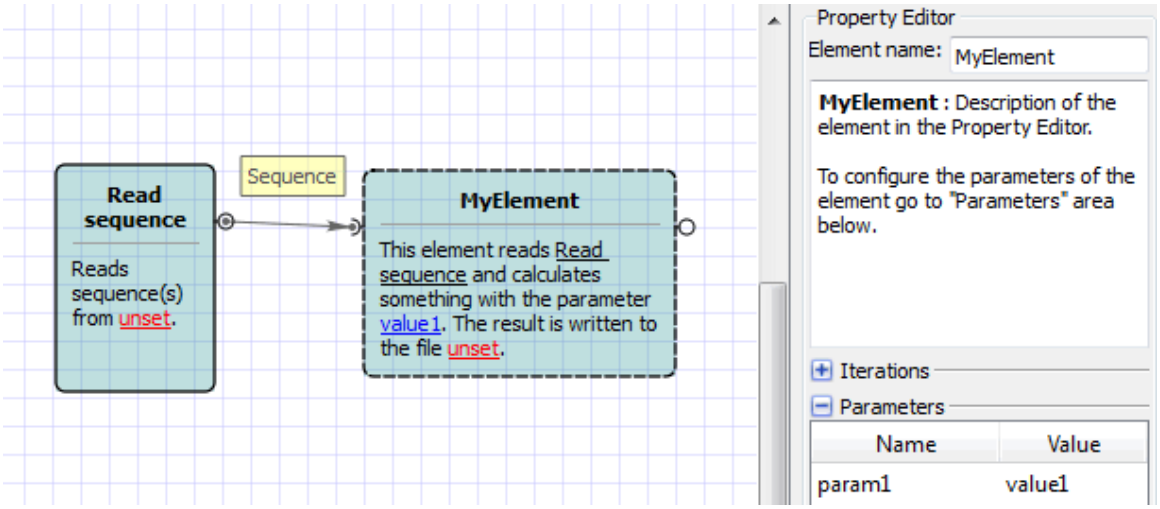
Here *infile* and *outfile* are input and output data set on the step 2, *param1* is an attribute set on the step 3.

On the same 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:



The element created looks as follows:



Editing Element

The element created appears in the *Custom Elements with CMD Tools* group on the *Palette*.

To edit an element select the *Edit* item in it's context menu in the *Palette* or select the *Edit configuration* item in it's 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 command line 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*.

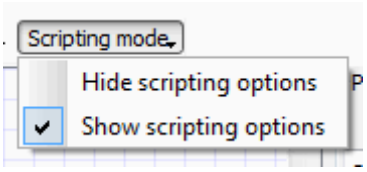
Removing Element

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.

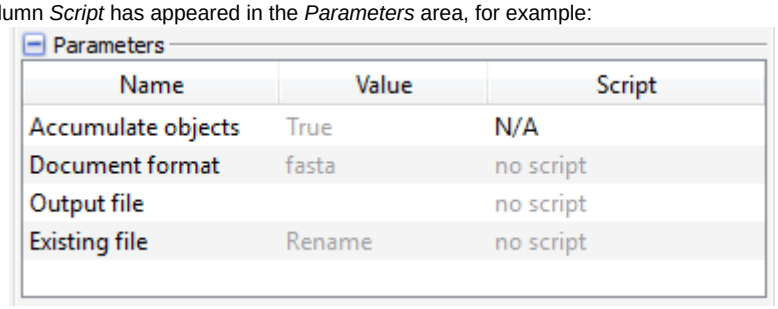
Using Script to Set Parameter Value

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.



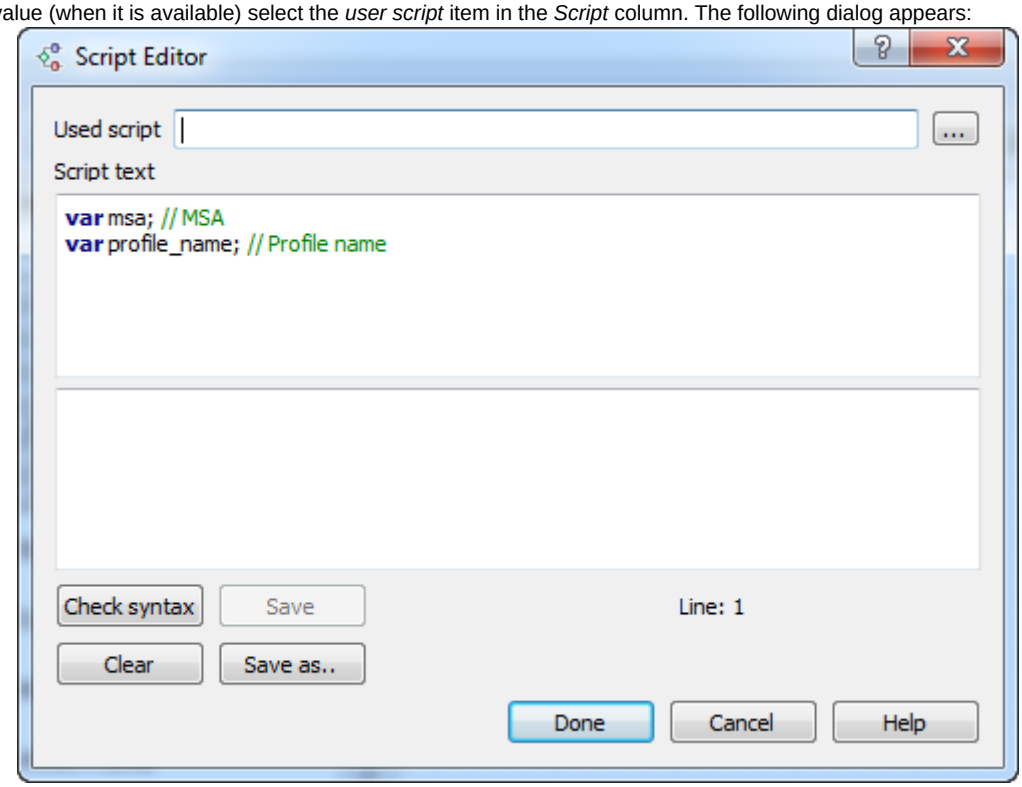
You can see that the third column *Script* has appeared in the *Parameters* area, for example:



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).

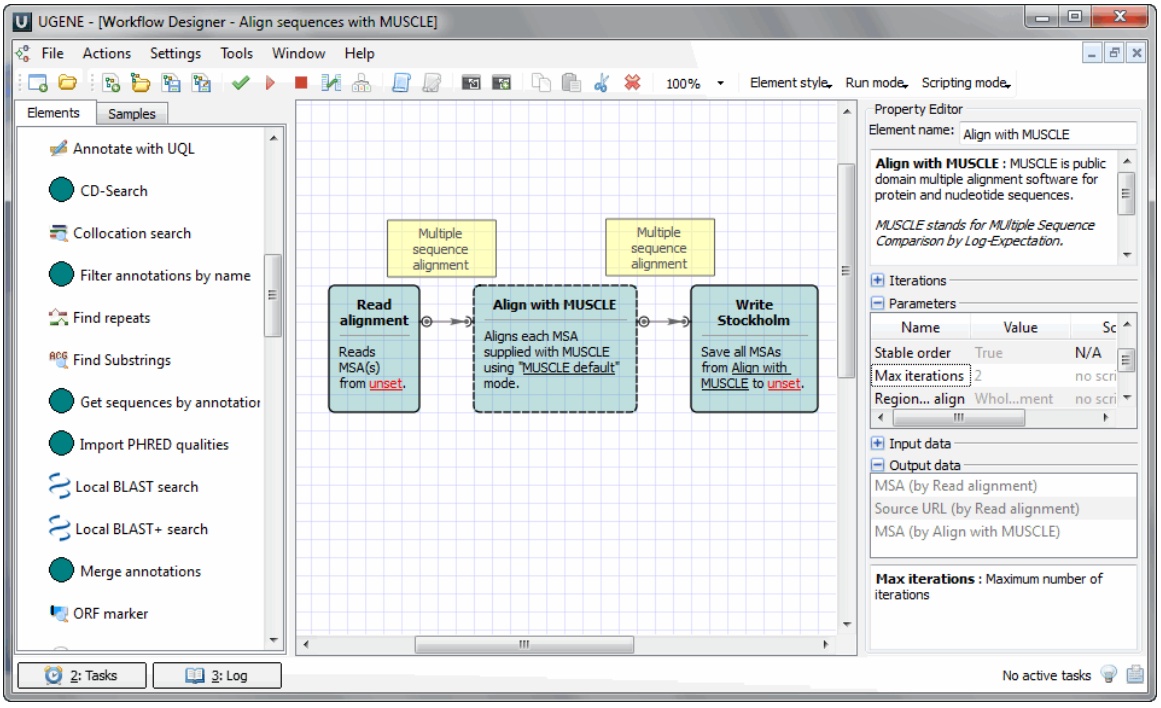
Running Workflow from the Command Line

UGENE provides command line interface (CLI). To learn more about UGENE CLI and commands available read main UGENE User Manual.

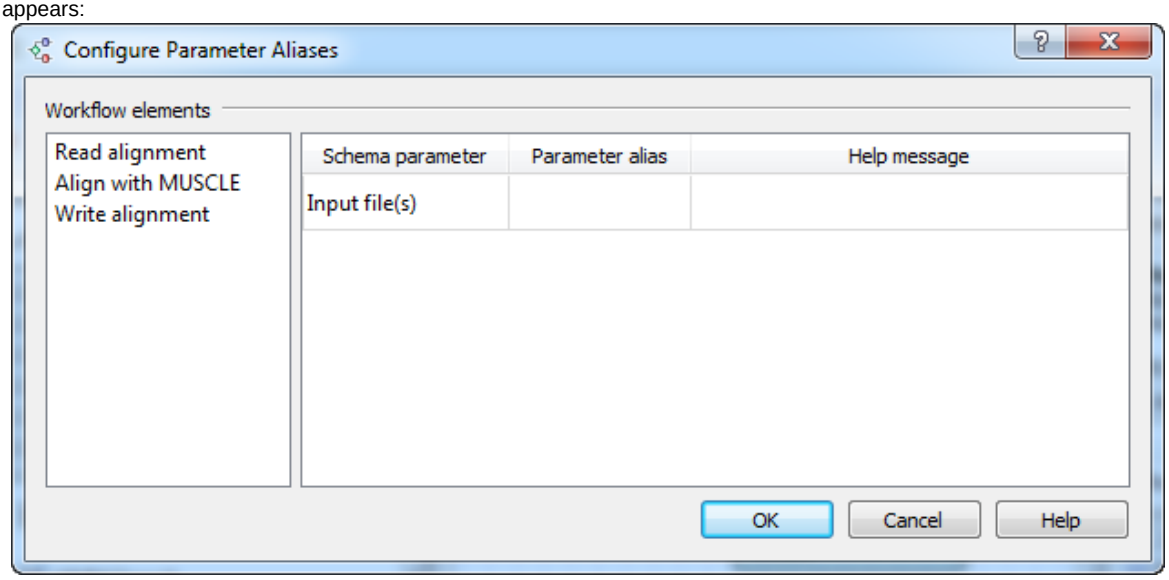
This chapter describes how you can create a new command using a *workflow*.

To run a workflow from the command line do the following:

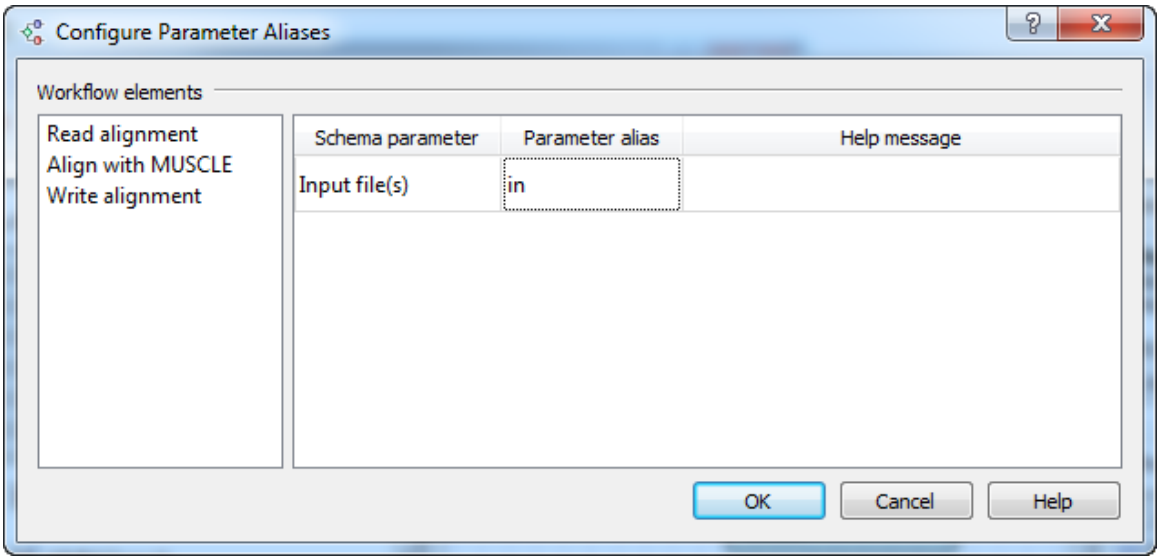
- Create the workflow in the Workflow Designer. For example on the image below the *Align sequences with MUSCLE* sample workflow is used:



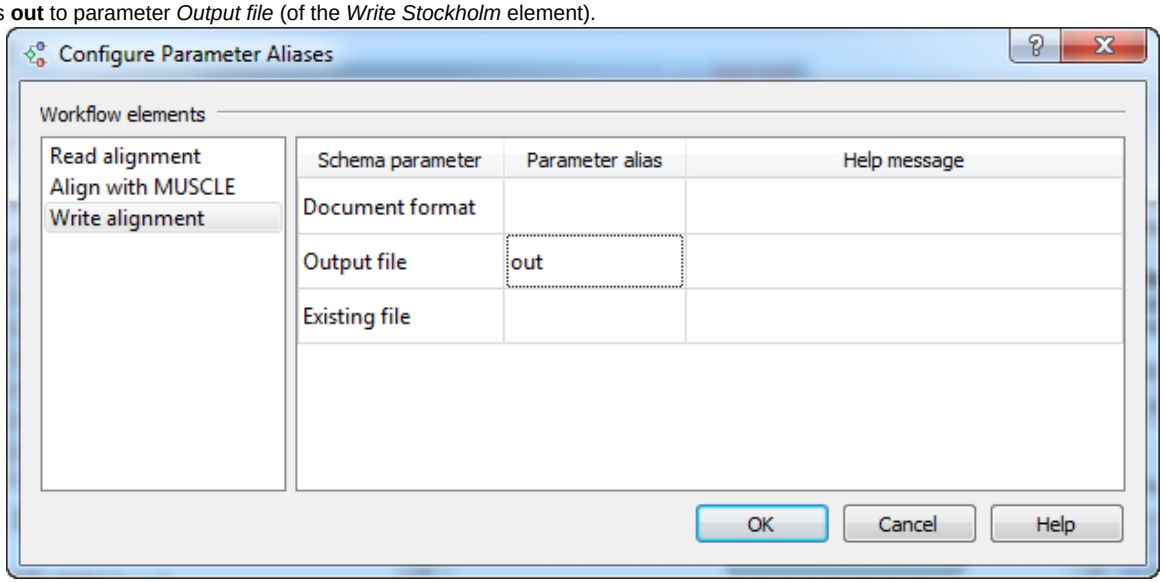
- Now you should configure aliases for those parameters and ports and slots that you are going to use from the command line. To do it select the *Actions Set parameter aliases...* item in the main menu or the *Set parameter aliases* toolbar button. The following dialog appears:



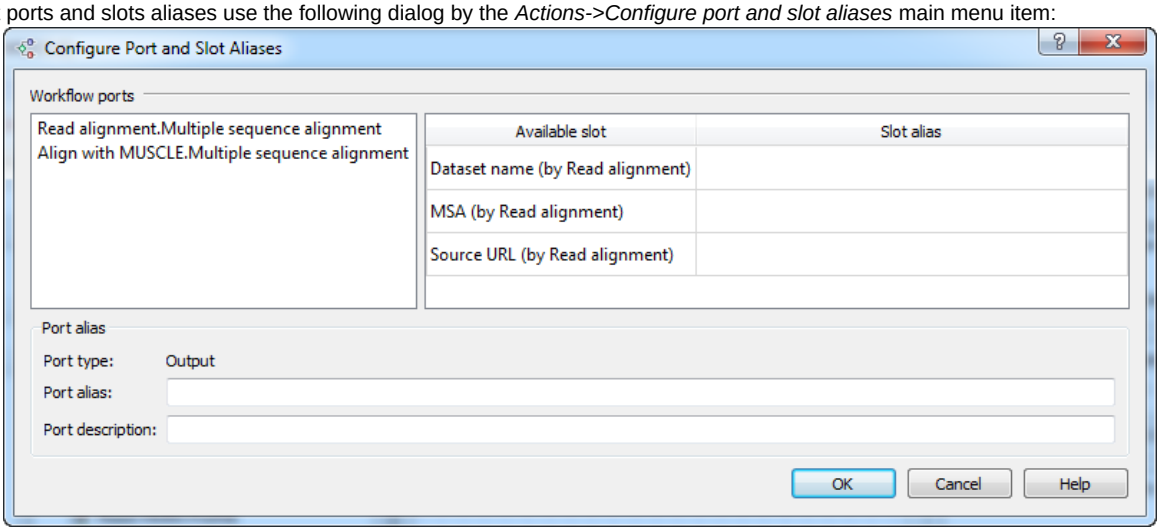
It contains the list of objects that corresponds to the *elements* of the workflow. For each object the list of parameters is available for which you can assign command line aliases. For example, assign alias **in** to parameter *Input file* (of the *Read alignment* element):



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.

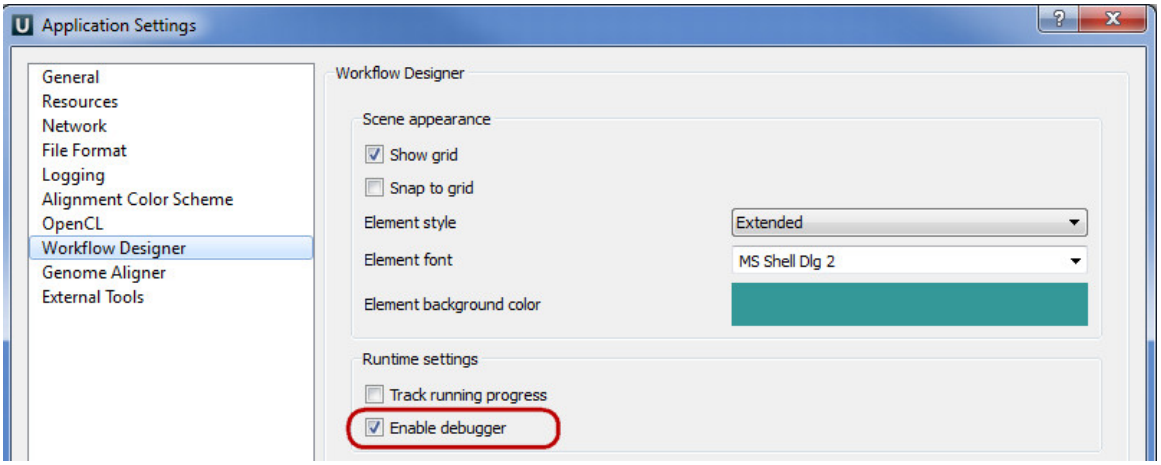
For example on Windows the command can look as follows:

```
ugene --task=C:\mySchema --in=C:\COI.aln --out=C:\COI.sto
```

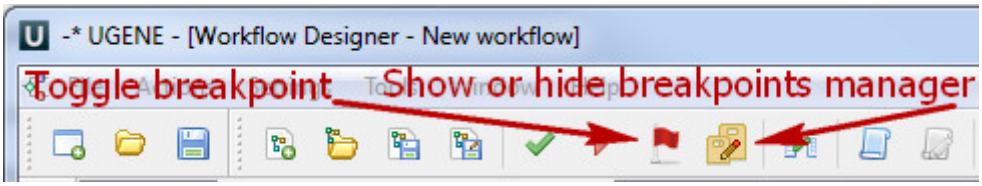

In this example the path to the directory with the UGENE executable is added to the system PATH variable.

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*:



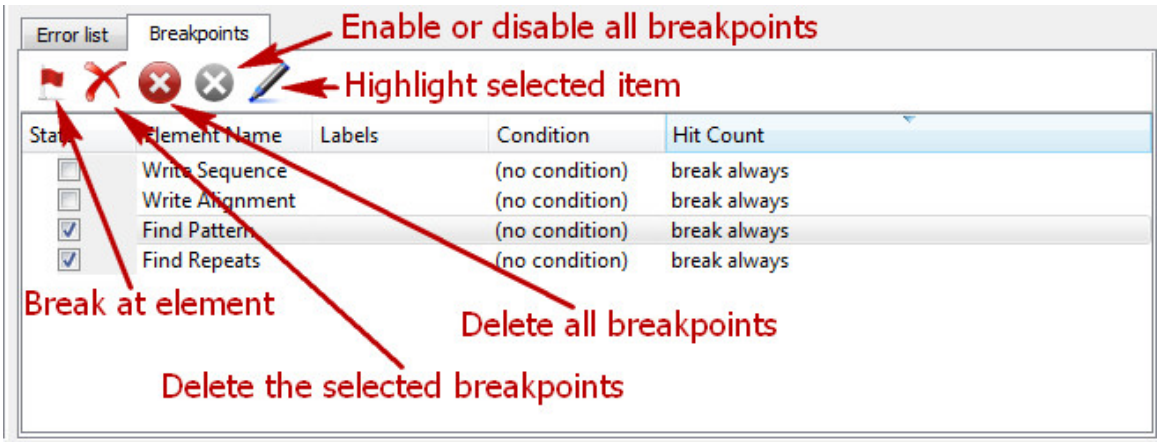
After that the two new buttons appears on the main toolbar:



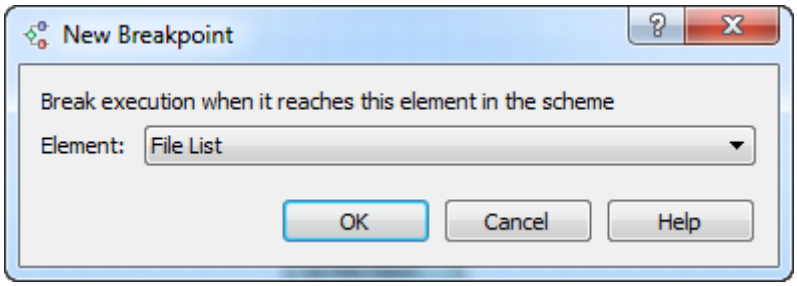
- Creating Breakpoints
- Manipulating Breakpoints

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.

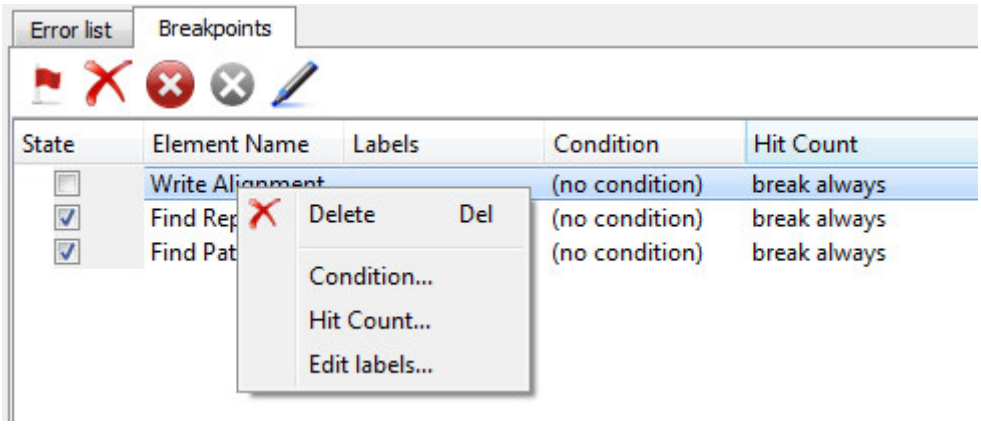
Delete all breakpoints - this button deletes all breakpoints.

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.

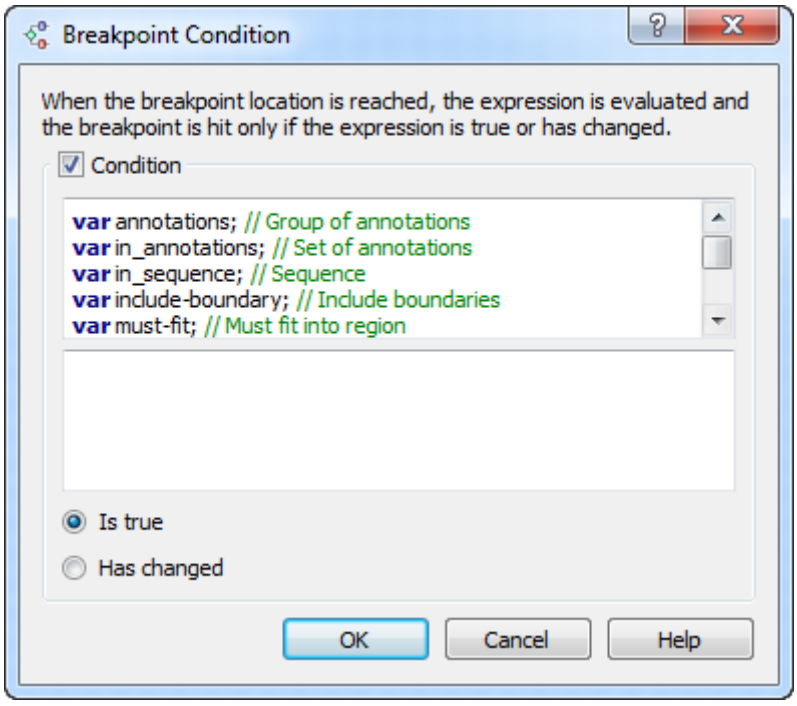
Manipulating Breakpoints

The following operations are available for each breakpoint:



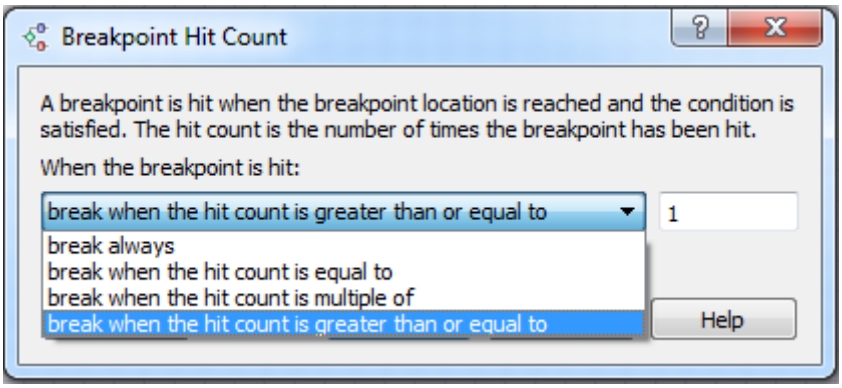
Delete - delete the selected breakpoint.

Condition - creates a breakpoint condition. Click on this menu item and the following dialog appears:



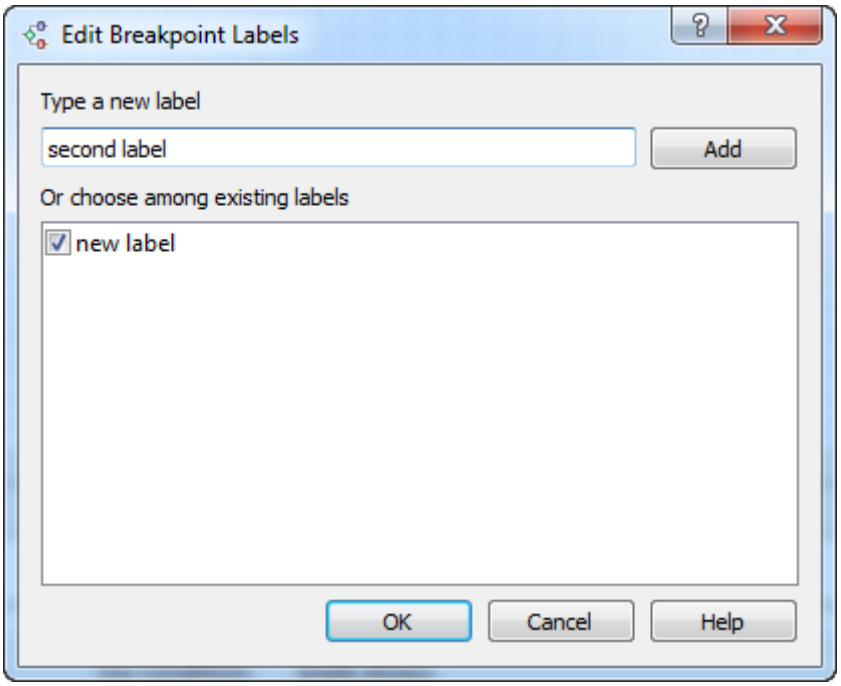
When the breakpoint location is reached, the expression is evaluated and the breakpoint is hit only if the expression is true or has changed.

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:


Edit Breakpoint Labels
Type a new label
second label
Add
Or choose among existing labels
new label
OK
Cancel
Help

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

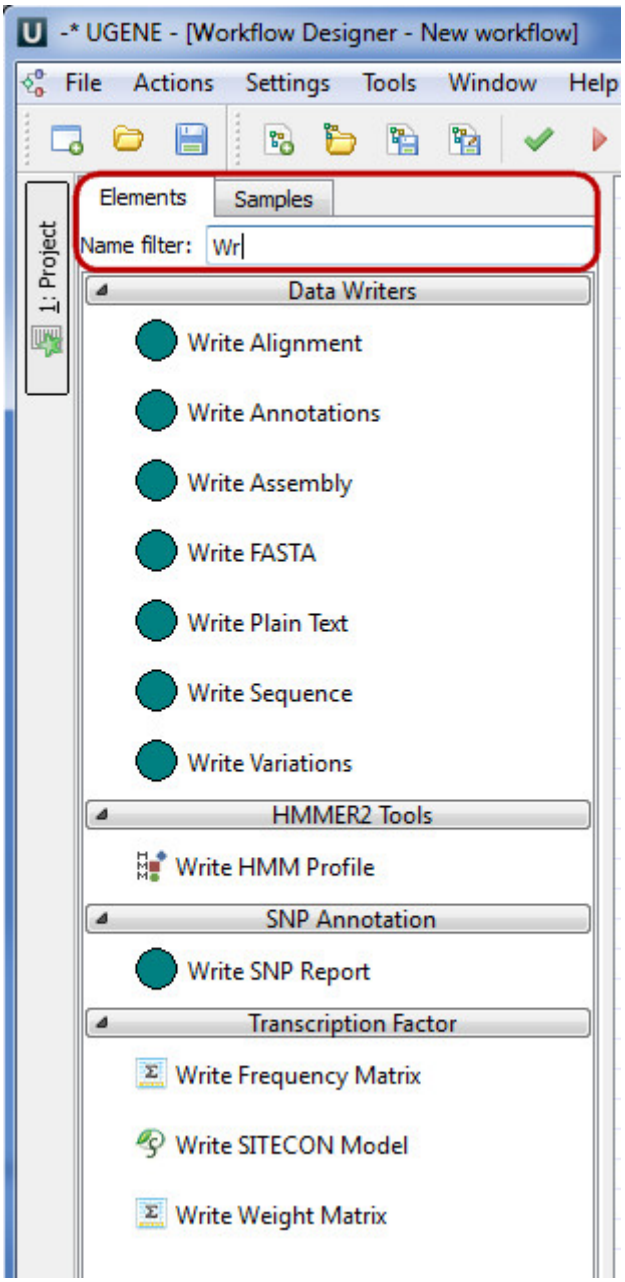
assembly

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
  - File List Element
  - Read Alignment Element
  - Read Annotations Element
  - Read Assembly Element
  - Read from Remote Database Element
  - Read Plain Text Element
  - Read Sequence Element
  - Read Variations Element
- Data Writers
  - Write Alignment Element
  - Write Annotations Element
  - Write Assembly Element
  - Write FASTA Element
  - Write Plain Text Element
  - Write Sequence Element
  - Write Variations 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
    - Local BLAST+ Search Element
    - Merge Annotations Element
    - ORF Marker Element
    - Remote BLAST 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 to Reference 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
    - Split Alignment into Sequences Element
- NGS: Align Short Reads
    - Align Reads with Bowtie Element
    - Align Reads with Bowtie2 Element
    - Align Reads with BWA Element
    - Align Reads with BWA-MEM Element
    - Align Reads with UGENE Genome Aligner Element
- NGS: Basic Functions
    - Assemble Genomes with SPAdes Element
    - 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
    - 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: RNA-Seq Analysis
    - Assembly Transcripts with Cufflinks Element

    - Extract Transcript Sequences with gffread Element
    - Find Splice Junction with TopHat Element
    - Merge Assemblies with Cuffmerge 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
    - Create VCF consensus
    - 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*.

- File List Element
- Read Alignment Element
- Read Annotations Element
- Read Assembly Element
- Read from Remote Database Element
- Read Plain Text Element
- Read Sequence Element
- Read Variations Element

File List Element

Gets the list of files in the specified directories.

Parameters in GUI

Parameter	Description	Default value
Input directory	Input directory.	
Absolute output paths	Specify whether to output absolute or relative paths of the files.	True
Recursive reading	Get files from all nested directories or just from the current one.	False
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.	
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.	

Parameters in Workflow File

Type: get-file-list

Parameter	Parameter in the GUI	Type
in-path	**Input directory**	*string*

absolute	**Absolute output paths**	*boolean*
recursive	**Recursive reading**	*boolean*
include-name-filter	**Include name filter**	*string*
exclude-name-filter	**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 Alignment Element

Reads multiple sequence alignments (MSAs) from local or remote files.

All MSA file formats supported by UGENE are allowed as input to this element.

See the list of all available formats here.

Parameters in GUI

Parameter	Description	Default value
Input files (required)	Semicolon-separated list of paths to the input files.	

Parameters in Workflow File

Type: read-msa

Parameter	Parameter in the GUI	Type
url-in	**Input files**	*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

Reads annotations from files.

All annotation file formats supported by UGENE are allowed as input to this element.

See the list of all available formats here.

Parameters in GUI

Parameter	Description	Default value
Input file(s)	Input files.	Dataset 1;
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

Parameters in Workflow File

Type: read-annotations

Parameter	Parameter in the GUI	Type
url-in	**Input file(s)**	*string*
mode	**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 Assembly Element

Reads assembly from files.

All assembly file formats supported by UGENE are allowed as input to this element.

See the list of all available formats here.

Parameters in GUI

Parameter	Description	Default value
Input file(s)	Input files.	Dataset 1;

Type: read-assembly

Parameter	Parameter in the GUI	Type
url-in	**Input file(s)**	*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 from Remote Database Element

Reads sequences and annotations if any from a remote database.

Parameters in GUI

Parameter	Description	Default value
Resource IDs (required)	Semicolon-separated list of resource IDs in the database.	
Database (required)	Name of the database to read from.	NCBI Genbank (DNA sequence)
Save file to directory	Directory to store a file loaded from the database.	default
Read resource ID(s) from source	The source to read resource IDs from: the list or a local file.	List of TDs

Parameters in Workflow File

Type: fetch-sequence

Parameter	Parameter in the GUI	Type
resource-id	**Resource IDs**	*string*
database	**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-dir	**Save file to directory**	*string*
ids-source	**Read resource ID(s) from 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 Plain Text Element

Reads text from local or remote files.

All text file formats supported by UGENE are allowed as input to this element.

See the list of all available formats here.

Parameters in GUI

Parameter	Description	Default value
Input files (required)	Semicolon-separated list of paths to the input files.	
Read by lines (required)	Specifies to read the input file line by line.	false

Parameters in Workflow File

Type: read-text

Parameter	Parameter in the GUI	Type
url-in	**Input files**	*string*
read-by-lines	**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

Reads sequences and annotations if any from local or remote files.

All sequence file formats supported by UGENE (e.g. GenBank, FASTA) are allowed as input to this element.

See the list of all available formats here.

Parameters in GUI

Parameter	Description	Default value
Input files	Semicolon-separated list of datasets to the input files.	
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
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

Sequence count limit	Split mode only. Read only first N sequences from each file. Set 0 value for reading all sequences.	0
Accession filter	Only reports a sequence with the specified accession (id).	

Parameters in Workflow File

Type: read-sequence

Parameter	Parameter in the GUI	Type
url-in	**Input files**	*string*
mode	**Mode**	*numeric* Available values are: • 0 - for split mode • 1 - for merge mode
merge-gap	**Merging gap**	*numeric*
sequence-count-limit	**Sequence count limit**	*numeric*
accept-accession	**Accession filter**	*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 Variations Element

Reads variations from files and produces variations tracks.

All variation file formats supported by UGENE are allowed as input to this element.

See the list of all available formats here.

Parameters in GUI

Parameter	Description	Default value
Input file(s)	Input file(s).	Dataset 1

Parameters in Workflow File

Type: read-variations

Parameter	Parameter in the GUI	Type
url-in	**Input file(s)**	*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 Assembly Element
- Write FASTA Element
- Write Plain Text Element
- Write Sequence Element
- Write Variations Element

Write Alignment Element

Writes all supplied alignments to file(s) in selected format.

Parameters in GUI

Parameter	Description	Default value
Data storage	Place to store workflow results: local file system or a database.	
Document format	Format of the output file.	clustal
Output file	Location of the output data file. If this parameter is set, then the “Location” slot is not taken into account.	
Output file suffix	This suffix will be used for generating the output file name.	
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

Parameters in Workflow File

Type: write-msa

Parameter	Parameter in the GUI	Type
data-storage	**Data storage**	*string*

document-format	**Document format**	*string* Available values are: • clustal • mega • msf • sam • srfasta • stockholm
url-out	**Output file**	*string*
url-suffix	**Output file suffix**	*string*
write-mode	**Existing file**	*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

Writes all supplied annotations to file(s) in the selected format.

Parameters in GUI

Parameter	Description	Default value
Data storage	Place to store workflow results: local file system or a database.	
Output file	Location of the output data file. If this attribute is set, slot “Location” in port will not be used.	
Output file suffix	This suffix will be used for generating the output file name.	
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
Document format	Document format of output file.	genbank
Annotations name	Object name of the annotations.	unknown feature
CSV separator	String which separates values in CSV file(s).	”,” (comma)
Write sequence name	Write sequence to CSV file(s).	False

Parameters in Workflow File

Type: write-annotations

Parameter	Parameter in the GUI	Type
data-storage	**Data storage**	*string*
url-out	**Output file**	*string*
url-suffix	**Output file suffix**	*string*
write-mode	**Existing file**	*numeric* Available values are: • 0 - for overwrite • 1 - for append • 2 - for rename
document-format	**Document format**	*string* Available values are: • CSV • GenBank • GFF
annotations-name	**Annotations name**	*string*
separator	**CSV separator**	*string*
write_names	**Write sequence name**	*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 Assembly Element

Writes all supplied assemblies to file(s) in selected format.

Parameters in GUI

Parameter	Description	Default value
Data storage	Place to store workflow results: local file system or a database.	
Document format	Document format of output file.	bam
Build index (BAM only)	Build BAM index for the target BAM file. The file .bai will be created in the same directory.	True
Output file	Location of output data file. If this attribute is set, slot "Location" in port will not be used.	

Output file suffix	This suffix will be used for generating the output file name.	
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

Parameters in Workflow File

Type: write-assembly

Parameter	Parameter in the GUI	Type
data-storage	**Data storage**	*string*
document-format	**Document format**	*string*
build-index	**Build index (BAM only)**	*boolean*
out-url	**Output file**	*string*
url-suffix	**Output file suffix**	*string*
write-mode	**Existing file**	*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 FASTA Element

Writes all supplied sequences to file(s) in FASTA format.

Parameters in GUI

Parameter	Description	Default value
Output file	Location of the output data file. If this attribute is set, then the “Location” slot is not taken into account.	
Output file suffix	This suffix will be used for generating the output file name.	
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
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

Parameters in Workflow File

Type: write-fasta

Parameter	Parameter in the GUI	Type
url-out	**Output file**	*string*
url-suffix	**Output file suffix**	*string*
write-mode	**Existing file**	*numeric* Available values are: • 0 - for overwrite • 1 - for append • 2 - for rename
accumulate	**Accumulate objects**	*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 Plain Text Element

Writes strings to a file.

Parameters in GUI

Parameter	Description	Default value
Data storage	Place to store workflow results: local file system or a database.	
Output file	Location of the output data file. If this attribute is set, then the “Location” slot is not taken into account.	
Output file suffix	This suffix will be used for generating the output file name.	
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
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

Parameters in Workflow File

Type: write-text

Parameter	Parameter in the GUI	Type

data-storage	**Data storage**	*string*
url-out	**Output file**	*string*
url-suffix	**Output file suffix**	*string*
write-mode	**Existing file**	*numeric* Available values are: • 0 - for overwrite • 1 - for append • 2 - for rename
accumulate	**Accumulate objects**	*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

Writes all supplied sequences to file(s) in selected format.

Parameters in GUI

Parameter	Description	Default value
Data storage	Place to store workflow results: local file system or a database.	
Output file	Location of the output data file. If this attribute is set, then the “Location” slot is not taken into account.	
Output file suffix	This suffix will be used for generating the output file name.	
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
Document format	Format of the output file.	fasta
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
Split sequence	Split each incoming sequence on several parts.	1

Parameters in Workflow File

Type: write-sequence

Parameter	Parameter in the GUI	Type

data-storage	**Data storage**	*string*
url-out	**Output file**	*string*
url-suffix	**Output file suffix**	*string*
write-mode	**Existing file**	*numeric* Available values are: • 0 - for overwrite • 1 - for append • 2 - for rename
document-format	**Document format**	*string* Available values are: • fasta • fastq • genbank • raw
accumulate	**Accumulate objects**	*boolean*
split	**Split sequence**	*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 Variations Element

Writes all supplied variations to file(s) in selected format.

Parameters in GUI

Parameter	Description	Default value
Data storage	Place to store workflow results: local file system or a database.	
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
Document format	Document format of output file.	snp
Output file	Location of output data file. If this attribute is set, slot "Location" in port will not be used.	
Output file suffix	This suffix will be used for generating the output file name.	

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

Parameters in Workflow File

Type: write-variations

Parameter	Parameter in the GUI	Type
data-storage	**Data storage**	*split*
accumulate	**Accumulate objects**	*boolean*
document-format	**Document format**	*string*
out-url	**Output file**	*string*
url-suffix	**Output file suffix**	*string*
write-mode	**Existing file**	*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).

Parameters in GUI

Parameter	Description	Default value
Filter by value(s)	Semicolon-separated list of values used to filter the input data.	

Parameters in Workflow File

Type: filter-by-values

Parameter	Parameter in the GUI	Type
text	**Filter by value(s)**	*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.

Parameters in GUI

To use the *Grouper* element connect the *Grouper's* input port to the required workflow element. Select the *Grouper* element on the *Scene* an d specify *Group 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

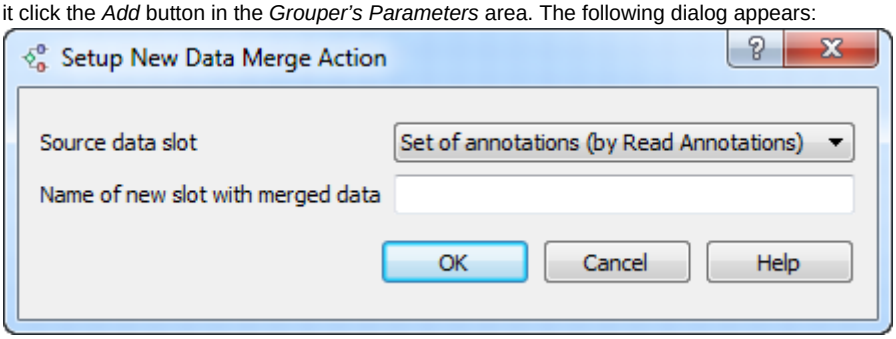
The *Group operation* specifies criteria to group data supplied to the *Group slot*. It can take the following values:

- *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.

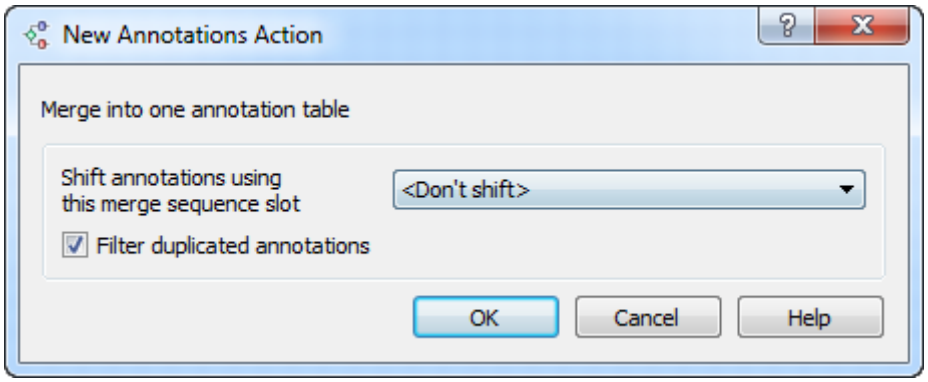
Output slots

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 **as sociated** 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*Grou per* element. To create it click the *Add* button in the *Grouper's Parameters* area. The following dialog appears:



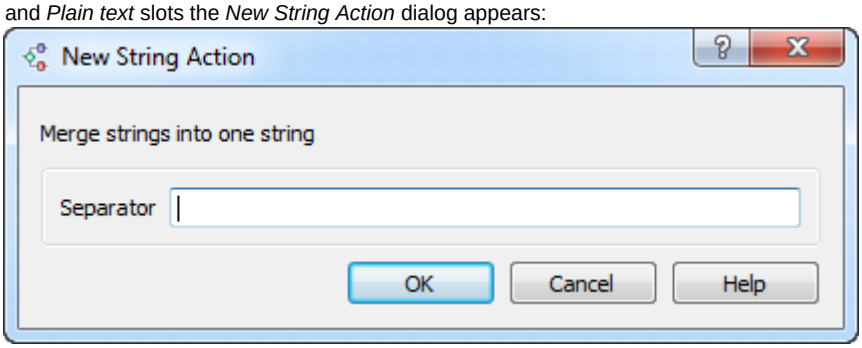
In the dialog you should select a *Source data slot* (i.e. a slot with the associated data) and input a name of the new slot. Click the OK button. A new dialog appears that specifies how the associated data should be merged. The view of the dialog and the available merge actions for different types of the *Source data slot* are the following:

- For a *Set of annotations* slot the *New Annotations Action* dialog appears:



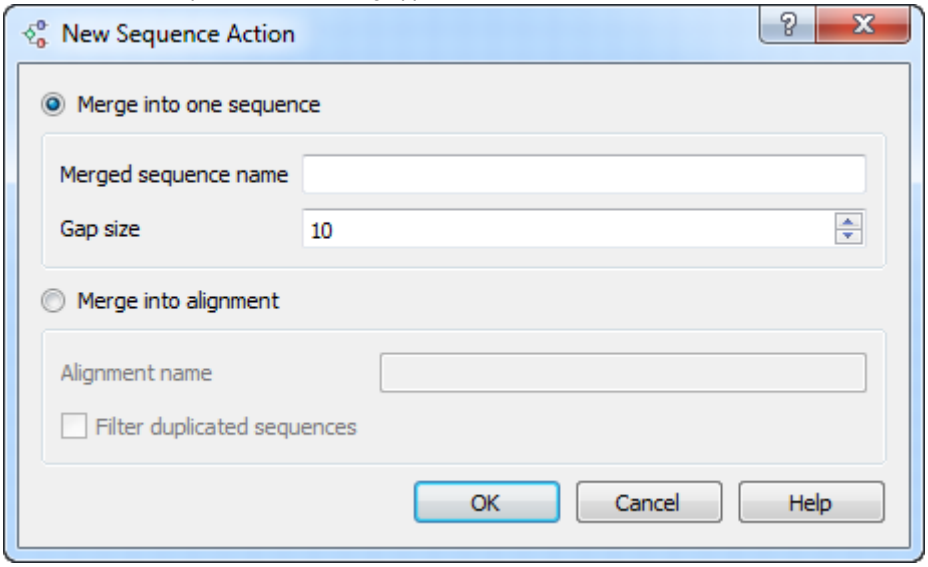
You can merge annotations into one annotation table and, optionally, filter duplicated annotations. Also, you can shift annotations. To do it, you need to create another output slot with type *Sequence* and *Merge into one sequence* option selected (see below). In other words you need to merge all sequences 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:



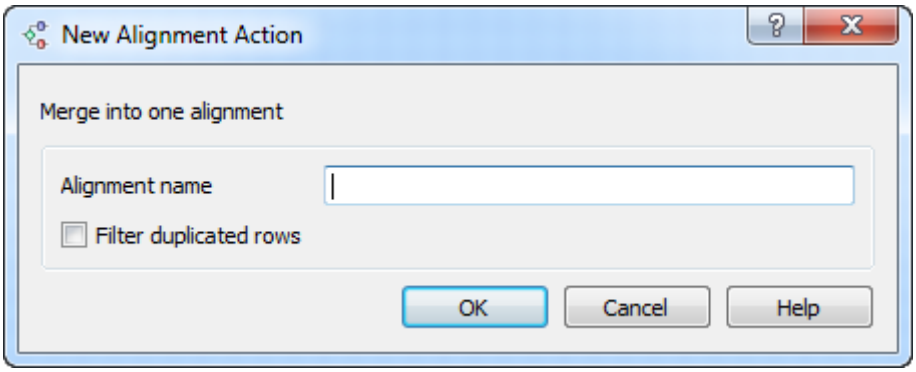
Using this dialog you can merge strings into one string. Optionally, you can specify an additional strings separator.

- 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* button.

Parameters in Workflow File

Type: grouper

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.

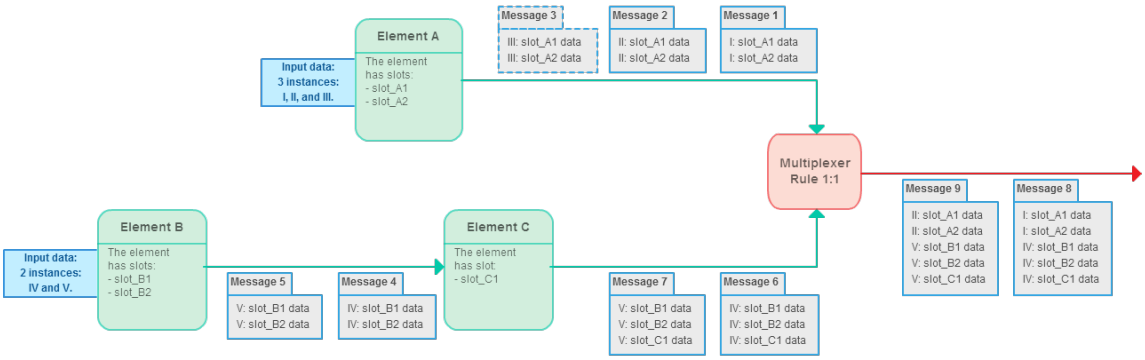
There are the following multiplexing rules:

- 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:



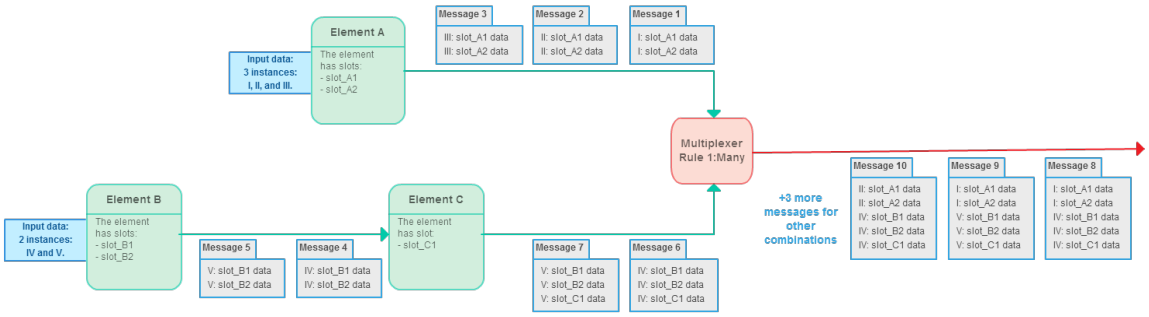
As you can see:

- 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"

Parameters in GUI

Parameter	Description	Default value
Multiplexing rule	Available values are: • 1 to 1 • 1 to many See the detailed description of the values above.	1 to 1

Parameters in Workflow File

Type: multiplexer

Parameter	Parameter in the GUI	Type
multiplexing-rule	**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:

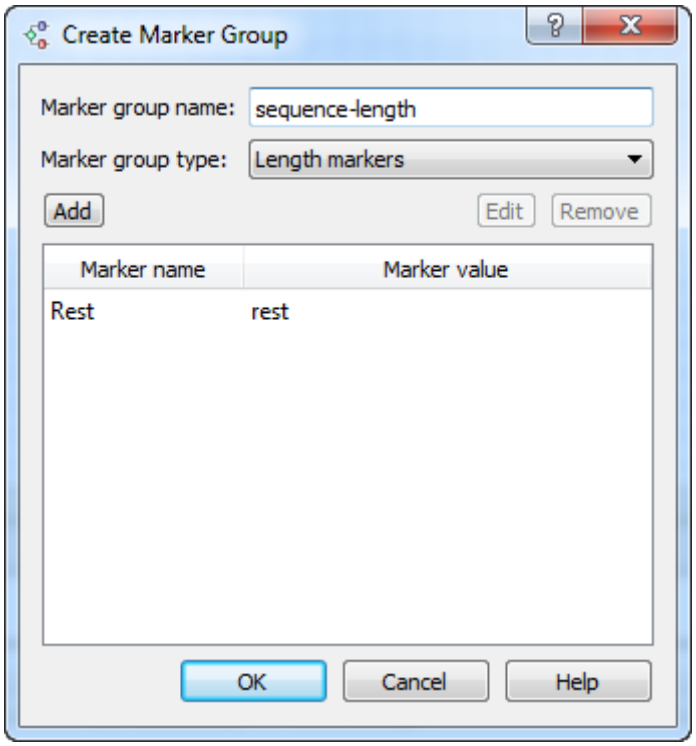
- Find Substrings in Sequences
- Merge Sequences and Annotations
- Search for TFBS

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.

Parameters in GUI

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.

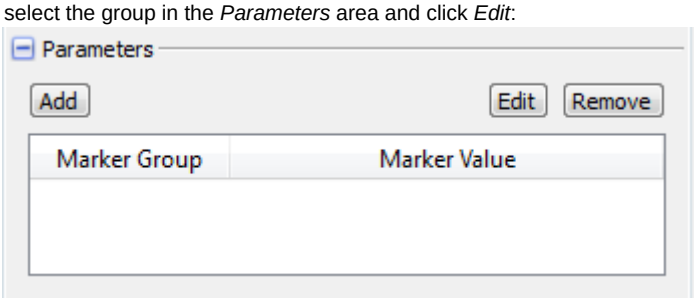
Qualifier float value markers — marks a sequence by the number of float qualifiers.

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*.

Parameters in Workflow File

Type: mark-sequence

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
- Local BLAST+ Search Element
- Merge Annotations Element
- ORF Marker Element
- Remote BLAST Element
- Smith-Waterman Search Element

Amino Translations Element

Translates a sequence into it's amino translation or translations.

Parameters in GUI

Parameter	Description	Default value
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
Auto selected genetic code	Specifies that genetic code should be selected automatically.	True
Genetic code	Genetic code that should be used to translate the input nucleotide sequence.	The Standard Genetic Code

Parameters in Workflow File

Type: sequence-translation

Parameter	Parameter in the GUI	Type
pos-2-translate	**Translate from**	*string* Available values are: • all • first • second • third
auto-translation	**Auto selected genetic code**	*boolean*
genetic-code	**Genetic code**	*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*

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.

Parameters in GUI

Parameter	Description	Default value
Workflow (required)	UQL workflow file.	
Merge	Merges regions of each result into a single annotation.	False
Offset	If the *Merge* parameter is set to *True*, adds left and right offsets of the specified length to the annotation.	0

Parameters in Workflow File

Type: query

Parameter	Parameter in the GUI	Type
schema	**Workflow**	*string*
merge	**Merge**	*boolean*
offset	**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.

Parameters in GUI

Parameter	Description	Default value
Annotate as	Name of the result annotations marking found conserved domains.	CDD result

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.	CDD Available values are: • CDD • Pfam • TIGRFAM • COG • KOG • Prk • SMART
Database directory	Specifies database directory for local search.	
Local search	Perform the search on local machine or submit the search to NCBI for remote execution.	True

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.	

Parameters in Workflow File

Type: cd-search

Parameter	Parameter in the GUI	Type
result-name	**Annotate as**	*string*
db-name	**Database**	*string*
db-path	**Database directory**	*string*
local-search	**Local search**	*boolean*
e-val	**Expect value**	*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.

Parameters in GUI

Parameter	Description	Default value
Result type	Copy original annotations or annotate found regions with new ones.	Create new annotations
Result annotation (required)	Name of the result annotation to mark found collocations.	misc_feature
Include boundaries	Include most left and most right boundary annotations regions into result or exclude them.	True
Group of annotations (required)	List of annotation names to search. Found regions will contain all the named annotations.	

Region size	Effectively this is the maximum allowed distance between the interesting annotations in a group.	1000
Must fit into region	Specifies whether the interesting annotations should entirely fit into the specified region to form a group.	False

Parameters in Workflow File

Type: collocated-annotation-search

Parameter	Parameter in the GUI	Type
result-type	**Result type**	*string*
result-name	**Result annotation**	*string*
annotations	**Group of annotations**	*string*
include-boundary	**Include boundaries**	*boolean*
region-size	**Region size**	*numeric*
must-fit	**Must fit into region**	*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.

Parameters in GUI

Parameter	Description	Default value
PHRED output	Path to file with PHRED quality scores.	

Parameters in Workflow File

Type: export-phred-qualities

Parameter	Parameter in the GUI	Type

url-out	**PHRED output**	*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.

Parameters in GUI

Parameter	Description	Default value
Save file to directory	The directory to store sequence files loaded from a database.	default
NCBI database	The database to read from.	nucleotide Available values are: • nucleotide • protein

Parameters in Workflow File

Type: fetch-sequence

Parameter	Parameter in the GUI	Type
save-dir	**Save file to directory**	*string*
database	**NCBI database**	*string*

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.

Parameters in GUI

Parameter	Description	Default value
Annotation names	List of annotation names, separated by spaces, that will be accepted or filtered.	
Annotation names file	File with annotation names, separated with whitespaces which will be accepted or filtered.	
Accept or filter	Selects the name filter: accept specified names or accept all except specified.	True

Parameters in Workflow File

Type: filter-annotations

Parameter	Parameter in the GUI	Type
annotation-names	**Annotation names**	*string*
annotation-names-file	**Annotation names file**	*string*
accept-or-filter	**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.

Parameters in GUI

Parameter	Description	Default value
Qualifier name	Name of the qualifier to use for filtering.	
Qualifier value	Text value of the qualifier to apply as filtering criteria.	
Accept or filter	Selects the name filter: accept specified names or accept all except specified.	True

Parameters in Workflow File

Type: filter-annotations-by-qualifier

Parameter	Parameter in the GUI	Type
qualifier-name	**Qualifier name**	*string*
qualifier-value	**Qualifier value**	*string*
accept-or-filter	**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.

Parameters in GUI

Parameter	Description	Default value
Output report file	Path to the report output file.	

Parameters in Workflow File

Type: find-primers

Parameter	Parameter in the GUI	Type
output-file	**Output report 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.

Parameters in GUI

Parameter	Description	Default value
Annotate as	Name of the result annotation.	misc_feature
Pattern(s)	Semicolon-separated list of patterns to search for.	
Pattern file	Load pattern from file in any sequence format or in newline-delimited format.	
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
Max Mismatches	Maximum number of mismatches between a substring and a pattern.	0
Search in	Specifies which strands should be searched: direct, complementary or both.	both strands
Allow Insertions/Deletions	Takes into account possibility of insertions/deletions when searching. By default substitutions are only considered.	False
Support ambiguous bases	Performs correct handling of ambiguous bases. When this option is activated insertions and deletions are not considered.	False
Search in Translation	Translates a supplied nucleotide sequence to protein and searches in the translated sequence.	False
Qualifier name for pattern name	Name of qualifier in result annotations which is containing a pattern name.	pattern_name

Parameters in Workflow File

Type: search

Parameter	Parameter in the GUI	Type
result-name	**Annotate as**	*string*
pattern	**Pattern(s)**	*string*
pattern_file	**Pattern file**	*string*
use-names	**Use pattern name**	*boolean*
max-mismatches-num	**Max Mismatches**	*numeric*
strand	**Search in**	*numeric* Available values are: • 0 - for searching in both strands • 1 - for searching in direct strand • 2 - for searching in complement strand
allow-ins-del	**Allow Insertions/Deletions**	*boolean*
ambiguous	**Support ambiguous bases**	*boolean*
amino	**Search in Translation**	*boolean*
pattern-name-qual	**Qualifier name for pattern name**	*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.

Parameters in GUI

Parameter	Description	Default value
Annotate as (required)	Name of the result annotation to mark found repeats.	repeat_unit
Algorithm	Control over variations of the algorithm.	Auto
Filter nested	Filters nested repeats.	True
Identity	Repeats identity in percents.	100
Inverted	Specifies to search for inverted repeats.	False
Max distance	Maximum distance between the repeats.	5000
Min distance	Minimum distance between the repeats.	0
Min length	Minimum length of the repeats.	5
Parallel threads	Number of parallel threads used for the task.	Auto

Parameters in Workflow File

Type: repeats-search

Parameter	Parameter in the GUI	Type
result-name	**Annotate as**	*string*
algorithm	**Algorithm**	*numeric* Available values are: • 0 - algorithm choosed automaticly • 1 - for diagonal algorithm • 2 - for suffix index algorithm

filter-nested	**Filter nested**	*boolean*
identity	**Identity**	*numeric*
max-distance	**Max distance**	*numeric*
min-distance	**Min distance**	*numeric*
min-length	**Min length**	*numeric*
threads	**Parallel 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.

Parameters in GUI

Parameter	Description	Default value
Output file	File to store a report.	
Annotation name	Annotation name used to compare genes and reference genomes..	blast-result
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
Identity cutoff	Identity between gene sequence length and annotation length in per cent. BLAST identity (if specified) is checked after	90.0000%

Parameters in Workflow File

Type: genebygene-report-id

Parameter	Parameter in the GUI	Type
output-file	**Output file**	*string*
annotation_name	**Annotation name**	*string*

existing	**Existing file**	*string*
identity	**Identity cutoff**	*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 input sequence.

Parameters in GUI

Parameter	Description	Default value
Translate	Translates the annotated regions if the corresponding annotation marks a protein subsequence.	False
Complement	Complements the annotated regions if the corresponding annotation is located on the complement strand.	False
Split joined	Split joined annotations to single region annotations.	False
Extend left	Extends the resulted regions to left.	0
Extend right	Extends the resulted regions to right.	0
Gap length	Inserts a gap of a specified length between the merged locations of the annotation.	0

Parameters in Workflow File

Type: extract-annotated-sequence

Parameter	Parameter in the GUI	Type
translate	**Translate**	*boolean*
complement	**Complement**	*boolean*
split-joined-annotations	**Split joined**	*boolean*
extend-left	**Extend left**	*numeric*
extend-right	**Extend right**	*numeric*
merge-gap-length	**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.

Parameters in GUI

Parameter	Description	Default value
Output report file	Path to the report output file.	

Parameters in Workflow File

Type: primers-grouper

Parameter	Parameter in the GUI	Type
output-file	**Output report 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.

Parameters in GUI

Parameter	Description	Default value
PHRED input (required)	Path to a file with PHRED quality scores.	
Quality format	Format to encode quality scores.	Sanger

Parameters in Workflow File

Type: import-phred-qualities

Parameter	Parameter in the GUI	Type
url-in	**PHRED input**	*string*
quality-format	**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.

Parameters in GUI

Parameter	Description	Default value
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

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
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%

Parameters in Workflow File

Type: intersect-annotations

Parameter	Parameter in the GUI	Type
report	**Result annotations**	*numeric*
unique	**Unique overlaps**	boolean
minimum-overlap	**Minimum overlap**	*numeric*

The element has 2 *input ports*:

Name in GUI: Annotations A

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 the supplied DNA sequence in local BLAST database.

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 in GUI

Parameter	Description	Default value
Search type	Selects the type of the BLAST searches.	blastn
Database path	Path to the database files.	
Database name	Base name for BLAST DB files.	
Tool path	Path to the BLAST executable.	default
Temporary directory	Directory for temporary files.	default
Expected value	Expectation threshold value.	10
Best hits limit	Specifies the number of best hits from a region of the query to keep. 0 turns it off. If used, 100 is recommended.	0
Annotate as	Name of the result annotations.	blast_result
Gapped alignment	Perform gapped alignment.	use
Gap costs	Cost to create and extend a gap in an alignment.	2 2
Match scores	Reward and penalty for matching and mismatching bases.	1 -3
BLAST output	Location of BLAST output file.	
BLAST output type	Type of BLAST output file.	XML (-m 7)

Parameters in Workflow File

Type: blast

Parameter	Parameter in the GUI	Type
blast-type	**Search type**	*string* Available values are: • blastn • blastp • blastx • tblastn • tblastx
db-path	**Database path**	*string*
db-name	**Database name**	*string*
tool-path	**Tool path**	*string*
temp-dir	**Temporary directory**	*string*
e-val	**Expected value**	*numeric*
max-hits	**Best hits limit**	*numeric*

result-name	**Annotate as**	*string*
gapped-aln	**Gapped alignment**	*boolean*
gap-costs	**Gap costs**	*string*
match-scores	**Match scores**	*string*
blast-output	**BLAST output**	*string*
type-output	**BLAST output type**	*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*

Local BLAST+ Search Element

Finds annotations for DNA sequence in a local BLAST database.

BLAST+ is a newer version of the BLAST package and is recommended to use by the NCBI.

> 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 in GUI

Parameter	Description	Default value
Search type	Selects the type of the BLAST searches.	blastn
Database path	Path to the database files.	
Database name	Base name for BLAST DB files.	
Tool path	Path to the BLAST executable.	default
Temporary directory	Directory for temporary files.	default
Expected value	Expectation threshold value.	10
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

Annotate as	Name of the result annotations.	blast_result
Gapped alignment	Perform gapped alignment.	use
Gap costs	Cost to create and extend a gap in an alignment.	2 2
Match scores	Reward and penalty for matching and mismatching bases.	1 -3
BLAST output	Location of BLAST output file.	
BLAST output type	Type of BLAST output file.	XML (-outfmt 5)

Parameters in Workflow File

Type: blast-plus

Parameter	Parameter in the GUI	Type
blast-type	**Search type**	*string* Available values are: • blastn • blastp • blastx • tblastn • tblastx
db-path	**Database path**	*string*
db-name	**Database name**	*string*
tool-path	**Tool path**	*string*
temp-dir	**Temporary directory**	*string*
e-val	**Expected value**	*numeric*
max-hits	**Culling limit**	*numeric*
result-name	**Annotate as**	*string*
gapped-aln	**Gapped alignment**	*boolean*
gap-costs	**Gap costs**	*string*
match-scores	**Match scores**	*string*
blast-output	**BLAST output**	*string*
type-output	**BLAST output type**	*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.

Parameters in GUI

Parameter	Description	Default value
Output file (required)	Location of the output data file. If this attribute is set, then the “Location” slot is not taken into account.	
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
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

Parameters in Workflow File

Type: write-fastq

Parameter	Parameter in the GUI	Type
url-out	**Output file**	*string*
write-mode	**Existing file**	*numeric* Available values are: • 0 - for overwrite • 1 - for append • 2 - for rename
accumulate	**Accumulate objects**	*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.

Parameters in GUI

Parameter	Description	Default value
Annotate as (required)	Name of the result annotations.	ORF
Search in	Specifies which strands should be searched: direct, complement or both.	both strands
Min length	Ignores ORFs shorter than the specified length.	100
Genetic code	Specifies which genetic code should be used for translating the input nucleotide sequence.	The Standard Genetic Code
Require init codon	Allows or not ORFs starting with any codon other than terminator.	True
Require stop codon	Ignores boundary ORFs which last beyound the search region (i.e. have no stop codon within the range).	False
Allow alternative codons	Allows ORFs starting with alternative initiation codons, accordingly to the current translation table.	False

Parameters in Workflow File

Type: orf-search

Parameter	Parameter in the GUI	Type
result-name	**Annotate as**	*string*
strand	**Search in**	*numeric* Available values are: • 0 - for searching in both strands • 1 - for searching in direct strand • 2 - for searching in complement strand
min-length	**Min length**	*numeric*
genetic-code	**Genetic code**	*string* Available values are: • NCBI-GenBank #1 • NCBI-GenBank #2 • etc.
require-init-codon	**Require init codon**	*boolean*
require-stop-codon	**Require stop codon**	*boolean*
allow-alternative-codons	**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.

Parameters in GUI

Parameter	Description	Default value
Database	Selects the database to search through. Available databases are blastn, blastp and cdd.	ncbi-blastn
Database	Select the database to search through.	
Expected value	This parameter specifies the statistical significance threshold of reporting matches against the database sequences.	10
Results limit	The maximum number of results.	10
Megablast	Use megablast.	False
Short sequence	Optimizes search for short sequences.	False
Entrez query	Enter an Entrez query to limit search.	
Annotate as	Name of the result annotations.	
BLAST output	Location of the BLAST output file. This parameter insignificant for cdd search.	
Gap costs	Cost to create and extend a gap in an alignment.	2 2
Match scores	Reward and penalty for matching and mismatching bases.	1 -3

Parameters in Workflow File

Type: blast-ncbi

Parameter	Parameter in the GUI	Type
db	**Database**	*string* Available values are: • ncbi-blastn • ncbi-blastp • ncbi-cdd
db	**Database**	*string*
e-val	**Expected value**	*string*
hits	**Results limit**	*numeric*
megablast	**Megablast**	*boolean*
short-sequence	**Short sequence**	*boolean*

entrez-query	**Entrez query**	*string*
result-name	**Annotate as**	*string*
blast-output	**BLAST output**	*string*
gap-costs	**Gap costs**	*string*
match-scores	**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*

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.

Parameters in GUI

Parameter	Description	Default value
Substitution Matrix	Describes the rate at which one character in a sequence changes to other character states over time.	Auto
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
Filter Results	Specifies either to filter the intersected results or to return all the results.	filter-intersections
Min Score	Minimal percent similarity between a sequence and a pattern.	90%
Search in	Specifies which strands should be searched: direct, complementary or both.	both strands
Search in Translation	Translates a supplied nucleotide sequence to protein and searches in the translated sequence.	False
Gap Open Score	Penalty for opening a gap.	-10.0
Gap Extension Score	Penalty for extending a gap.	-1.0

Use Pattern Names	Use a pattern name as an annotation name.	True
Annotate as	Name of the result annotations.	misc_feature
Qualifier name for pattern name	Name of qualifier in result annotations which is containing a pattern name.	pattern name

Parameters in Workflow File

Type: ssearch

Parameter	Parameter in the GUI	Type
matrix	**Substitution Matrix**	*string* Available values are: • Auto - for auto detecting matrix • blosum60 • dna • rna • ...
algorithm	**Algorithm**	*string* Available values are: • Classic 2 • SSE2 • OpenCL • CUDA
filter-strategy	**Filter Results**	*string* Available values are: • filter-intersections • none
min-score	**Min Score**	*numeric*
strand	**Search in**	*numeric* Available values are: • 0 - for searching in both strands • 1 - for searching in direct strand • 2 - for searching in complement strand
amino	**Search in Translation**	*boolean*
gap-open-score	**Gap Open Score**	*numeric*
gap-ext-score	**Gap Extension Score**	*numeric*
use-names	**Use Pattern Names**	*boolean*
result-name	**Annotate as**	*string*
pattern-name-qual	**Qualifier name for pattern name**	*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.

Parameters in GUI

Parameter	Description	Default value
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
Custom directory	Specify the output directory.	
Genome	File with genome length.	human.hg18
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.	

Block size	Number of items to bundle in r-tree (-blockSize).	256
Items per slot	Number of data points bundled at lowest level (-itemsPerSlot).	1024
Uncompressed	If set, do not use compression.(-unc).	False

Parameters in Workflow File

Type: bgtbw-bam

Parameter	Parameter in the GUI	Type
out-mode	**Output directory**	*numeric*
custom-dir	**Custom directory**	*string*
genome	**Genome**	*string*
out-name	**Output name**	*string*
bs	**Block size**	*numeric*
its	**Items per slot**	*numeric*
unc	**Uncompressed**	*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.

Parameters in GUI

Parameter	Description	Default value
Sequence name (required)	Result sequence name.	*Sequence*

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*
Skip unknown symbols	If *True*, ignores all symbols that are not presented in the sequence alphabet selected.	*True*
Replace unknown symbols with	Replaces all unknown symbols with the specified symbol.	*N*

Parameters in Workflow File

Type: convert-text-to-sequence

Parameter	Parameter in the GUI	Type
sequence-name	**Sequence name**	*string*
alphabet	**Alphabet**	*string*
skip-unknown	**Skip unknown symbols**	*boolean*
replace-unknown-with	**Replace unknown symbols 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 selected format if it is not excluded.

Parameters in GUI

Parameter	Description	Default value
Document format	Document format of output file.	

Excluded formats	Input file won't be converted to any of selected formats.	

Parameters in Workflow File

Type: files-conversion

Parameter	Parameter in the GUI	Type
document-format	**Document format**	*string*
excluded-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.

Parameters in GUI

Parameter	Description	Default value
Operation type	Selects either to produce the reverse, complement, or reverse-complement sequence.	Reverse Complement

Parameters in Workflow File

Type: reverse-complement

Parameter	Parameter in the GUI	Type
op-type	**Operation 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).

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 to 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.

Parameters in GUI

Parameter	Description	Default value
Output file	Write assembly results to this output file in ACE format..	result.ace
Quality cutoff for clipping	Base quality cutoff for clipping (-c).	12

Clipping range	Set a number which unit is base. It will get the refGenes in n bases from peak center. (--distance).	100
Quality cutoff for differeneces	Base quality cutoff for differences (-b).	20
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
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
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
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
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 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 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

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
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
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
Assembly reverse reads	Specifies whether to consider reads in reverse orientation for assembly (originally, parameter -r of CAP3).	True
CAP3 tool path	The path to the CAP3 external tool in UGENE.	default
Temporary directory	The directory for temporary files.	default

Parameters in Workflow File

Type: cap3

Parameter	Parameter in the GUI	Type
out-file	**Output file**	*string*
clipping-cutoff	**Quality cutoff for clipping**	*numeric*
clipping-range	**Clipping range**	*numeric*
diff-cutoff	**Quality cutoff for differeneces**	*numeric*
diff-max-qscore	**Maximum difference score**	*numeric*
match-score-factor	**Match score factor**	*numeric*
mismatch-score-factor	**Mismatch score factor**	*numeric*
gap-penalty-factor	**Gap penalty factor**	*numeric*
overlap-sim-score-cutoff	**Overlap similarity score cutoff**	*numeric*

overlap-length-cutoff	**Overlap length cutoff**	*numeric*
overlap-perc-id-cutoff	**Overlap percent identity cutoff**	*numeric*
max-num-word-matches	**Max number of word matches**	*numeric*
band-exp-size	**Band expansion size**	*numeric*
max-gap-in-overlap	**Max gap length in an overlap**	*numeric*
assembly-reverse	**Assembly reverse reads**	*boolean*
path	**CAP3 tool path**	*string*
tmp-dir	**Temporary directory**	*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.

Parameters in GUI

Parameter	Description	Default value
Profile name	Descriptive name of the HMM profile.	
HMM strategy	Specifies the kind of alignments you want to allow.	hmmls
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
Parallel calibration	Number of parallel threads that the calibration will run in.	1
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

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
Mean length of samples	Mean length of the synthetic sequences, positive real number.	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.	5000
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

Parameters in Workflow File

Type: hmm2-build

Parameter	Parameter in the GUI	Type
profile-name	**Profile name**	*string*
strategy	**HMM strategy**	*numeric* Available values are: • 0 - for hmms • 1 - for hmmls • 2 - for hmmfs • 3 - for hmmsw
calibrate	**Calibrate profile**	*boolean*
calibration-threads	**Parallel calibration**	*numeric*
deviation	**Standard deviation**	*numeric*
fix-samples-length	**Fixed length of samples**	*numeric*
mean-samples-length	**Mean length of samples**	*numeric*
samples-num	**Number of samples**	*numeric*
seed	**Random 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 outputs united set of annotations for each sequence

Parameters in GUI

Parameter	Description	Default value
Result annotation	Name of the result annotations.	hmm_signal
Filter by high E-value	E-value filtering can be used to exclude low-probability hits from result.	1e-1
Number of seqs	Calculates the E-value scores as if we had seen a sequence database of sequences.	1
Filter by low score	Score based filtering is an alternative to E-value filtering to exclude low-probability hits from result.	-1000000000.0

Parameters in Workflow File

Type: hmm2-search

Parameter	Parameter in the GUI	Type
result-name	**Result annotation**	*string*
e-val	**Filter by high E-value**	*numeric*
seqs-num	**Number of seqs**	*numeric*
score	**Filter by low 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.

Parameters in GUI

Parameter	Description	Default value
Input files (required)	Semicolon-separated list of paths to the input files.	

Parameters in Workflow File

Type: hmm2-read-profile

Parameter	Parameter in the GUI	Type
url-in	**Input files**	*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 specified location.

Parameters in GUI

Parameter	Description	Default value
Output file (required)	Location of the output data file. If this attribute is set, the “Location” slot is not taken into account.	
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

Parameters in Workflow File

Type: hmm2-write-profile

Parameter	Parameter in the GUI	Type
url-out	**Output file**	*string*
write-mode	**Existing file**	*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.

Parameters in GUI

Parameter	Description	Default value
Random seed	Random generator seed. 0 - means that one-time arbitrary seed will be used.	0

Parameters in Workflow File

Type: hmm3-build

Parameter	Parameter in the GUI	Type
seed	**Random 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 outputs united set of annotations for each sequence.

Parameters in GUI

Parameter	Description	Default value
Result annotation	Name of the result annotations.	hmm_signal
Seed	Random generator seed. 0 - means that one-time arbitrary seed will be used.	0
Filter by high E-value	E-value filtering can be used to exclude low-probability hits from result.	1e-1
Filter by low score	Score based filtering is an alternative to E-value filtering to exclude low-probability hits from result.	0.01

Parameters in Workflow File

Type: hmm3-search

Parameter	Parameter in the GUI	Type
result-name	**Result annotation**	*string*
seed	**Seed**	*numeric*
seqs-num	**Number of seqs**	*numeric*
score	**Filter by low 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.

Parameters in GUI

Parameter	Description	Default value
Input files (required)	Semicolon-separated list of paths to the input files.	

Parameters in Workflow File

Type: hmm3-read-profile

Parameter	Parameter in the GUI	Type
url-in	**Input files**	*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 specified location.

Parameters in GUI

Parameter	Description	Default value
Output file	Location of the output data file. If this attribute is set, the “Location” slot is not taken into account.	
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

Parameters in Workflow File

Type: hmm3-write-profile

Parameter	Parameter in the GUI	Type

url-out	**Output file**	*string*
write-mode	**Existing file**	*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 to Reference 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
- 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 to Reference Element

Align input sequences (e.g. Sanger reads) to the reference sequence.

Parameters in GUI

Parameter	Description	Default value
Reference URL	A URL to the file with a reference sequence.	

Parameters in Workflow File

Type: align-to-reference

Parameter	Parameter in the GUI	Type
reference	**Reference URL**	*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*

Align with ClustalO Element

Aligns multiple sequence alignments (MSAs) supplied with ClustalO.

Parameters in GUI

Parameter	Description	Default value
Number of iterations	Number of (combined guide-tree/HMM) iterations.	1
Number of guidetree iterations	Maximum number guidetree iterations.	0
Number of HMM iterations	Maximum number of HMM iterations.	0
Set auto options	Set options automatically (might overwrite some of your options).	False

Tool path	Path to the ClustalO tool. The default path can be set in the UGENE application settings.	Default
Temporary directory	Directory to store temporary files.	Default

Parameters in Workflow File

Type: ClustalO

Parameter	Parameter in the GUI	Type
num-iterations	**Number of iterations**	*numeric*
max-guidetree-iterations	**Number of guidetree iterations**	*numeric*
max-hmm-iterations	**Number of HMM iterations**	*numeric*
set-auto	**Set auto options**	*boolean*
path	**Tool path**	*string*
temp-dir	**Temporary directory**	*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 main UGENE User Manual.

Parameters in GUI

Parameter	Description	Default value

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
End gaps	The penalty for closing a gap.	False
Gap distance	The gap separation penalty. Tries to decrease the chances of gaps being too close to each other.	4.42
Gap extension penalty	The penalty for extending a gap.	8.52
Gap open penalty	The penalty for opening a gap.	53.90
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
Residue-specific gaps off	Residue-specific penalties are amino specific gap penalties that reduce or increase the gap opening penalties at each position in the alignment.	False
Iteration type	Alignment improvement iteration type.	None
Number of iterations	The maximum number of iterations to perform.	3
Tool path (required)	Path to the ClustalW tool. The default path can be set in the UGENE Application Settings.	default
Temporary directory	Directory to store temporary files.	default

Parameters in Workflow File

Type: clustalw

Parameter	Parameter in the GUI	Type
matrix	**Weight 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
close-gap-penalty	**End gaps**	*boolean*
gap-distance	**Gap distance**	*numeric*
gap-ext-penalty	**Gap extension penalty**	*numeric*
gap-open-penalty	**Gap open penalty**	*numeric*
no-hydrophilic-gaps	**Hydrophilic gaps off**	*boolean*
no-residue-specific-gaps	**Residue-specific gaps off**	*boolean*

iteration-type	**Iteration type**	*numeric* Available values are: • 0 - for None • 1 - for Tree • 2 - for Alignment
iterations-max-num	**Number of iterations**	*numeric*
path	**Tool path**	*string*
temp-dir	**Temporary directory**	*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.

Parameters in GUI

Parameter	Description	Default value
Gap extension penalty	The penalty for extending a gap.	8.52
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
Terminal gap penalty	The penalty to extend gaps from the N/C terminal of protein or 5'/3' terminal of nucleotide sequences.	4.42
Bonus score	A bonus score that is added to each pair of aligned residues.	0.02

Parameters in Workflow File

Type: kalign

Parameter	Parameter in the GUI	Type
gap-ext-penalty	**Gap extension penalty**	*numeric*
gap-open-penalty	**Gap open penalty**	*numeric*
terminal-gap-penalty	**Terminal gap penalty**	*numeric*
bonus-score	**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.

Parameters in GUI

Parameter	Description	Default value
Offset	Works like gap extension penalty.	0
Gap open penalty	Gap open penalty.	1.53
Max iteration	Maximum number of iterative refinement.	0
Tool path (default)	Path to the ClustalW tool. The default path can be set in the UGENE application settings.	default
Temporary directory	Directory to store temporary files.	default

Parameters in Workflow File

Type: mafft

Parameter	Parameter in the GUI	Type
gap-ext-penalty	**Offset**	*numeric*
gap-open-penalty	**Gap open penalty**	*numeric*
iterations-max-num	**Max iteration**	*numeric*
path	**Tool path**	*string*
temp-dir	**Temporary directory**	*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.

Parameters in GUI

Parameter	Description	Default value
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
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

Parameters in Workflow File

Type: muscle

Parameter	Parameter in the GUI	Type

mode	**Mode**	*numeric* Availables values are: • 0 - for MUSCLE default • 1 - for Large alignment • 2 - for Refine only
stable	**Stable order**	*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.

Parameters in GUI

Parameter	Description	Default value
Gap extension penalty	Gap Extension Penalty. Positive values give rewards to gaps and prevent the alignment of unrelated segments.	0
Gap open penalty	Gap open penalty. Must be negative, best matches get a score of 1000.	-50
Max iteration	Number of iteration on the progressive alignment. 0 - no iteration, -1 - Nseq iterations.	0
Tool path (required)	Path to the ClustalW tool. The default path can be set in the UGENE Application Settings.	default
Temporary directory	Directory to store temporary files.	default

Parameters in Workflow File

Type: tcoffee

Parameter	Parameter in the GUI	Type
gap-ext-penalty	**Offset**	*numeric*
gap-open-penalty	**Gap open penalty**	*numeric*
iterations-max-num	**Max iteration**	*numeric*
path	**Tool path**	*string*
temp-dir	**Temporary directory**	*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.

Parameters in GUI

Parameter	Description	Default value
Algorithm	The algorithm of consensus extracting.	
Threshold	The threshold of the algorithm.	100
Keep gaps	Set this parameter if the result consensus must keep the gaps.	True

Parameters in Workflow File

Type: extract-msa-consensus-sequence

Parameter	Parameter in the GUI	Type
algorithm	**Algorithm**	*string*
threshold	**Threshold**	*numeric*
keep-gaps	**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.

Parameters in GUI

Parameter	Description	Default value
Algorithm	The algorithm of consensus extracting.	
Threshold	The threshold of the algorithm.	100

Parameters in Workflow File

Type: extract-msa-consensus-string

Parameter	Parameter in the GUI	Type
algorithm	**Algorithm**	*string*
threshold	**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.

Parameters in GUI

Parameter	Description	Default value
Primers URL	A URL to the input file with primer pairs.	
Report URL	A URL to the output file with the PCR report.	
Mismatches	Number of allowed mismatches.	3
Min perfect match	Number of bases that match exactly on 3' end of primers.	15
Max product size	Maximum size of amplified region.	5000

Parameters in Workflow File

Type: in-silico-pcr

Parameter	Parameter in the GUI	Type
primers-url	**Primers URL**	*string*
report-url	**Report URL**	*string*
mismatches	**Mismatches**	*numeric*
perfect-match	**Min perfect match**	*numeric*
max-product	**Max product size**	*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*

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: Align Short Reads

- Align Reads with Bowtie Element
- Align Reads with Bowtie2 Element
- Align Reads with BWA Element
- Align Reads with BWA-MEM Element
- Align Reads with UGENE Genome Aligner Element

Align Reads with Bowtie Element

Performs alignment of short reads with Bowtie.

Parameters in GUI

Parameter	Description	Default value
Output directory	Directory to save Bowtie output files.	

Reference genome	Path to indexed reference genome.	
Output file name	Base name of the output file. 'out.sam' by default.	out.sam
Library	Is this library mate-paired?	single-end
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 number	Mismatches number.	2
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
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
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
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
Seed	Use as the seed for pseudo-random number generator.	0

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
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
No forward orientation	If --nofw is specified, bowtie will not attempt to align against the forward reference strand.	False
No reverse-complement orientation	If --norc is specified, bowtie will not attempt to align against the reverse-complement reference strand.	False
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
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
All alignment	Report all valid alignments per read or pair.	False

Parameters in Workflow File

Type: align-reads-with-bowtie

Parameter	Parameter in the GUI	Type
output-dir	**Output directory**	*string*
reference	**Reference genome**	*string*
outname	**Output file name**	*string*
library	**Library**	*string*
mismatches_type	**Mode**	*string*
mismatches_number	**Mismatches number**	*numeric*
maqerr	**Mismatches number**	*numeric*
seedLen	**Seed length**	*numeric*
maxbts	**Maximum of backtracks**	*numeric*
chunkmbs	**Best hits**	*numeric*

seed	**Seed**	*numeric*
colorspace	**Colorspace**	*boolean*
nomaqround	**No Maq rounding**	*boolean*
nofw	**No forward orientation**	*boolean*
norc	**No reverse-complement orientation**	*boolean*
tryhard	**Try as hard**	*boolean*
best	**Best alignments**	*boolean*
all	**All alignment**	*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*

Align Reads with Bowtie2 Element

Performs alignment of short reads with Bowtie2.

Parameters in GUI

Parameter	Description	Default value
Output directory	Directory to save Bowtie2 output files.	
Reference genome	Path to indexed reference genome.	
Output file name	Base name of the output file. 'out.sam' by default.	out.sam
Library	Is this library mate-paired?	single-end

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
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
Seed length (--L)	Sets the length of the seed substrings to align. Smaller values make alignment slower but more senstive.	20
Add columns to allow gaps (--dpad)	"Pads" dynamic programming problems by specified number of columns on either side to allow gaps.	15
Disallow gaps (--gbar)	Disallow gaps within specified number of positions of the beginning or end of the read.	4
Seed (--seed)	Use as the seed for pseudo-random number generator.	0
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
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 alignment status of pairs per se, not individual mates.	False
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
No forward orientation (--nofw)	If --nofw is specified, bowtie will not attempt to align against the forward reference strand.	False
No reverse-complement orientation (--norc)	If --norc is specified, bowtie will not attempt to align against the reverse-complement reference strand.	False

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
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

Parameters in Workflow File

Type: align-reads-with-bowtie2

Parameter	Parameter in the GUI	Type
output-dir	**Output directory**	*string*
reference	**Reference genome**	*string*
outname	**Output file name**	*string*
library	**Library**	*string*
mode	**Mode**	*string*
mismatches_number	**Number of mismatches**	*numeric*
seed_len	**Seed length (--L)**	*numeric*
dpad	**Add columns to allow gaps (--dpad)**	*numeric*
gbar	**Disallow gaps (--gbar)**	*numeric*
seed	**Seed (--seed)**	*numeric*
threads	**Threads**	*numeric*
nomixed	**No unpaired alignments (--no-mixed)**	*boolean*
nodiscordant	**No discordant alignments (--no-discordant)**	*boolean*
nofw	**No forward orientation (--nofw)**	*boolean*
norc	**No reverse-complement orientation (--norc)**	*boolean*
nooverlap	**No overlapping mates (--no-overlap)**	*boolean*
nocontain	**No mates containing one another (--no-contain)**	*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*

Align Reads with BWA Element

Performs alignment of short reads with BWA.

Parameters in GUI

Parameter	Description	Default value
Output directory	Directory to save BWA-MEM output files.	
Reference genome	Path to indexed reference genome.	
Output file name	Base name of the output file. 'out.sam' by default.	out.sam
Library	Is this library mate-paired?	single-end
Use missing prob	Use missing prob instead maximum edit distance.	True
Missing prob	Missing prob (-n).	0.04
Seed length	Seed length (-l).	32
Max gap opens	Max gap opens (-o).	1
Index algorithm	Index algorithm (-a).	is
Best hits	Best hits (-R).	30
Long-scaled gap penalty for long deletions	Long-scaled gap penalty for long deletions (-L).	False
Non iterative mode	Non iterative mode (-N).	False
Enable long gaps	Enable long gaps.	True
Max gap extensions	Max gap extensions (-e).	0
Indel offset	Indel offset (-i).	5
Max long deletions extensions	Max long deletions extensions(-d).	10
Barcode length	Barcode length (-B).	0
Max queue entries	Max queue entries (-m).	2000000
Threads	Threads (-t).	4
Max seed differencies	Max seed differencies (-k).	2
Mismatch penalty	Mismatch penalty (-M).	3
Gap open penalty	Gap open penalty (-O).	11
Gap extension penalty	Gap extension penalty; a gap of size k cost (-E).	4
Quality threshold	Quolity threshold (-q).	0

Parameters in Workflow File

Type: align-reads-with-bwa

Parameter	Parameter in the GUI	Type
output-dir	**Output directory**	*string*
reference	**Reference genome**	*string*
outname	**Output file name**	*string*
library	**Library**	*string*
use-miss-prob	**Use missing prob**	*boolean*
missing-prob	**Missing prob**	*numeric*
seed-length	**Seed length**	*numeric*
max-gap	**Max gap opens**	*numeric*
index-alg	**Index algorithm**	*string*
best-hits	**Best hits**	*numeric*
scaled-gap	**Long-scaled gap penalty for long deletions**	*boolean*
non-iterative	**Non iterative mode**	*boolean*
enable-long-gaps	**Enable long gaps**	*boolean*
gap-extensions	**Max gap extensions**	*numeric*
indel-offset	**Indel offset**	*numeric*
long-deletions	**Max long deletions extensions**	*numeric*
barcode-length	**Barcode length**	*numeric*
max-queue	**Max queue entries**	*numeric*
num-threads	**Threads**	*numeric*
max-seed	**Max seed differencies**	*numeric*
mistmatch-penalty	**Mismatch penalty**	*numeric*
gap-open-penalty	**Gap open penalty**	*numeric*
gap-ext-penalty	**Gap extension penalty**	*numeric*
quality-threshold	**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*

Align Reads with BWA-MEM Element

Performs alignment of short reads with BWA-MEM.

Parameters in GUI

Parameter	Description	Default value
Output directory	Directory to save BWA-MEM output files.	
Reference genome	Path to indexed reference genome.	
Output file name	Base name of the output file. 'out.sam' by default.	out.sam
Library	Is this library mate-paired?	single-end
Number of threads	Number of threads (-t).	1
Min seed length	Path to indexed reference genome (-k).	19
Index algorithm	Index algorithm (-a).	autodetect
Band width	Band width for banded alignment (-w).	100
Dropoff	Off-diagonal X-dropoff (-d).	100
Internal seed length	Look for internal seeds inside a seed longer than {-k} (-r).	1.50000
Skip seed threshold	Skip seeds with more than INT occurrences (-c).	10000
Drop chain threshold	Drop chains shorter than FLOAT fraction of the longest overlapping chain (-D).	0.5
Rounds of made rescues	Perform at most INT rounds of mate rescues for each read (-m).	100
Skip mate rescue	Skip mate rescue (-S).	False
Skip pairing	Skip pairing; mate rescue performed unless -S also in use (-P).	False
Matching score	Score for a sequence match (-A).	1
Mismatch penalty	Penalty for a mismatch (-B).	4
Gap open penalty	Gap open penalty (-O).	6
Gap extension penalty	Gap extension penalty; a gap of size k cost {-O} (-E).	1
Penalty for clipping	Penalty for clipping (-L).	5
Penalty unpaired	Penalty for an unpaired read pair (-U).	17
Score threshold	Minimum score to output (-T).	30

Parameters in Workflow File

Type: bwamem-id

Parameter	Parameter in the GUI	Type
output-dir	**Output directory**	*string*
reference	**Reference genome**	*string*
outname	**Output file name**	*string*
library	**Library**	*string*
threads	**Number of threads**	*numeric*
min-seed	**Min seed length**	*numeric*
index-alg	**Index algorithm**	*string*
band-width	**Band width**	*numeric*
dropoff	**Dropoff**	*numeric*
seed-lookup	**Internal seed length**	*numeric*
seed-threshold	**Skip seed threshold**	*numeric*
drop-chains	**Drop chain threshold**	*numeric*
mate-rescue	**Rounds of made rescues**	*numeric*
skip-mate-rescues	**Skip mate rescue**	*boolean*
skip-pairing	**Skip pairing**	*boolean*
match-score	**Matching score**	*numeric*
mistmatch-penalty	**Mismatch penalty**	*numeric*
gap-open-penalty	**Gap open penalty**	*numeric*
gap-ext-penalty	**Gap extension penalty**	*numeric*
clipping-penalty	**Penalty for clipping**	*numeric*
inpaired-panalty	**Penalty unpaired**	*numeric*
score-threshold	**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*

Align Reads with UGENE Genome Aligner Element

Unique UGENE algorithm for aligning short reads to reference genome.

Parameters in GUI

Parameter	Description	Default value
Output file name	Base name of the output file. 'out.sam' by default.	out.sam
Reference genome	Path to indexed reference genome.	
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
Absolute mismatches	Number of mismatches allowed while aligning reads.	0
Align reverse complement reads	Set this option to align both direct and reverse complement reads.	False
Use "best"-mode	Report only the best alignment for each read (in terms of mismatches).	True
Omit reads with qualities lower than	Omit reads with qualities lower than the specified value. Reads that have no qualities are not omited. Set "0" to switch off this option.	0

Parameters in Workflow File

Type: genome-aligner

Parameter	Parameter in the GUI	Type
outname	**Output file name**	*string*
reference	**Reference genome**	*string*
if-absolute-mismatches-value	**Is absolute mismatches values?**	*boolean*
absolute-mismatches	**Absolute mismatches**	*numeric*
reverse	**Align reverse complement reads**	*boolean*
best	**Use "best"-mode**	*boolean*
quality-threshold	**Omit reads with qualities lower than**	*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*

NGS: Basic Functions

- Assemble Genomes with SPAdes Element
- 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
- Merge BAM Files Element
- Remove Duplicates in BAM Files Element
- Slopbed Element
- Sort BAM Files Element

Assemble Genomes with SPAdes Element

Performes assembly of input short reads.

Parameters in GUI

Parameter	Description	Default value
Output directory	Directory to save Spades output files.	
Dataset type	Input dataset type.	Multi Cell
Running mode	Running mode.	Error correction and assembly
K-mers	k-mer sizes (-k).	auto
Number of threads	Number of threads (-t).	16
Memory limit (Gb)	Memory limit (-m).	250

Parameters in Workflow File

Type: spades-id

Parameter	Parameter in the GUI	Type
output-dir	**Output directory**	*string*
dataset-type	**Dataset type**	*string*
running-mode	**Running mode**	*string*
k-mer	**K-mers**	*numeric*
threads	**Number of threads**	*numeric*
memlimit	**Memory limit (Gb)**	*numeric*

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*

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.

Parameters in GUI

Parameter	Description	Default value
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
Custom directory	Specify the output directory.	
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.	

Parameters in Workflow File

Type: CASAVAFilter

Parameter	Parameter in the GUI	Type
out-mode	**Output directory**	*numeric*
custom-dir	**Custom directory**	*string*
out-name	**Output file 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.

Parameters in GUI

Parameter	Description	Default value
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
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.	
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.	
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.	
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.	

Parameters in Workflow File

Type: CutAdaptFastq

Parameter	Parameter in the GUI	Type
out-mode	**Output directory**	*string*
out-name	**Output file name**	*string*
adapters-url	**FASTA file with 3' adapters**	*string*
front-url	**FASTA file with 5' adapters**	*string*
anywhere-url	**FASTA file with 5' and 3' adapters**	*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.

Parameters in GUI

Parameter	Description	Default value
Algorithm	The algorithm of consensus extracting.	Default
Keep gaps	Set this parameter if the result consensus must keep the gaps.	True

Parameters in Workflow File

Type: extract-consensus

Parameter	Parameter in the GUI	Type
algorithm	**Algorithm**	*string*
keep-gaps	**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.

Parameters in GUI

Parameter	Description	Default value
Output file	Location of output data file. If this attribute is set, slot "Location" in port will not be used.	assembly_coverage.txt
Export	Data type to export.	coverage
Treshold	The minimum coverage value to export.	1

Parameters in Workflow File

Type: extract-assembly-coverage

Parameter	Parameter in the GUI	Type
url-out	**Output file**	*string*
export-type	**Export**	*string*
threshold	**Treshold**	*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.

Parameters in GUI

Parameter	Description	Default value
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.	
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.	

Parameters in Workflow File

Type: MergeFastq

Parameter	Parameter in the GUI	Type
out-mode	**Output directory**	*string*
out-name	**Output file 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.

Parameters in GUI

Parameter	Description	Default value

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
Custom directory	Specify the output directory.	
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.	
Quality threshold	Quality threshold for trimming.	30
Min Length	Too short reads are discarded by the filter.	0
Trim both ends	Trim the both ends of a read or not. Usually, you need to set True for Sanger se quencing and False for NGS	True

Parameters in Workflow File

Type: QualityTrim

Parameter	Parameter in the GUI	Type
out-mode	**Output directory**	*numeric*
custom-dir	**Custom directory**	*string*
out-name	**Output file name**	*string*
qual-id	**Quality threshold**	*numeric*
len-id	**Min Length**	*numeric*
both-ends	**Trim 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.

Parameters in GUI

Parameter	Description	Default value
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
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.	

Parameters in Workflow File

Type: fastqc

Parameter	Parameter in the GUI	Type
out-mode	**Output directory**	*string*
adapter	**List of adapters**	*string*
contaminants	**List of 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.

Parameters in GUI

Parameter	Description	Default value
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.	
Custom directory	Custom output directory.	

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.	
Output format	Format of an output assembly file.	bam
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.	0
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.	

Parameters in Workflow File

Type: filter-bam

Parameter	Parameter in the GUI	Type
out-mode	**Output directory**	*numeric*
custom-dir	**Custom directory**	*string*
out-name	**Output name**	*string*
out-format	**Output format**	*string*
region	**Region**	*string*
mapq	**MAPQ threshold**	*numeric*
flag	**Skip 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.

Parameters in GUI

Parameter	Description	Default value
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
Custom directory	Specify the output directory.	
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.	
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
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
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
Strand	Calculate coverage of intervals from a specific strand. With BED files, requires at least 6 columns (strand is column 6) (-strand).	False
5 prime	Calculate coverage of 5â positions (instead of entire interval) (-5).	False
3 prime	Calculate coverage of 3â positions (instead of entire interval) (-3).	False
Max	Combine all positions with a depth >= max into a single bin in the histogram (-max).	2147483647

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
Trackline	Adds a UCSC/Genome-Browser track line definition in the first line of the output (-trackline).	False
Trackopts	Writes additional track line definition parameters in the first line (-trackopts).	

Parameters in Workflow File

Type: genomecov

Parameter	Parameter in the GUI	Type
out-mode	**Output directory**	*numeric*
custom-dir	**Custom directory**	*string*
out-name	**Output file name**	*string*
genome	**Genome**	*string*
mode-id	**Report mode**	*numeric*
split-id	**Split**	*boolean*
strand-id	**Strand**	*boolean*
prime5-id	**5 prime**	*boolean*
prime3-id	**3 prime**	*boolean*
max-id	**Max**	*numeric*
scale-id	**Scale**	*numeric*
trackline-id	**Trackline**	*boolean*
trackopts-id	**Trackopts**	*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*

Merge BAM Files Element

Merge BAM files using SAMTools merge.

Parameters in GUI

Parameter	Description	Default value
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.	
Custom directory	Custom output directory.	
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.	

Parameters in Workflow File

Type: merge-bam

Parameter	Parameter in the GUI	Type
out-mode	**Output directory**	*numeric*
custom-dir	**Custom directory**	*string*
out-name	**Output 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.

Parameters in GUI

Parameter	Description	Default value

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
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.	
Remove for single-end reads	Remove duplicate for single-end reads. By default, the command works for paired-end reads only (-s).	False
Treat as single-end	Treat paired-end reads and single-end reads (-S).	False

Parameters in Workflow File

Type: rmdup-bam

Parameter	Parameter in the GUI	Type
out-mode	**Output directory**	*numeric*
out-name	**Output file name**	*string*
remove-single-end	**Remove for single-end reads**	*boolean*
treat_reads	**Treat as single-end**	*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.

Parameters in GUI

Parameter	Description	Default value

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
Custom directory	Specify the output directory.	
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.	
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
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
Substract from start	The number of base pairs to subtract from the start coordinate. Enter 0 to disable (-l).	0
Add to end	The number of base pairs to add to the end coordinate. Enter 0 to disable (-r).	0
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
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
Print header	Print the header from the input file prior to results (-header).	False

Parameters in Workflow File

Type: slopbed

Parameter	Parameter in the GUI	Type
out-mode	**Output directory**	*numeric*
custom-dir	**Custom directory**	*string*
out-name	**Output file name**	*string*
genome-id	**Genome**	*string*
b-id	**Each direction increase**	*numeric*
l-id	**Substract from start**	*numeric*
r-id	**Add to end**	*numeric*
s-id	**Strand-based**	*boolean*
pct-id	**As fraction**	*boolean*
header-id	**Print header**	*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.

Parameters in GUI

Parameter	Description	Default value
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
Custom directory	Specify the output directory.	
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.	
Build index	Build index for the sorted file with SAMTools index.	human.hg18

Parameters in Workflow File

Type: Sort-bam

Parameter	Parameter in the GUI	Type
out-mode	**Output directory**	*numeric*
custom-dir	**Output BAM name**	*string*
out-name	**Output file name**	*string*
index	**Build 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.

Parameters in GUI

Parameter	Description	Default value
Genome file	Select a genome file (sqlite3 file) to search refGenes. (--genome).	hg19
Output file	Select which type of genes need to output. up for genes upstream to peak summit, do wn for genes downstream to peak summit, all for both up and down. (--op).	all
Official gene symbols	Output official gene symbol instead of refse q name. (--symbol).	False
Distance	Set a number which unit is base. It will get the refGenes in n bases from peak center. (--distance).	3000

Parameters in Workflow File

Type: peak2gene-id

Parameter	Parameter in the GUI	Type
genome	**Genome file**	*string*
outpos	**Output file**	*string*
symbol	**Official gene symbols**	*boolean*
distance	**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 scores profiles.

Parameters in GUI

Parameter	Description	Default value
Output file	File to store phastcons results (BMP).	
Title	Title of the figure (--title).	Average Phastcons around the Center of Sites
Label	Label of data in the figure (--bed-label).	Conservation_at_peak_summits
Assembly version	The directory to store phastcons scores (--phasdb).	hg19
Window width	Window width centered at middle of regions (-w).	1000
Height	Height of plot (--height).	1000
Width	Width of plot (--width).	1000

Parameters in Workflow File

Type: conservation_plot-id

Parameter	Parameter in the GUI	Type
output-file	**Output file**	*string*
title	**Title**	*string*
label	**Label**	*string*
assembly_version	**Assembly version**	*string*
windos_s	**Window width**	*numeric*
height	**Height**	*numeric*
width	**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.

Parameters in GUI

Parameter	Description	Default value
Output directory	The directory to store seqpos results.	
Genome assembly version	UCSC database version (GENOME).	hg19
Output file name	Name of the output file which stores new motifs found during a de novo search (-n).	Default
De novo motifs	Run de novo motif search (-d).	False
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
Region width	Width of the region to be scanned for motifs; depends on a resolution of assay (-w).	600
Pvalue cutoff	Pvalue cutoff for the motif significance (-p).	0.001

Parameters in Workflow File

Type: seqpos-id

Parameter	Parameter in the GUI	Type
output-dir	**Output directory**	*string*
assembly	**Genome assembly version**	*string*
out_name	**Output file name**	*string*
de_novo	**De novo motifs**	*boolean*
motif_db	**Motif database**	*string*
reg_width	**Region width**	*numeric*
p_val	**Pvalue cutoff**	*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.

Parameters in GUI

Parameter	Description	Default value
Output directory	The directory to store Conduct GO results.	
Title	Title is used to name the output files - so make it meaningful.	Default
Gene Universe	Select a gene universe.	hgu133a.db

Parameters in Workflow File

Type: conduct-go-id

Parameter	Parameter in the GUI	Type
output-dir	**Output directory**	*string*
title	**Title**	*string*
gene-universe	**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

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.

Parameters in GUI

Parameter	Description	Default value
Output report file	Path to the report output file. Result for CEAS analysis.	
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).	
Gene annotations table	Path to gene annotation table (e.g. a refGene table in sqlite3 db format (--gt).	hg19
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

Wiggle profiling resolution	Wiggle profiling resolution. WARNING: Value smaller than the wig interval (resolution) may cause aliasing error. (--pf-res).	50
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
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
Relative distance	Relative distance to TSS/TTS in WIGGLE file profiling (--rel-dist).	3000
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).	
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).	

Parameters in Workflow File

Type: ceas-report

Parameter	Parameter in the GUI	Type
image-file	**Output report file**	*string*
anns-file	**Output annotations file**	*string*
anns-table	**Gene annotations table**	*string*
span	**Span size**	*numeric*
profiling-resolution	**Wiggle profiling resolution**	*numeric*
promoter-sizes	**Promoter/downstream interval**	*numeric*
promoter-bisizes	**BiPromoter ranges**	*string*
relative-distance	**Relative distance**	*string*
group-files	**Gene group files**	*string*
group-names	**Gene 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.

Parameters in GUI

Parameter	Description	Default value
Output directory	Directory to save MACS output files.	
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).	
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 (--wig) (--single-profile).	hg19
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
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
P-value	P-value cutoff. Default is 0.00001, for looser results, try 0.001 instead (--pvalue).	3000
Tag size (optional)	Length of reads. Determined from first 10 reads if not specified (input 0) (--tsize).	5000
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

Use model	Whether or not to use MACS paired peaks model (--nomodel).	
Model fold	Select the regions within MFOLD range of high-confidence enrichment ratio against. M odel 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).	
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).	
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).	
Use lambda	Whether to use local lambda model which can use the local bias at peak regions to throw out false positives (--nolambda).	
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).	
Large nearby region	The large nearby region in basepairs to calculate dynamic lambda. This is used to capture the surround bias (--llocal).	
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).	
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).	

Parameters in Workflow File

Type: macs-id

Parameter	Parameter in the GUI	Type
output-dir	**Output directory**	*string*

file-names	**Name**	*string*
wiggle-output	**Wiggle output**	*boolean*
wiggle-space	**Wiggle space**	*numeric*
genome-size	**Genome size (Mbp)**	*numeric*
p-value	**P-value**	*numeric*
tag-size	**Tag size (optional)**	*numeric*
keep-duplicates	**Keep duplicates**	*string*
use-model	**Use model**	*boolean*
model-fold	**Model fold**	*string*
shift-size	**Shift size**	*numeric*
band-width	**Band width**	*numeric*
use-lambda	**Use lambda**	*boolean*
small-nearby	**Small nearby region**	*numeric*
large-nearby	**Large nearby region**	*numeric*
auto_bimodal	**Auto bimodal**	*boolean*
scale_large	**Scale to large**	*boolean*

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: RNA-Seq Analysis

- Assembly Transcripts with Cufflinks Element
- Extract Transcript Sequences with gffread Element
- Find Splice Junction with TopHat Element
- Merge Assemblies with Cuffmerge Element

- Test for Diff. Expression with Cuffdiff Element

Assembly Transcripts with Cufflinks Element

Cufflinks accepts aligned RNA-Seq reads and assembles the alignments into a parsimonious set of transcripts. Cufflinks then estimates the relative abundances of these transcripts based on how many reads support each one, taking into account biases in library preparation protocols.

Parameters in GUI

Parameter	Description	Default value
Output directory	Directory to save MACS output files.	
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 assembly. Output will include all reference transcripts as well as any novel genes and isoforms that are assembled.	
Library type	Specifies RNA-Seq protocol.	Standart Illumina
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.	False
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
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.	0.15
Cufflinks tool path	The path to the Cufflinks external tool in UGENE.	default
Temporary directory	The directory for temporary files.	default

Parameters in Workflow File

Type: cufflinks

Parameter	Parameter in the GUI	Type
out-dir	**Output directory**	*string*
ref-annotation	**Reference annotation**	*string*
rabt-annotation	**RABT annotation**	*string*
library-type	**Library type**	*numeric*
mask-file	**Mask file**	*string*
multi-read-correct	**Multi-read correct**	*boolean*
min-isoform-fraction	**Min isoform fraction**	*numeric*
frag-bias-correct	**Frag bias correct**	*string*
pre-mrna-fraction	**Pre-mRNA fraction**	*numeric*
path	**Cufflinks tool path**	*string*
tmp-dir	**Temporary directory**	*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.

Parameters in GUI

Parameter	Description	Default value
Output sequences	The url to the output file with the extracted sequences.	

Parameters in Workflow File

Type: gffread

Parameter	Parameter in the GUI	Type
url-out	**Output sequences**	*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*

Find Splice Junction with TopHat Element

TopHat is a fast splice junction mapper for RNA-Seq reads. It aligns RNA-Seq reads to mammalian-sized genomes using the ultra high-throughput short read aligner Bowtie, and then analyzes the mapping results to identify splice junctions between exons.

Parameters in GUI

Parameter	Description	Default value
Output directory	Directory to save MACS output files.	
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.	
Mate inner distance	The expected (mean) inner distance between mate pairs.	200

Mate standard deviation	The standard deviation for the distribution on inner distances between mate pairs.	20
Library type	Specifies RNA-Seq protocol.	Standard Illumins
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.	False
Raw junctions	The list of raw junctions.	
Known transcript file	A set of gene model annotations and/or known transcripts.	
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
Segment length	Each read is cut up into segments, each at least this long. These segments are mapped independently.	25
Fusion search	Turn on fusion mapping.	False
Transcriptome only	Only align the reads to the transcriptome and report only those mappings as genomic mappings.	False
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
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
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
Splice mismatches	The maximum number of mismatches that may appear in the anchor region of a spliced alignment.	0
Read mismatches	Final read alignments having more than these many mismatches are discarded.	2
Segment mismatches	Read segments are mapped independently, allowing up to this many mismatches in each segment alignment.	2

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
Bowtie version	Specifies which Bowtie version should be used.	Bowtie2
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 tool path	The path to the Bowtie external tool.	default
SAMtools tool path	The path to the SAMtools tool. Note that the tool is available in the UGENE External Tool Package.	default
TopHat tool path	The path to the TopHat external tool in UGENE.	default
Temporary directory	The directory for temporary files.	default

Parameters in Workflow File

Type: tophat

Parameter	Parameter in the GUI	Type
out-dir	**Output directory**	*string*
bowtie-index-dir	**Bowtie index directory**	*string*
bowtie-index-basename	**Bowtie index basename**	*string*
mate-inner-distance	**Mate inner distance**	*numeric*
mate-standard-deviation	**Mate standard deviation**	*numeric*
library-type	**Library type**	*numeric*
no-novel-junctions	**No novel junctions**	*boolean*
raw-junctions	**Raw junctions**	*string*
known-transcript	**Known transcript file**	*string*
max-multihits	**Max multihits**	*numeric*
segment-length	**Segment length**	*numeric*
fusion-search	**Fusion search**	*boolean*
transcriptome-only	**Transcriptome only**	*boolean*
transcriptome-max-hits	**Transcriptome max hits**	*numeric*
prefilter-multihits	**Prefilter multihits**	*boolean*
min-anchor-length	**Min anchor length**	*numeric*
splice-mismatches	**Splice mismatches**	*numeric*
read-mismatches	**Read mismatches**	*numeric*
segment-mismatches	**Segment mismatches**	*numeric*
solexa-1-3-quals	**Solexa 1.3 quals**	*boolean*
bowtie-version	**Bowtie version**	*numeric*

bowtie-n-mode	**Bowtie -n mode**	*numeric*
bowtie-tool-path	**Bowtie tool path**	*string*
samtools-tool-path	**SAMtools tool path**	*string*
path	**TopHat tool path**	*string*
temp-dir	**Temporary directory**	*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
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*

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.

Parameters in GUI

Parameter	Description	Default value
Output directory	Directory to save MACS output files.	
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.	
Minimum isoform fraction	Discard isoforms with abundance below this.	0.05

Cuffcompare tool path	The path to the Cuffcompare external tool in UGENE.	default
Cuffmerge tool path	The path to the Cuffmerge external tool in UGENE.	default
Temporary directory	The directory for temporary files.	default

Parameters in Workflow File

Type: cuffmerge

Parameter	Parameter in the GUI	Type
out-dir	**Output directory**	*string*
ref-annotation	**Reference annotation**	*string*
ref-seq	**Reference sequence**	*string*
min-isoform-fraction	**Minimum isoform fraction**	*numeric*
cuffcompare-tool-path	**Cuffcompare tool path**	*string*
path	**Cuffmerge tool path**	*string*
tmp-dir	**Temporary directory**	*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*

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.

Parameters in GUI

Parameter	Description	Default value
Output directory	Directory to save MACS output files.	

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
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
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 Compatibl e 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
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..	
Multi read correct	Do an initial estimation procedure to more accurately weight reads mapping to multiple locations in the genome.	False
Library type	Specifies RNA-Seq protocol.	Standard Illumina
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 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
FDR	The allowed false discovery rate used in testing.	0.05
Max MLE iterations	Sets the number of iterations allowed during maximum likelihood estimation of abundances.	5000

Emit count tables	Include information about the fragment counts, fragment count variances, and fitted variance model into the report.	False
Cuffdiff tool path	The path to the Cuffdiff external tool in UGENE.	defaul
Temporary directory	The directory for temporary files.	default

Parameters in Workflow File

Type: cuffdiff

Parameter	Parameter in the GUI	Type
out-dir	**Output directory**	*string*
time-series-analysis	**Time series analysis**	*boolean*
upper-quartile-norm	**Upper quartile norm**	*boolean*
hits-norm	**Hits norm**	*numeric*
frag-bias-correct	**Frag bias correct**	*string*
multi-read-correct	**Multi read correct**	*boolean*
library-type	**Library type**	*numeric*
mask-file	**Mask file**	*numeric*
min-alignment-count	**Min alignment count**	*string*
fdr	**FDR**	*numeric*
max-mle-iterations	**Max MLE iterations**	*numeric*
emit-count-tables	**Emit count tables**	*boolean*
path	**Cuffdiff tool path**	*string*
temp-dir	**Temporary directory**	*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
- Create VCF consensus
- SnpEff Annotation and Filtration Element

Call Variants with SAMtools Element

Calls SNPs and INDELS with SAMtools mpileup and bcftools.

Parameters in GUI

Parameter	Description	Default value
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
Count anomalous read pairs	Do not skip anomalous read pairs in variant calling (mpileup)(-A).	False
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
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
Max number of reads per input BAM	At a position, read maximally the number of reads per input BAM (mpileup)(-d).	250
Extended BAQ computation	Extended BAQ computation. This option helps sensitivity especially for MNPs, but may hurt specificity a little bit (mpileup)(-E).	False
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).	0
Minimum base quality	Minimum base quality for a base to be considered (mpileup)(-Q).	13
Gap extension error	Phred-scaled gap extension sequencing error probability. Reducing INT leads to longer indels (mpileup)(-e).	20
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
No INDELs	Do not perform INDEL calling (mpileup)(-I).	False
Max INDEL depth	Skip INDEL calling if the average per-sample depth is above INT (mpileup)(-L).	250
Gap open error	Phred-scaled gap open sequencing error probability. Reducing INT leads to more indel calls (mpileup)(-o).	40
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).	
Retain all possible alternate	Retain all possible alternate alleles at variant sites. By default, the view command discards unlikely alleles. (bcf view)(-A).	False
Indicate PL	Indicate PL is generated by r921 or before (ordering is different) (bcf view)(-F).	False
No genotype information	Suppress all individual genotype information (bcf view)(-G).	False
A/C/G/T only	Skip sites where the REF field is not A/C/G/T (bcf view)(-N).	False
List of sites	List of sites at which information are outputted (bcf view)(-l).	
QCALL likelihood	Output the QCALL likelihood format (bcf view)(-Q).	False
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).	
Min samples fraction	skip loci where the fraction of samples covered by reads is below FLOAT (bcf view)(-d).	0
Per-sample genotypes	Call per-sample genotypes at variant sites. (bcf view)(-g).	True

INDEL-to-SNP Ratio	Ratio of INDEL-to-SNP mutation rate. (bcf view)(-i).	-1
Max P(ref|D)	A site is considered to be a variant if P(ref|D)	0.5
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
Mutation rate	Scaled mutation rate for variant calling (bcf view)(-t).	0.001
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).	
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
N permutations	Number of permutations for association test (effective only with -1) (bcf view)(-U).	0
Min P(chi^2)	Only perform permutations for P(chi^2).	0.01
Minimum RMS quality	Minimum RMS mapping quality for SNPs (varFilter) (-Q).	10
Minimum read depth	Minimum read depth (varFilter) (-d).	2
Maximum read depth	Maximum read depth (varFilter) (-D).	10000000
Alternate bases	Minimum number of alternate bases (varFilter) (-a).	2
Gap size	SNP within INT bp around a gap to be filtered (varFilter) (-w).	3
Window size	Window size for filtering adjacent gaps (varFilter) (-W).	10
Strand bias	Minimum P-value for strand bias (given PV4) (varFilter) (-1).	0.0001
BaseQ bias	Minimum P-value for baseQ bias (varFilter) (-2).	1e-100
MapQ bias	Minimum P-value for mapQ bias (varFilter) (-3).	0
End distance bias	Minimum P-value for end distance bias (varFilter) (-4).	0.0001
HWE	Minimum P-value for HWE (plus F).	0.0001

Log filtered	Print filtered variants into the log (varFilter) (-p).	False

Parameters in Workflow File

Type: call_variants

Parameter	Parameter in the GUI	Type
illumina13-encoding	Illumina-1.3+ encoding	*boolean*
use_orphan	Count anomalous read pairs	*boolean*
disable_baq	Disable BAQ computation	*boolean*
capq_thres	Mapping quality downgrading coefficient	*numeric*
max_depth	Max number of reads per input BAM	*numeric*
ext_baq	Extended BAQ computation	*boolean*
bed	BED or position list file	*string*
reg	Pileup region	*string*
min_mq	Minimum mapping quality	*numeric*
min_baseq	Minimum base quality	*numeric*
extQ	Gap extension error	*numeric*
tandemQ	Homopolymer errors coefficient	*numeric*
no_indel	No INDELs	*boolean*
max_indel_depth	Max INDEL depth	*numeric*
openQ	Gap open error	*numeric*
pl_list	List of platforms for indels	*string*
keepalt	Retain all possible alternate	*boolean*
fix_pl	Indicate PL	*boolean*
no_geno	No genotype information	*boolean*
acgt_only	A/C/G/T only	*boolean*
bcf_bed	List of sites	*string*
qcall	QCALL likelihood	*boolean*
samples	List of samples	*string*
min_smpl_frac	Min samples fraction	*numeric*
call_gt	Per-sample genotypes	*boolean*
indel_frac	INDEL-to-SNP Ratio	*numeric*
pref	Max P(ref|D)	*numeric*
ptype	Prior allele frequency spectrum	*string*
theta	Mutation rate	*numeric*
ccall	Pair/trio calling	*string*
n1	N group-1 samples	*numeric*
n_perm	N permutations	*numeric*
min_perm_p	Min P(chi^2)	*numeric*

min-qual	Minimum RMS quality	*numeric*
min-dep	Minimum read depth	*numeric*
max-dep	Maximum read depth	*numeric*
min-alt-bases	Alternate bases	*numeric*
gap-size	Gap size	*numeric*
window"	Window size	*numeric*
min-strand	Strand bias	*numeric*
min-baseQ	BaseQ bias	*string*
min-mapQ	MapQ bias	*numeric*
min-end-distance	End distance bias	*numeric*
min-hwe	HWE	*numeric*
print-filtered	Log 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.

Parameters in GUI

Parameter	Description	Default value
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".	

Parameters in Workflow File

Type: rename-chromosome-in-variation

Parameter	Parameter in the GUI	Type
prefixes-to-replace	**Replace prefixes**	*string*
prefix-replace-with	**Replace by**	*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

Parses information, added to variations by SnpEff, into standard annotations.

Parameters in GUI

Parameter	Description	Default value
Output 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.	genbank

Parameters in Workflow File

Type: convert-snpeff-variations-to-annotations

Parameter	Parameter in the GUI	Type
url-out	**Output file**	*string*
document-format	**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

Apply VCF variants to a fasta file to create consensus sequence.

Parameters in GUI

Parameter	Description	Default value
Output FASTA consensus	The url to the output file with the result consensus.	

Parameters in Workflow File

Type: vcf-consensus

Parameter	Parameter in the GUI	Type
consensus-url	**Output FASTA consensus**	*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.

Parameters in GUI

Parameter	Description	Default value
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
Input format	Select the input format of variations.	VCF
Output format	Select the format of annotated output files.	VCF
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
Upstream/downstream length	Upstream and downstream interval size. Eliminate any upstream and downstream effect by using 0 length.	No upstream/downstream interval (0 bases)
Cannonical transcripts	Use only cannonical transcripts.	False
HGVS nomenclature	Annotate using HGVS nomenclature.	False
Annotate loss of function	Annotate Loss of function (LOF) and Nonsense mediated decay (NMD).	False
Annotate TFBSs motifs	Annotate transcription factor binding site motifs (only available for latest GRCh37).	False

Parameters in Workflow File

Type: seff

Parameter	Parameter in the GUI	Type
out-mode	**Output directory**	*string*
inp-format	**Input format**	*string*
out-format	**Output format**	*string*
genome	**Genome**	*string*
updown-length	**Upstream/downstream length**	*numeric*
canon	**Cannonical transcripts**	*boolean*
hgvs	**HGVS nomenclature**	*boolean*
lof	**Annotate loss of function**	*boolean*
motif	**Annotate TFBSs motifs**	*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.

Parameters in GUI

Parameter	Description	Default value
Matrix type	Dinucleic matrices are more detailed, while mononucleic one are more useful for small input data sets.	Mononucleic

Parameters in Workflowa File

Type: fmatrix-build

Parameter	Parameter in the GUI	Type
type	**Matrix 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.

Parameters in GUI

Parameter	Description	Default value
Weight algorithm	Optional feature, in most cases applying no weight will fit. In some cases choosing algorithm 2 will increase the recognition quality.	None
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
Calibration length	Length of random synthetic sequences used to calibrate the profile. Should not be less than window size.	1M
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

Parameters in Workflow File

Type: sitecon-build

Parameter	Parameter in the GUI	Type
weight-algorithm	**Weight algorithm**	*boolean* Available values are: • 0 - for None • 1 - for Algorithm2
window-size	**Window size, bp**	*numeric*
calibrate-length	**Calibration length**	*numeric*
seed	**Random 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 weight matrix. Weight matrices are used for probabilistic recognition of transcription factor binding sites.

Parameters in GUI

Parameter	Description	Default value
Matrix type (required)	Dinucleic matrices are more detailed, while mononucleic one are more useful for small input data sets.	Mononucleic
Weight algorithm	Different weight algorithms uses 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

Parameters in Workflow File

Type: wmatrix-build

Parameter	Parameter in the GUI	Type
type	**Matrix type**	*boolean* Available values are: • true - for Dinucleic • false - for Mononucleic
weight-algorithm	**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.

Parameters in GUI

Parameter	Description	Default value
Matrix type (required)	Dinucleic matrices are more detailed, while mononucleic one are more useful for small input data sets.	Mononucleic
Weight algorithm	Different weight algorithms uses 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

Parameters in Workflow File

Type: fmatrix-to-wmatrix

Parameter	Parameter in the GUI	Type
type	**Matrix type**	*boolean* Available values are: • true - for Dinucleic • false - for Mononucleic
weight-algorithm	**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.

Parameters in GUI

Parameter	Description	Default value
Input files (required)	Semicolon-separated list of paths to the input files.	

Parameters in Workflow File

Type: fmatrix-read

Parameter	Parameter in the GUI	Type
url-in	**Input files**	*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.

Parameters in GUI

Parameter	Description	Default value
Input files (required)	Semicolon-separated list of paths to the input files.	

Parameters in Workflow File

Type: sitecon-read

Parameter	Parameter in the GUI	Type
url-in	**Input files**	*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.

Parameters in GUI

Parameter	Description	Default value
Input files (required)	Semicolon-separated list of paths to the input files.	

Parameters in Workflow File

Type: wmatrix-read

Parameter	Parameter in the GUI	Type
url-in	**Input files**	*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.

Parameters in GUI

Parameter	Description	Default value
Result annotation	Name of the result annotations.	misc_feature
Search in	Specifies which strands should be searched: direct, complement or both.	both strands
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 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
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

Parameters in Workflow File

Type: sitecon-search

Parameter	Parameter in the GUI	Type
result-name	**Result annotation**	*string*
strand	**Search in**	*numeric* Available values are: • 0 - for searching in both strands • 1 - for searching in direct strand • 2 - for searching in complement strand
min-score	**Min score**	*numeric*
err1	**Min err1**	*numeric*
err2	**Max 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.

Parameters in GUI

Parameter	Description	Default value
Result annotation	Name of the result annotations.	misc_feature

Search in	Specifies which strands should be searched: direct, complement or both.	both strands
Min score	Minimum score to detect transcription factor binding site in percents.	85

Parameters in Workflow File

Type: wmatrix-search

Parameter	Parameter in the GUI	Type
result-name	**Result annotation**	*string*
strand	**Search in**	*numeric* Available values are: • 0 - for searching in both strands • 1 - for searching in direct strand • 2 - for searching in complement strand
min-score	**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.

Parameters in GUI

Parameter	Description	Default value

Output file (required)	Location of the output data file. If this attribute is set, the “Location” slot is not taken into account.	
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

Parameters in Workflow File

Type: fmatrix-write

Parameter	Parameter in the GUI	Type
url-out	**Output file**	*string*
write-mode	**Existing file**	*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 specified location.

Parameters in GUI

Parameter	Description	Default value
Output file (required)	Location of the output data file. If this attribute is set, the “Location” slot is not taken into account.	
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

Parameters in Workflow File

Type: sitecon-write

Parameter	Parameter in the GUI	Type
url-out	**Output file**	*string*

write-mode	**Existing file**	*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.

Parameters in GUI

Parameter	Description	Default value
Output file (required)	Location of the output data file. If this attribute is set, the “Location” slot is not taken into account.	
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

Parameters in Workflow File

Type: wmatrix-write

Parameter	Parameter in the GUI	Type
url-out	**Output file**	*string*
write-mode	**Existing file**	*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.

Parameters in GUI

Parameter	Description	Default value
GC-content	Evaluate GC-content.	True
GC1-content	Evaluate GC1-content.	True
GC2-content	Evaluate GC2-content.	True
GC3-content	Evaluate GC3-content.	True

Parameters in Workflow File

Type: dna-stats

Parameter	Parameter in the GUI	Type
gc-content	**GC-content**	*boolean*
gc1-content	**GC1-content**	*boolean*
gc2-content	**GC2-content**	*boolean*
gc3-content	**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.

Parameters in GUI

Parameter	Description	Default value

Length	Length of the resulted sequence or sequences.	1000 bp
Count	Number of sequences to generate.	1
Seed	Value to initialize the random generator. By default (seed = -1) the generator is initialized with the system time.	-1
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
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*parameter is used to generate the sequence(s).	GC Content
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
Reference	Path to the reference file (could be a sequence or an alignment).	
A	Adenine content.	25%
C	Cytosine content.	25%
G	Guanine content.	25%
T	Thymine content.	25%
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

Parameters in Workflow File

Type: generate-dna

Parameter	Parameter in the GUI	Type
length	**Lenght**	*numeric*
count	**Count**	*numeric*
seed	**Seed**	*numeric*
content	**Countent**	*string*
algorithm	**Algorithm**	*string* Available values are: • gc-content • gc-skew
window-size	**Window size**	*numeric*
reference-url	**Reference**	*string* Available values are: • manual • reference

percent-a	**A**	*numeric*
percent-c	**C**	*numeric*
percent-g	**G**	*numeric*
percent-t	**T**	*numeric*
gc-skew	**GC Skew**	*numeric*

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *Sequences*

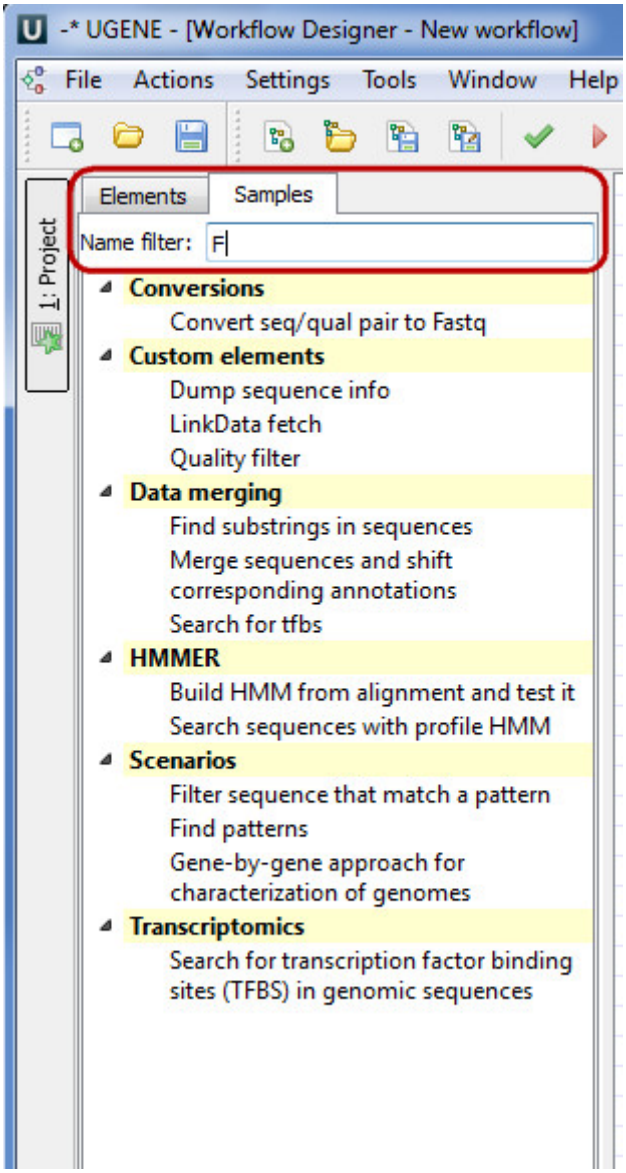
Name in Workflow File: out-sequence

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 *Ctrl+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 Sequences by Annotation Number
  - Marking Sequences 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
  - De novo Assembly with Spades
  - Call Variants with SAMtools
  - ChIP-Seq Coverage
  - ChIP-seq Analysis with Cistrome Tools
  - Extract Consensus from Assembly
  - Extract Coverage from Assembly
  - Extract Transcript Sequences
  - Quality Control by FastQC
  - Raw ChIP-Seq Data Processing
  - Raw DNA-Seq Data Processing
  - Raw RNA-Seq Data Processing
  - RNA-seq Analysis with Tuxedo Tools
  - Get Unmappet Reads
  - Variation Annotation with SnpEff
- 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
  - Merge Sequences and Annotations
  - In Silico PCR
  - 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.

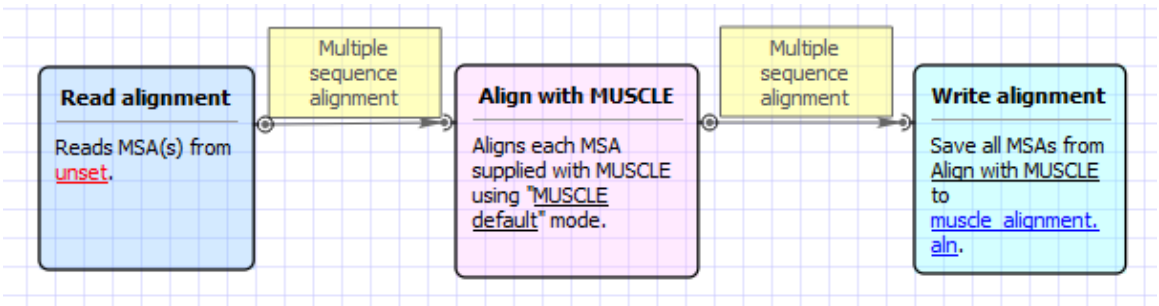
> **How to Use This Sample**
> If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Align Sequences with MUSCLE" can be found in the "Alignment" section of the Workflow Designer samples.

Workflow Image

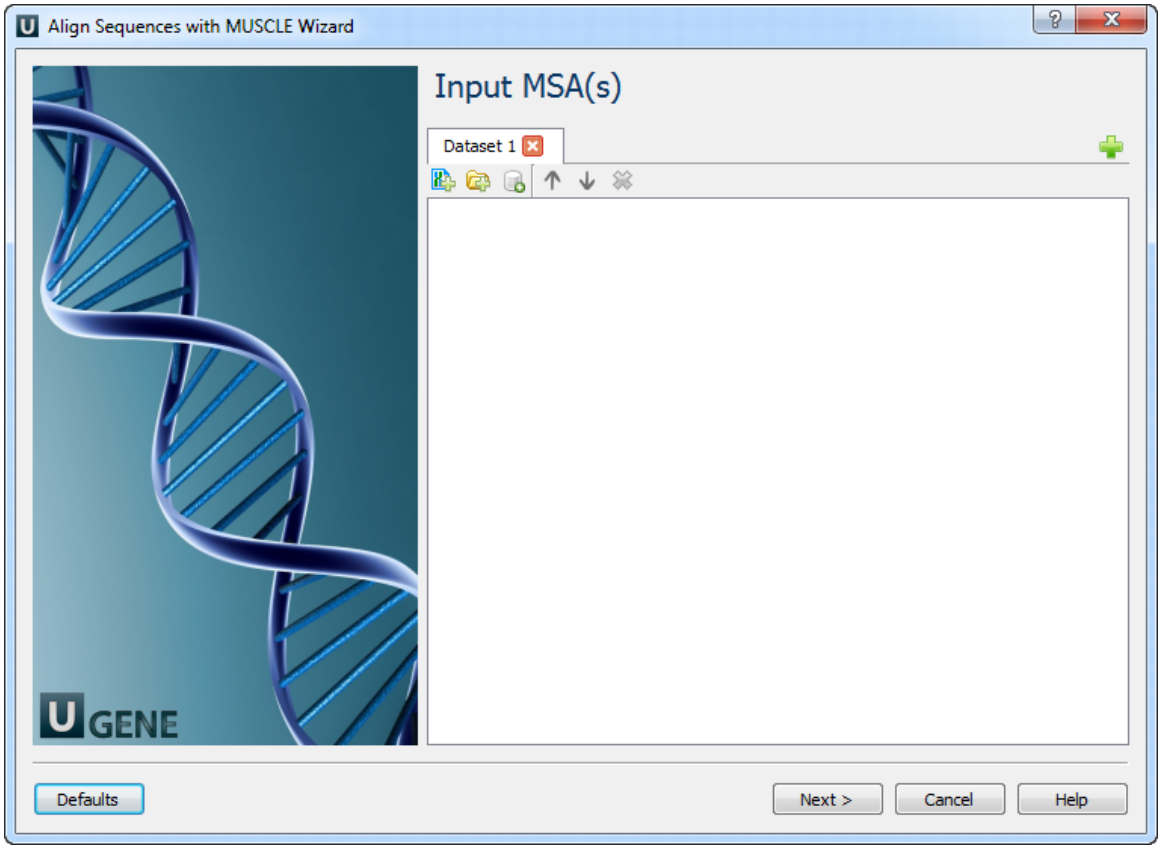
The workflow looks as follows:



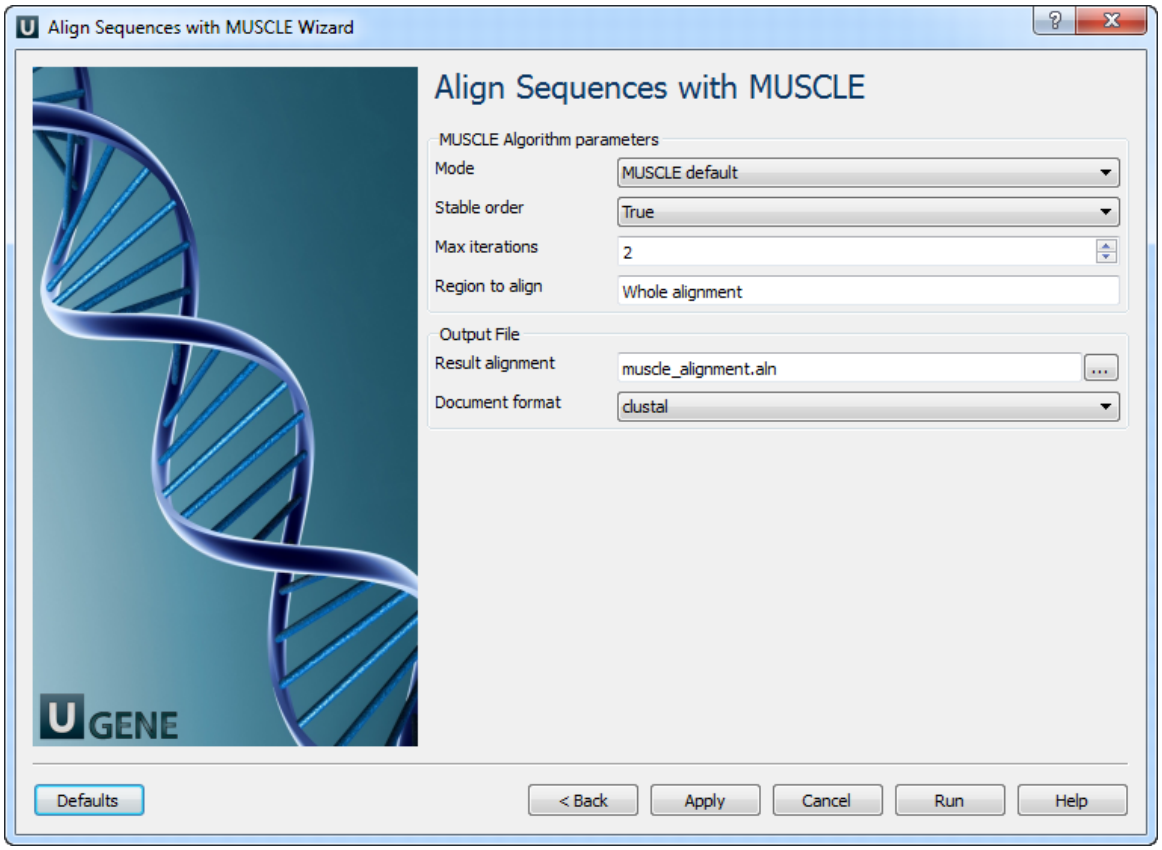
Workflow Wizard

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.

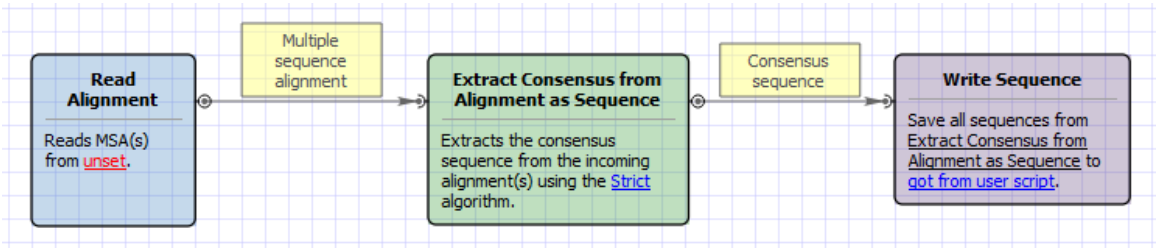
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Extract Consensus as Sequence" can be found in the "Alignment" section of the Workflow Designer samples.

Workflow Image

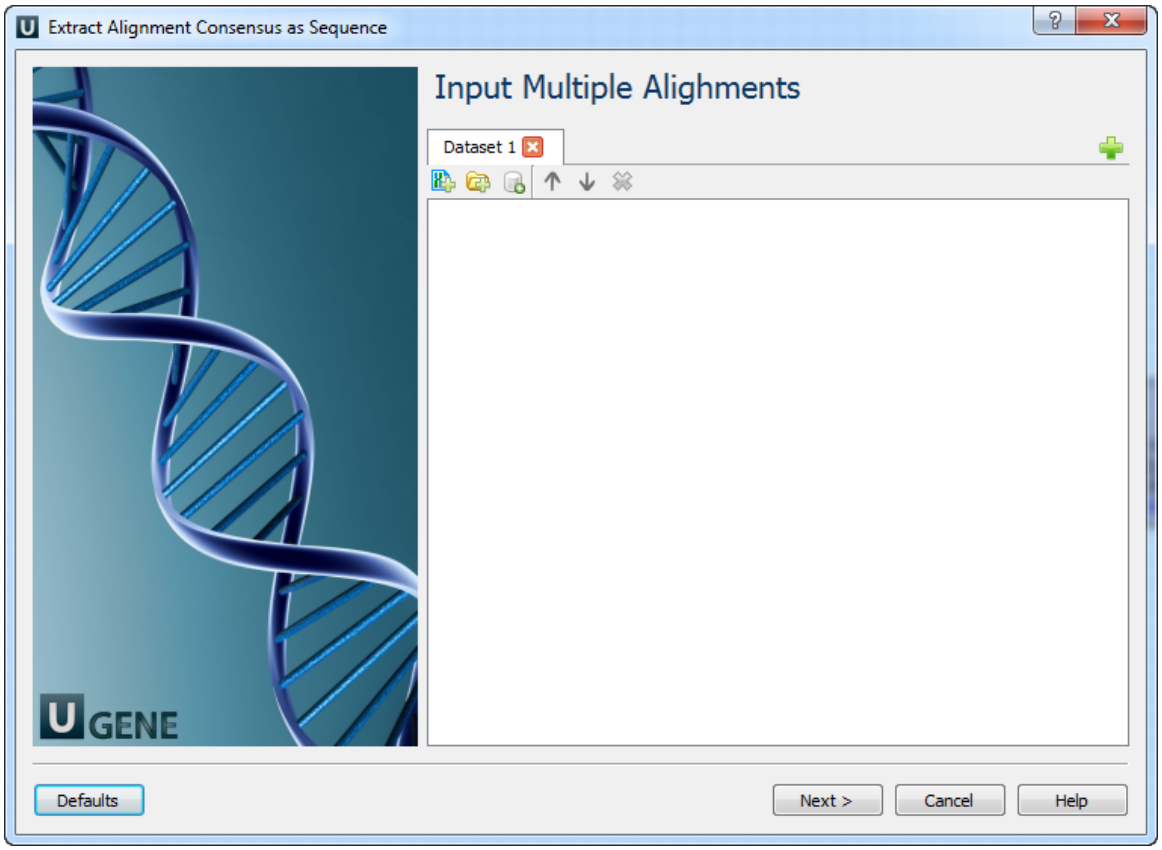
The workflow looks as follows:



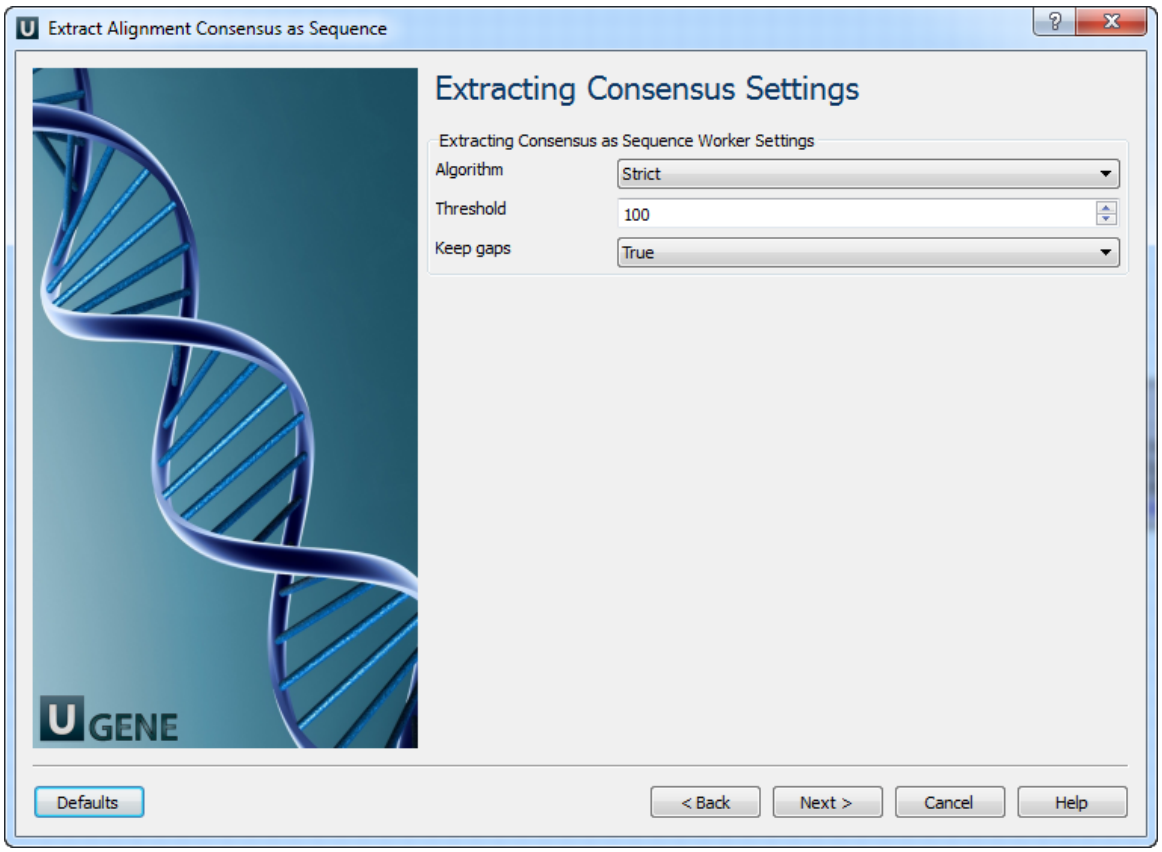
Workflow Wizard

The wizard has 3 pages.

1. Input Multiple Alignments: On this page you must input multiple alignments file(s).



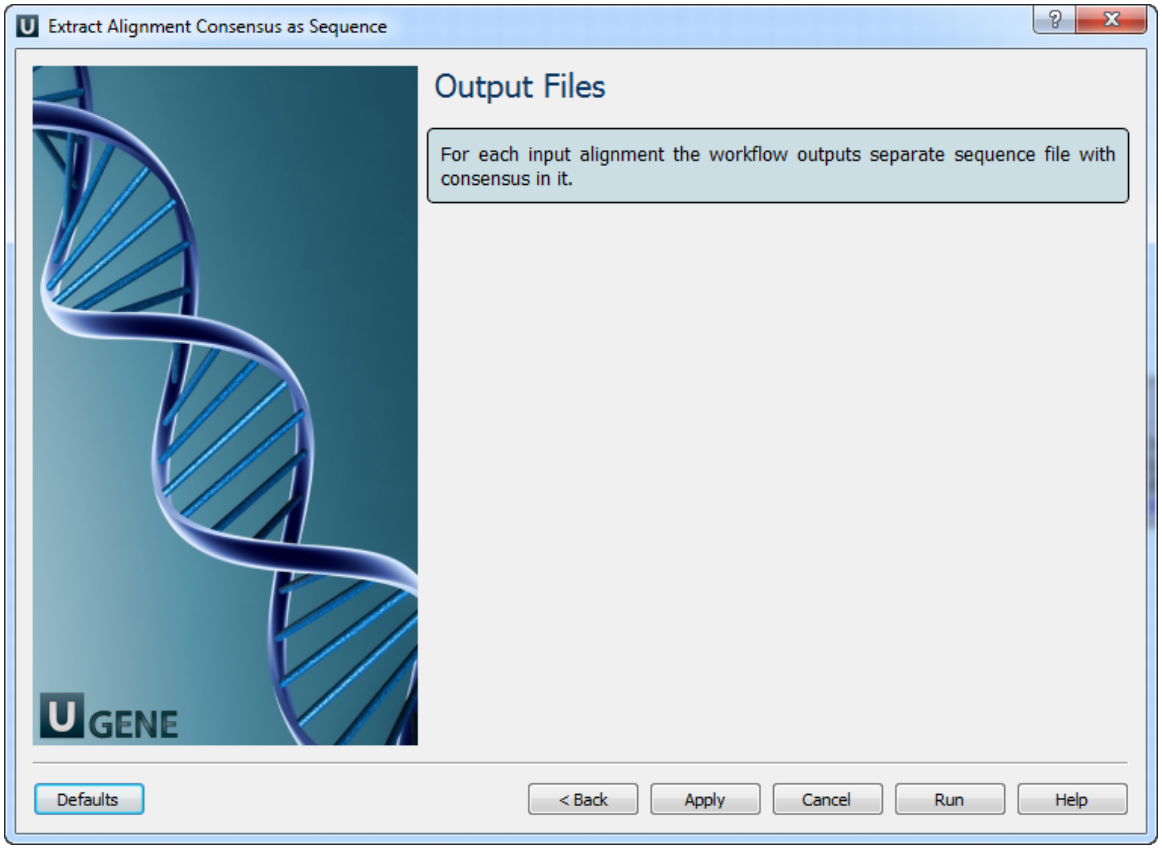
2. Extracting Consensus Settings: On this page you can modify extracting settings.



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.

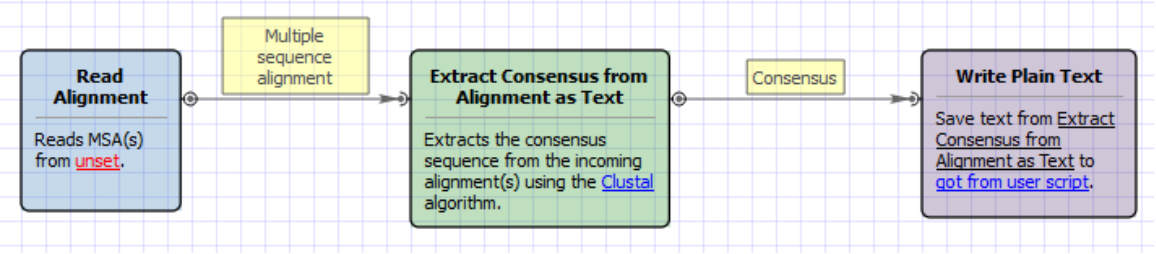
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Extract Consensus as Text" can be found in the "Alignment" section of the Workflow Designer samples.

Workflow Image

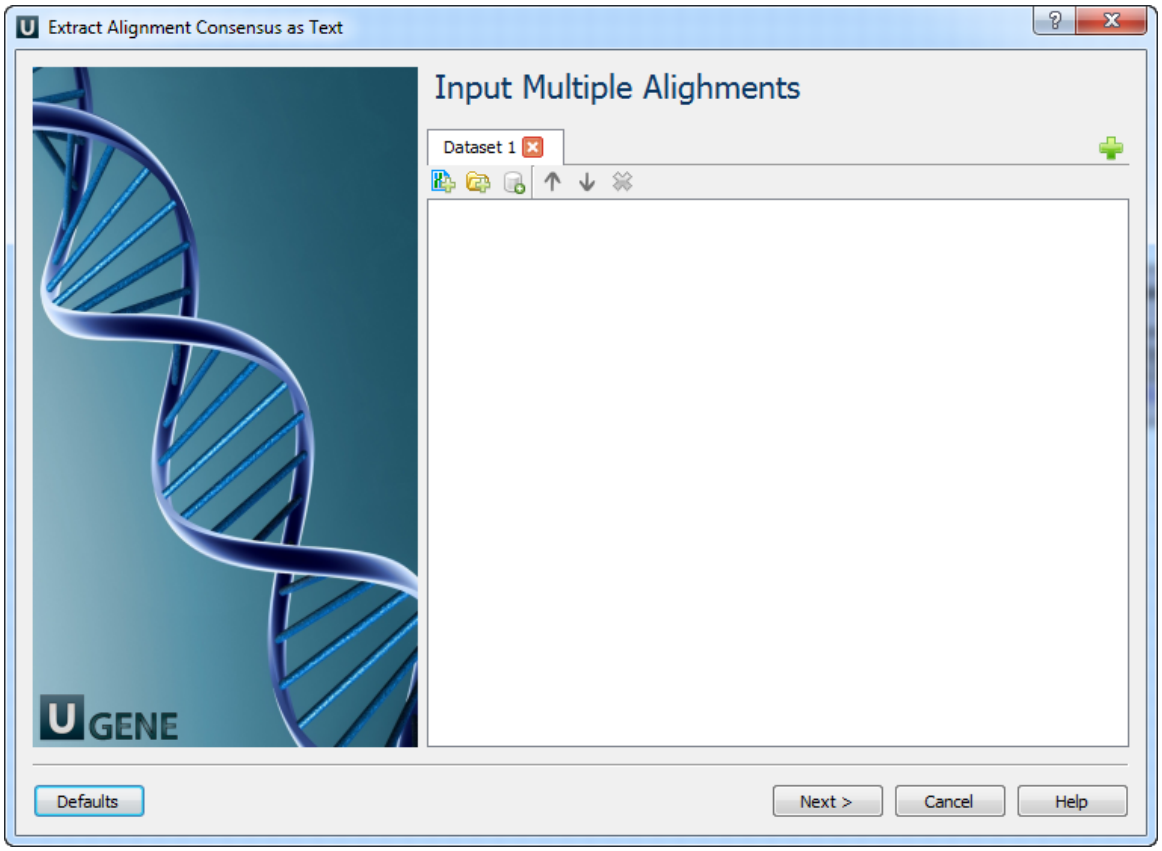
The workflow looks as follows:



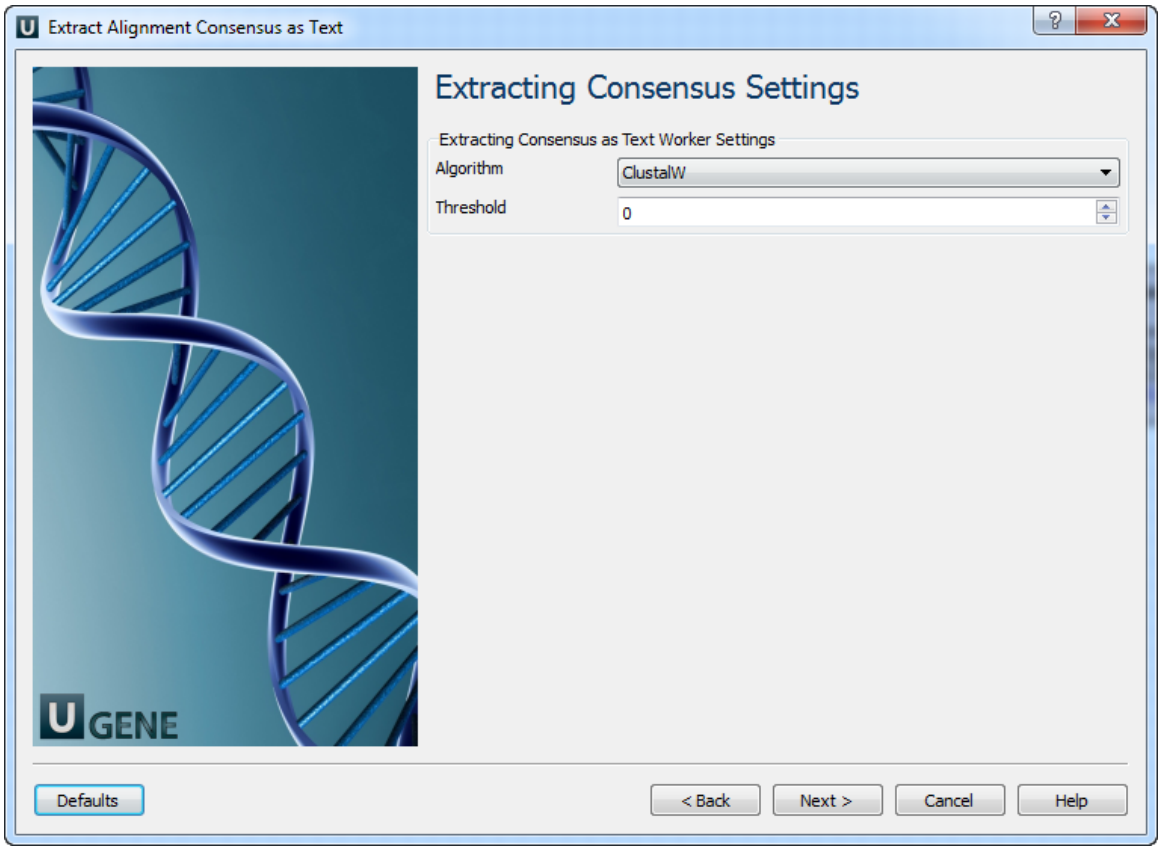
Workflow Wizard

The wizard has 3 pages.

1. Input Multiple Alignments: On this page you must input multiple alignments file(s).



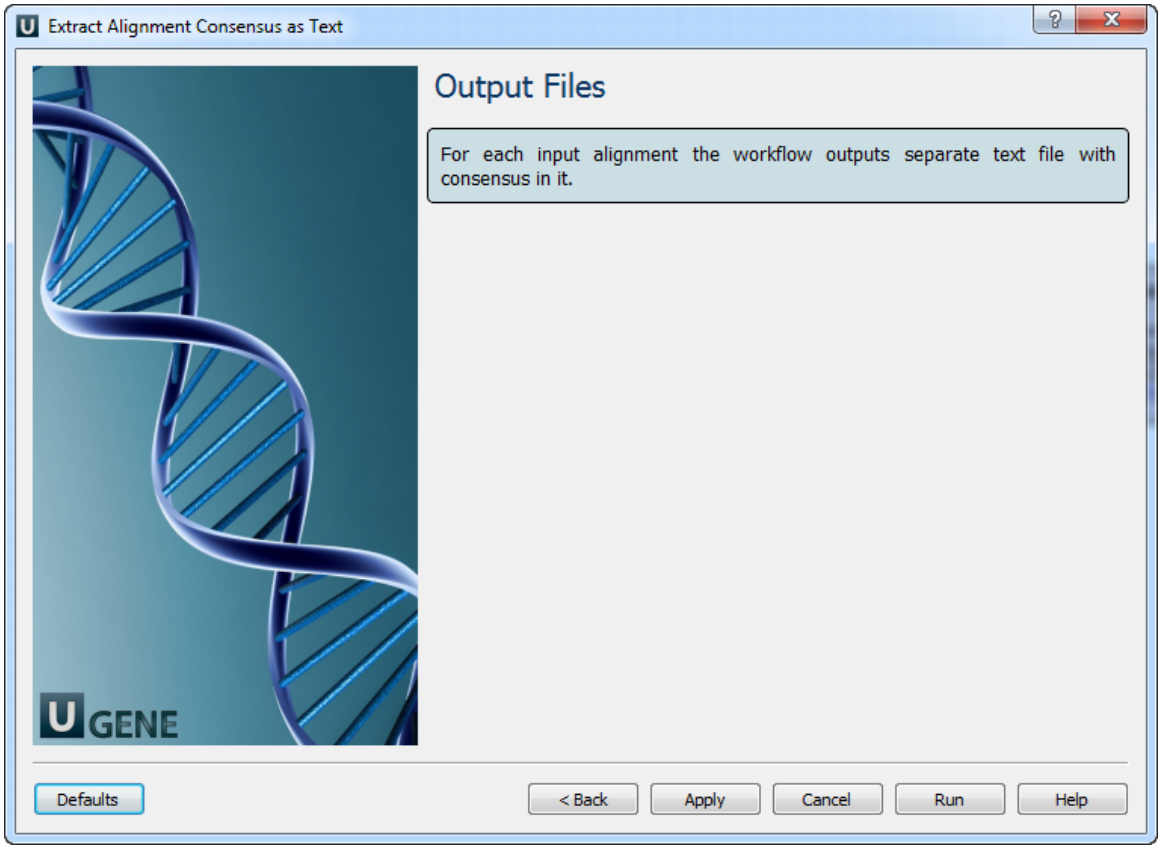
2. Extracting Consensus Settings: On this page you can modify extracting settings.



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.

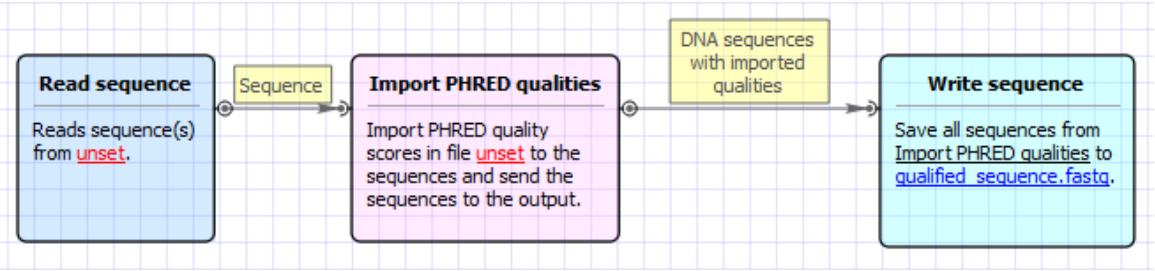
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

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

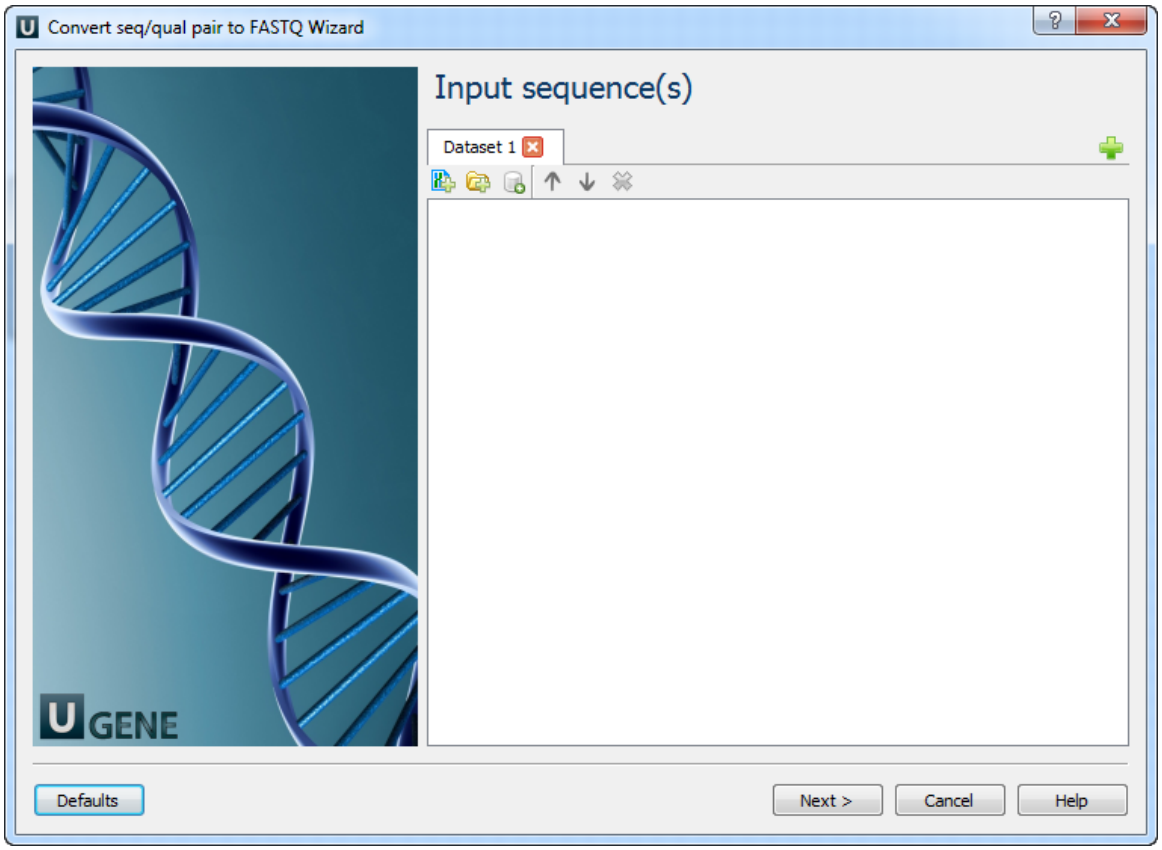
The workflow looks as follows:



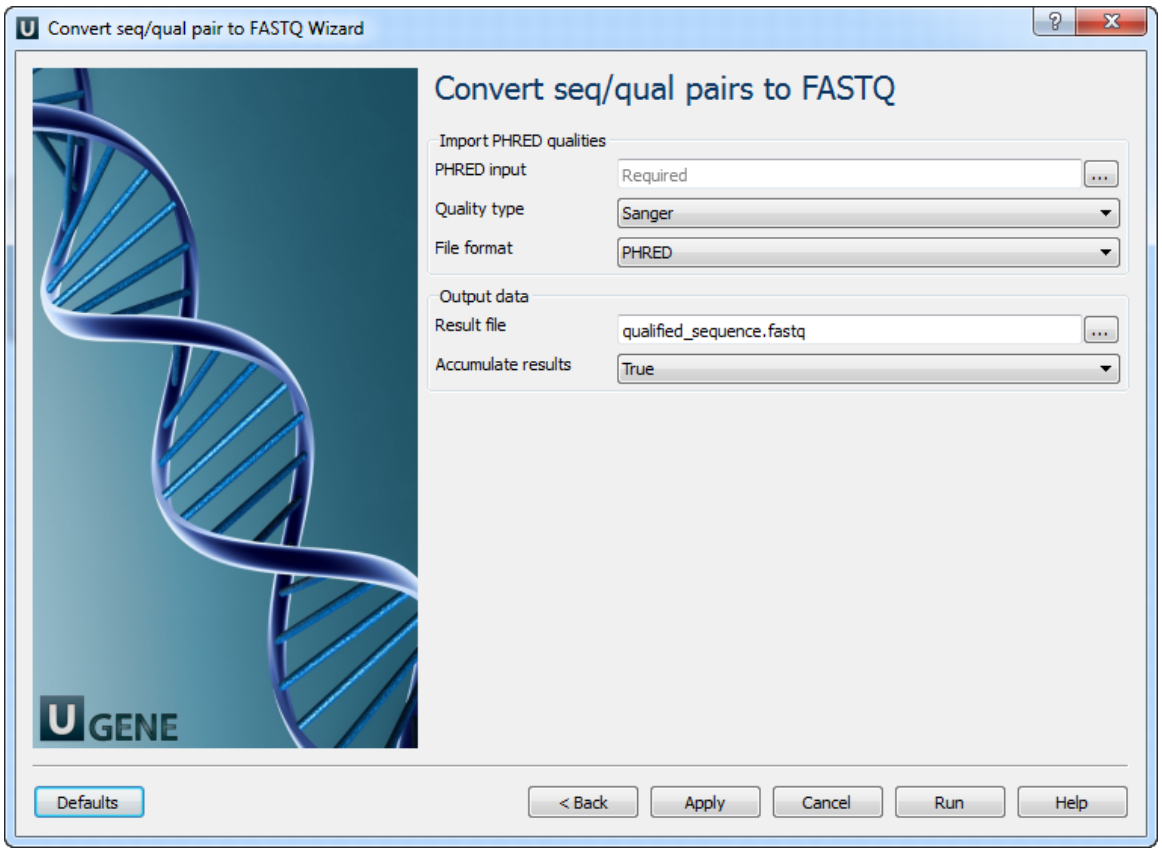
Workflow Wizard

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.

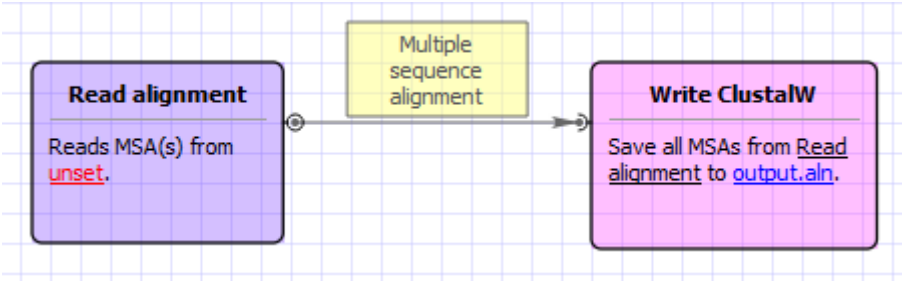
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Convert Alignments to ClustalW" can be found in the "Conversions" section of the Workflow Designer samples.

Workflow Image

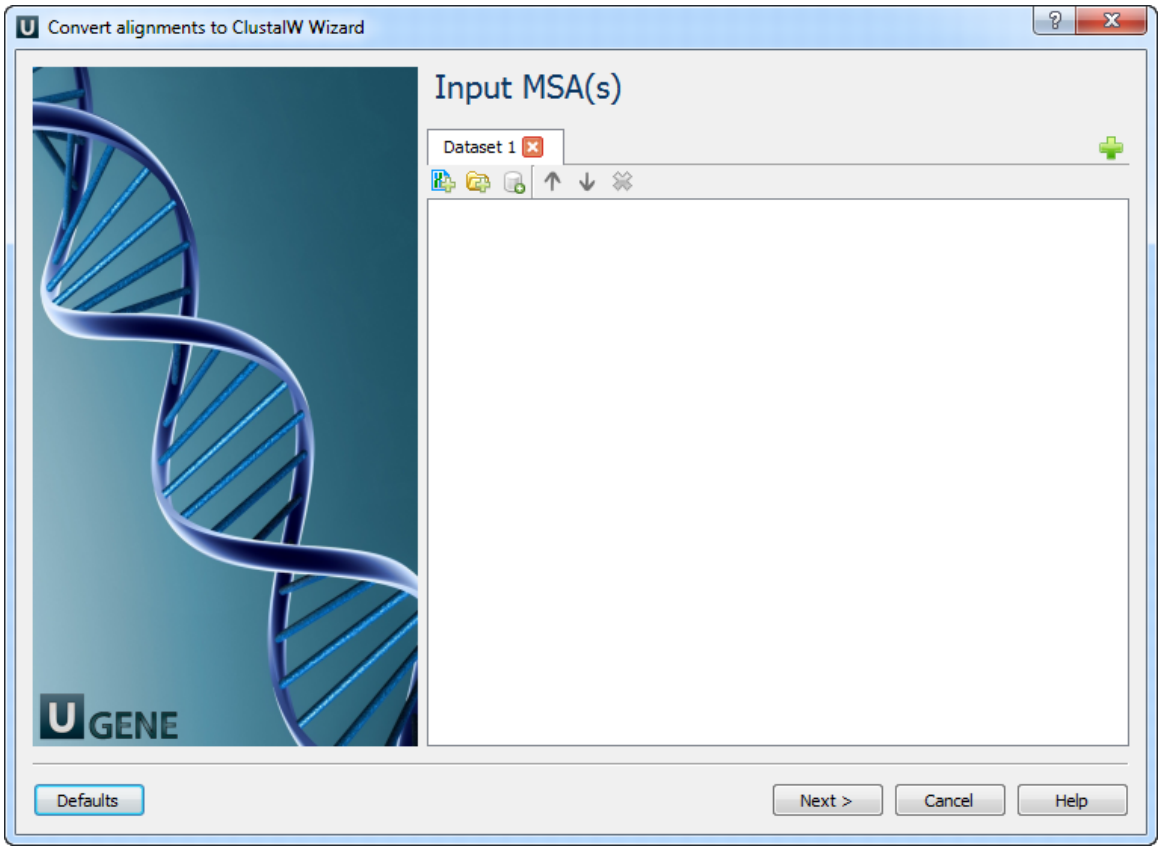
The workflow looks as follows:



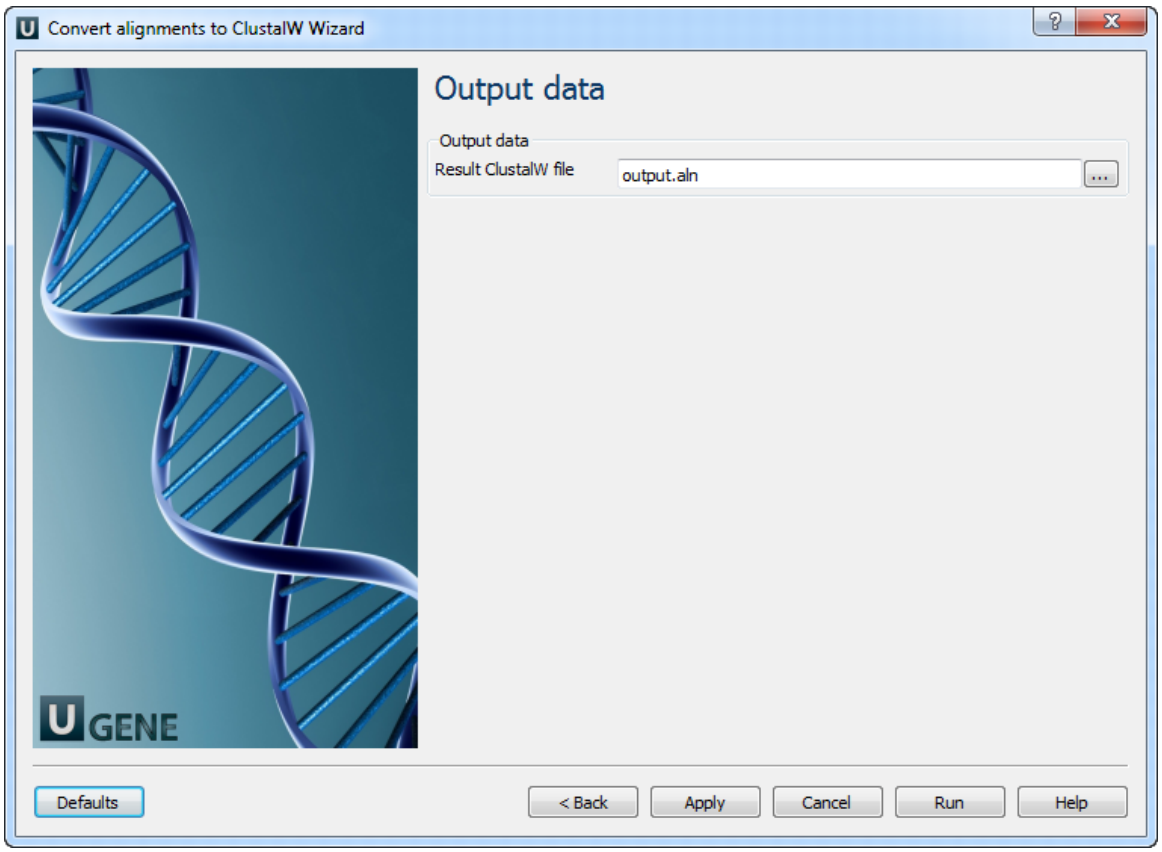
Workflow Wizard

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.

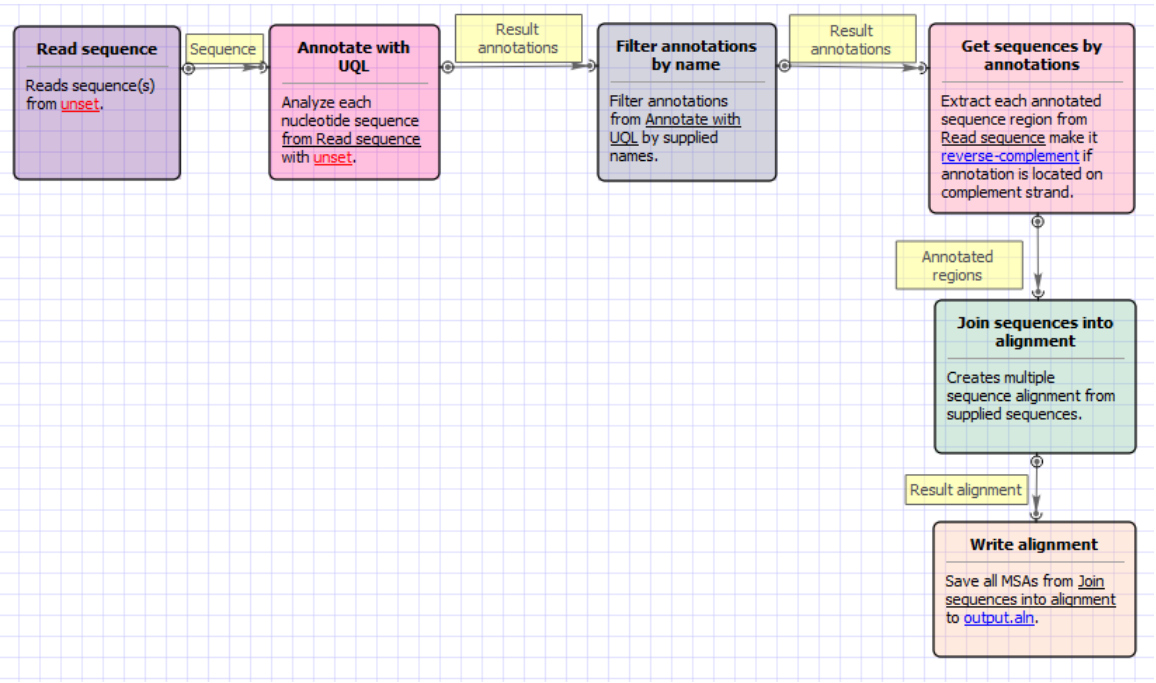
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

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

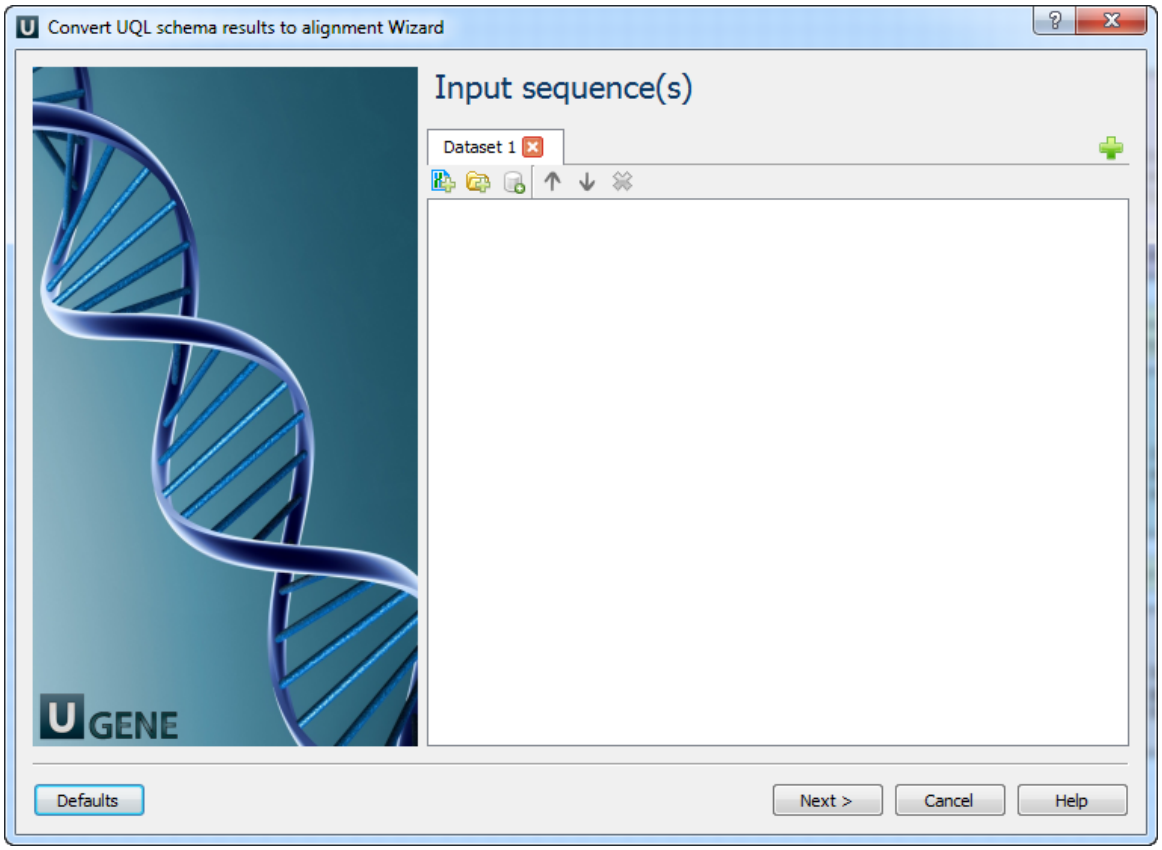
The workflow looks as follows:



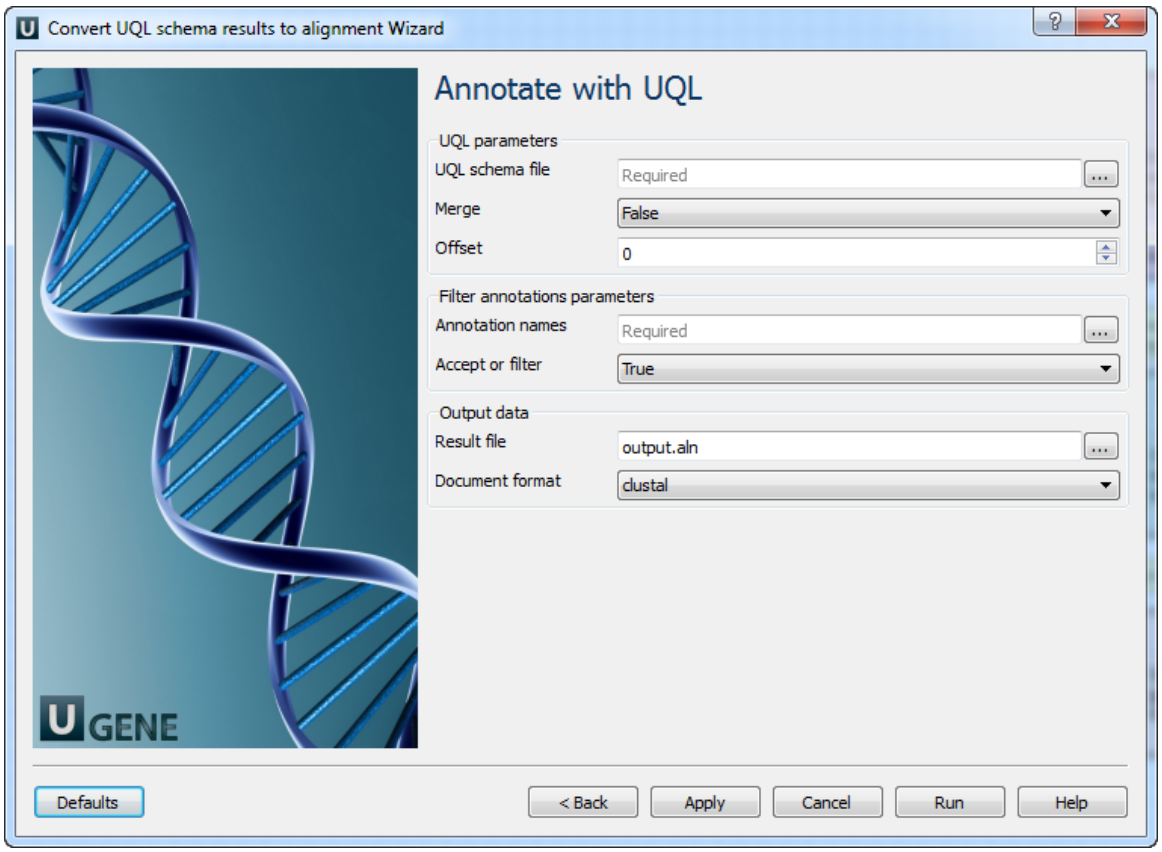
Workflow Wizard

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.

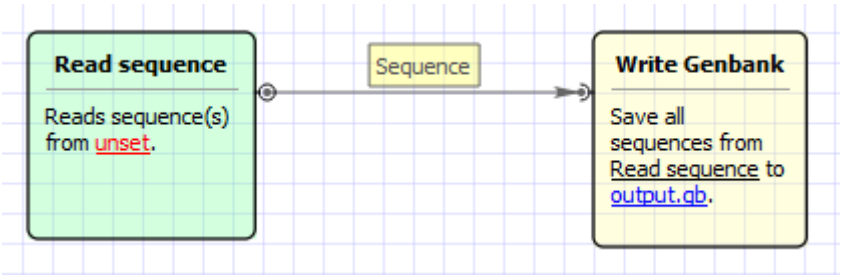
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Convert Sequence to Genbank" can be found in the "Conversions" section of the Workflow Designer samples.

Workflow Image

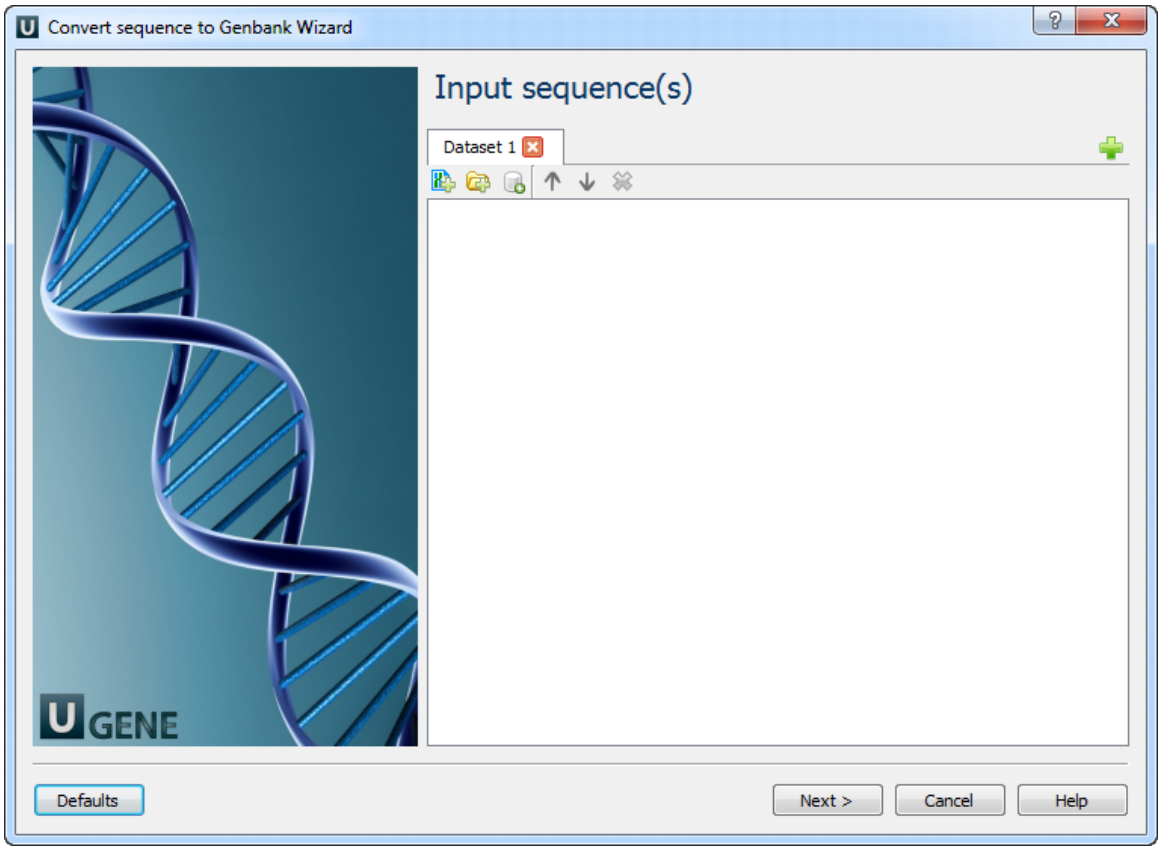
The workflow looks as follows:



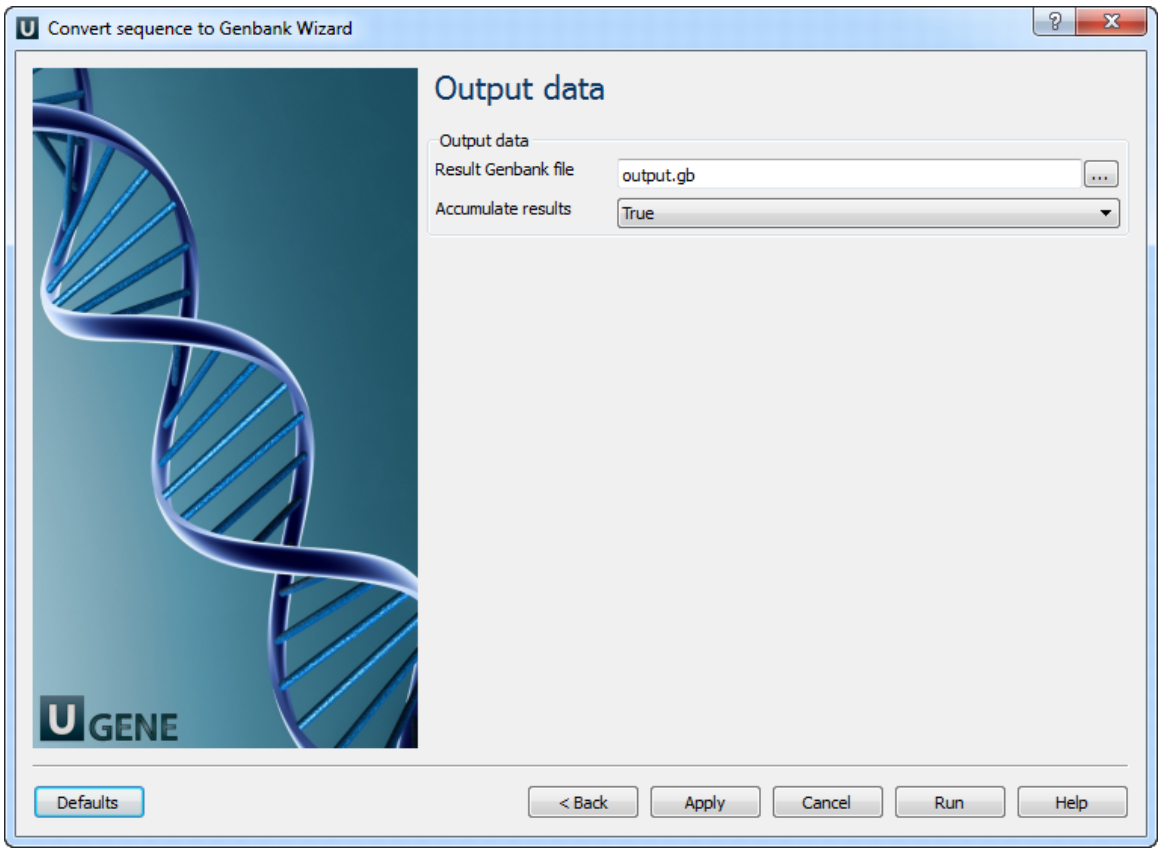
Workflow Wizard

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.

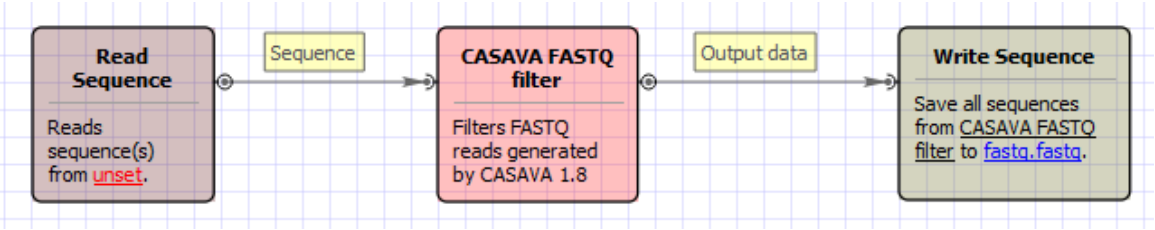
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

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.

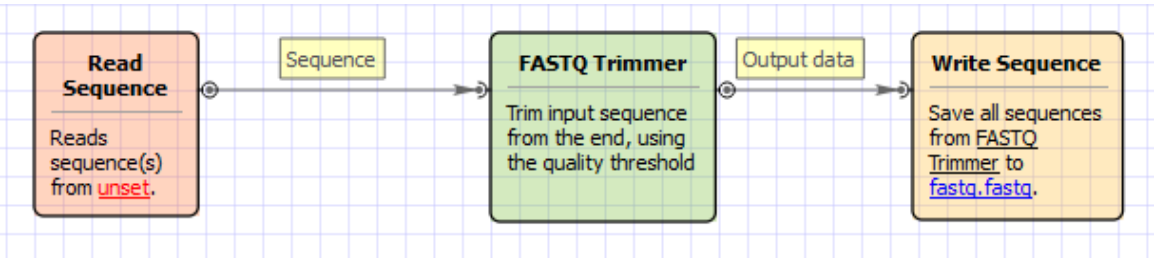
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

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.

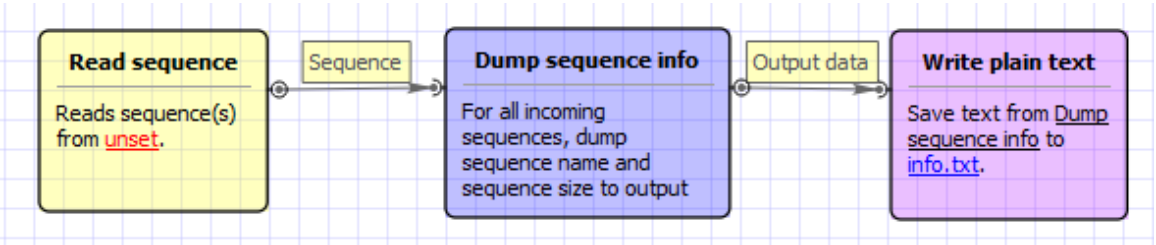
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Dump Sequence Info" can be found in the "Custom Elements" section of the Workflow Designer samples.

Workflow Image

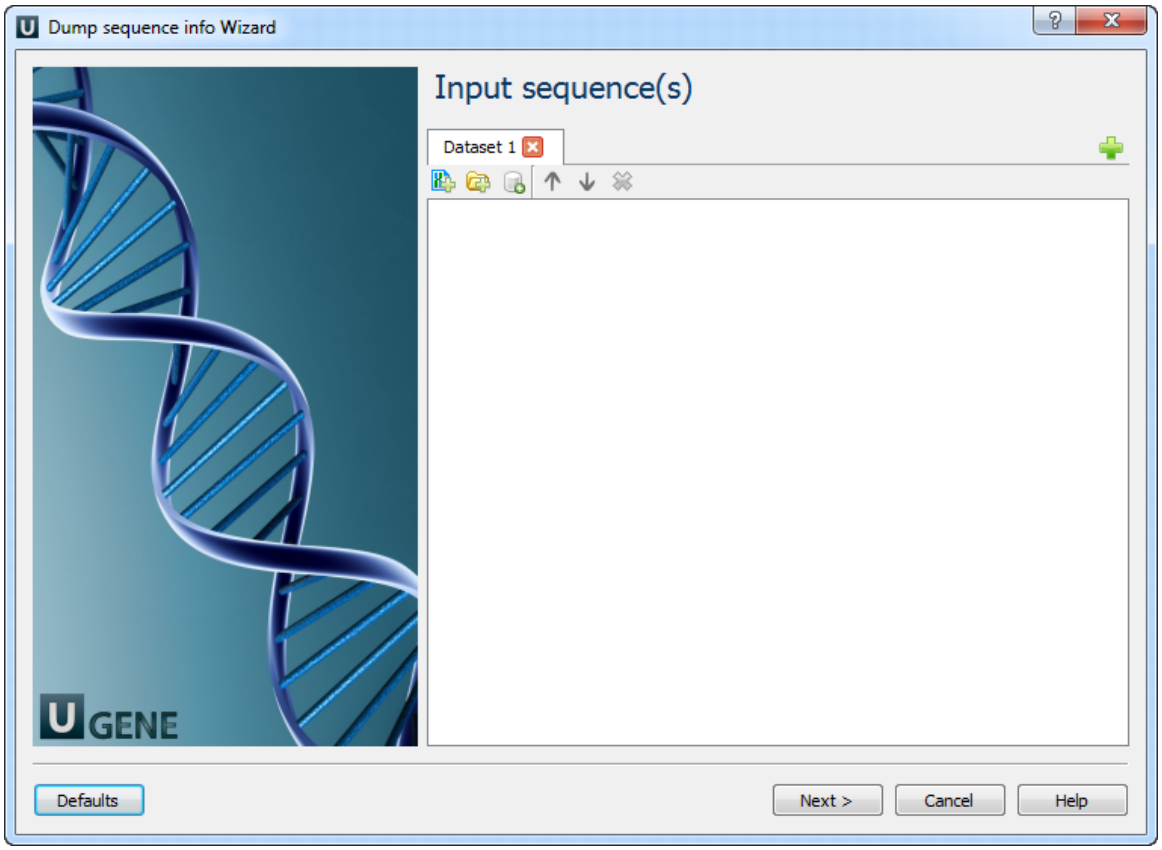
The workflow looks as follows:



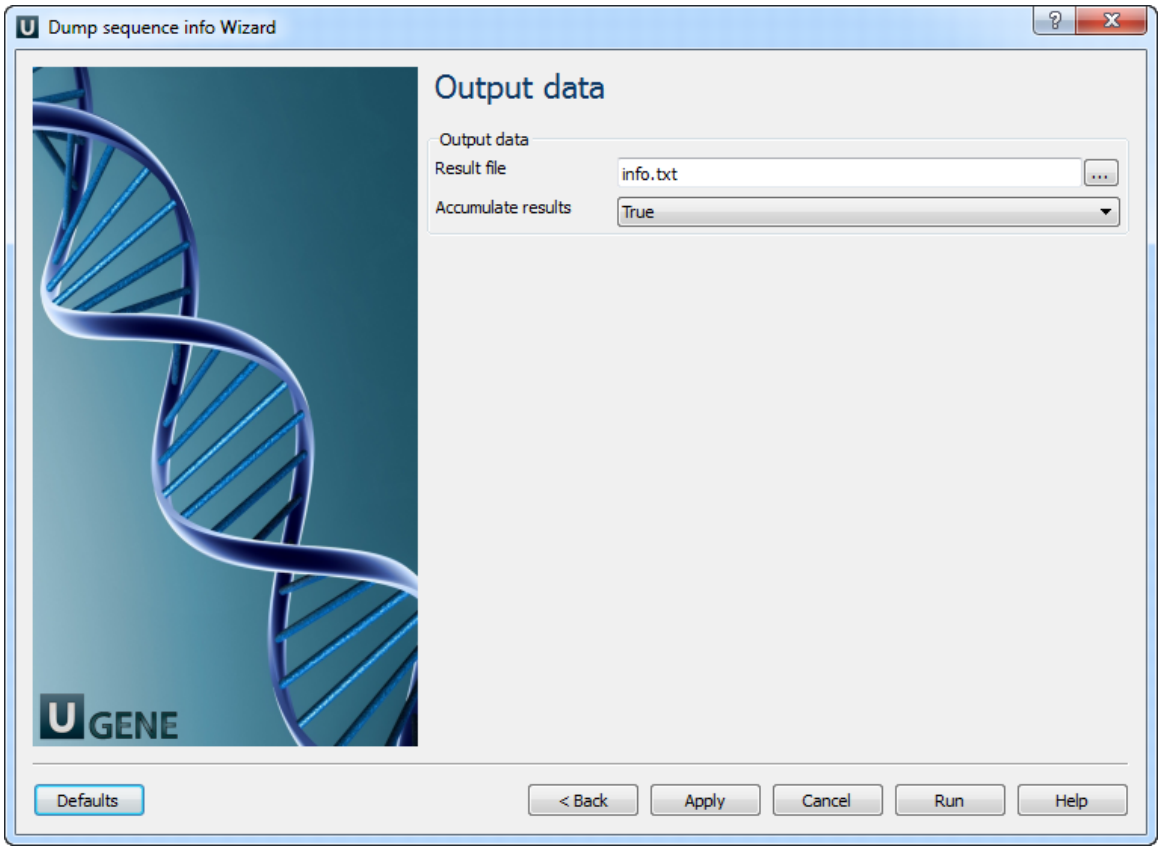
Workflow Wizard

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.

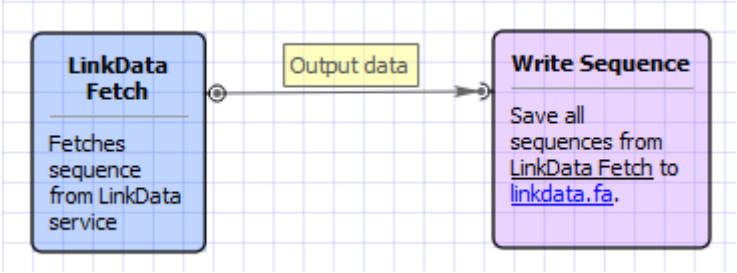
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "LinkData Fetch" can be found in the "Custom Elements" section of the Workflow Designer samples.

Workflow Image

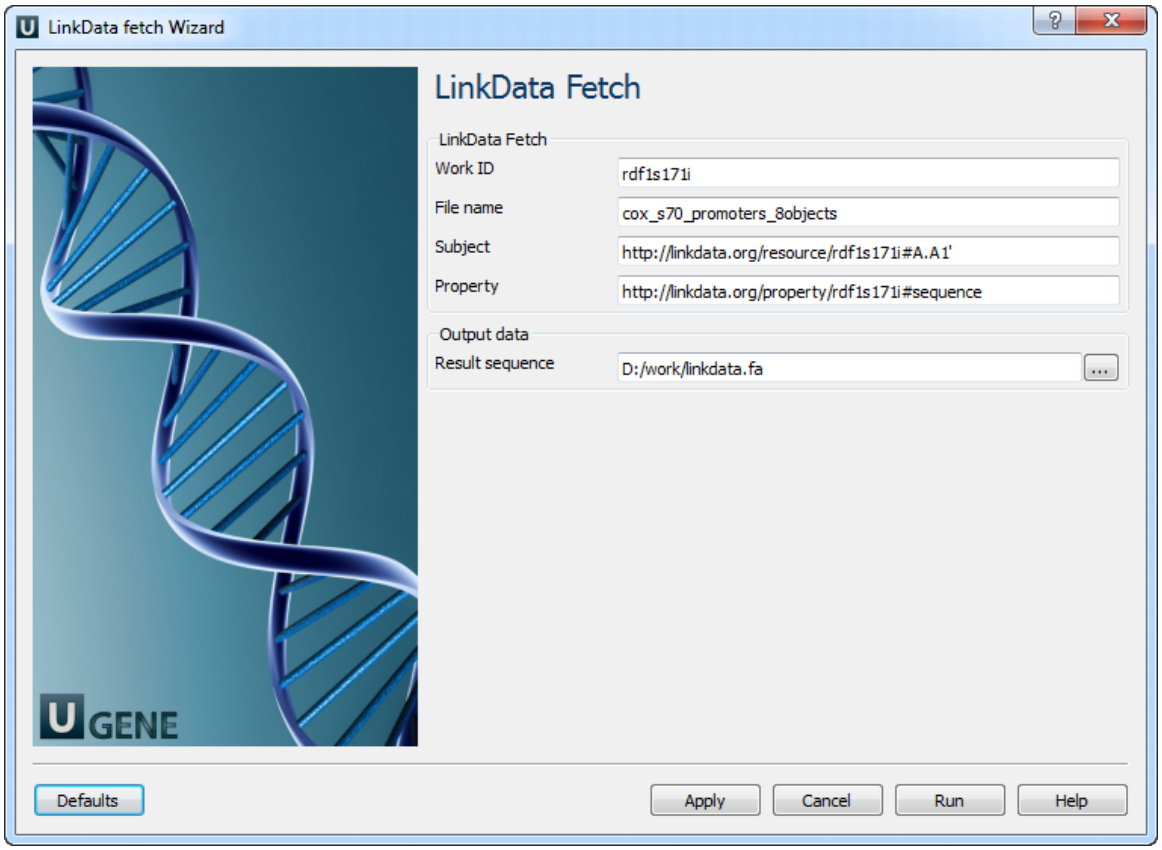
The workflow looks as follows:



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.

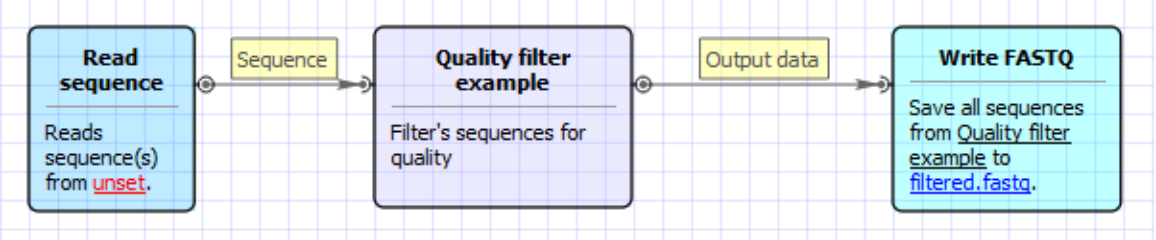
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Quality Filter" can be found in the "Custom Elements" section of the Workflow Designer samples.

Workflow Image

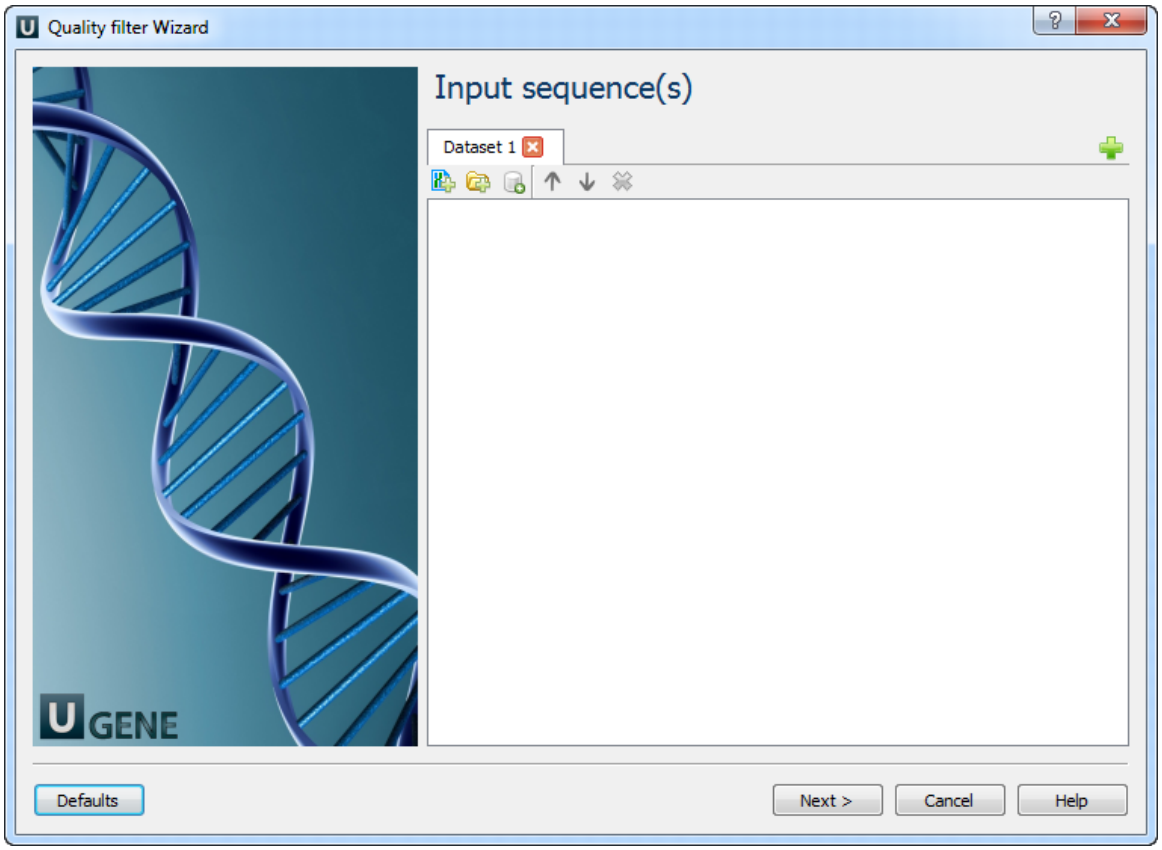
The workflow looks as follows:



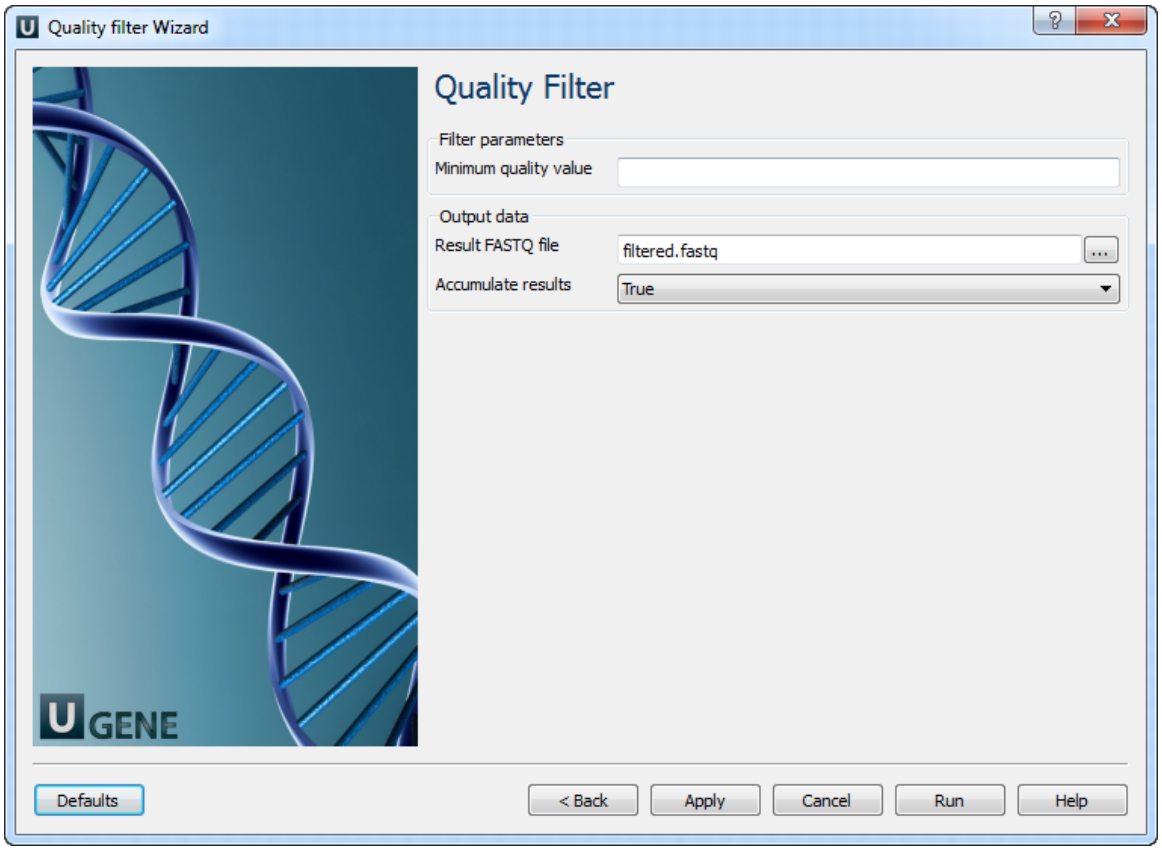
Workflow Wizard

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 Sequences by Annotation Number
- Marking Sequences by Length

Marking Sequences 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 marks 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.

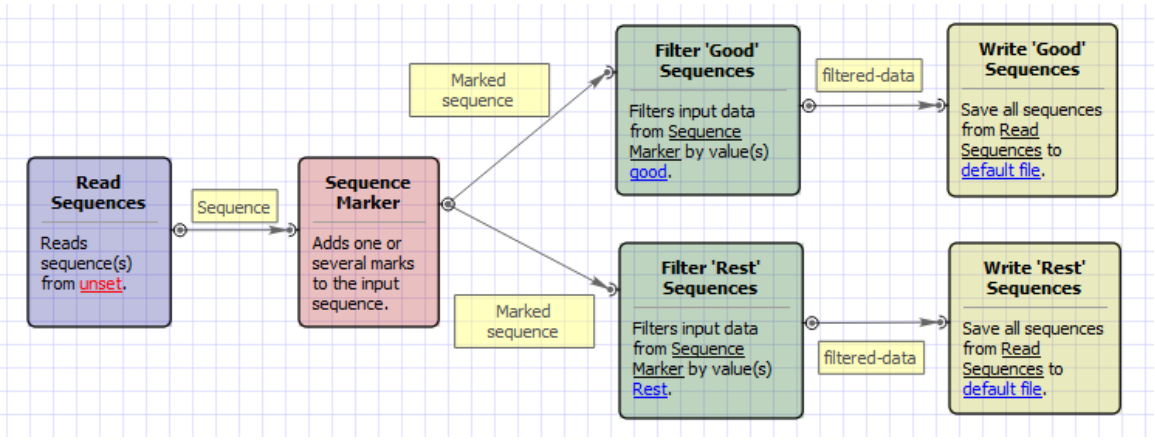
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

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 Sequences 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 marks as “Short”. A sequence with a length more than 200 bp is marks as “Long”. You can configure this value in the *Sequence Marker* element parameters.

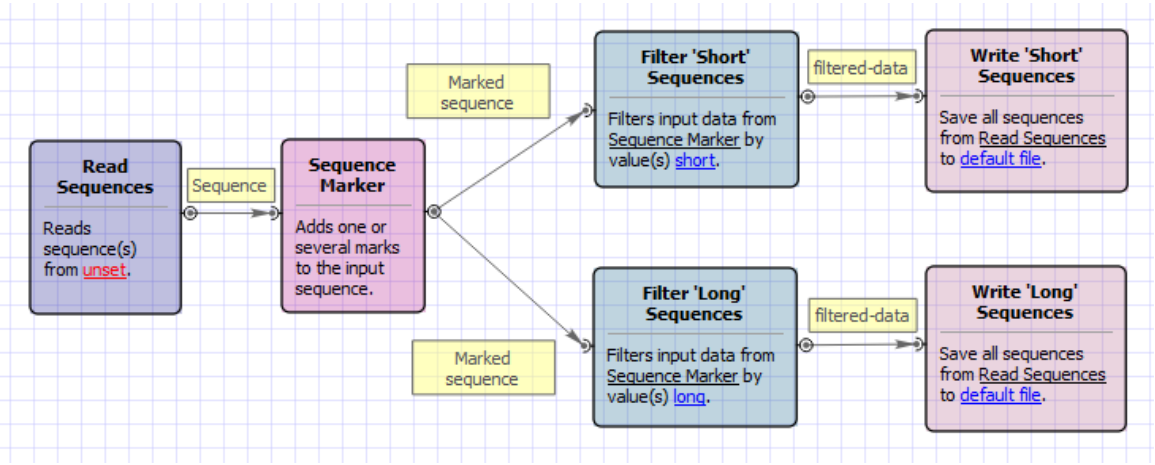
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

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).

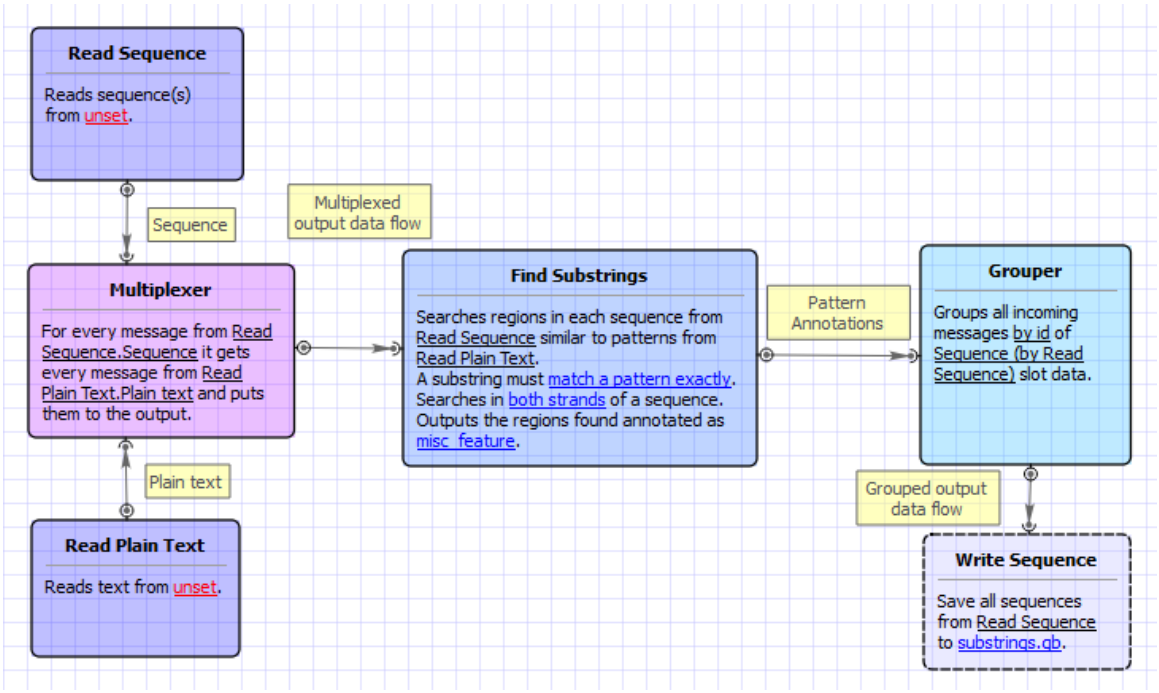
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

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

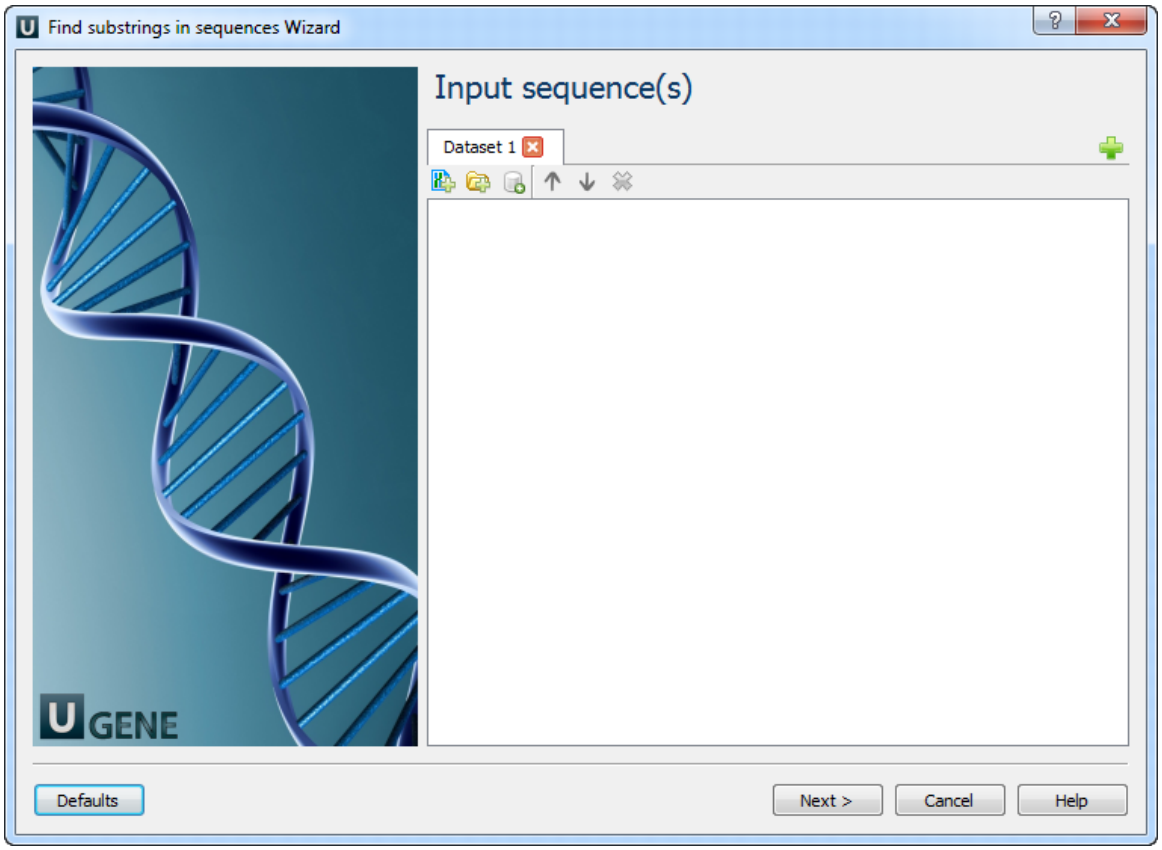
The workflow looks as follows:



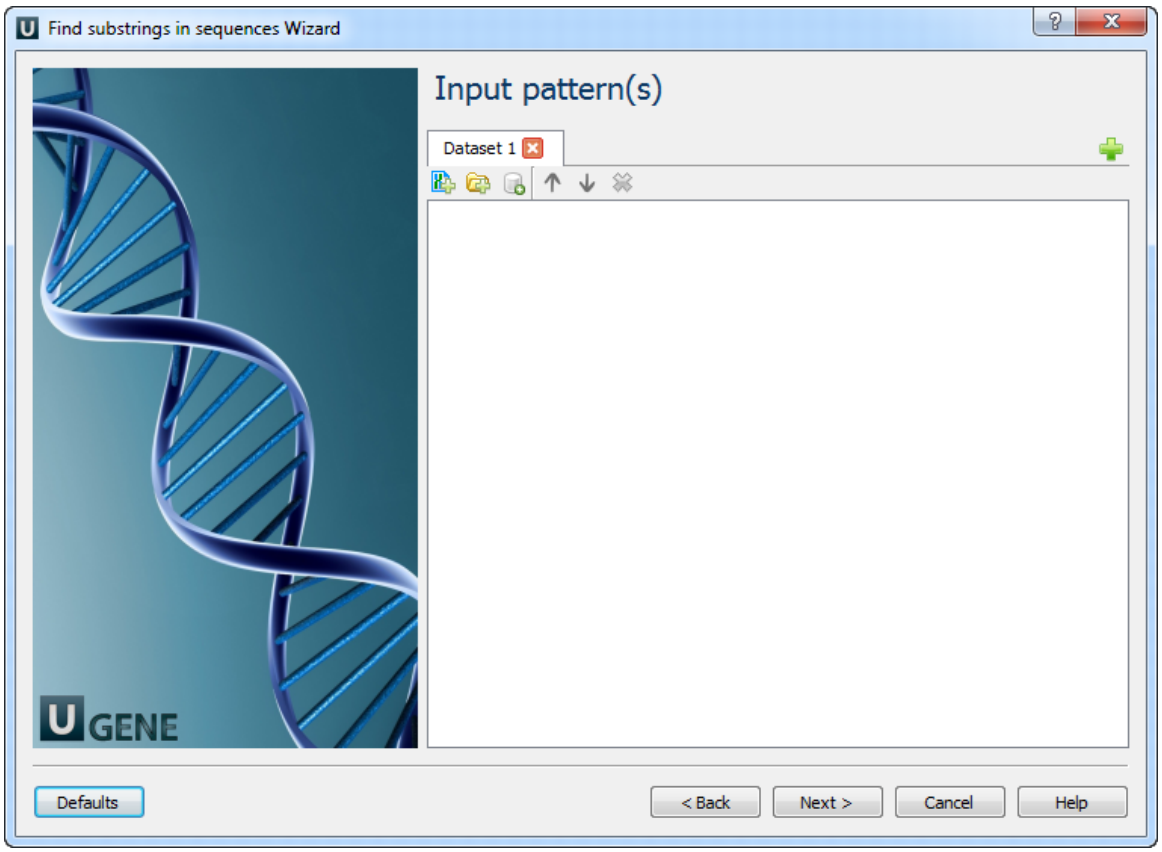
Workflow Wizard

The wizard has 3 pages.

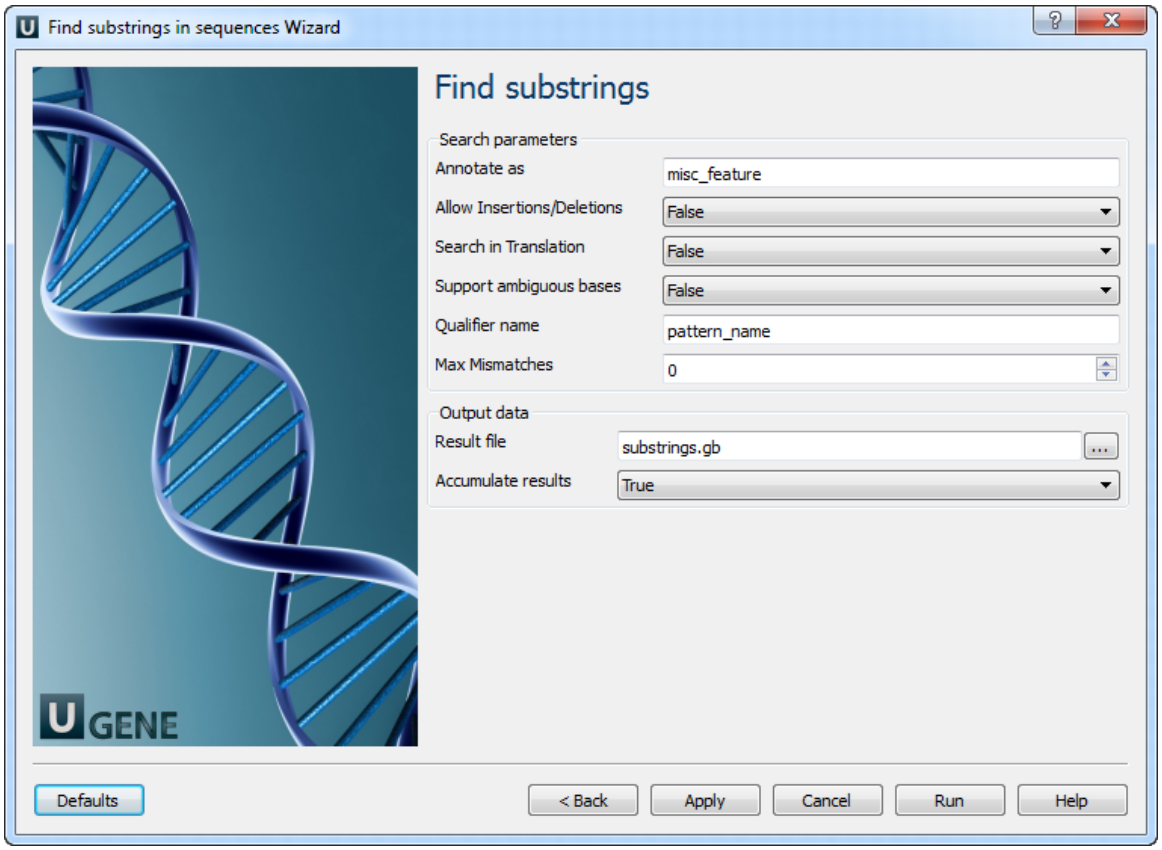
1. Input sequence(s): On this page you must input sequence(s).



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

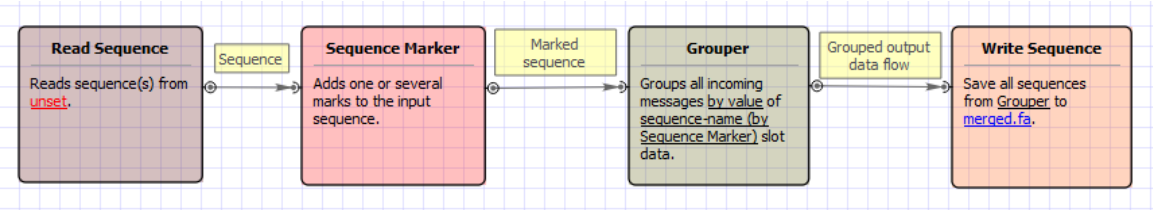
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

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).

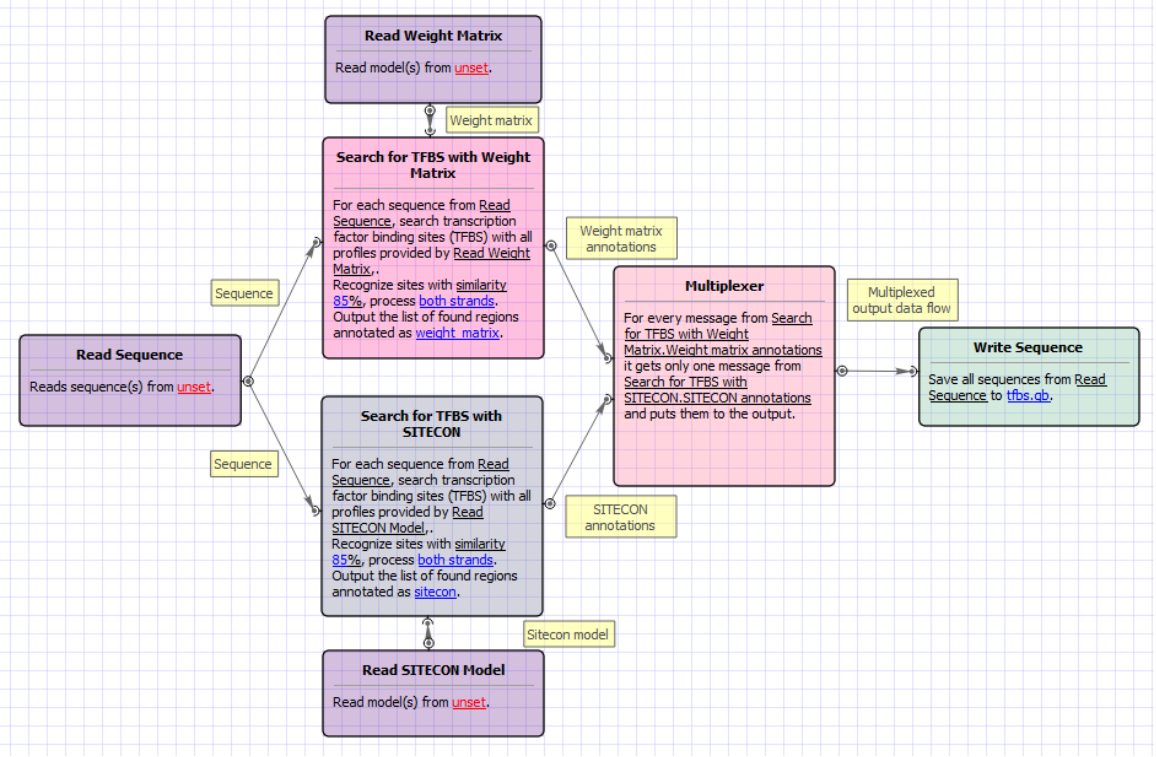
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Search for TFBS" can be found in the "Data Merging" section of the Workflow Designer samples.

Workflow Image

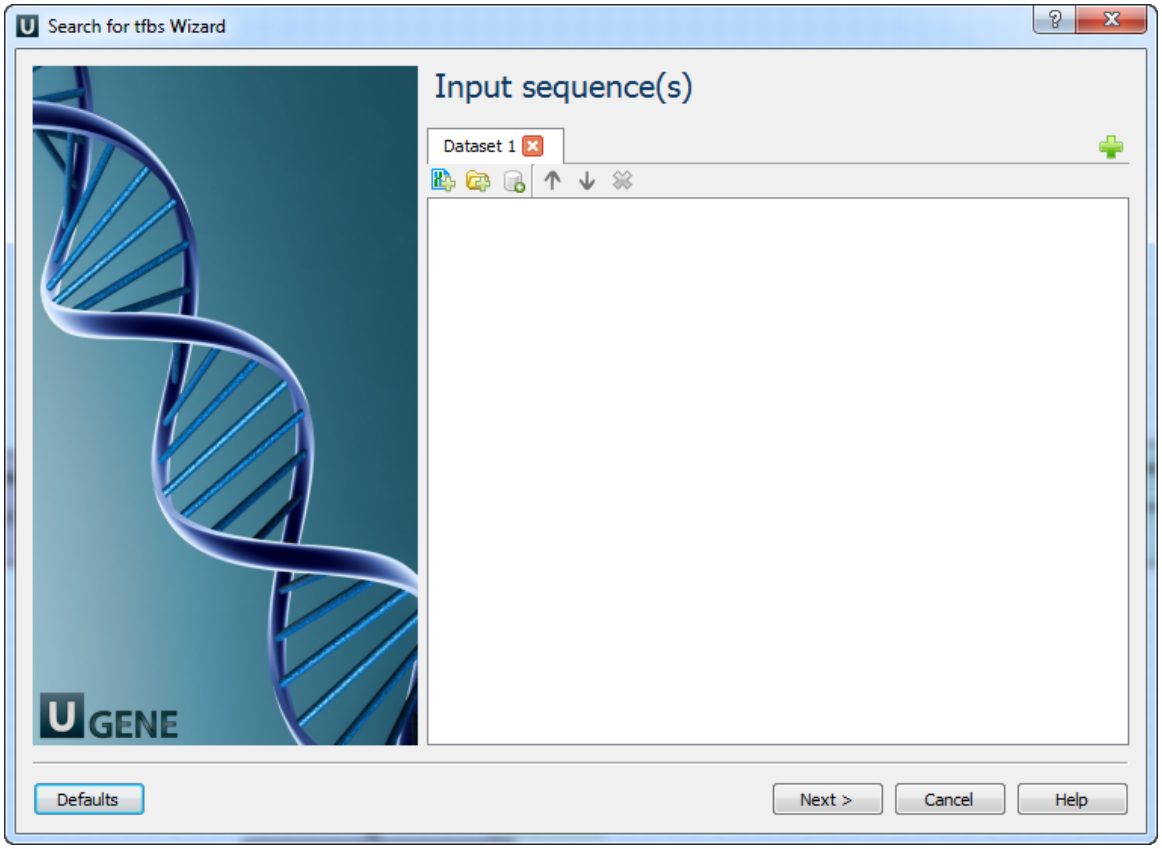
The workflow looks as follows:



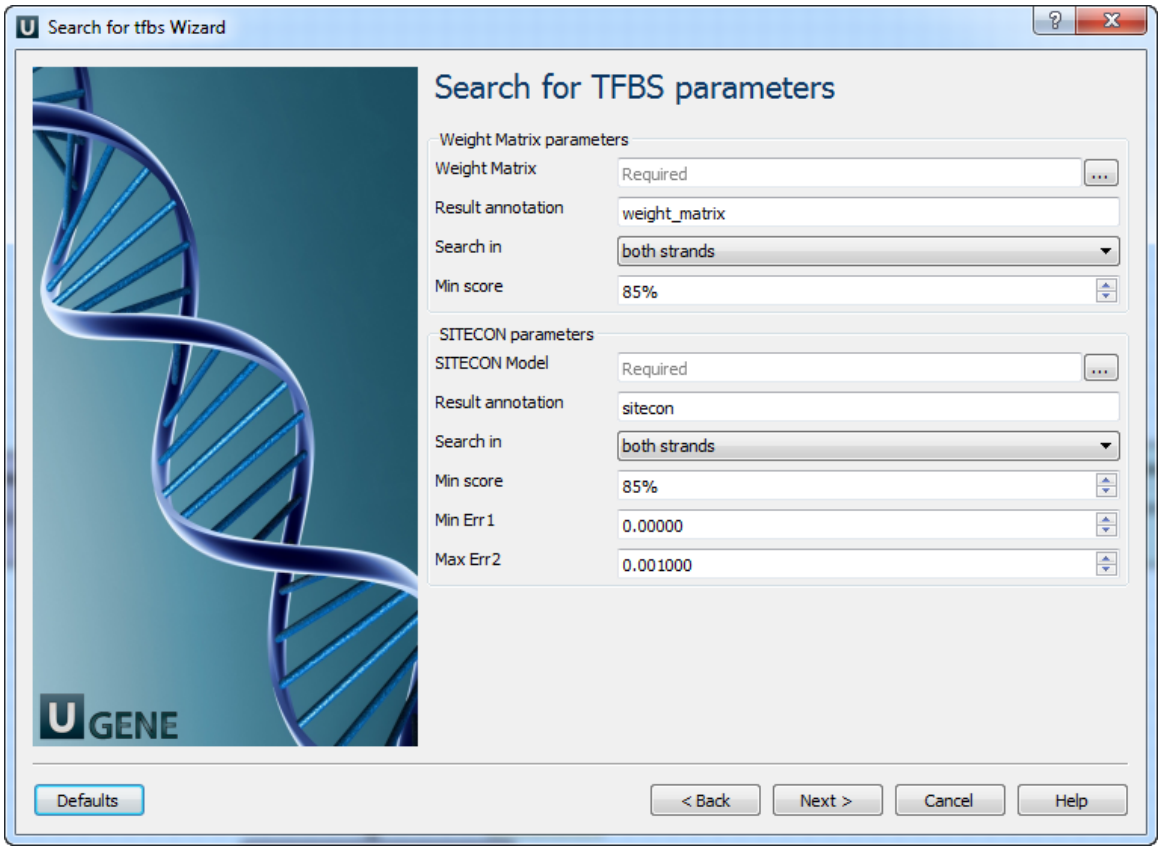
Workflow Wizard

The wizard has 3 pages.

1. Input sequence(s): On this page you must input sequence(s).



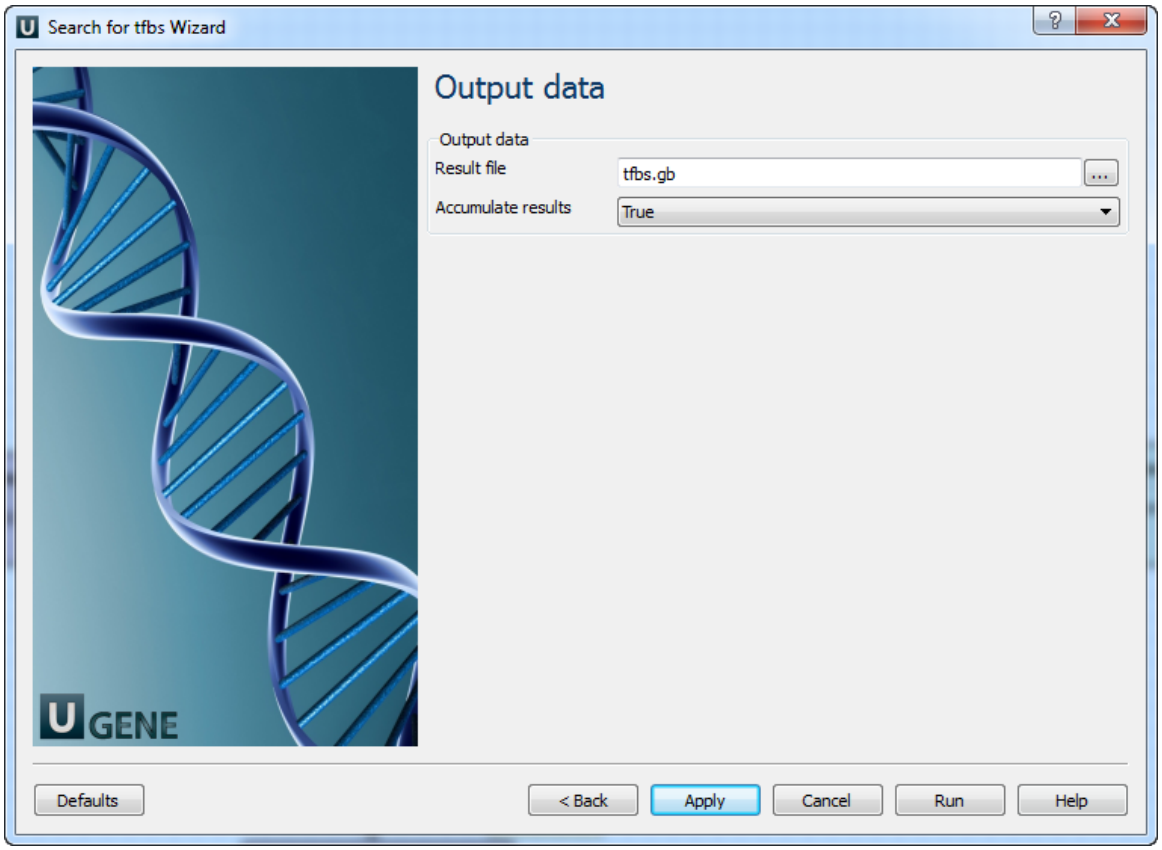
2. Search for TFBS parameters: On this page you can modify search for TFBS parameters.



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.

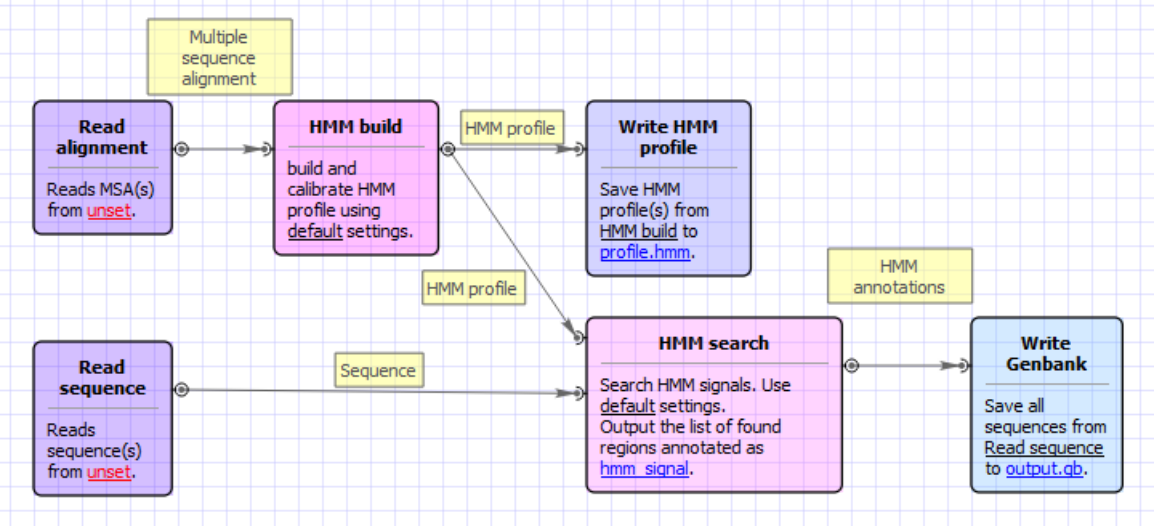
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

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

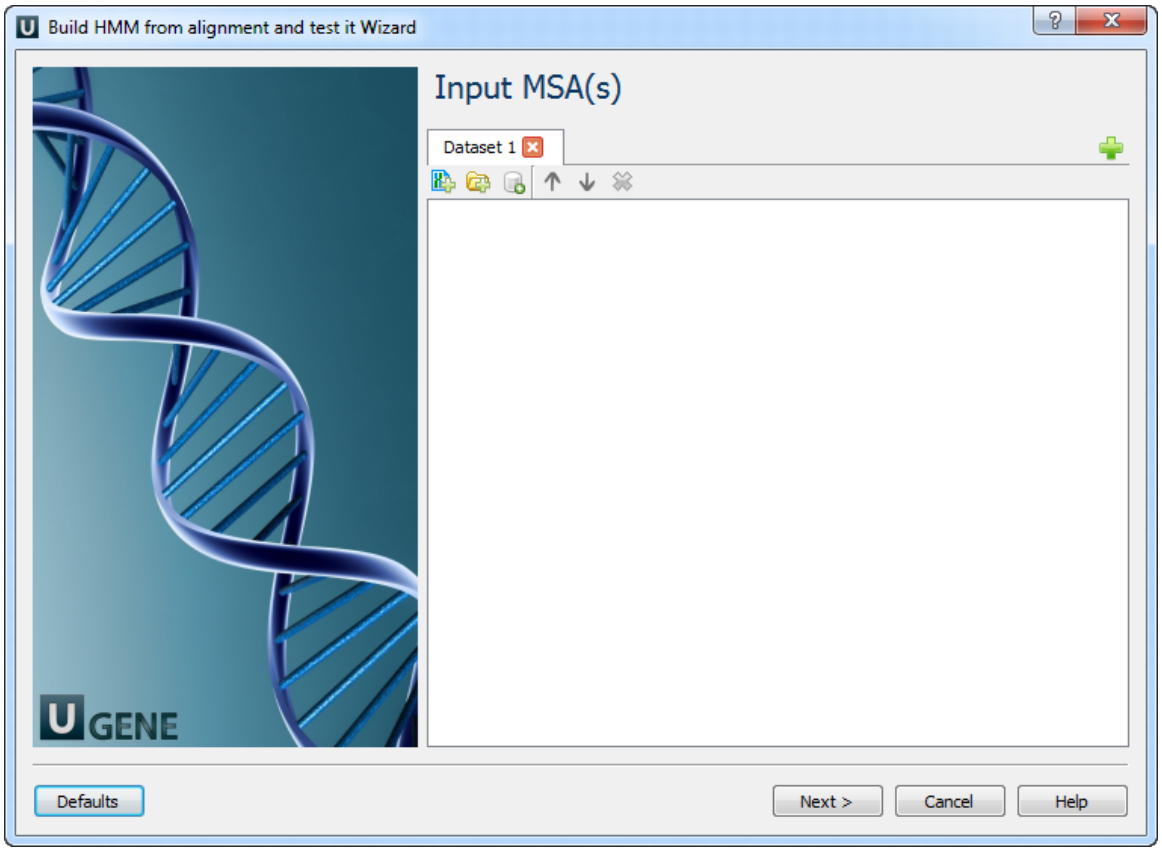
The workflow looks as follows:



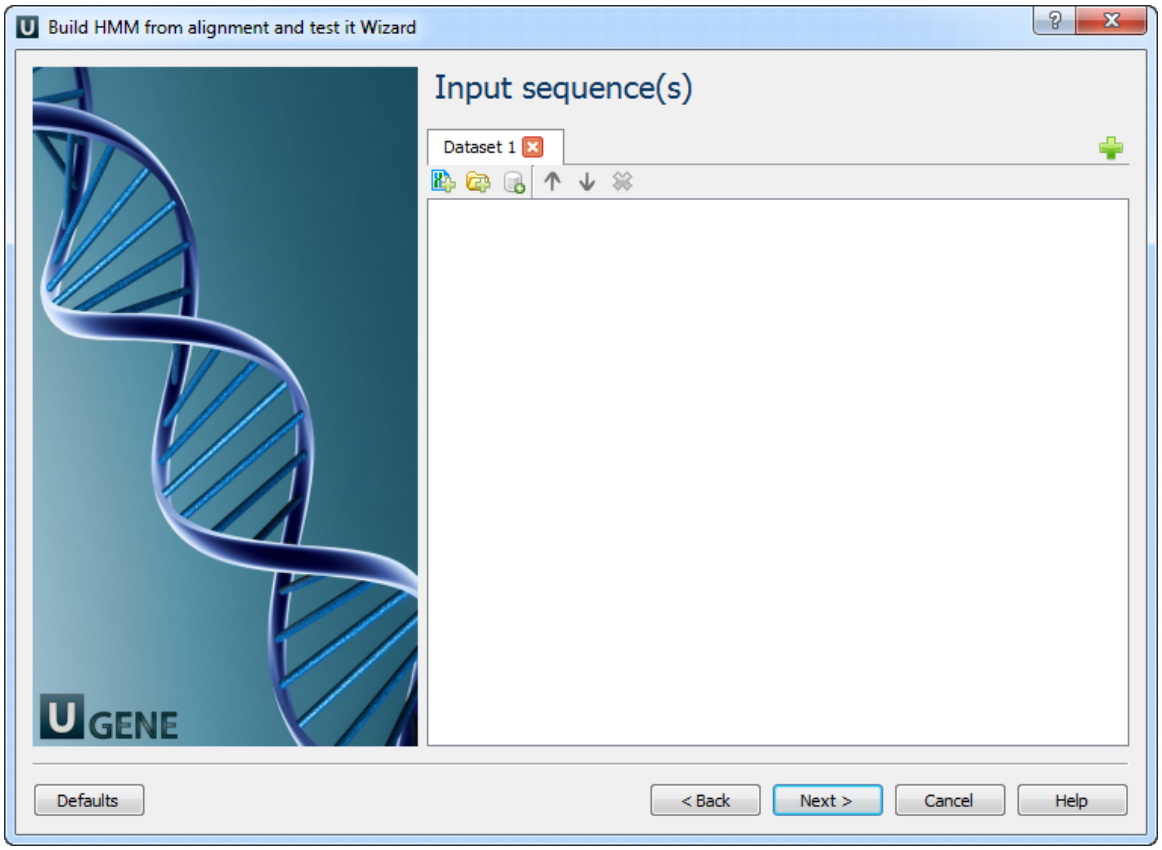
Workflow Wizard

The wizard has 4 pages.

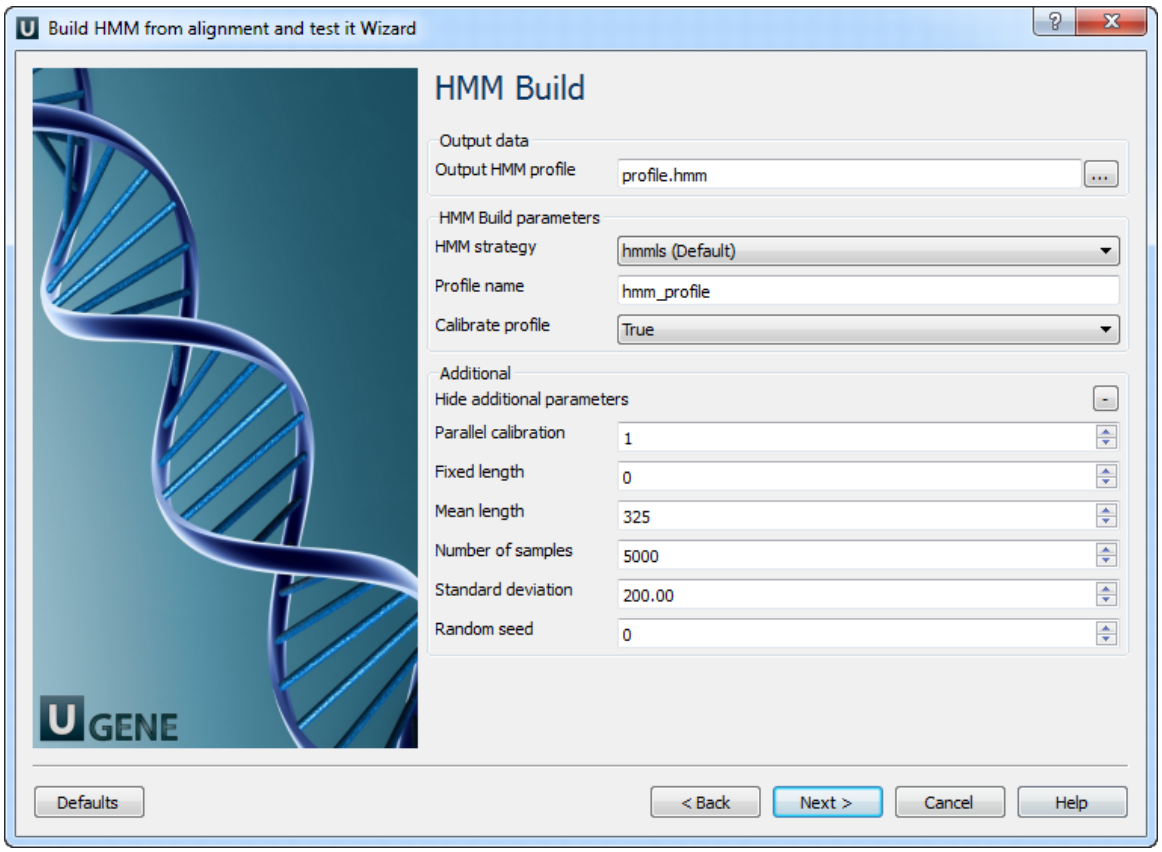
1. Input MSA(s): On this page you must input MSA(s).



2. Input sequence(s): On this page you must input sequence(s).



3. HMM build: On this page you can modify HMM build parameters.

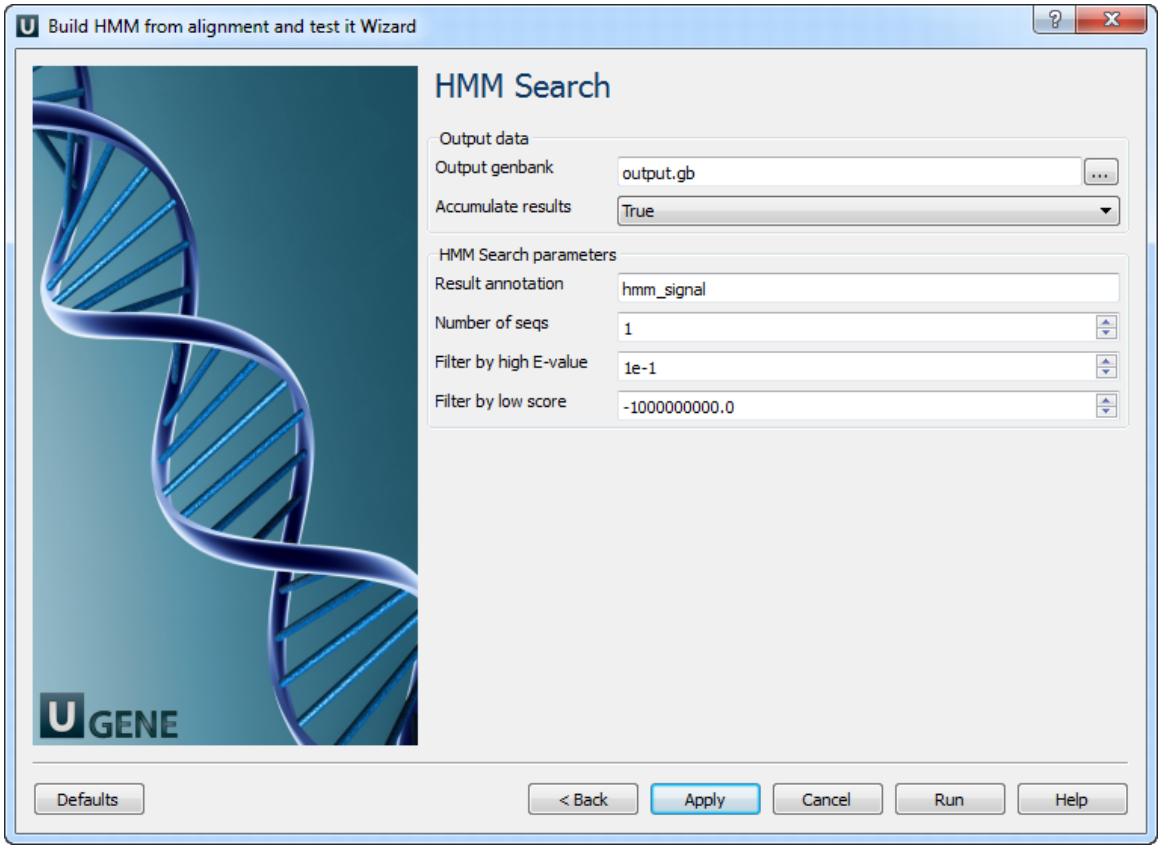


The following parameters are available:

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.

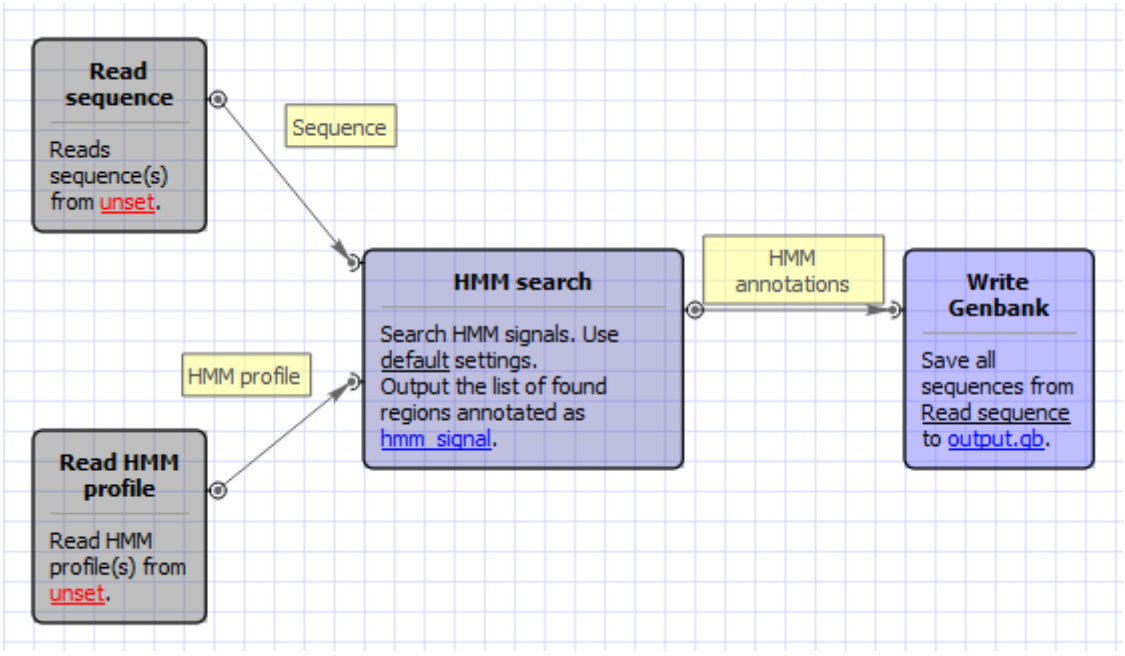
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

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

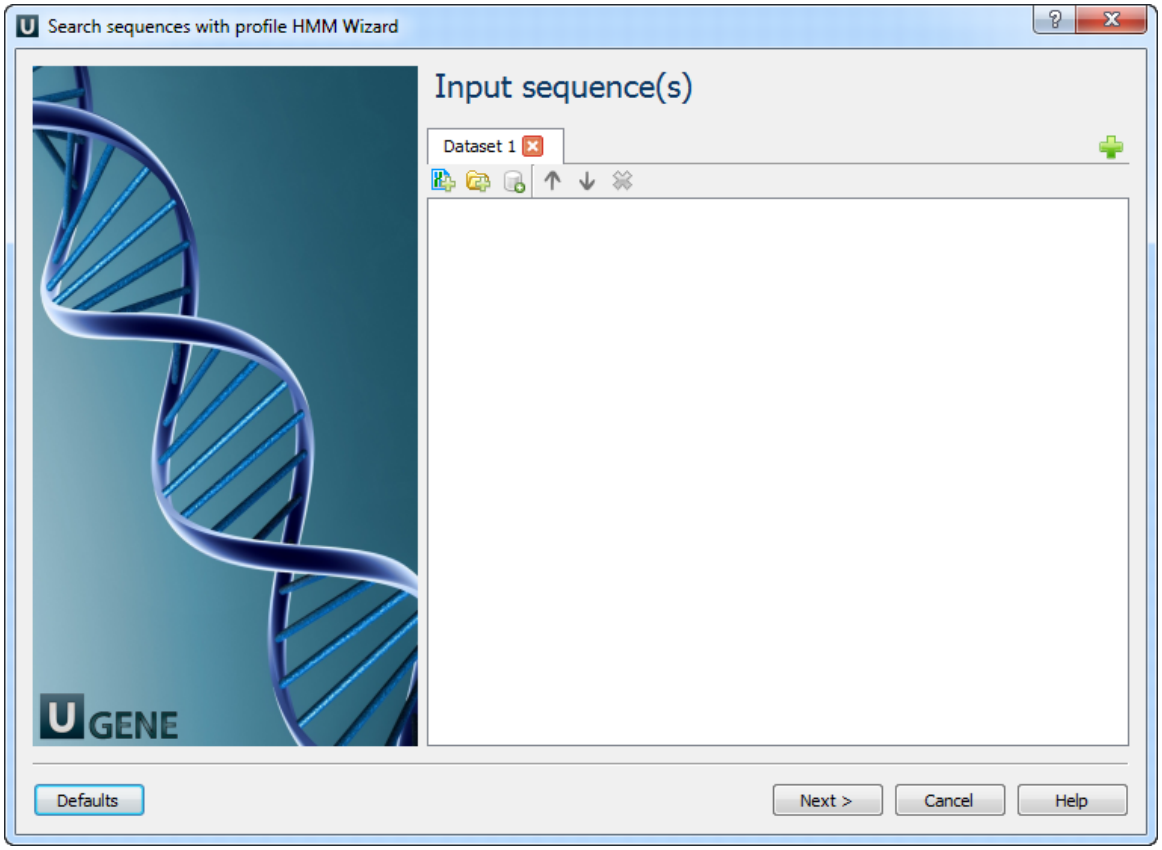
The workflow looks as follows:



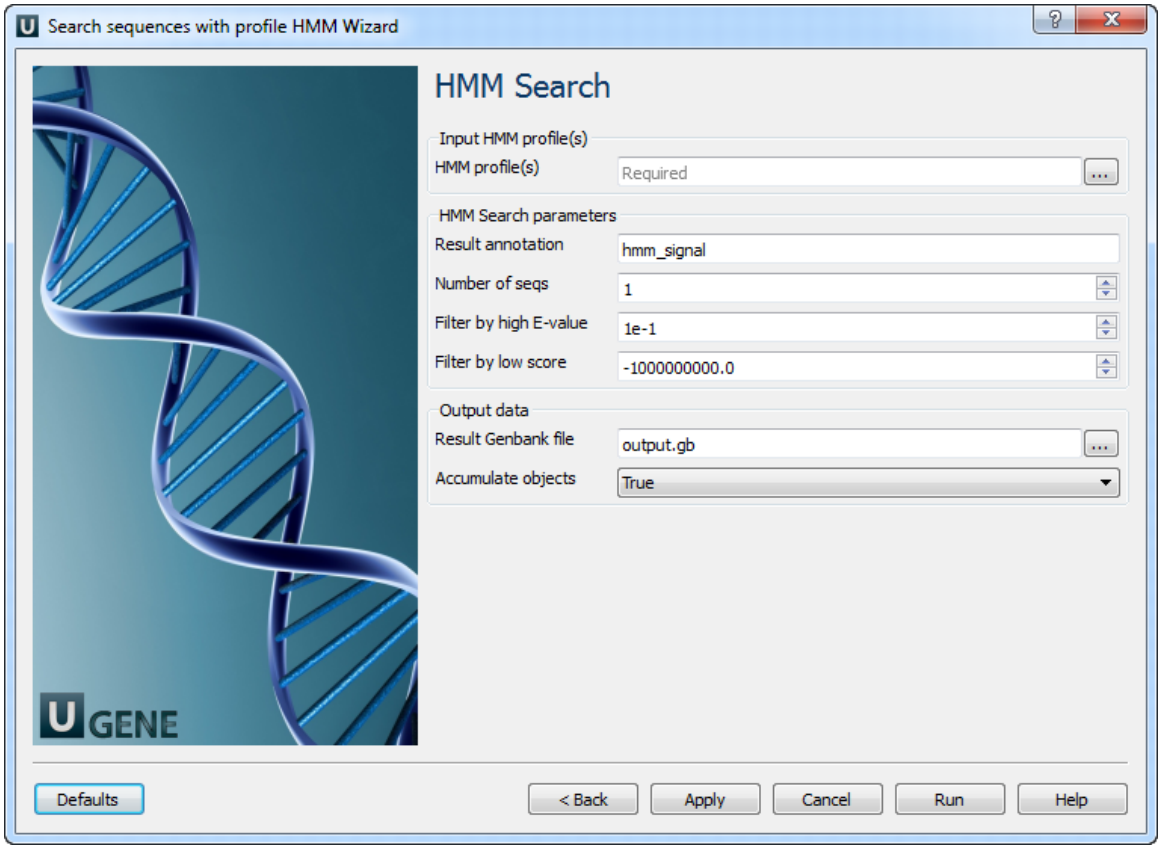
Workflow Wizard

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.
Filter by low score	Score based filtering is an alternative to E-value filtering to exclude low-probability hits from result.
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

- De novo Assembly with Spades
- Call Variants with SAMtools
- ChIP-Seq Coverage
- ChIP-seq Analysis with Cistrome Tools
- Extract Consensus from Assembly
- Extract Coverage from Assembly
- Extract Transcript Sequences
- Quality Control by FastQC
- Raw ChIP-Seq Data Processing
- Raw DNA-Seq Data Processing
- Raw RNA-Seq Data Processing
- RNA-seq Analysis with Tuxedo Tools
- Get Unmappet Reads
- Variation Annotation with SnpEff

De novo Assembly with Spades

Use this workflow sample to assemble genomes from short reads with Spades.

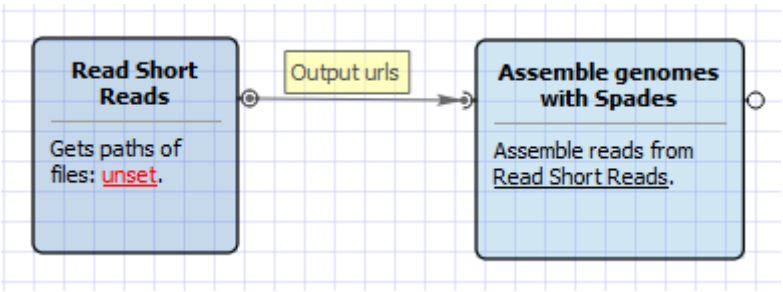
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

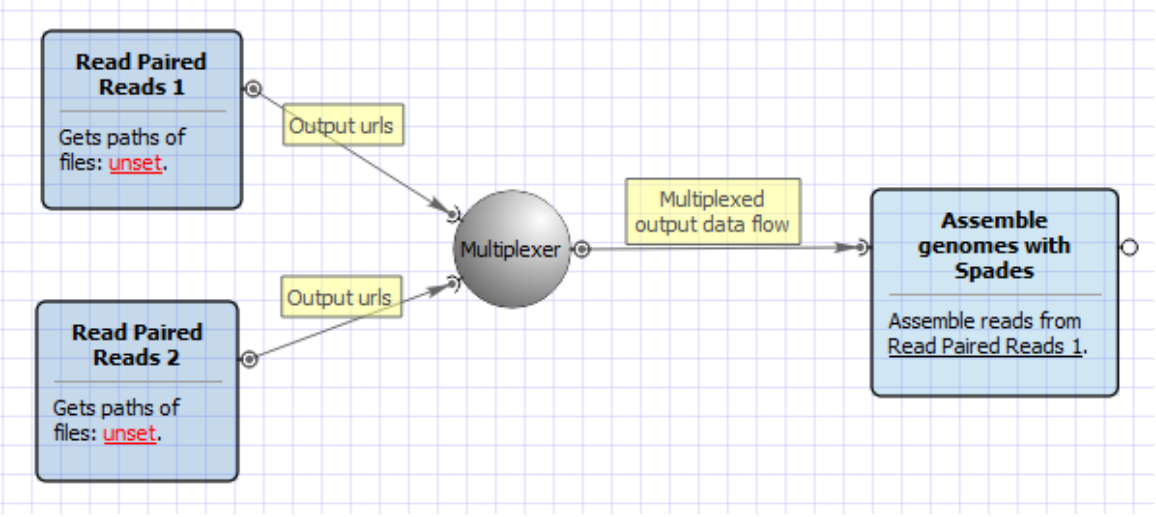
The workflow sample "Assembly with Spades" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

There are two versions of the workflow available. The workflow for single tags looks as follows:



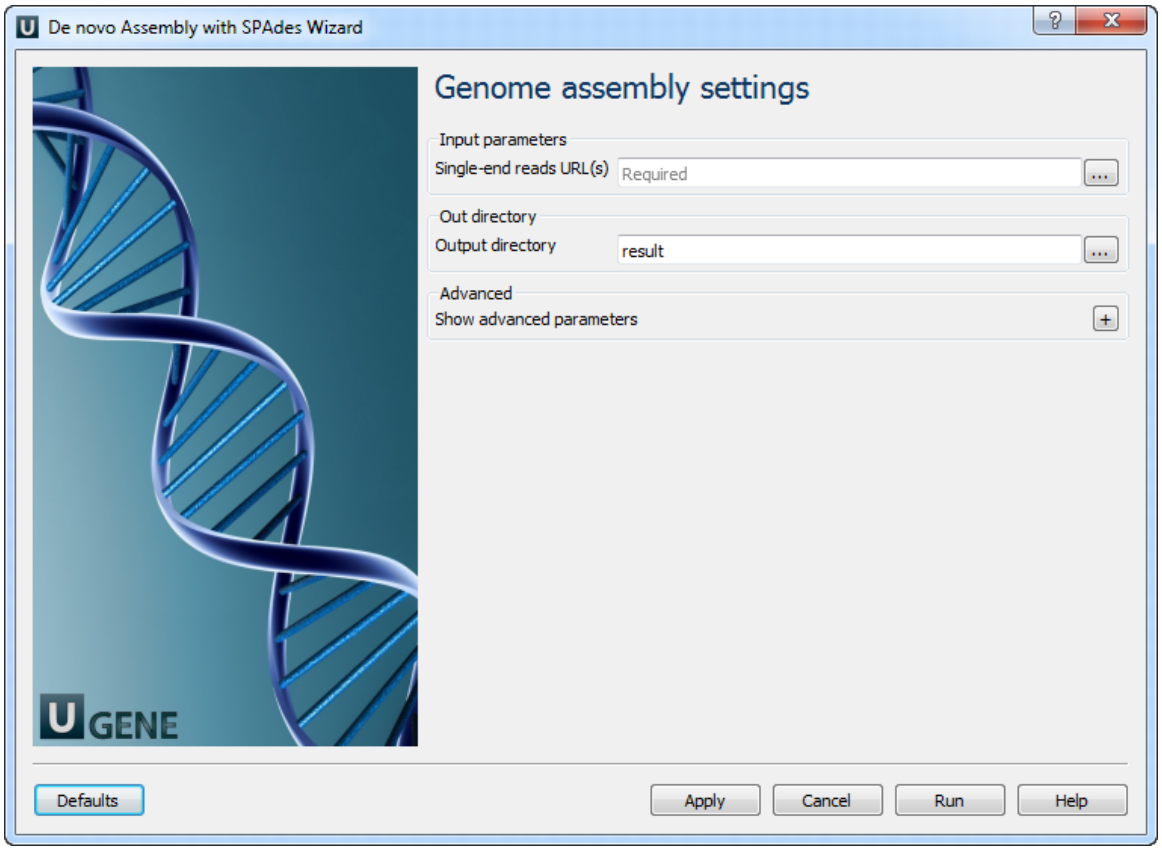
The workflow for paired tags appearance is the following:



Workflow Wizard

The wizard for single tags has 1 page.

1. Genome assembly settings: On this page you must input single reads and optionally modify advanced parameters.

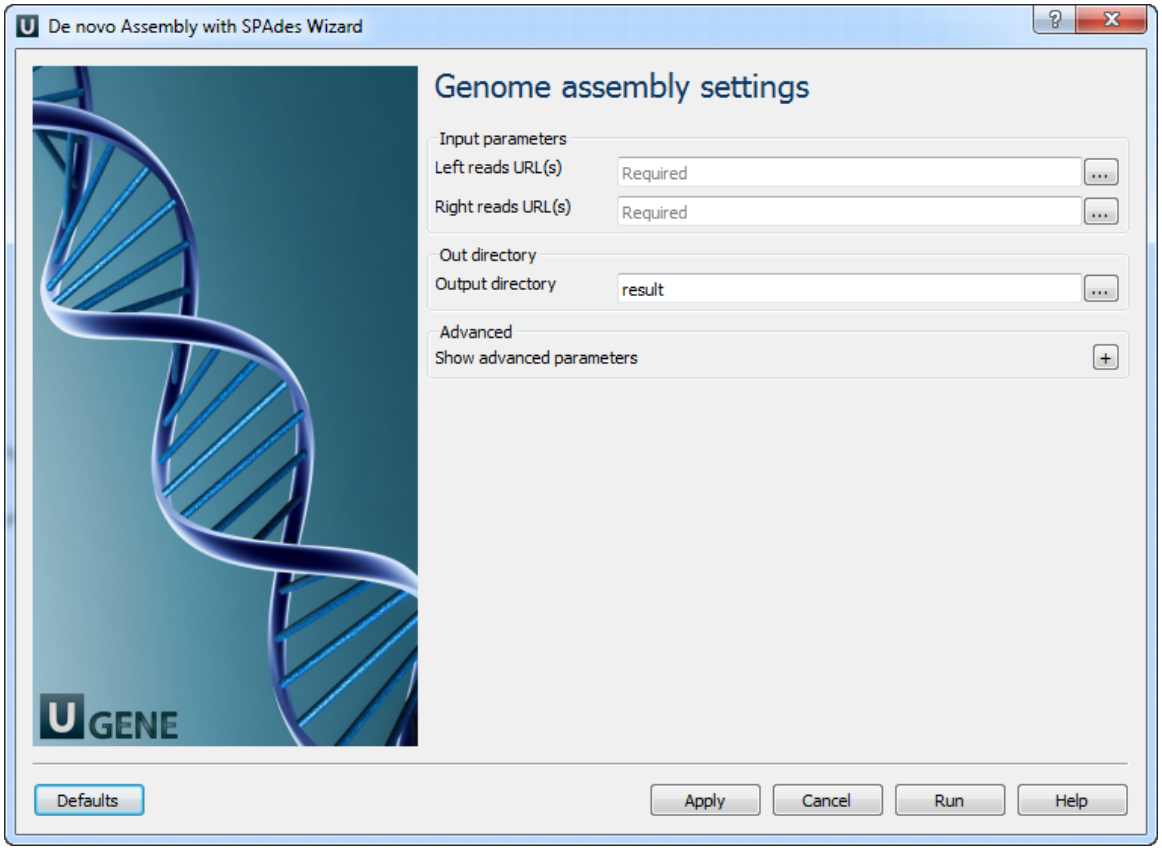


The following parameters are available:

Single-end reads URL(s)	Semicolon-separated list of pathes to the input files.
Output directory	Directory to save Spades output files.
Dataset type	Input dataset type.
Running mode	Running mode.
K-mers	k-mer sizes (-k).

The wizard for paired tags has 1 page.

1. Genome assembly settings: On this page you must input paired reads and optionally modify advanced parameters.



The following parameters are available:

Left reads URL(s)	Semicolon-separated list of pathes to the input files.
Right reads URL(s)	Semicolon-separated list of pathes to the input files.
Output directory	Directory to save Spades output files.
Dataset type	Input dataset type.
Running mode	Running mode.
K-mers	k-mer sizes (-k).

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.

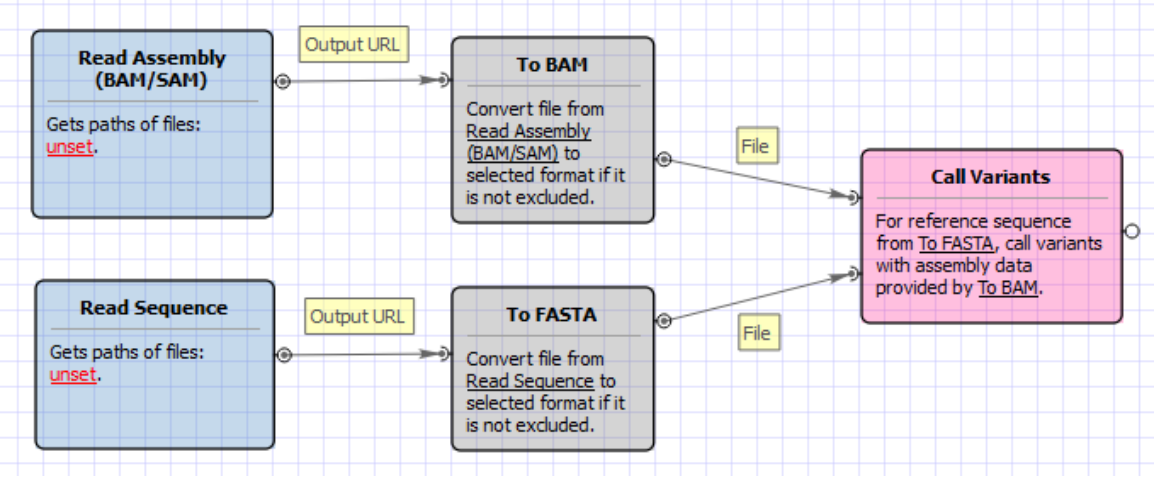
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Call Variants with SAMtools" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

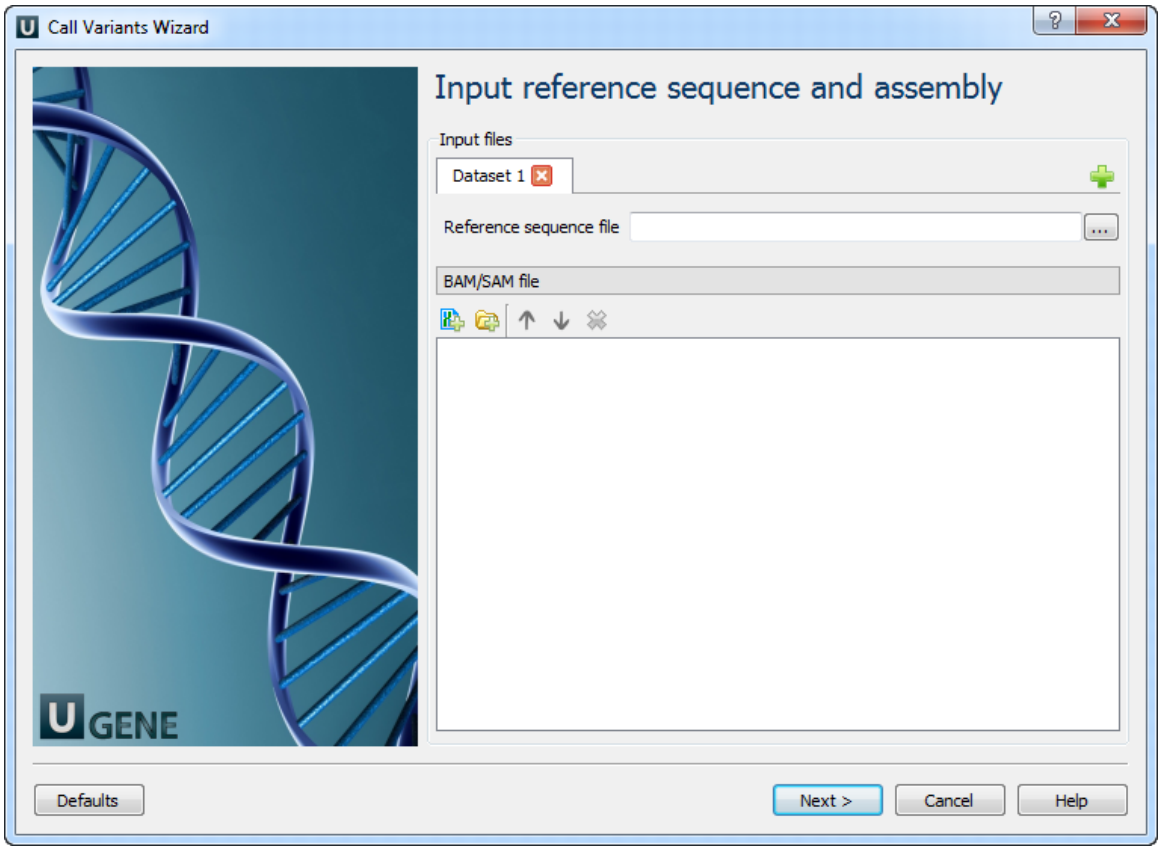
The workflow looks as follows:



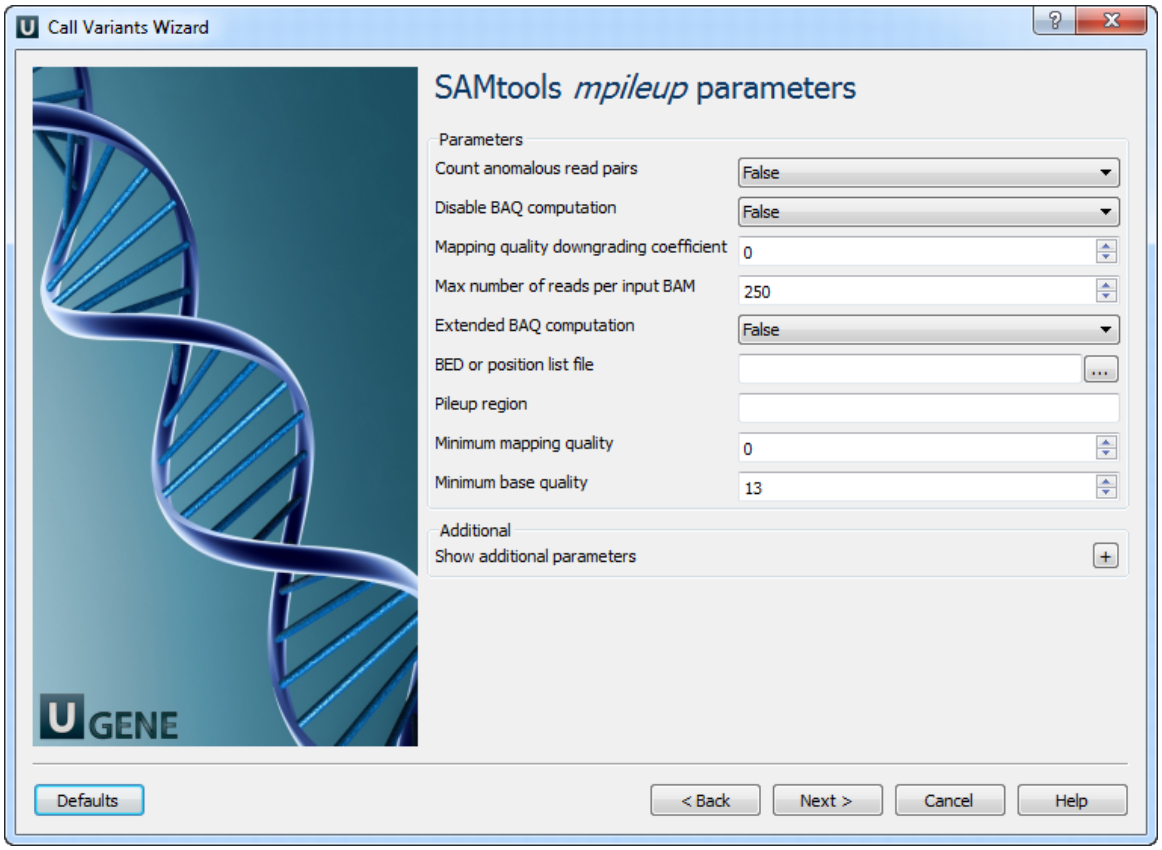
Workflow Wizard

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.

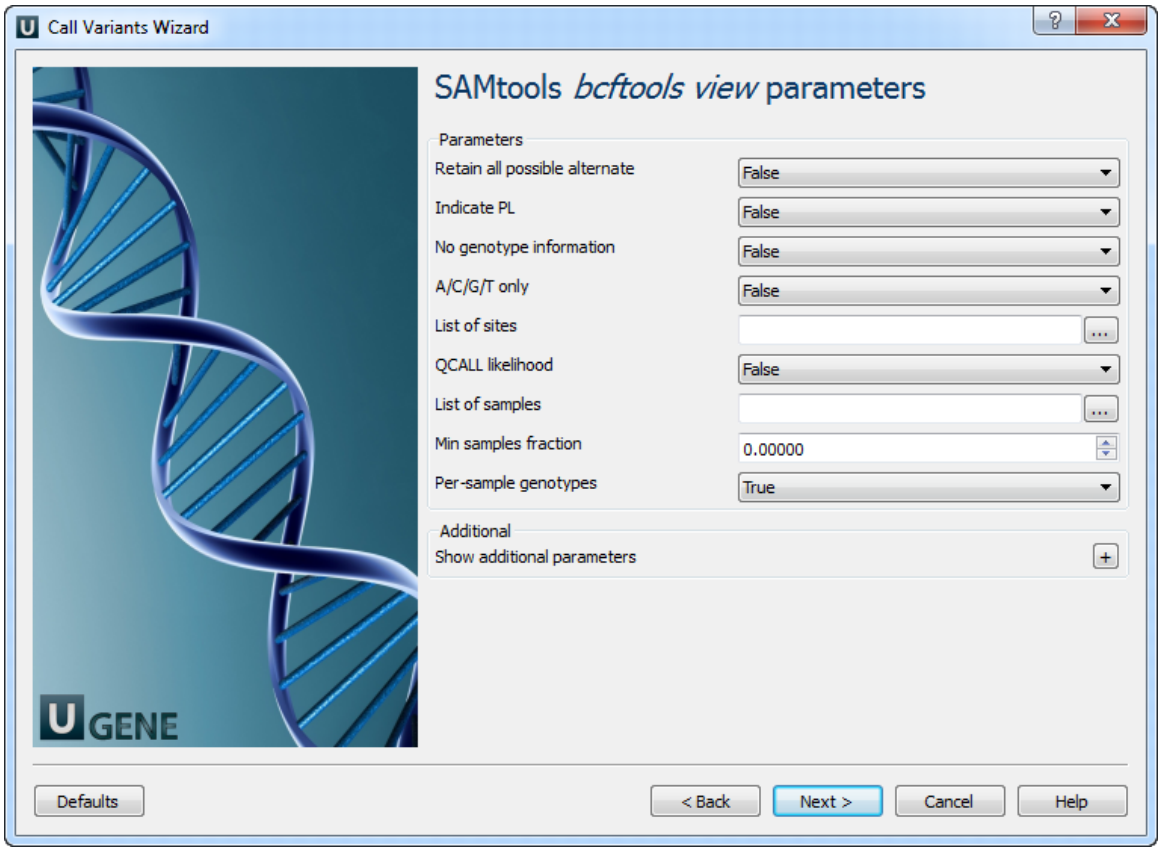


The following parameters are available:

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.

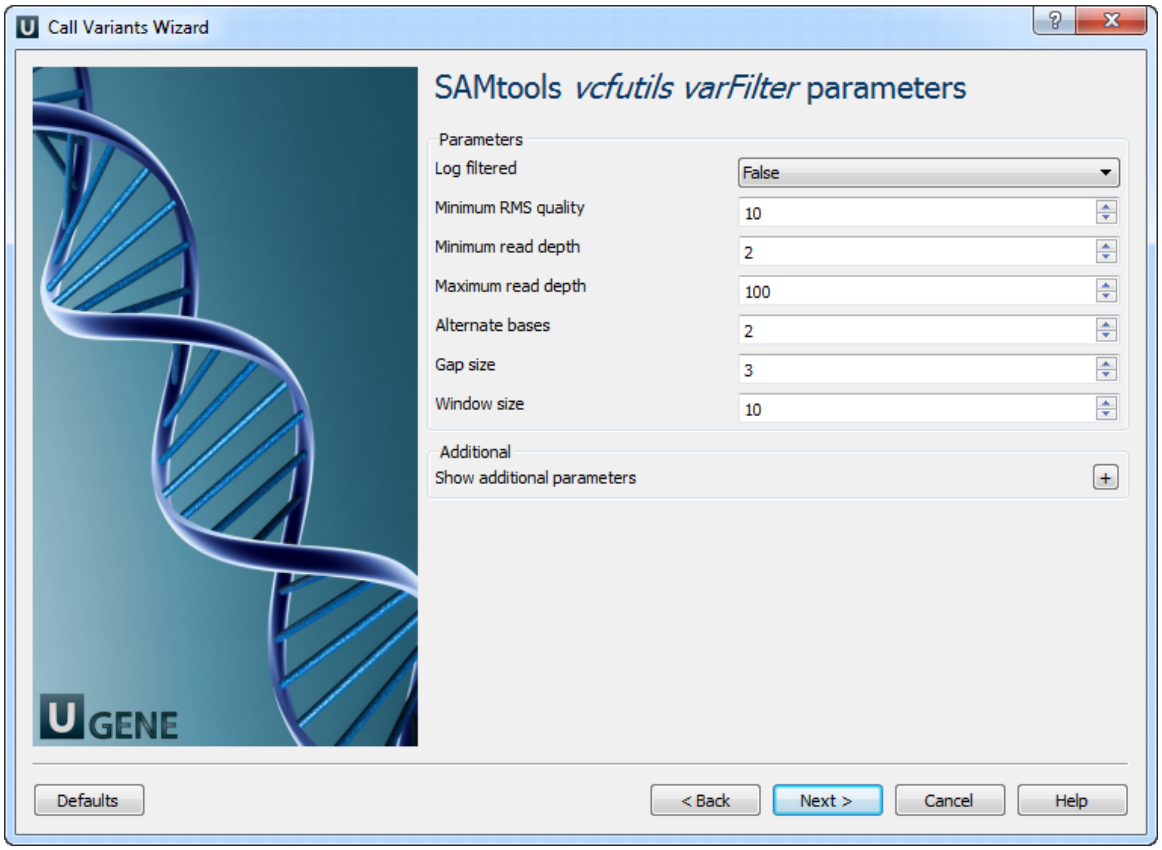


The following parameters are available:

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.
Min samples fraction	Skip loci where the fraction of samples covered by reads is below FLOAT.
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).

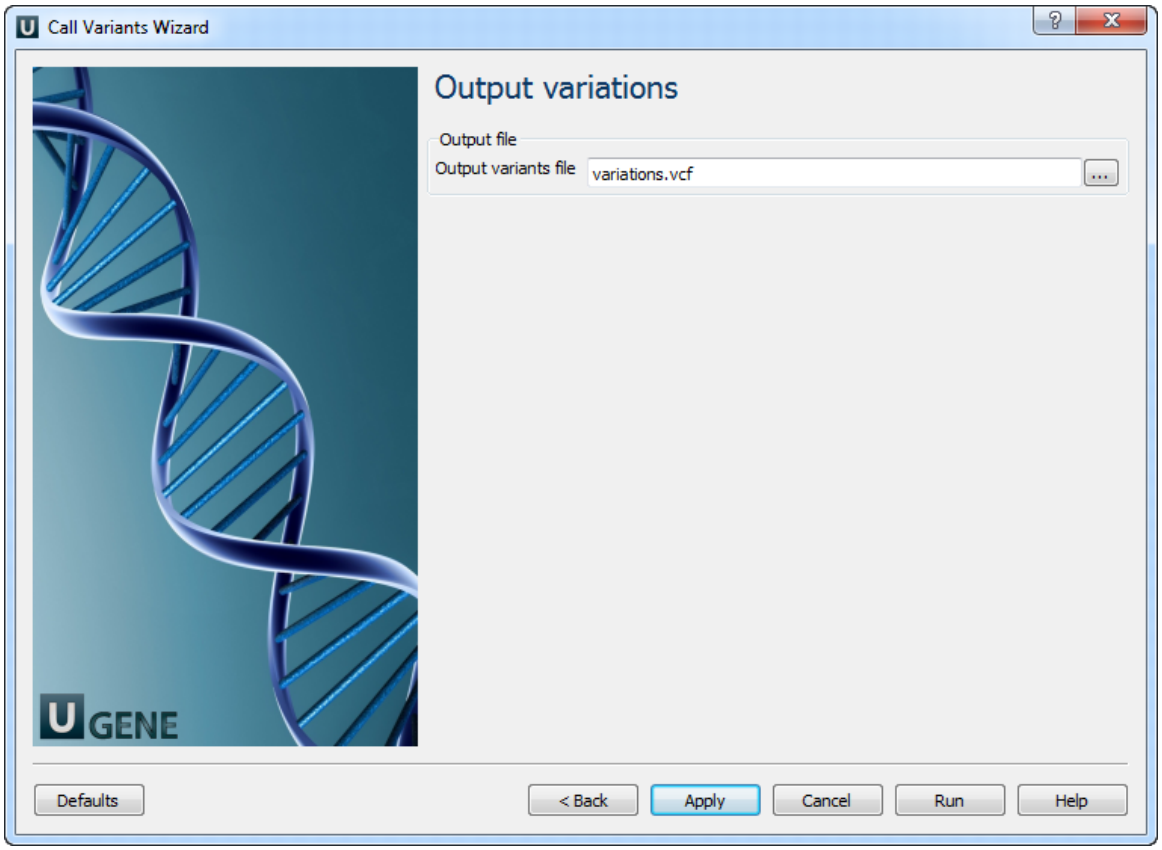
4. SAMTolls *vcfutils varFilter* parameters: The next page allows one to configure SAMtools vcfutils parameters.



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.

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.

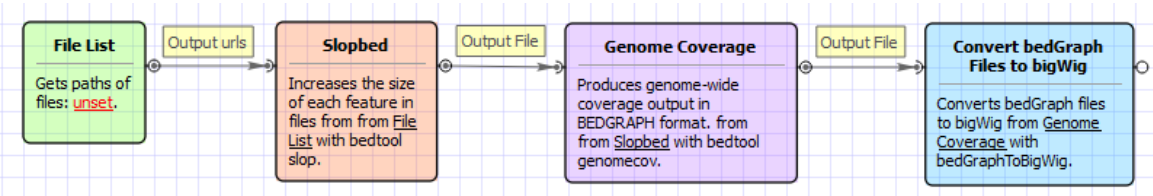
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "ChIP-Seq Coverage" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

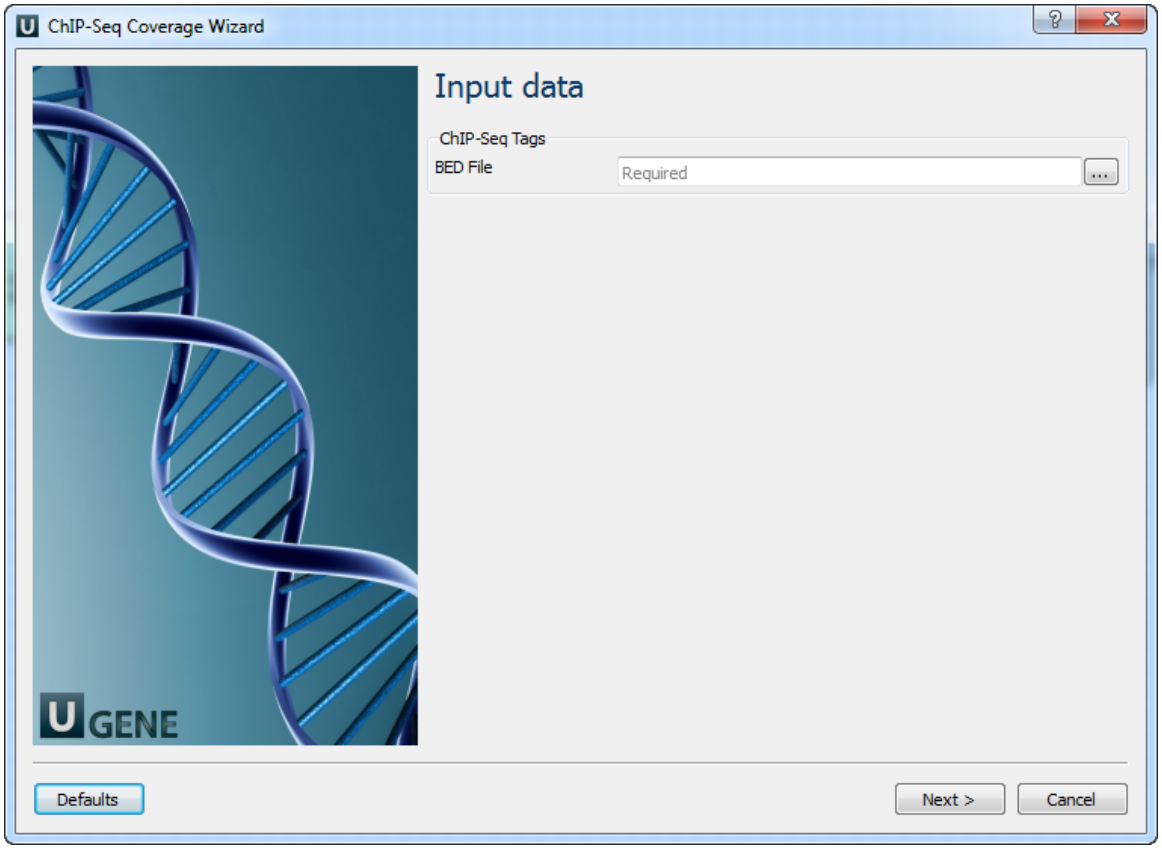
The opened workflow looks as follows:



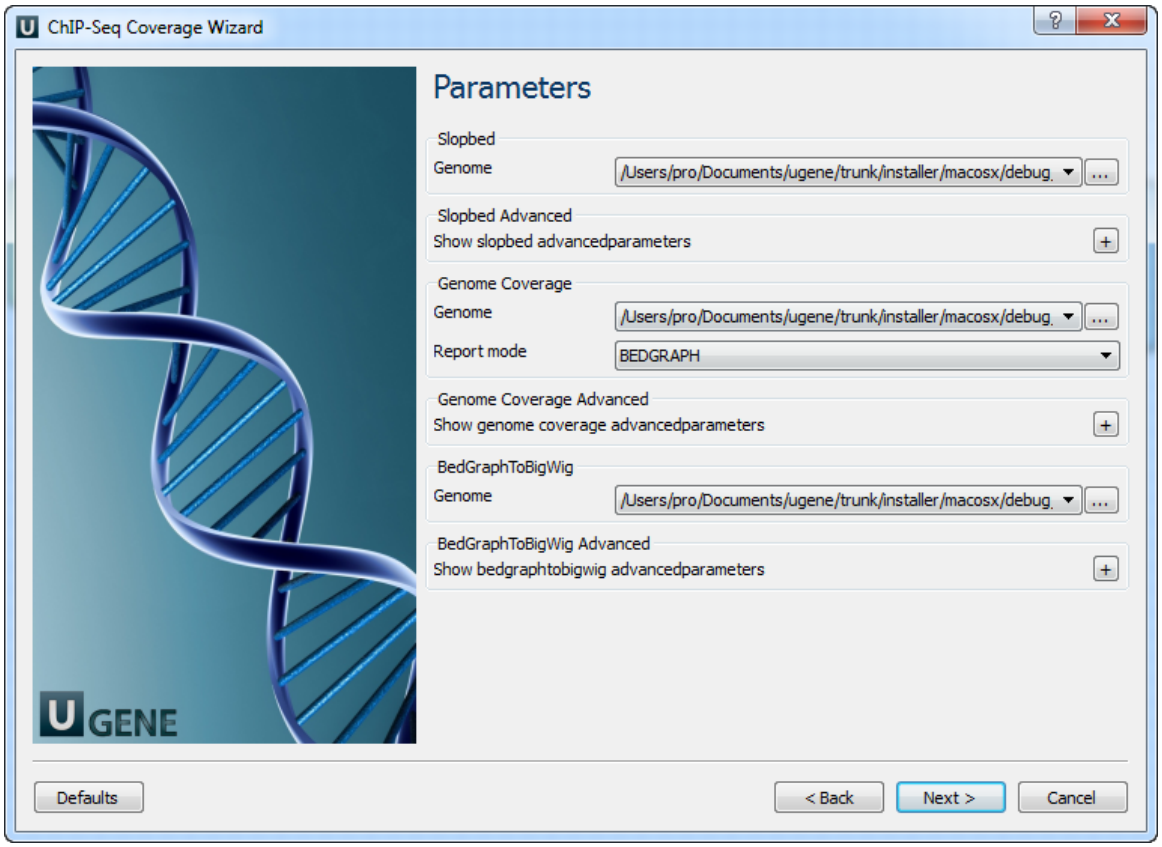
Workflow Wizard

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.

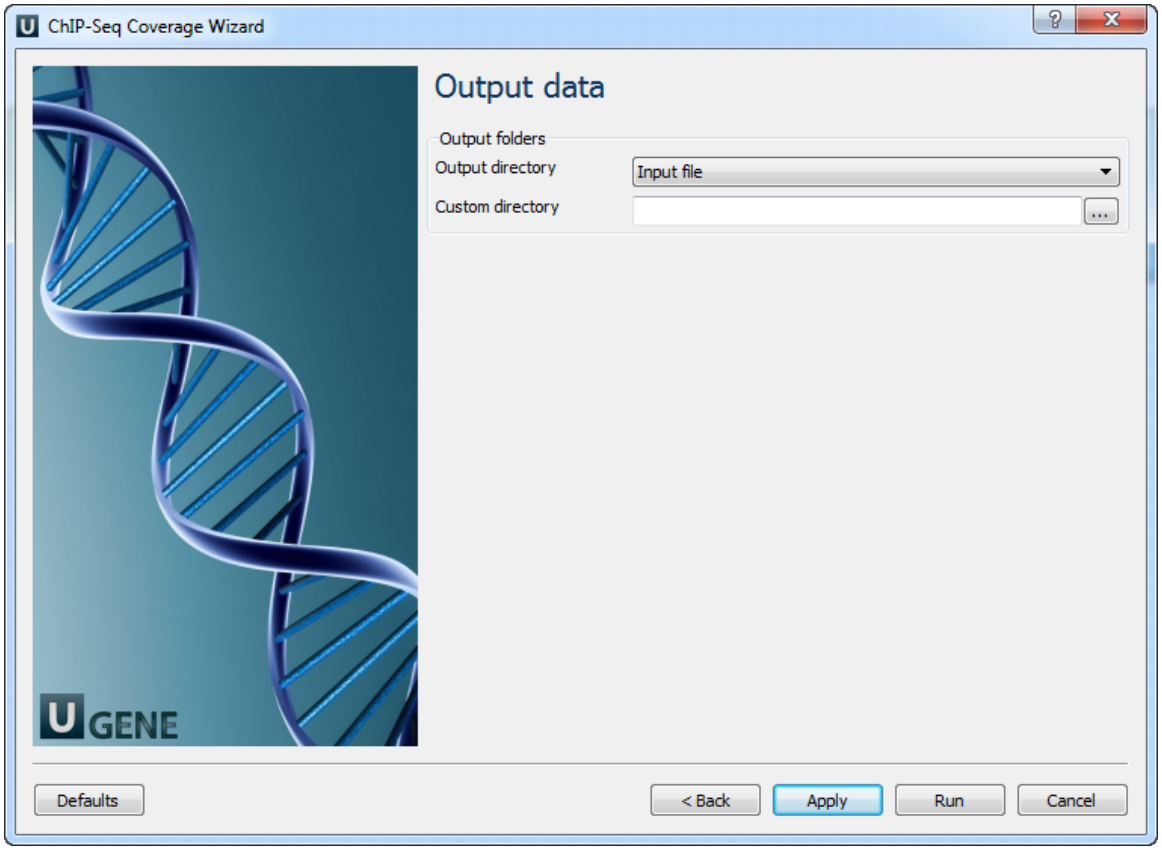


The following parameters are available:

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)
Substract 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

Download and install the UGENE NGS package to use this pipeline.

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 which genes are nearby so can be regarded as potential regulated genes, then perform GO analysis;
2. to check the conservation scores at the binding sites;
3. 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.

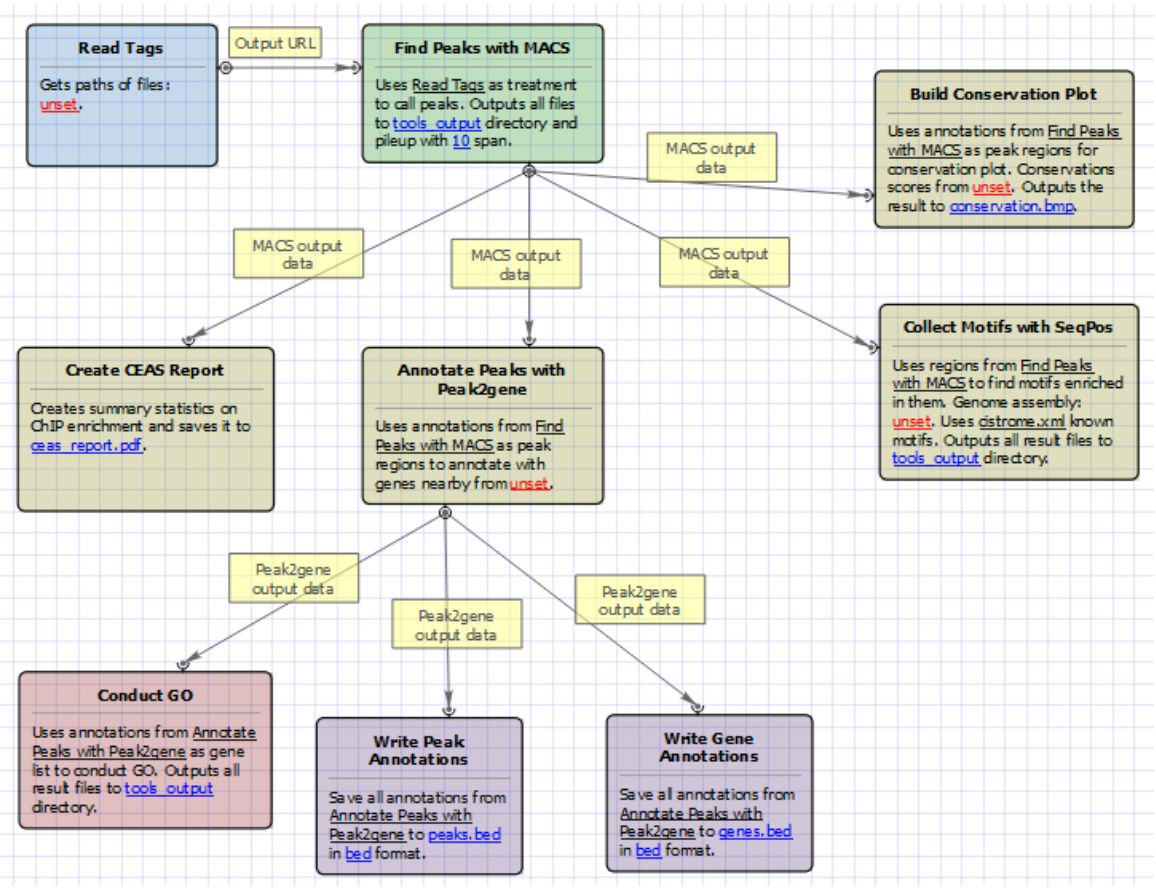
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

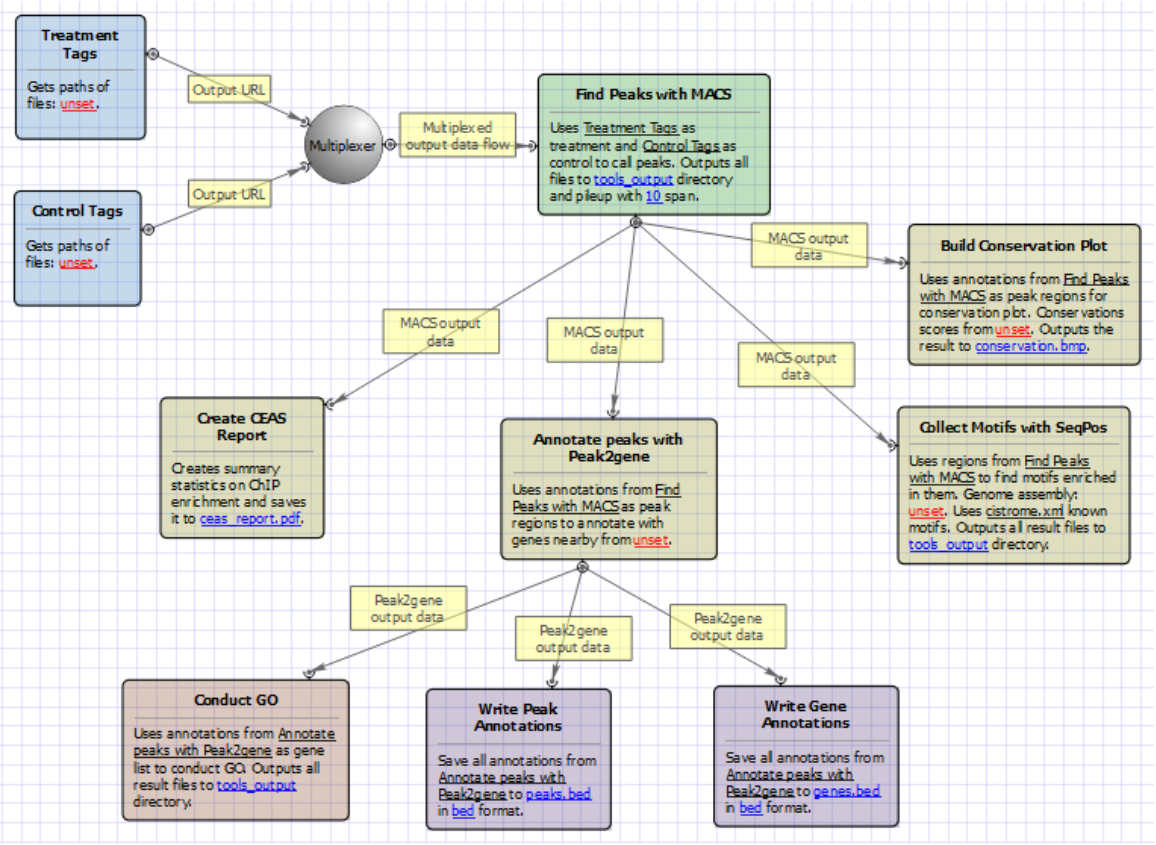
The workflow sample "ChIP-seq Analysis with Cistrome Tools" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

For treatment tags only analysis type the workflow looks as follows:



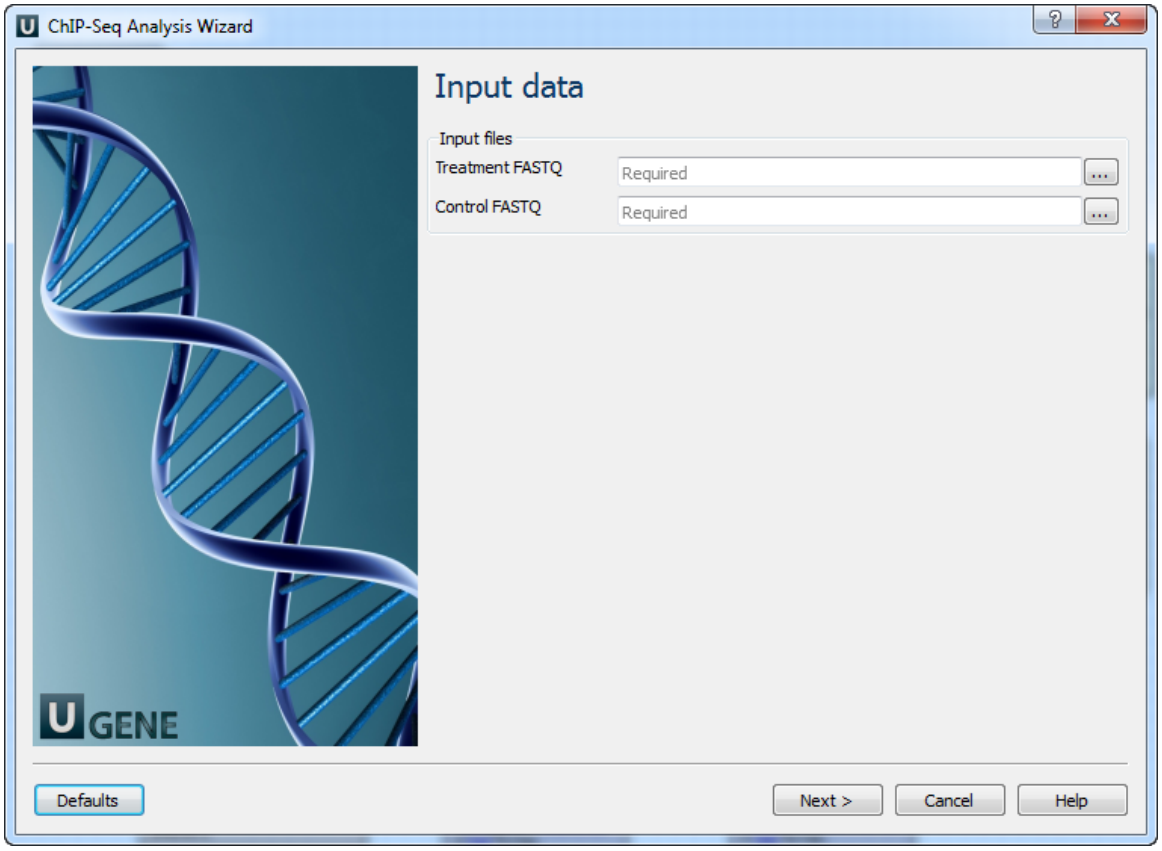
For treatment and control tags analysis type the workflow looks as follows:



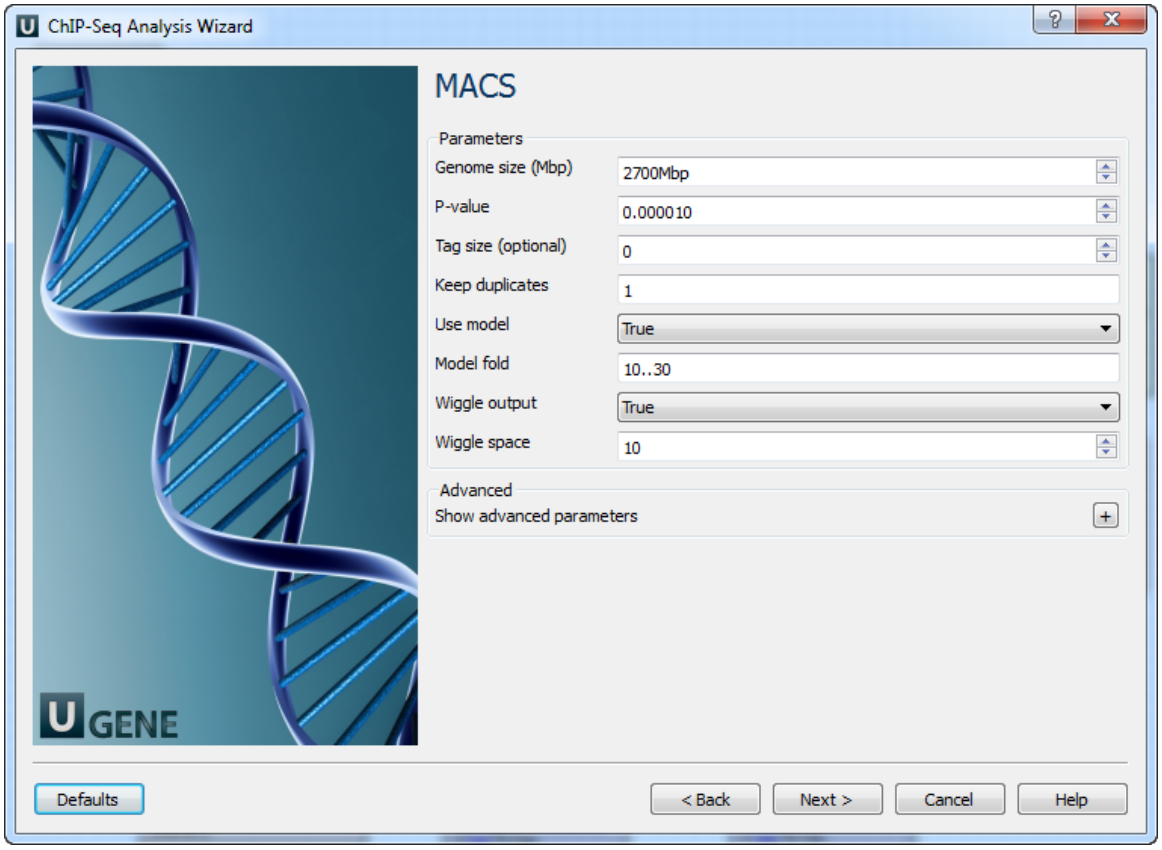
Workflow Wizard

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.

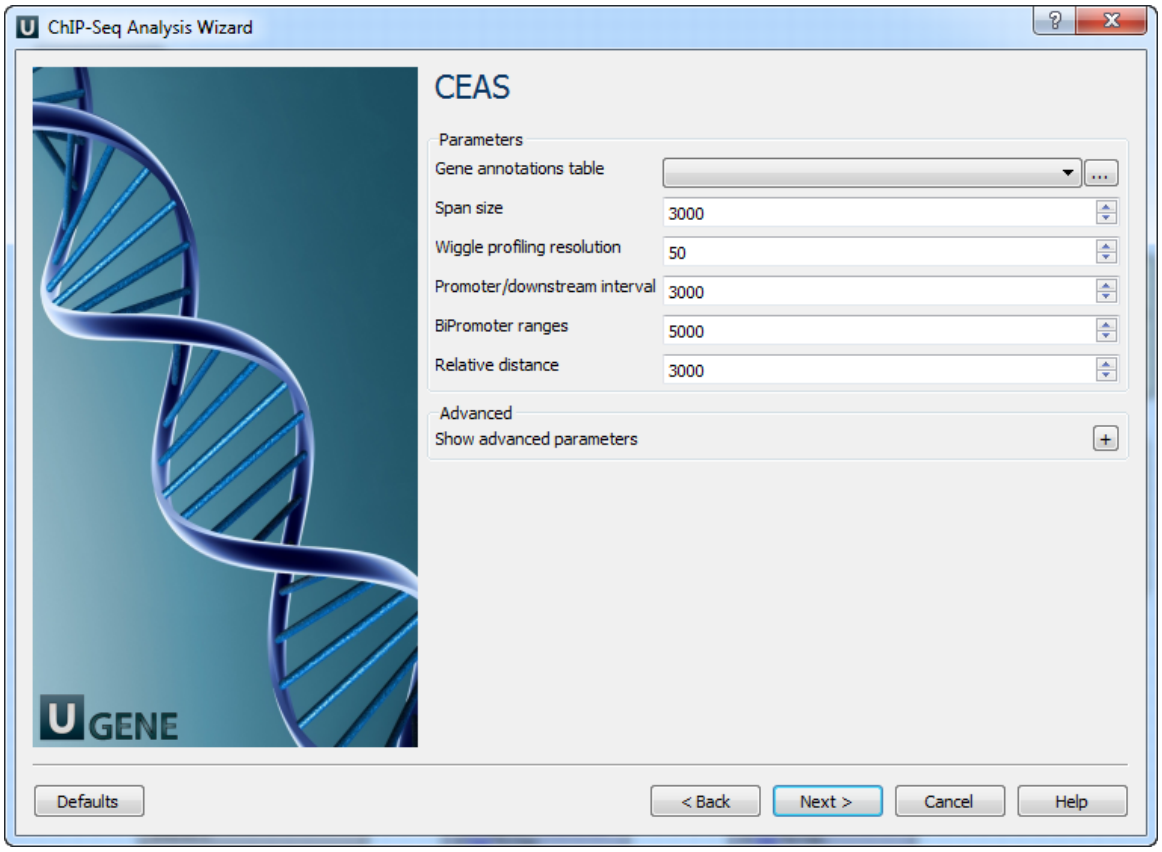


The following parameters are available:

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.

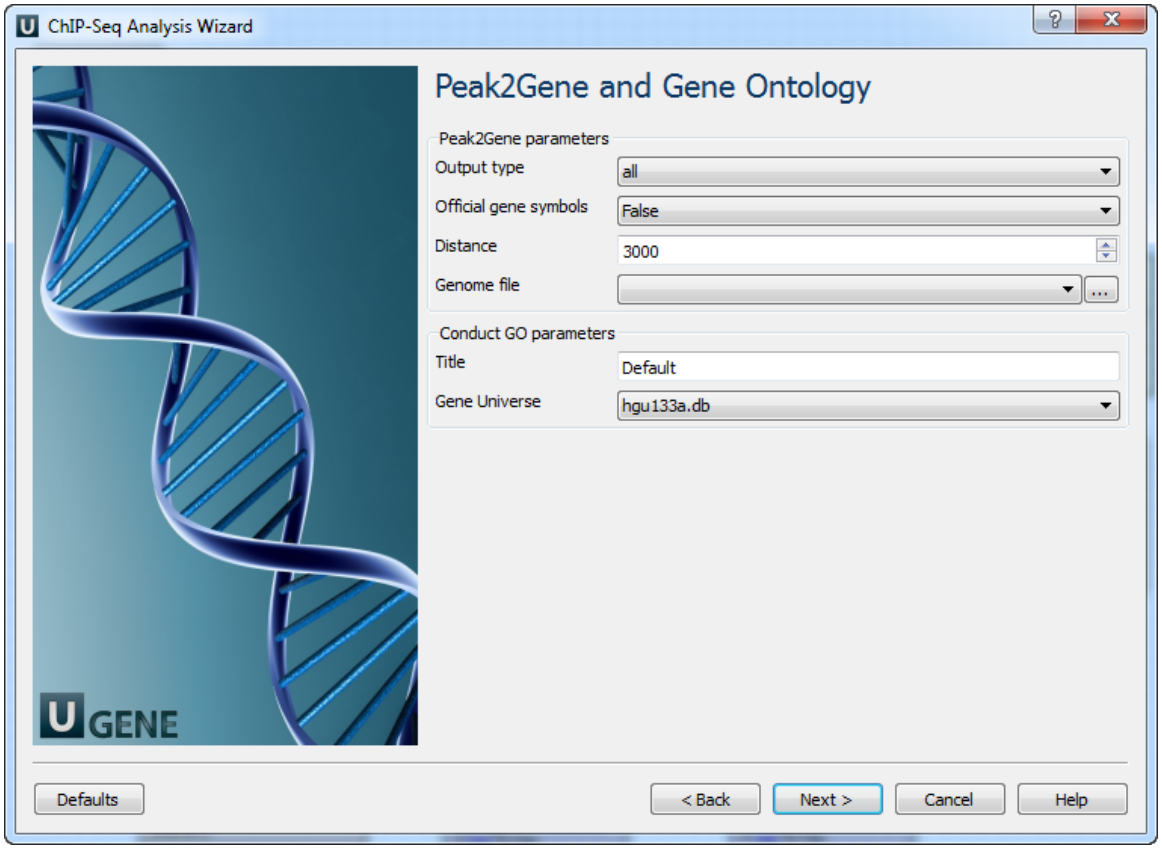


The following parameters are available:

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'.

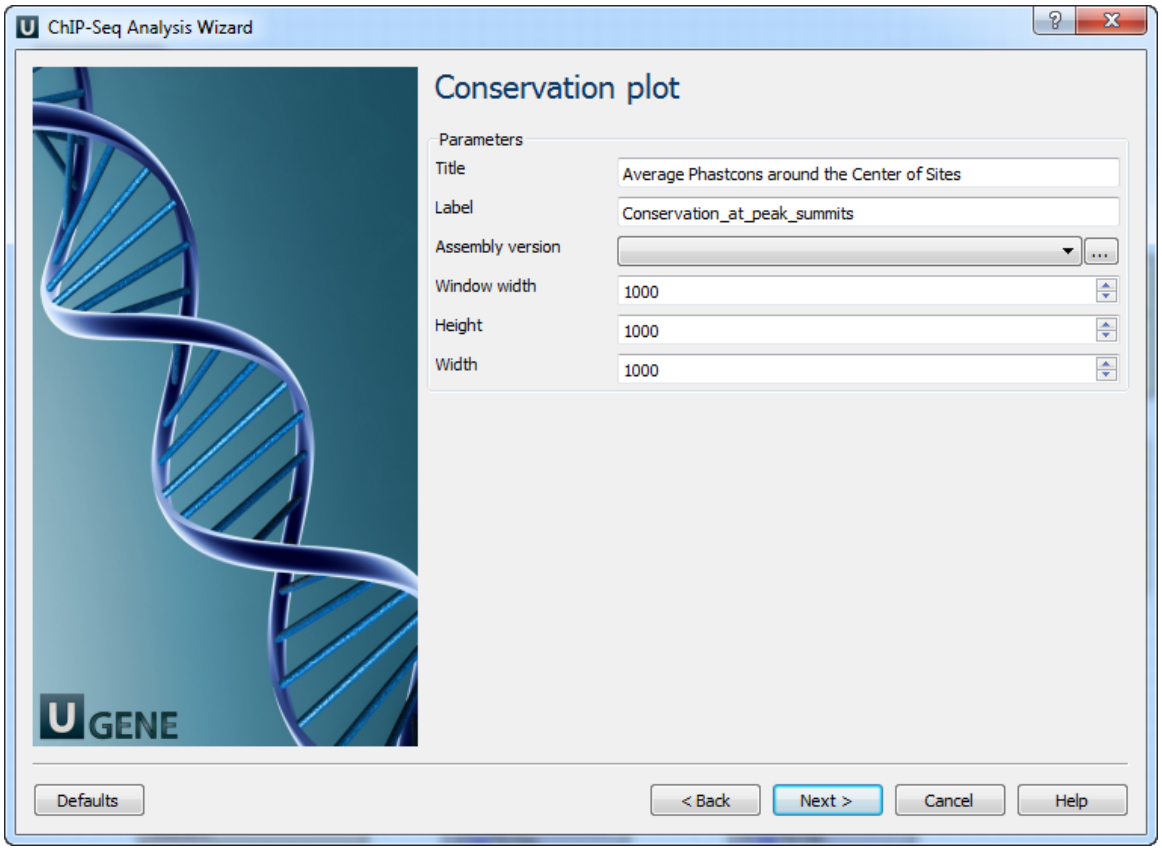
4. Peak2Gene and Gene Ontology: The next page allows to configure Peak2Gene and Gene Ontology parameters.



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.

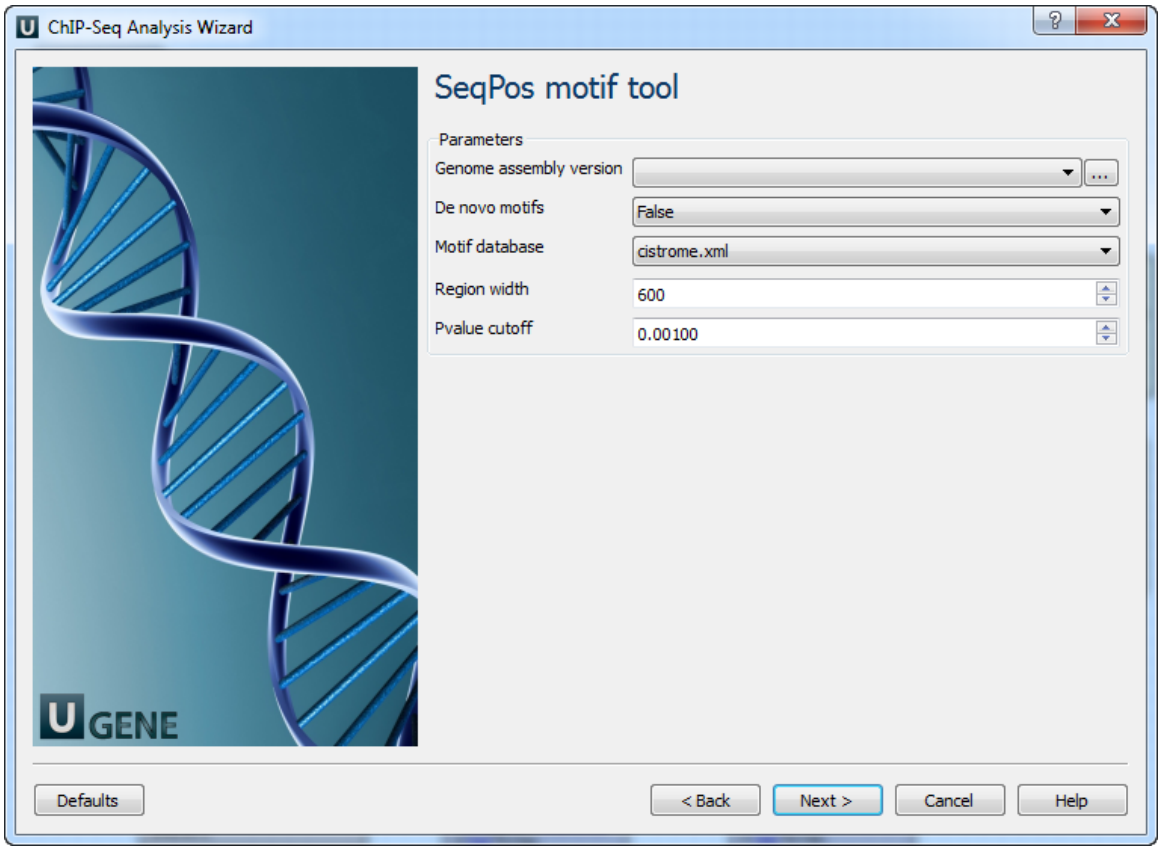
5. Conservation plot: On this page you can modify Conservation Plot parameters.



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.

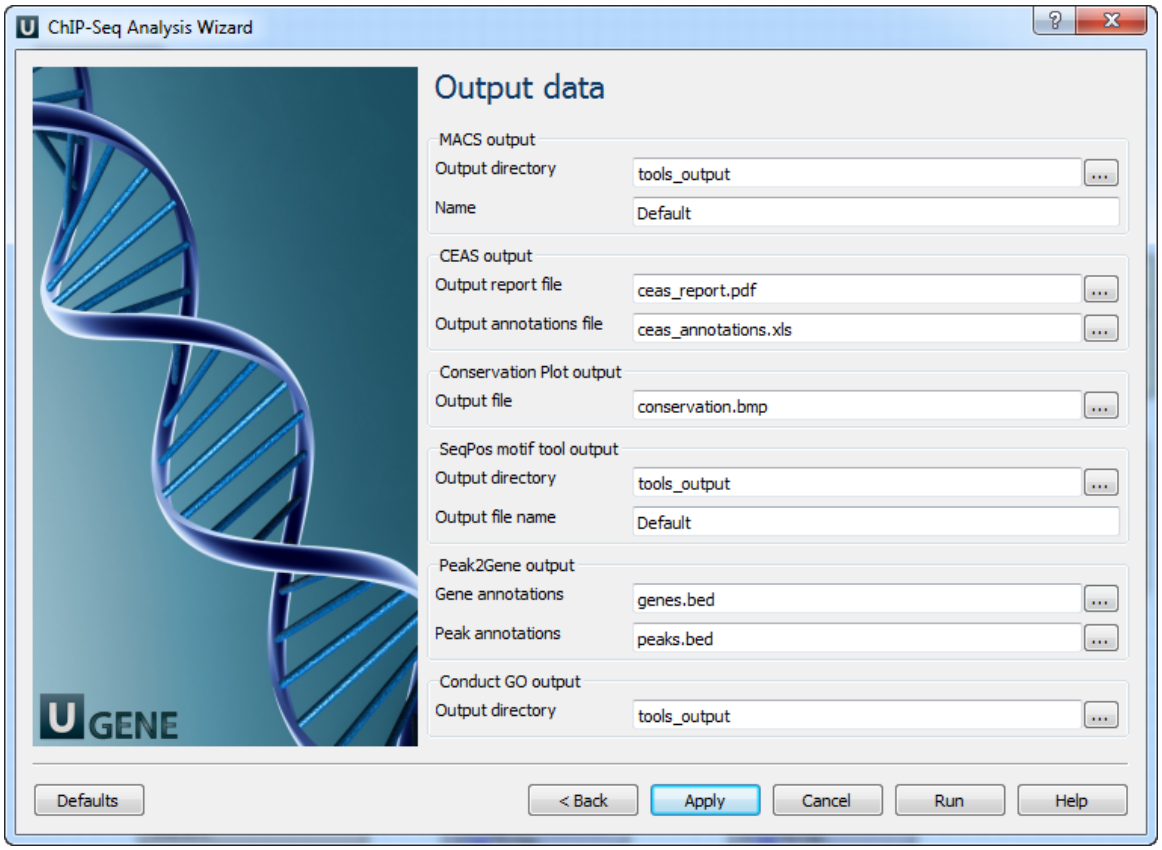
6. SeqPos motif tool: On this page you can modify SeqPos motif parameters.



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.

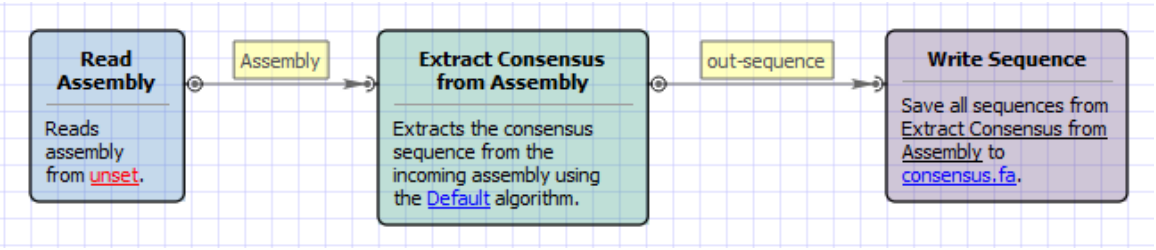
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Extract Consensus from Assembly" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

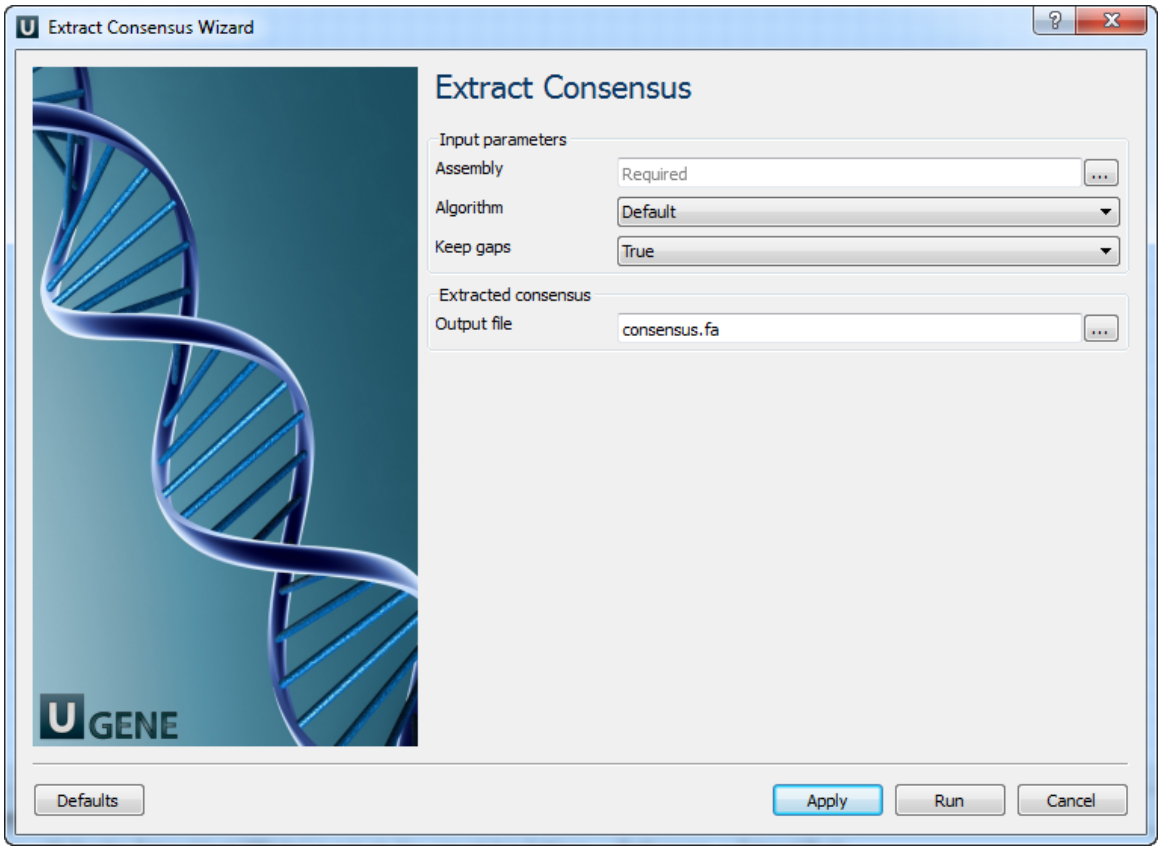
The opened workflow looks as follows:



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.

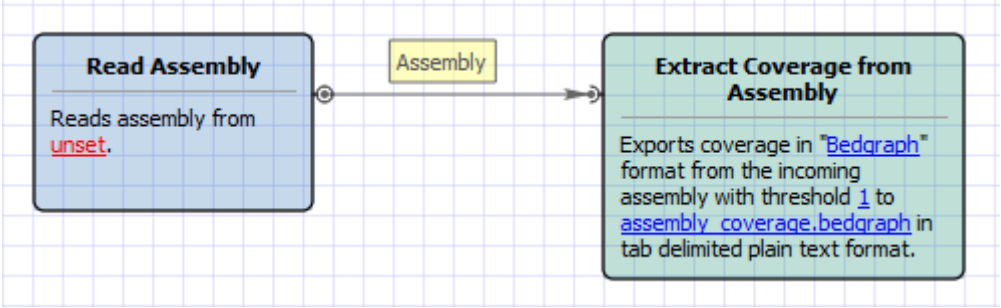
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Extract Coverage from Assembly" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

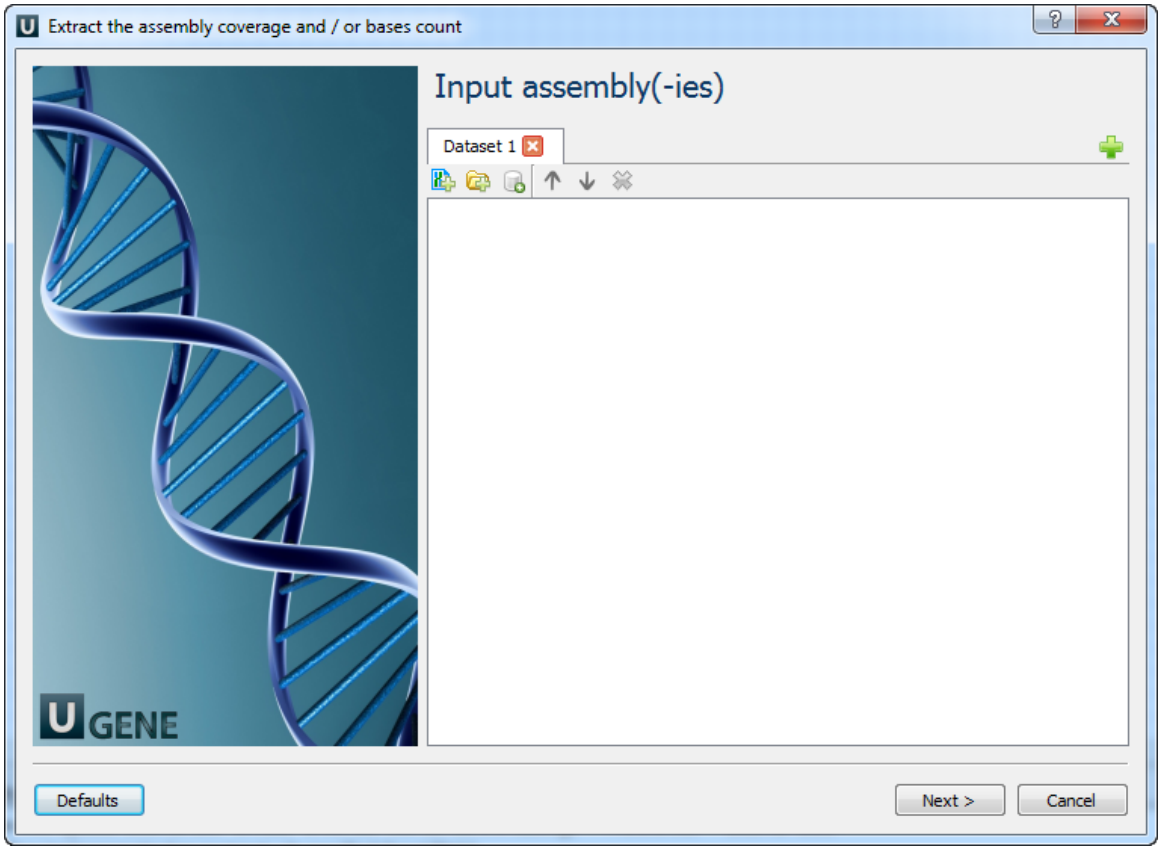
The opened workflow looks as follows:



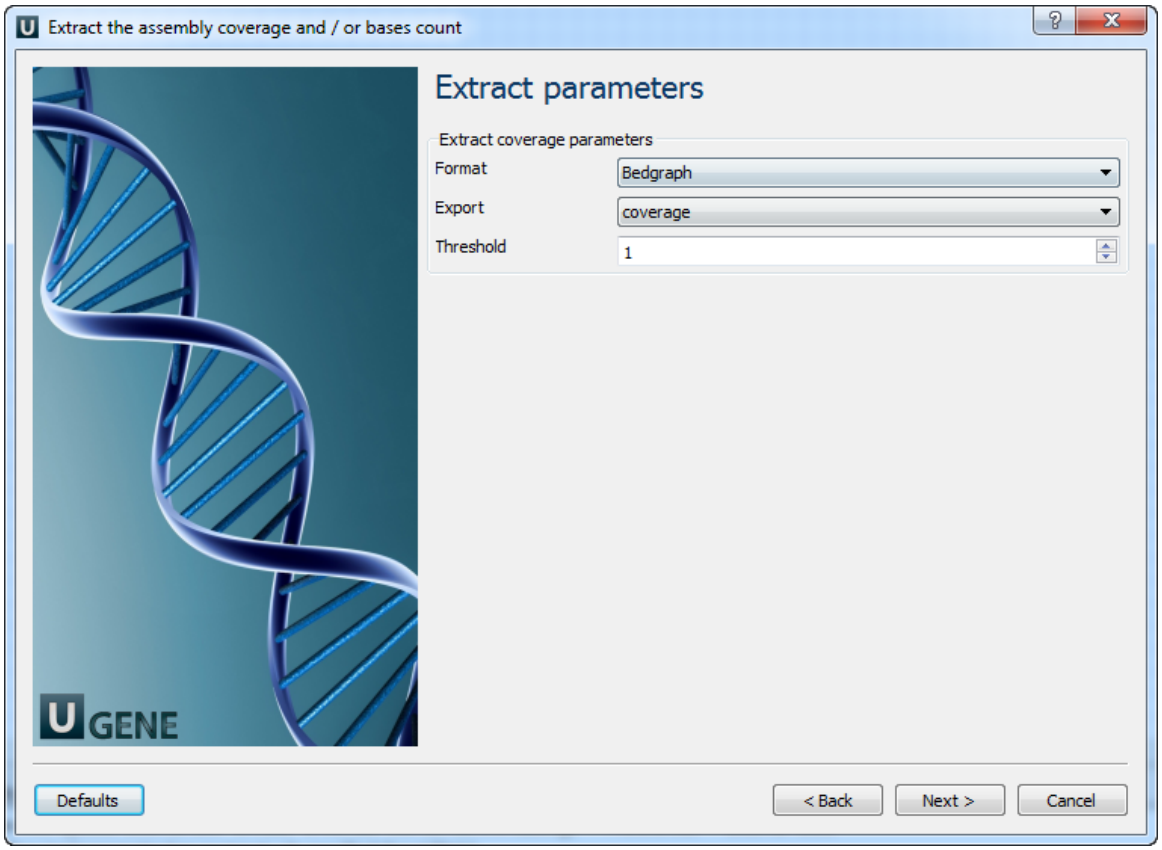
Workflow Wizard

The wizard has 3 pages.

1. Input assembly (-ies) Page: On this page you must input assembly(-ies).



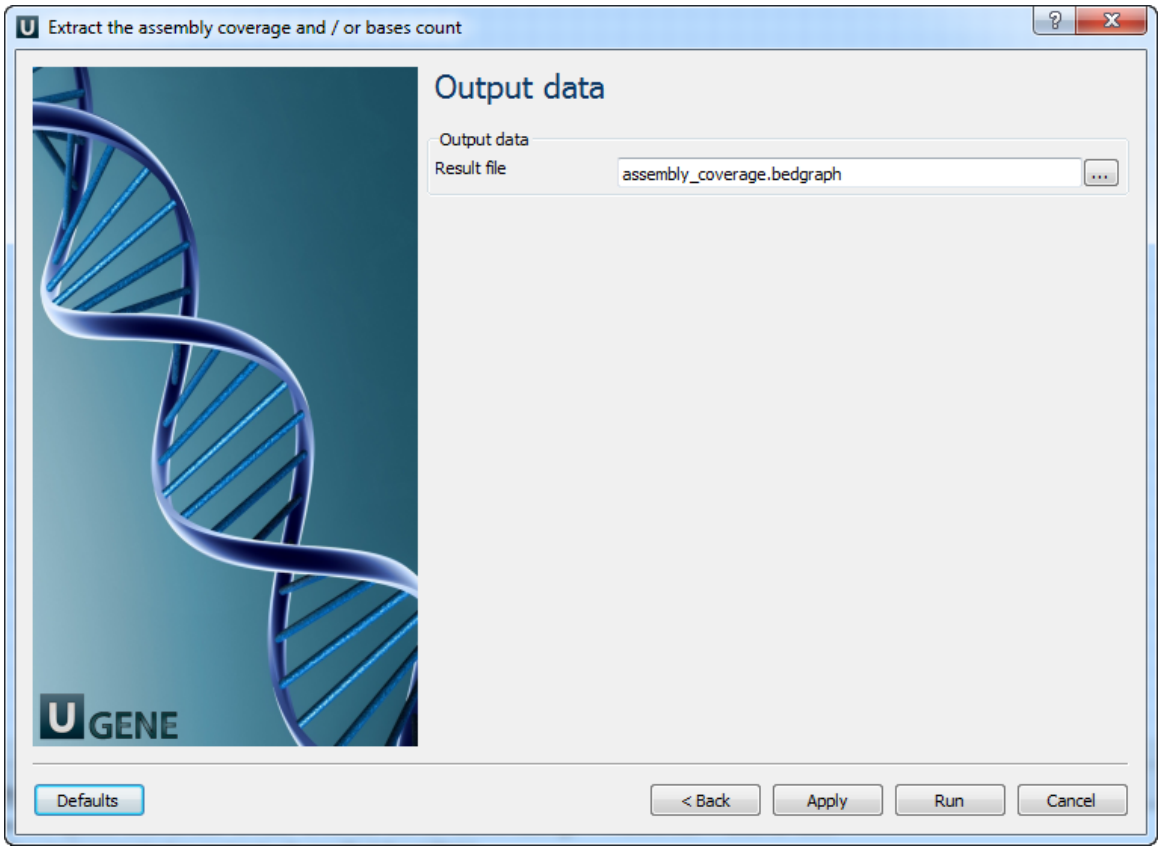
2. Extract parameters Page: Here you can optionally modify extract parameters.



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).

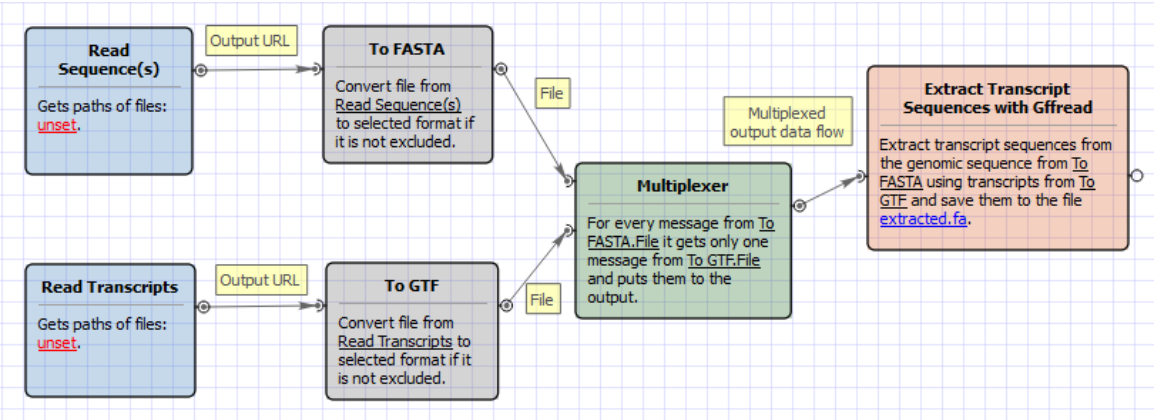
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

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.

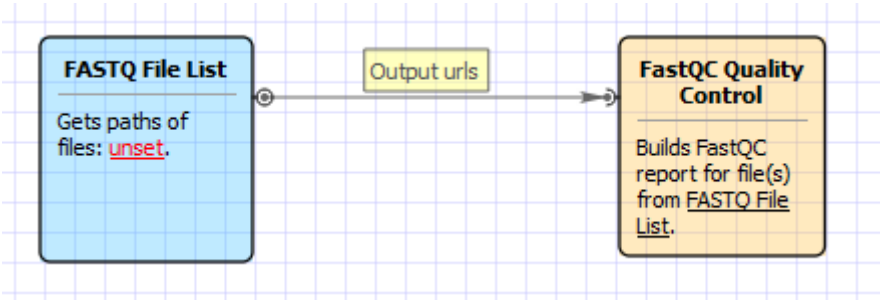
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Quality Control by FastQC" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

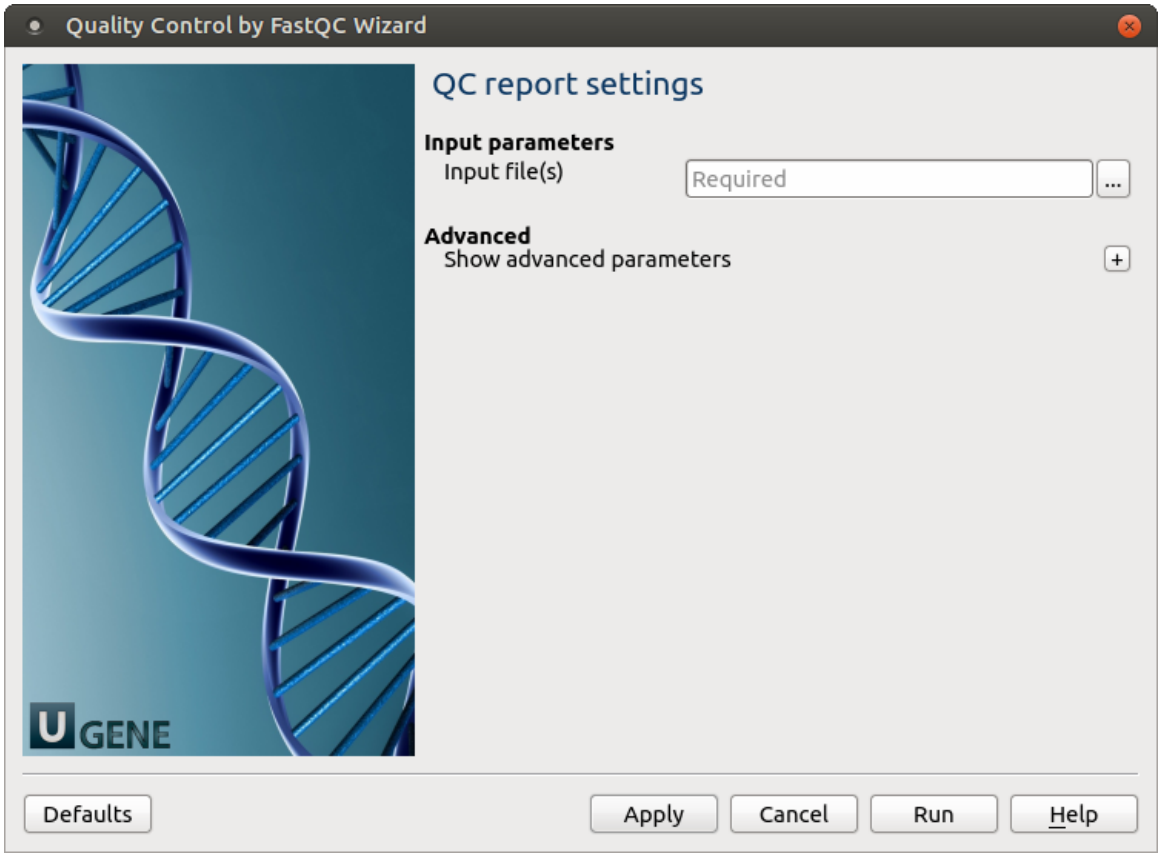
The workflow is the following:



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 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.
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.

Raw ChIP-Seq Data Processing

Download and install the UGENE NGS package to use this pipeline.

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.

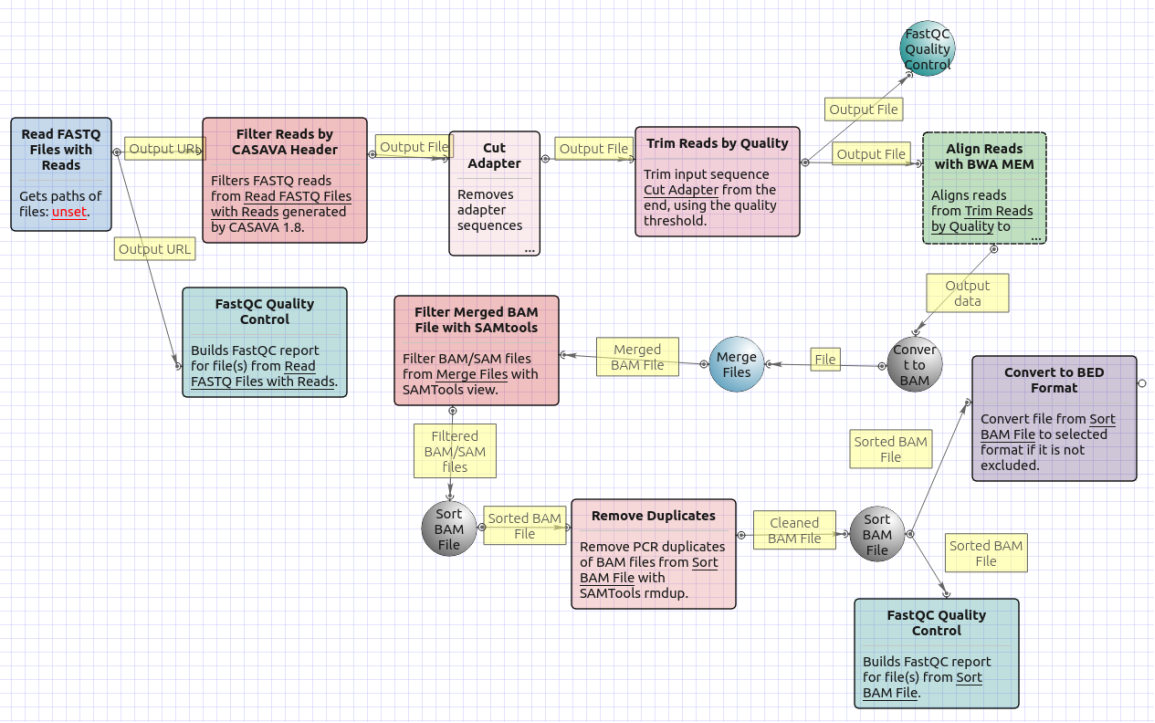
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

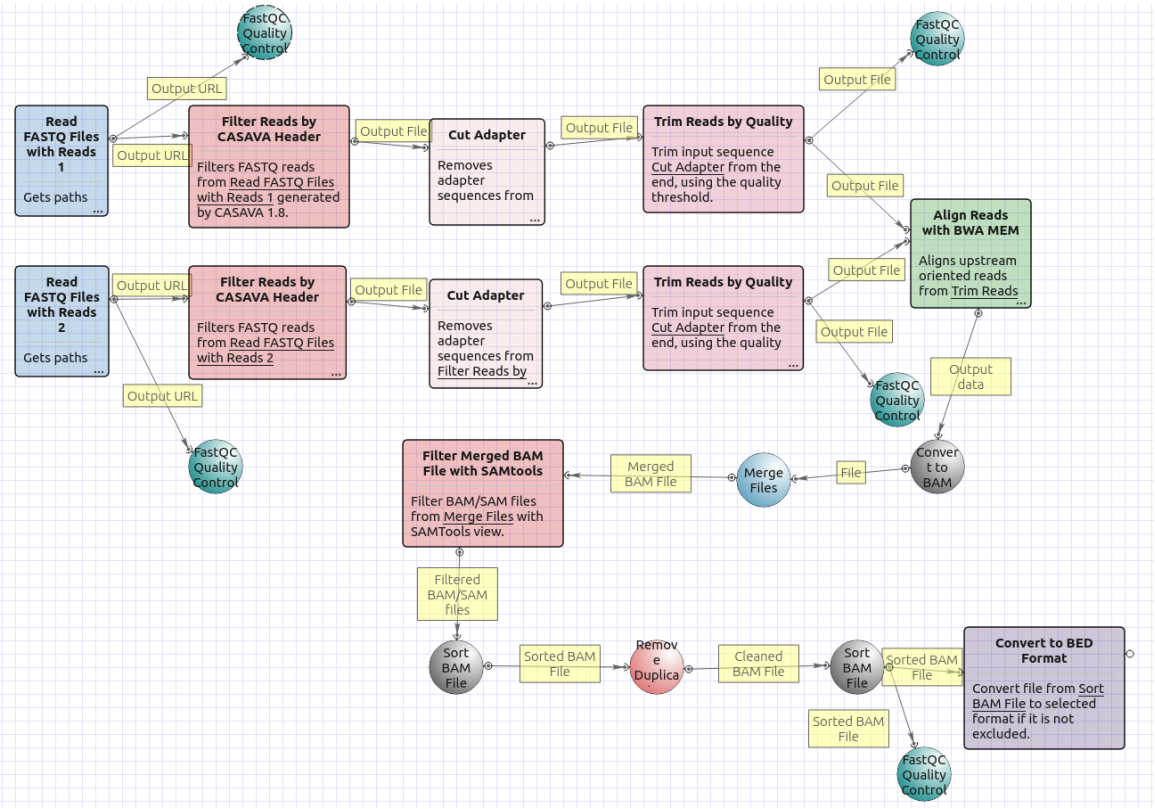
The workflow sample "Raw ChIP-Seq processing" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

There are two versions of the workflow available. The workflow for single-end reads looks as follows:



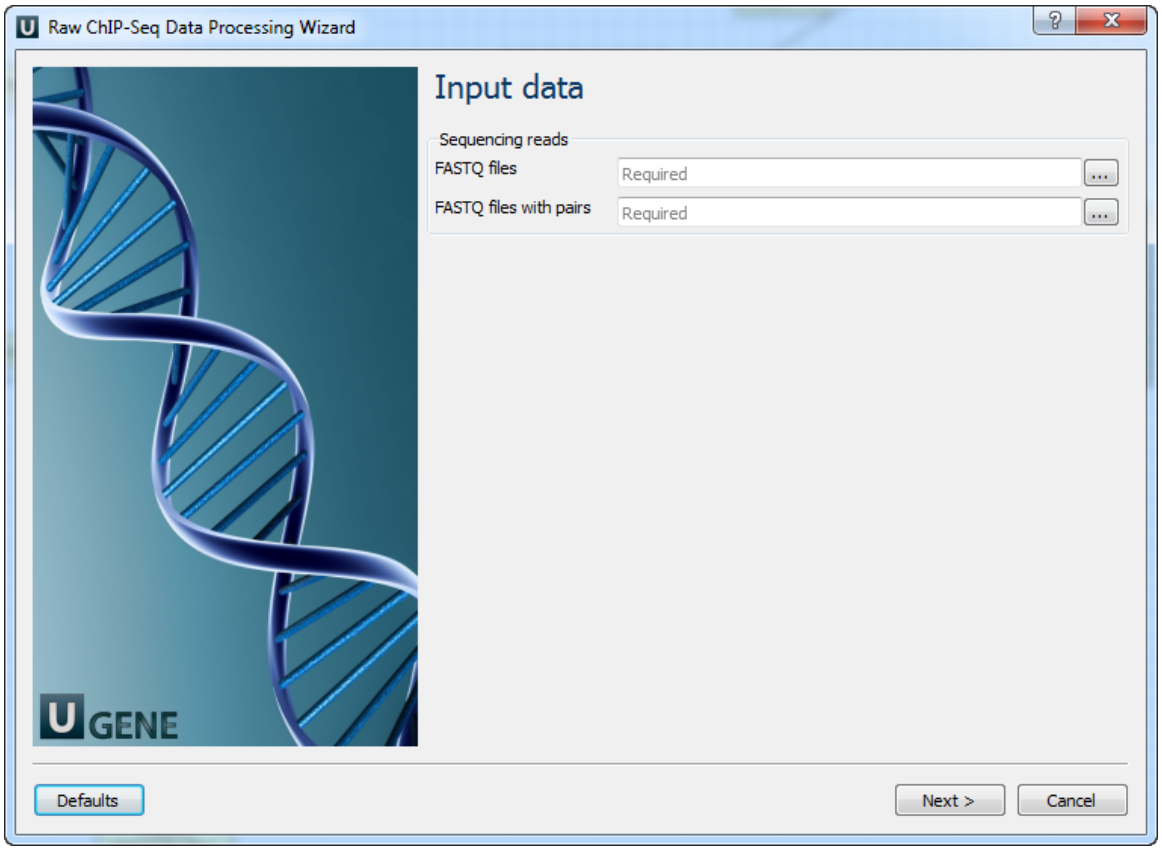
The workflow for paired-end short appearance is the following:



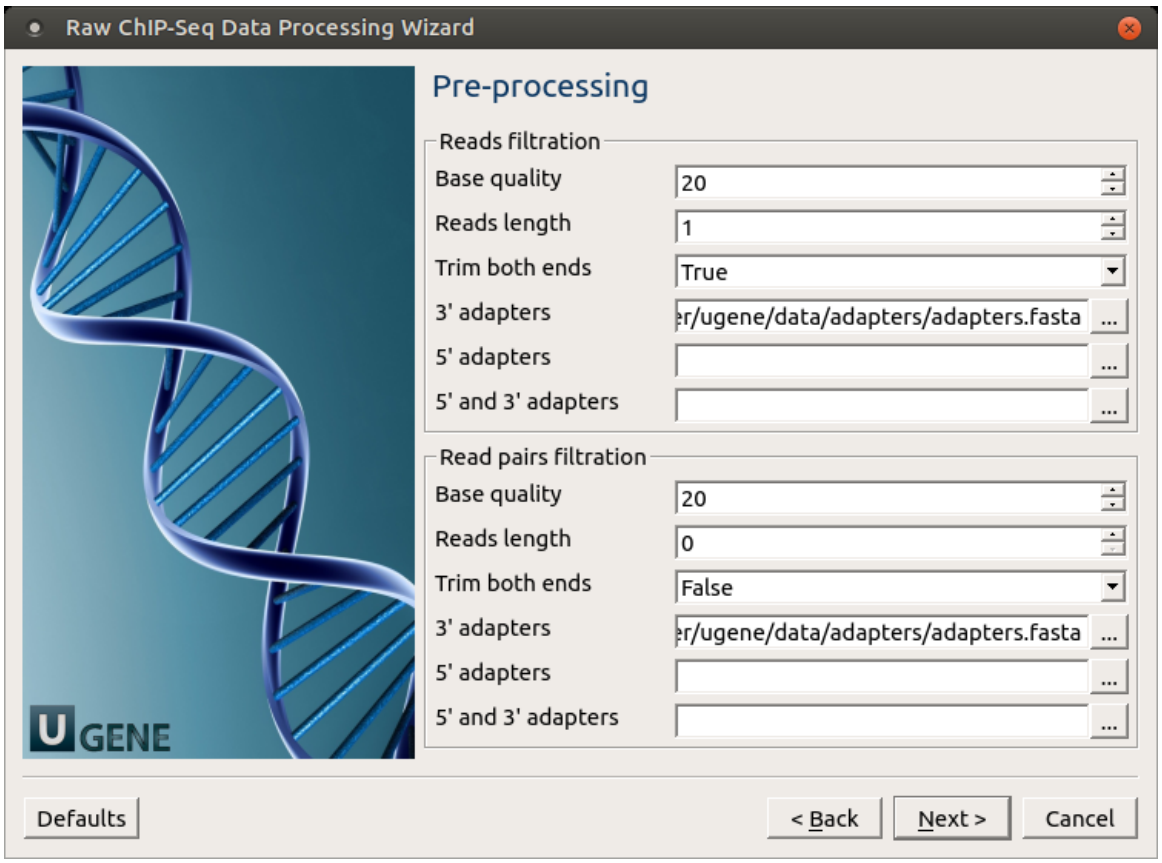
Workflow Wizard

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.

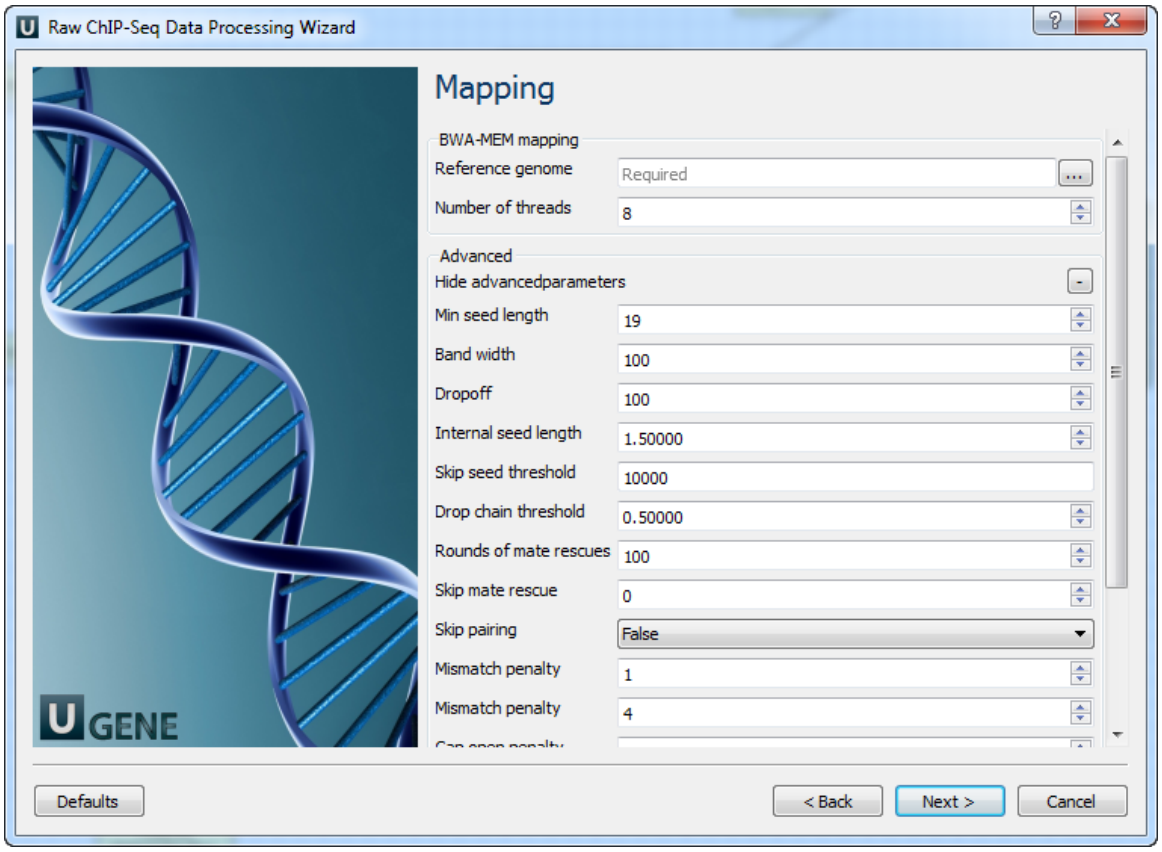


The following parameters are available for reads and reads pairs filtration:

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 Tr ue 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 t he character ^, the adapter is 'anchored'. An anchored adapter must appear in its entirety at the 5' end o f 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.

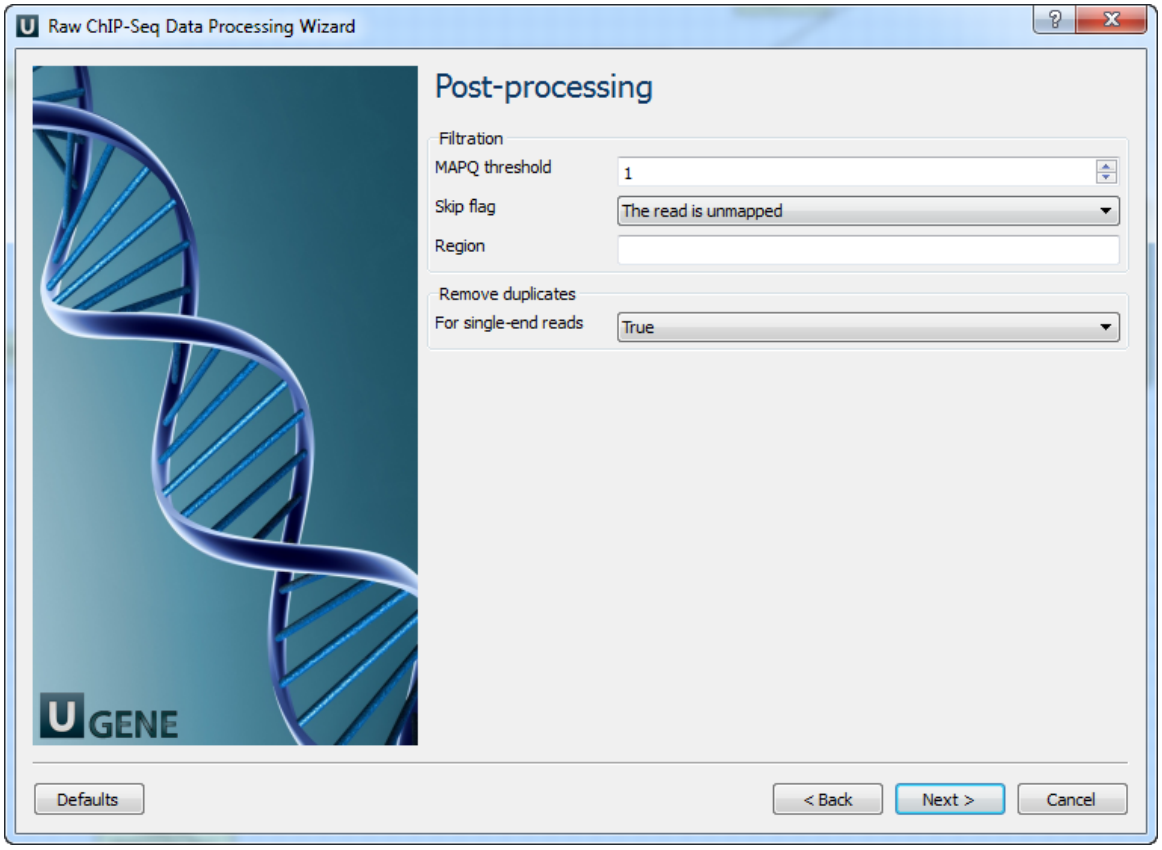


The following parameters are available:

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.

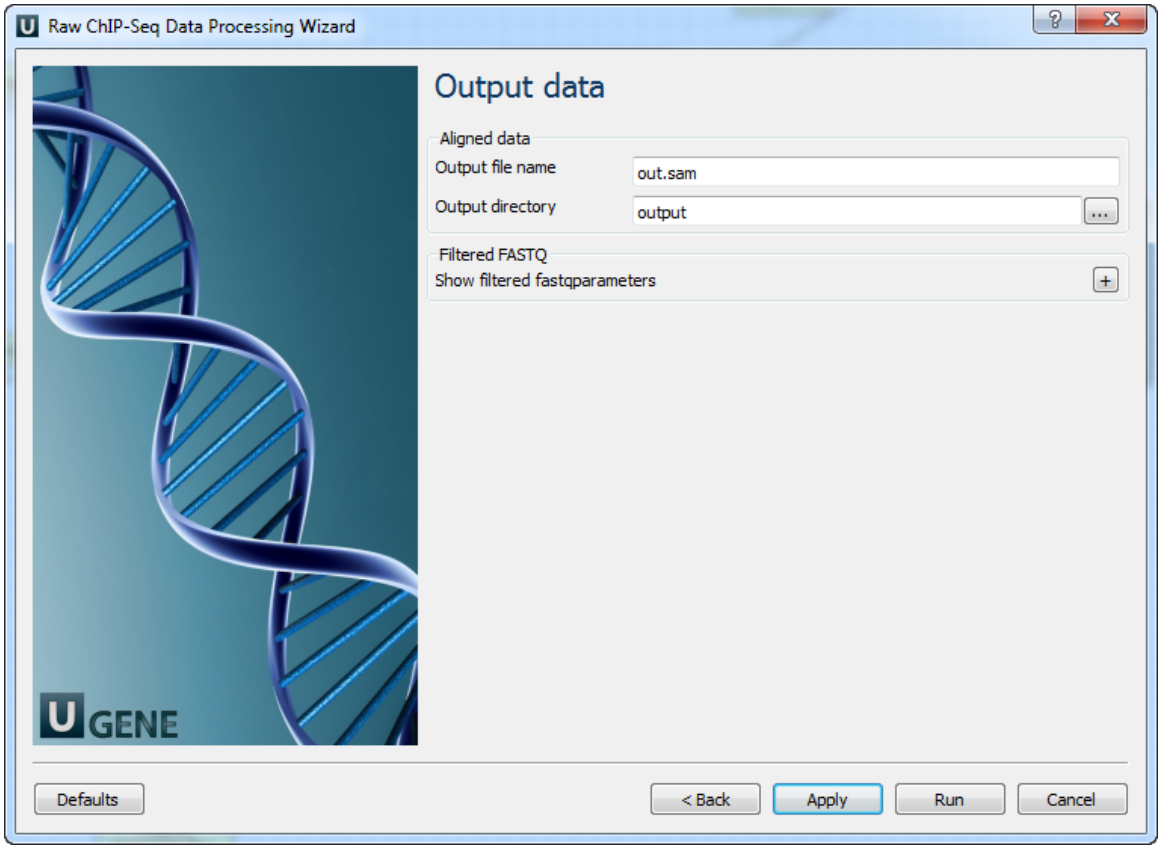


The following parameters are available:

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

Download and install the UGENE FULL or NGS package to use this pipeline.

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.

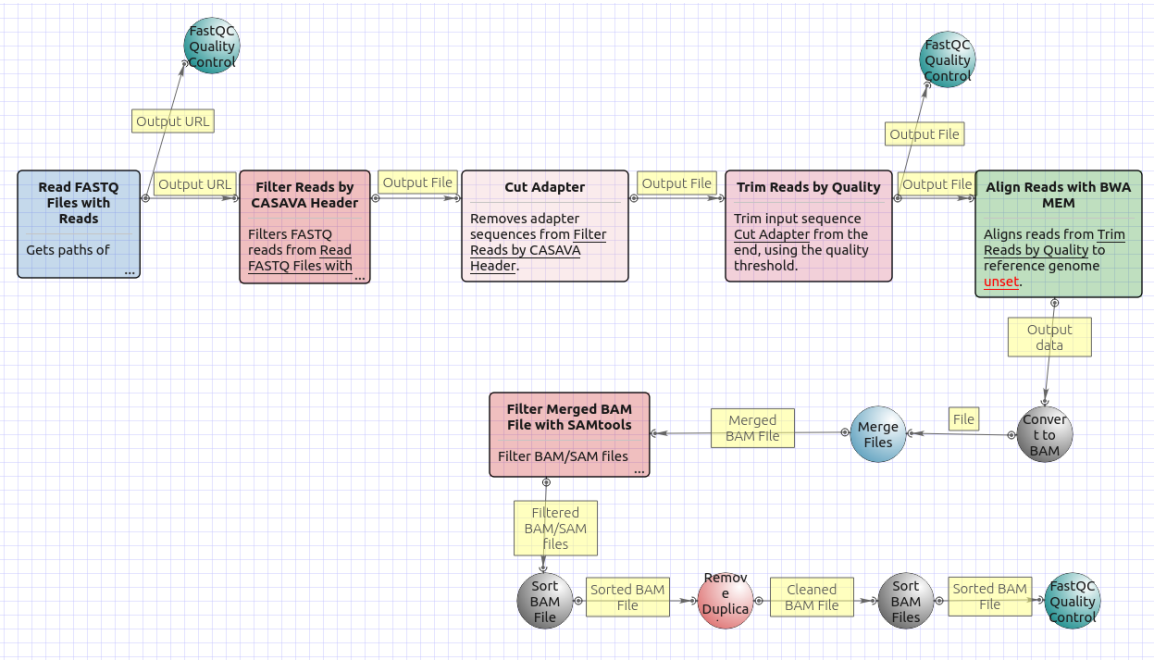
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

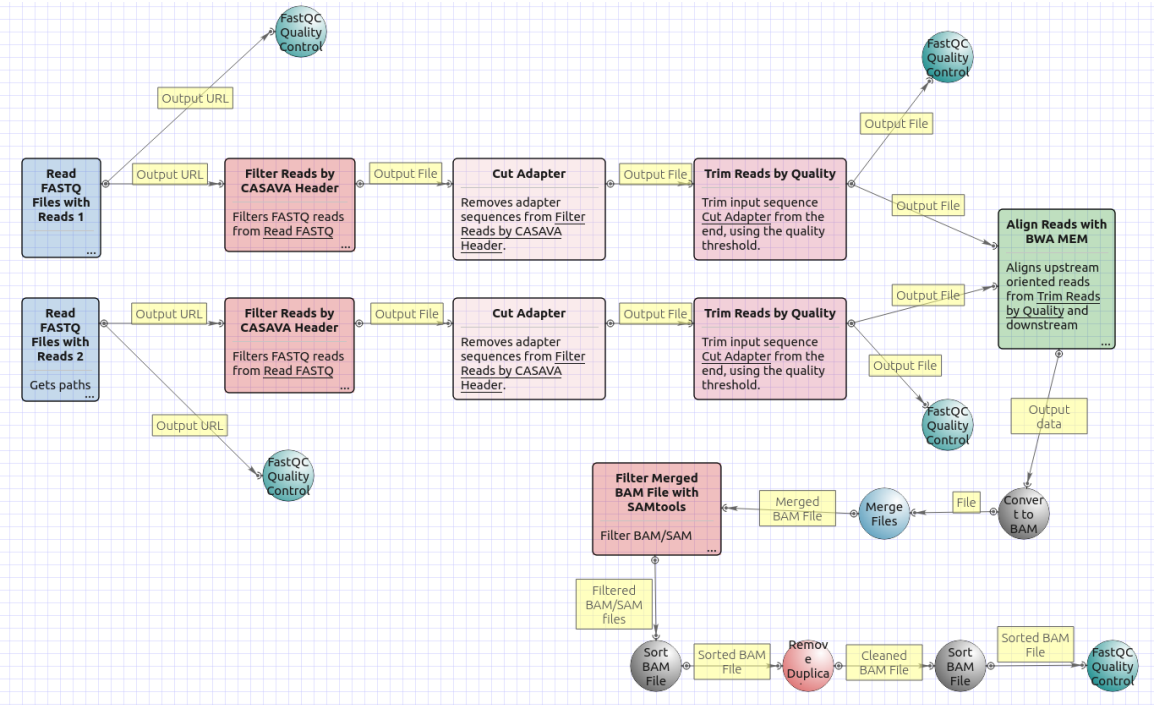
The workflow sample "Raw DNA-Seq processing" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

There are two versions of the workflow available. The workflow for single-end reads looks as follows:



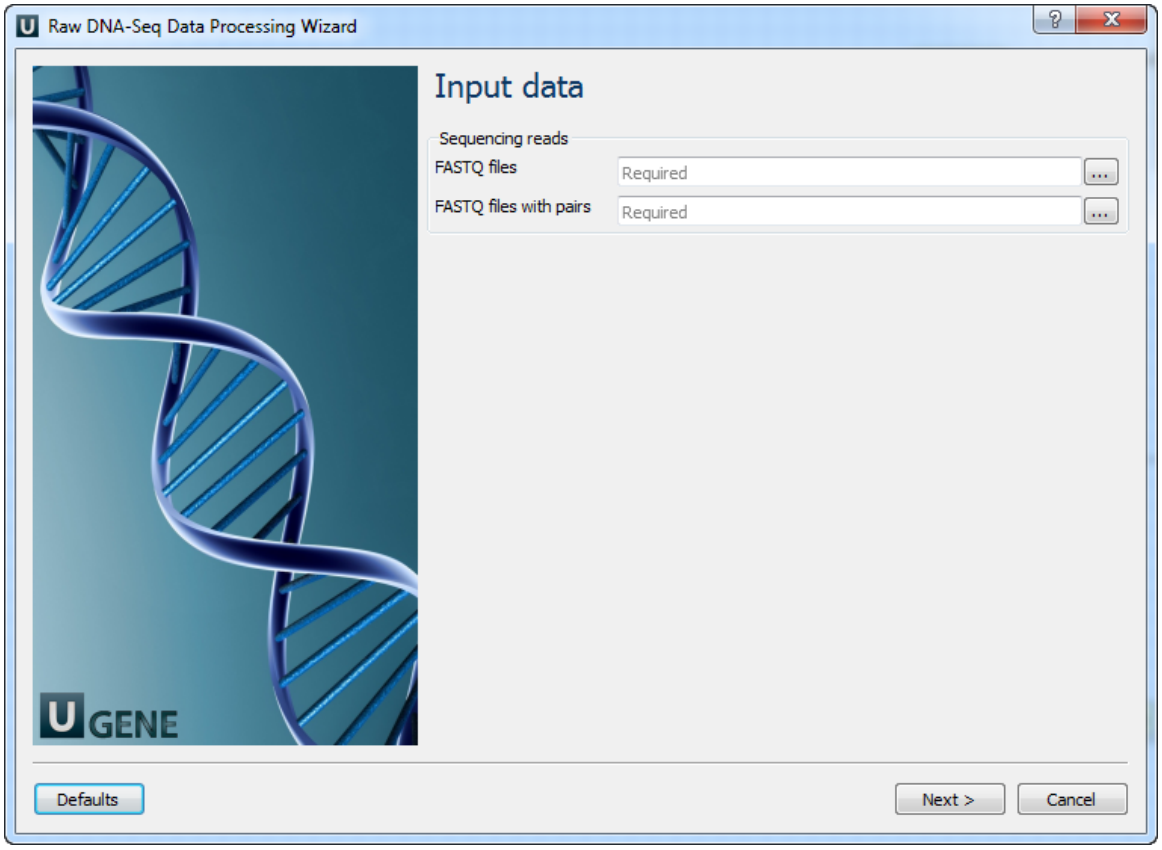
The workflow for paired-end short appearance is the following:



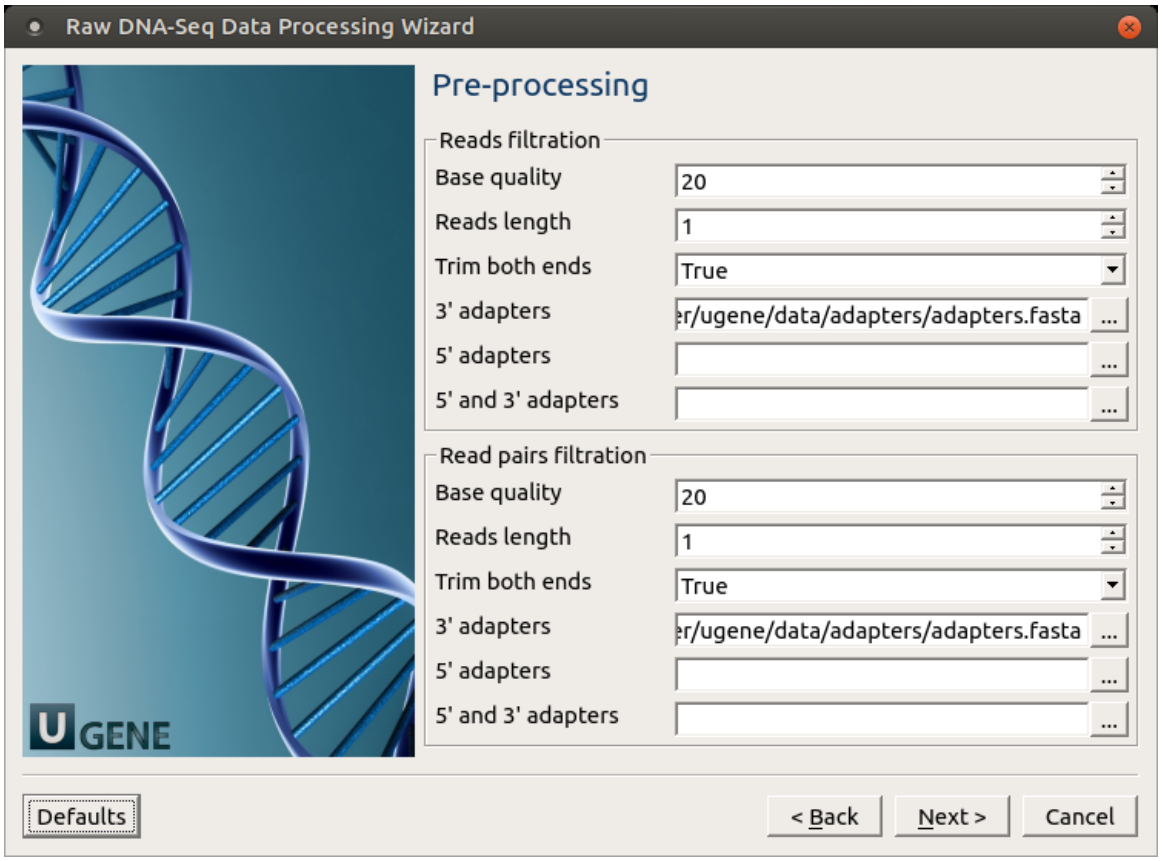
Workflow Wizard

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).



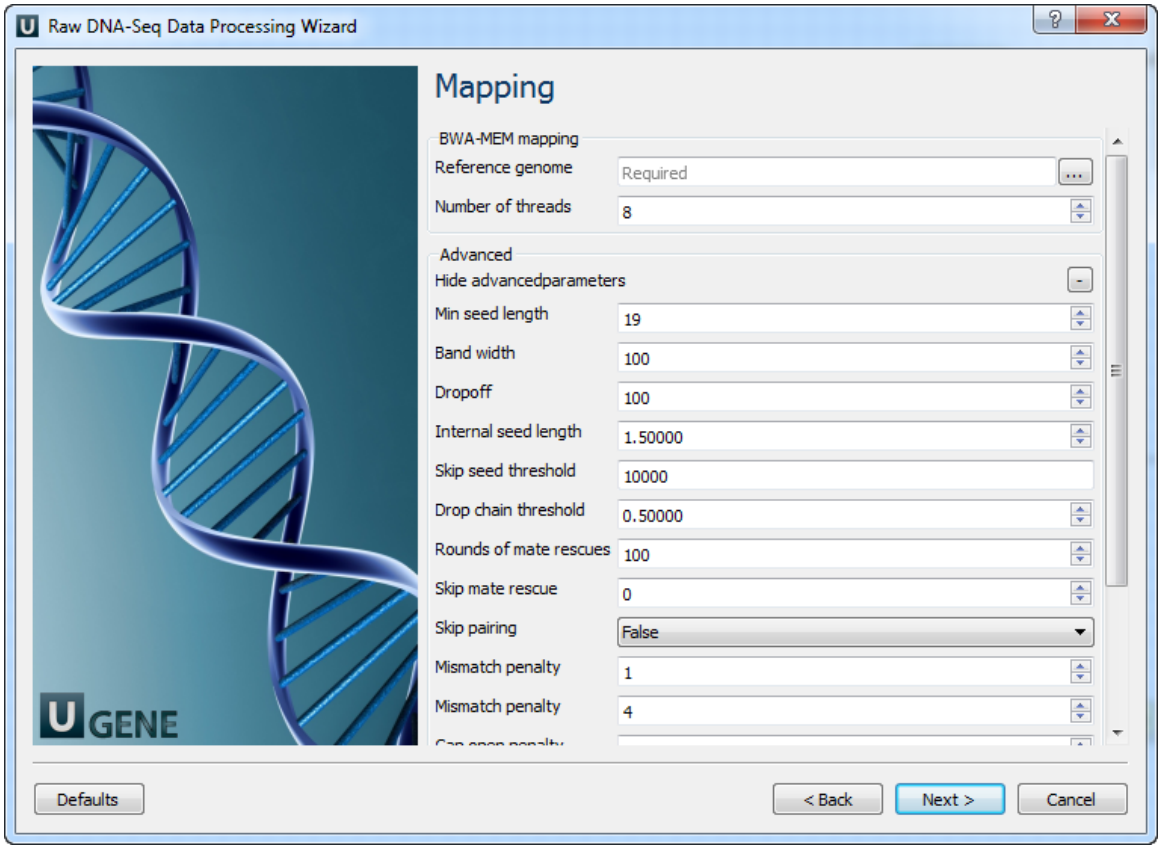
2. Pre-processing: On this page you can modify filtration parameters.



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 t he character ^, the adapter is 'anchored'. An anchored adapter must appear in its entirety at the 5' end o f 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.

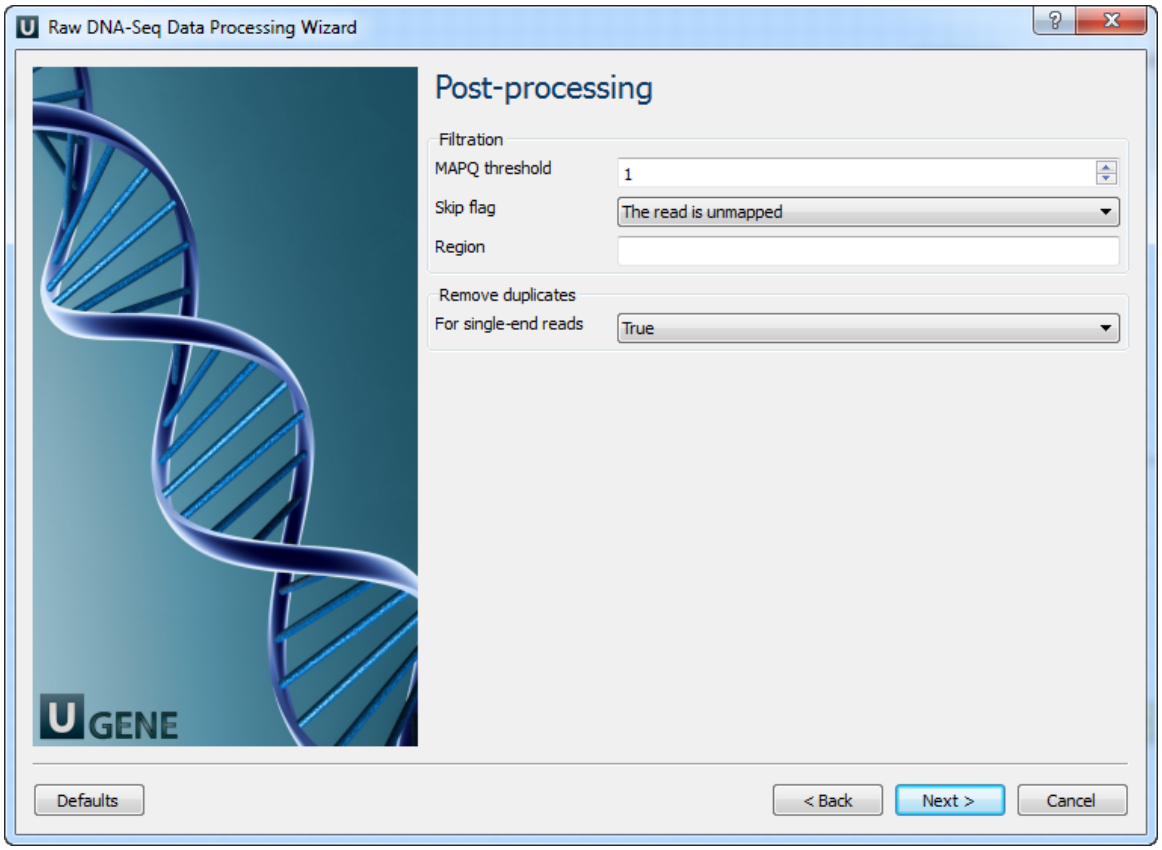


The following parameters are available:

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.

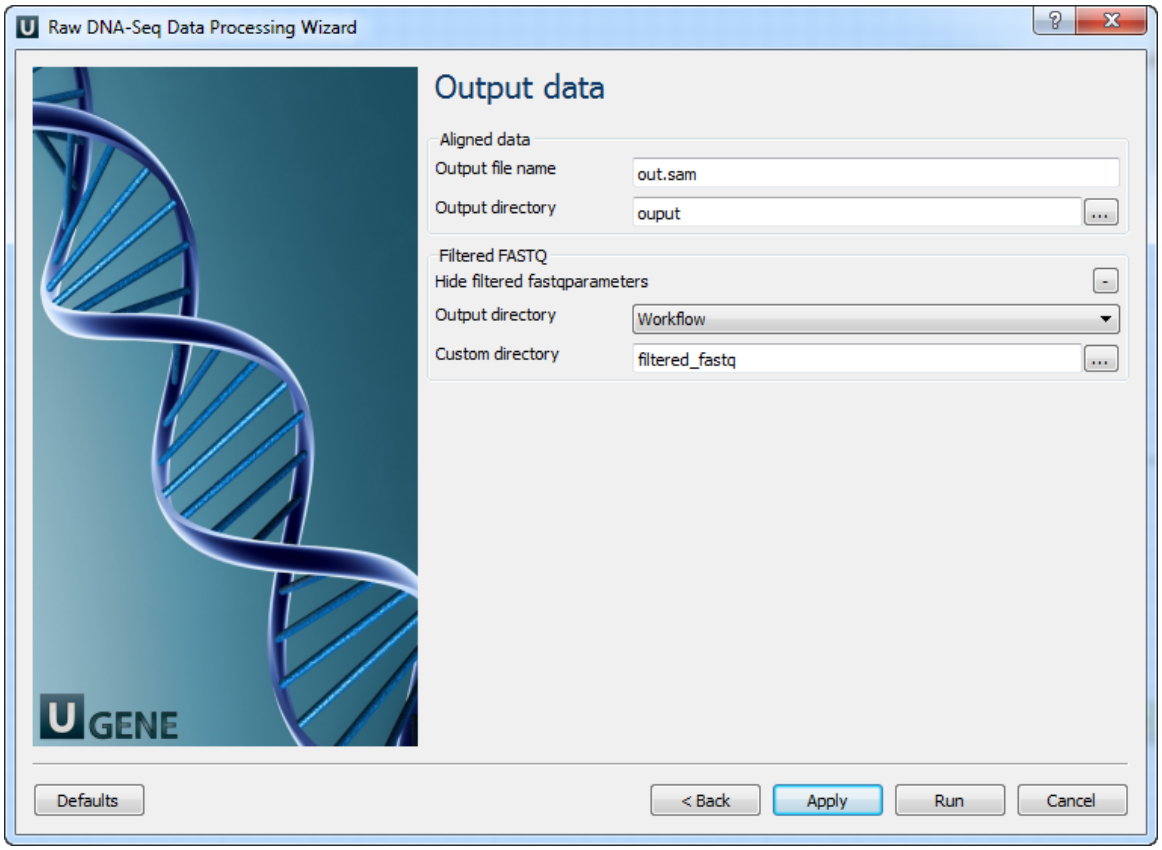


The following parameters are available:

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

Download and install the UGENE FULL or NGS package to use this pipeline.

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?

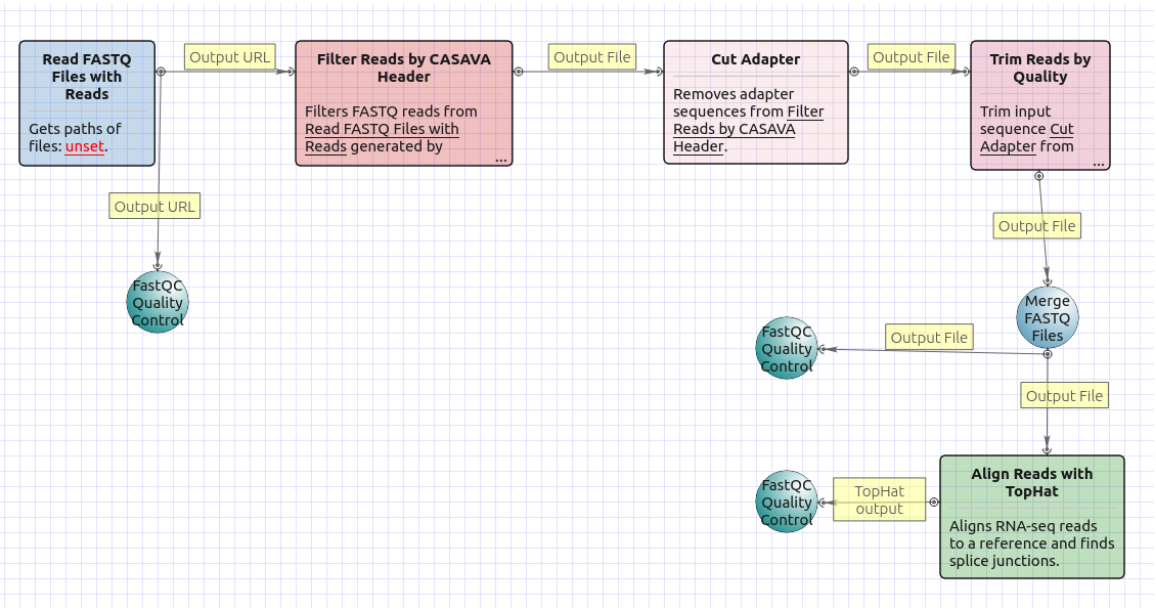
The Tuxedo workflow can be used to analyze the filtered RNA-seq data. In this case the mapping step of this workflow can be skipped, as it also present in the Tuxedo pipeline.

Workflow Sample Location

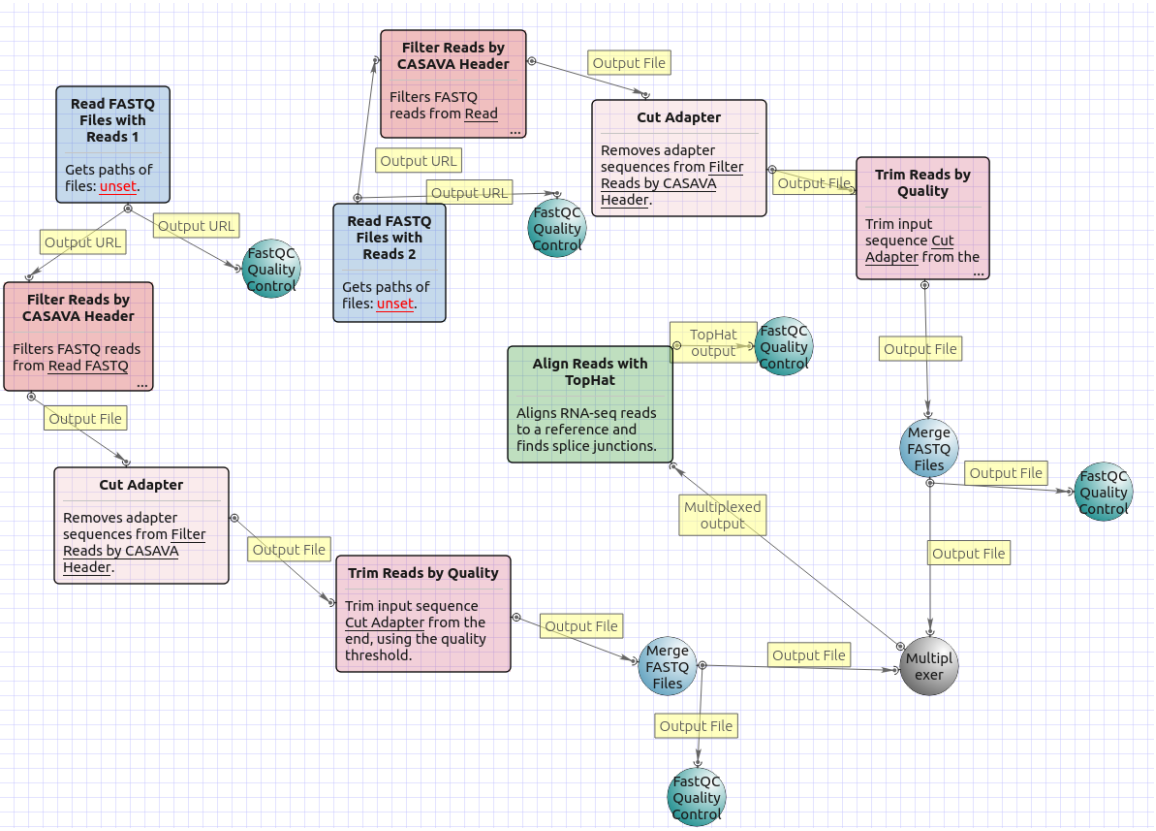
The workflow sample "Raw DNA-Seq processing" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

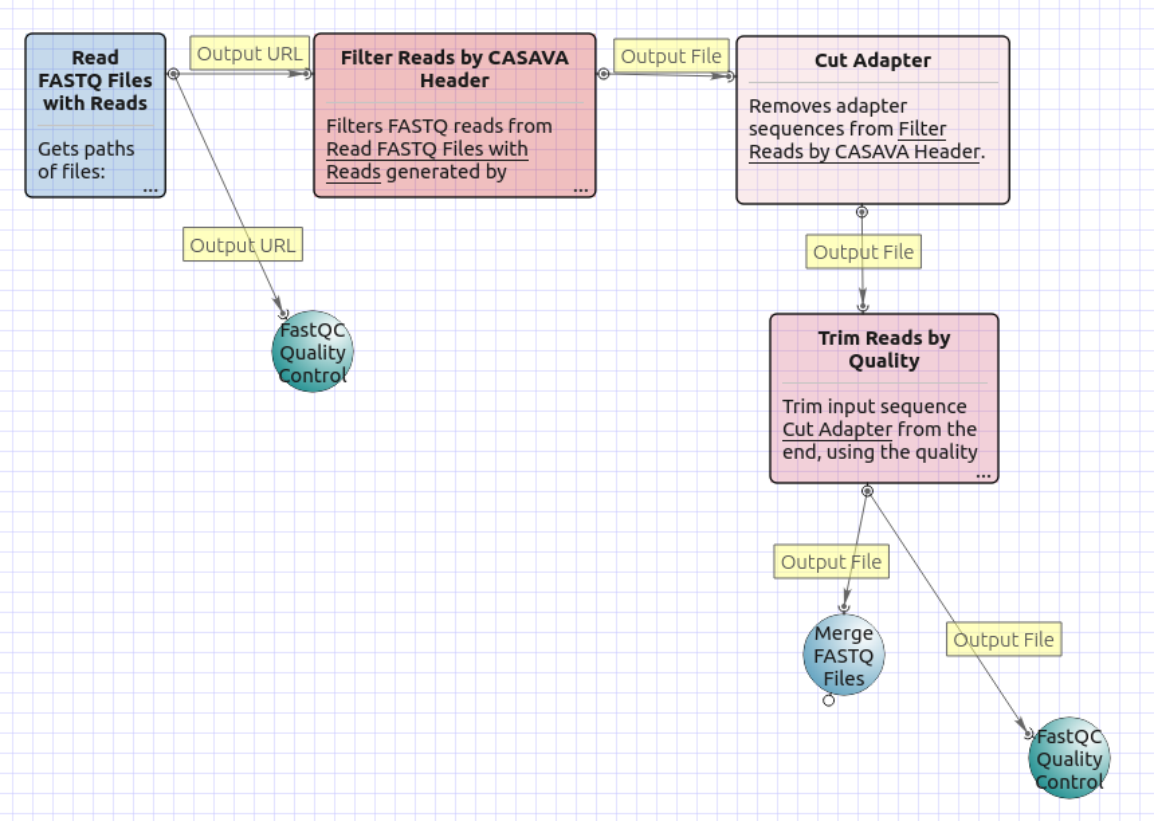
There are four versions of the workflow available. The workflow with mapping for single-end reads looks as follows:



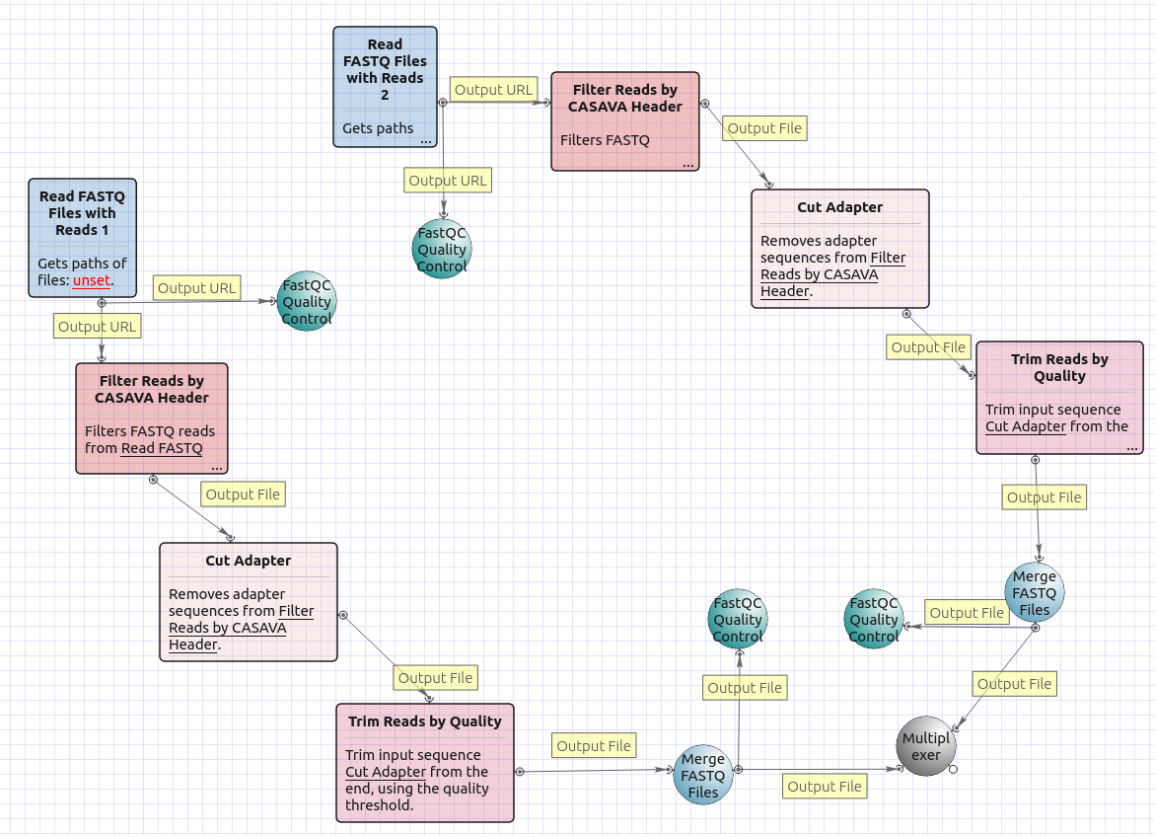
The workflow with mapping for paired-end short appearance is the following:



The workflow without mapping for single-end short appearance is the following:



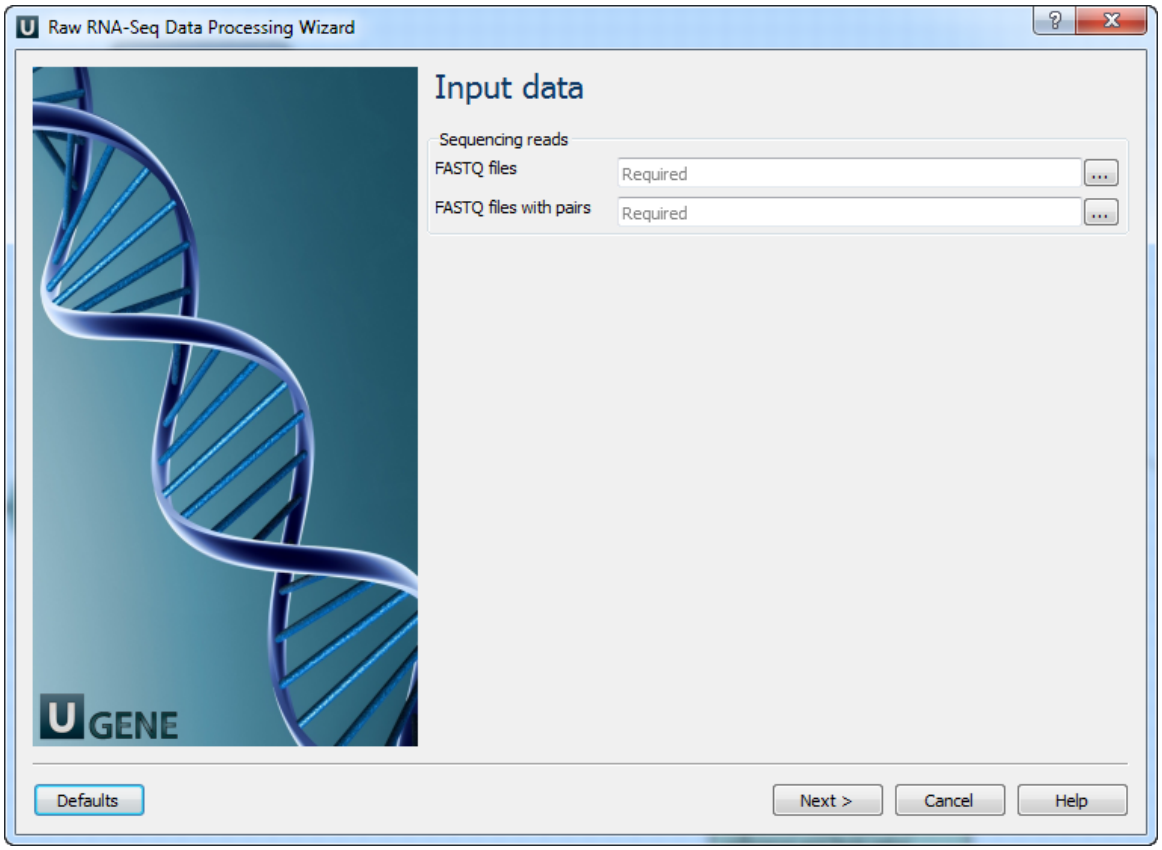
The workflow without mapping for paired-end short appearance is the following:



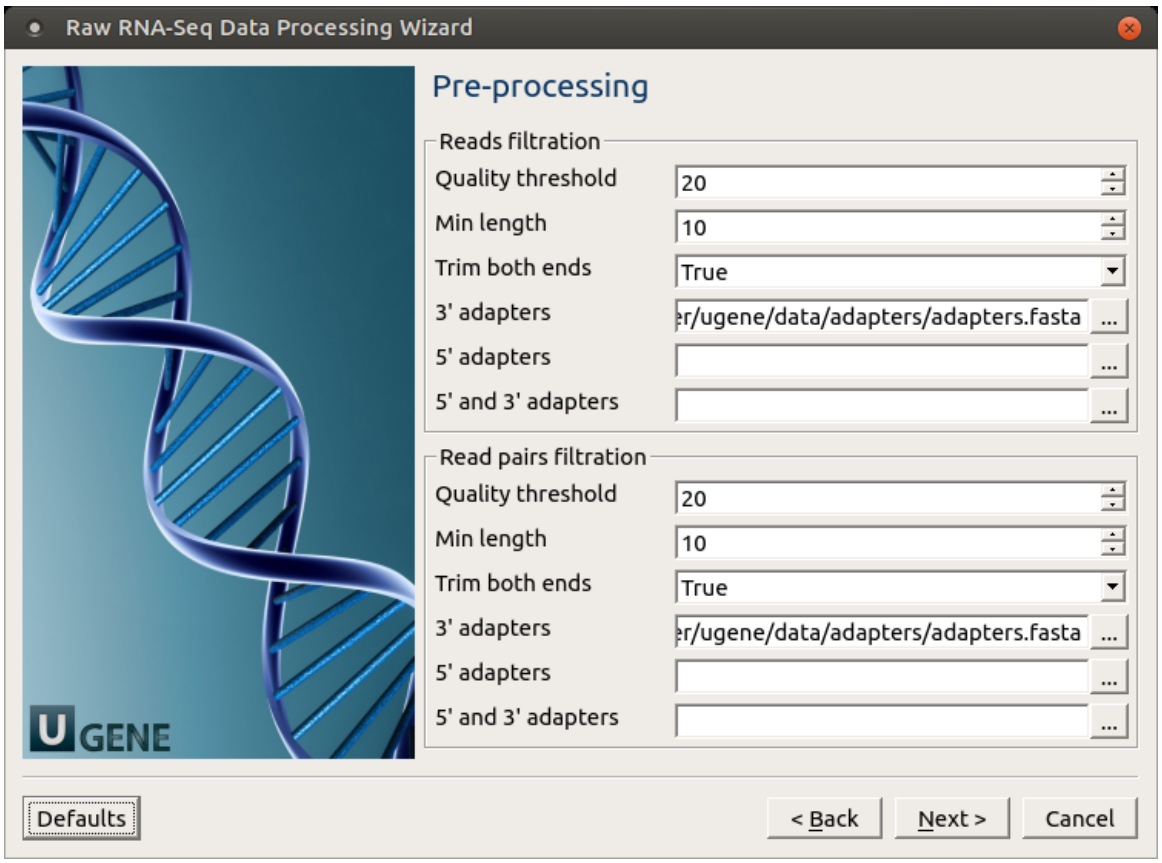
Workflow Wizard

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.

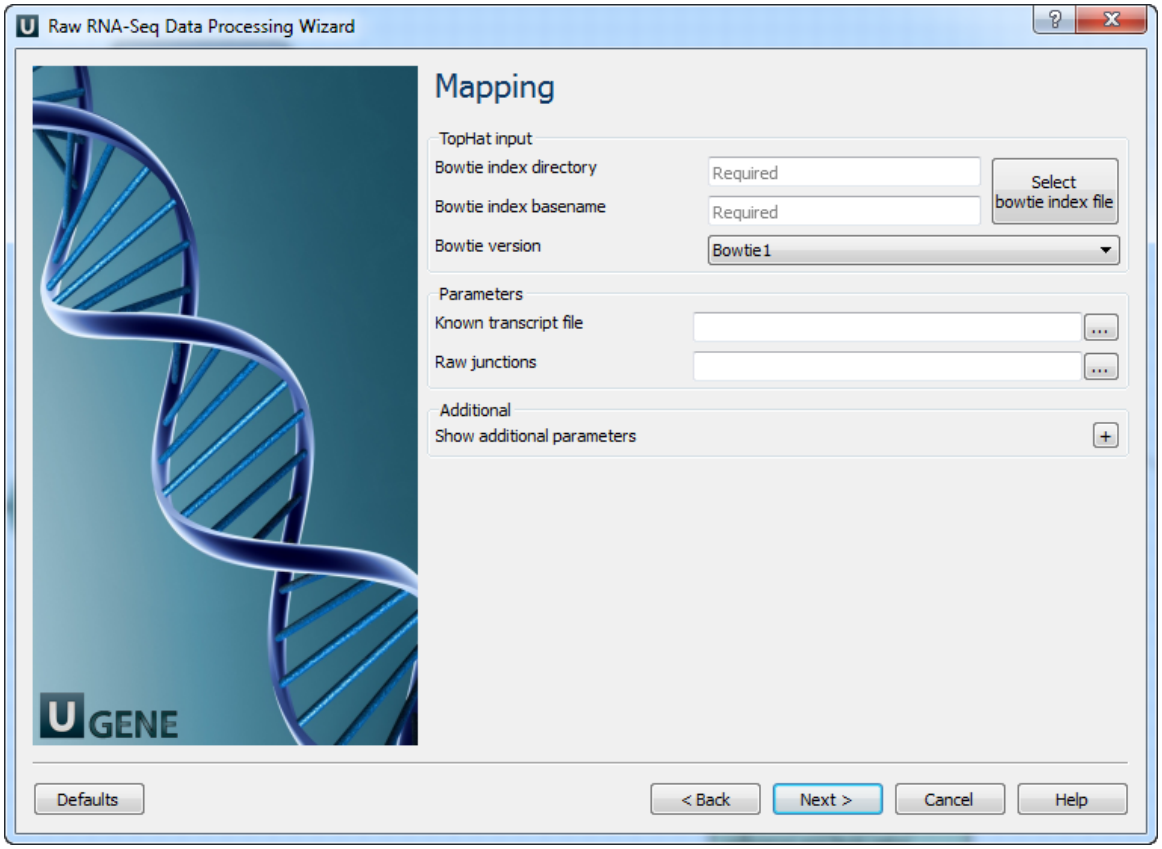


The following parameters are available for reads and reads pairs filtration:

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' 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 t he character ^, the adapter is 'anchored'. An anchored adapter must appear in its entirety at the 5' end o f 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.

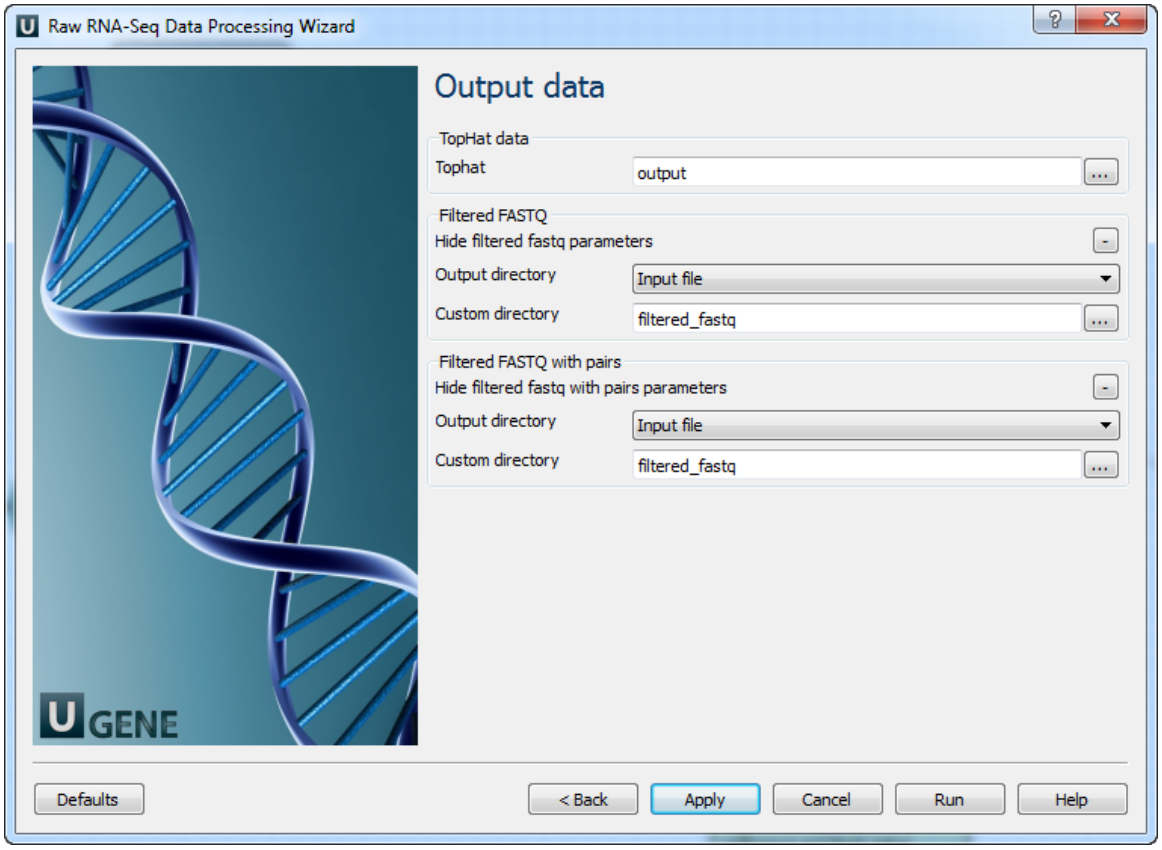


The following parameters are available:

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.

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 multihints	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 multihints	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.



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 Mac OS X 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

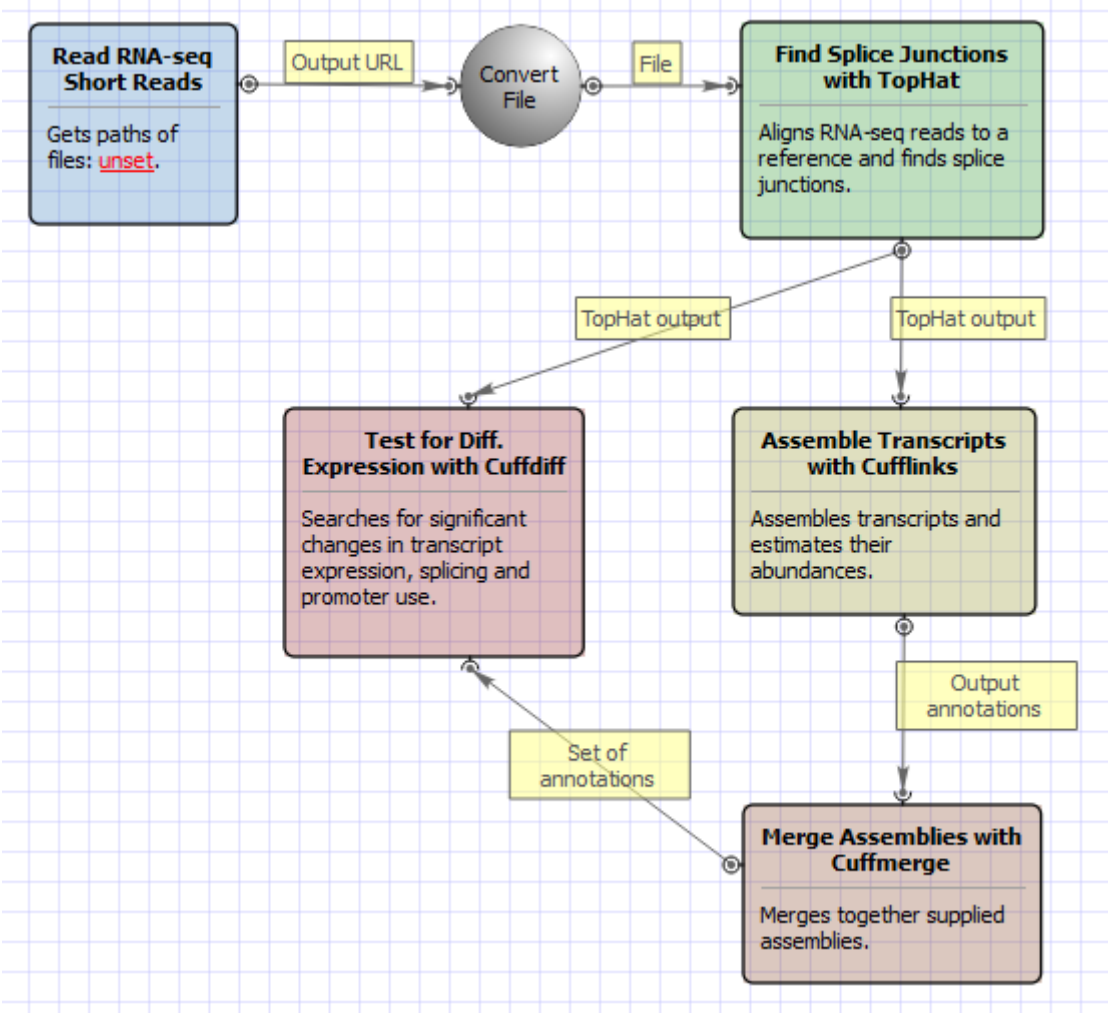
The workflow sample "RNA-seq Analysis with Tuxedo Tools" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

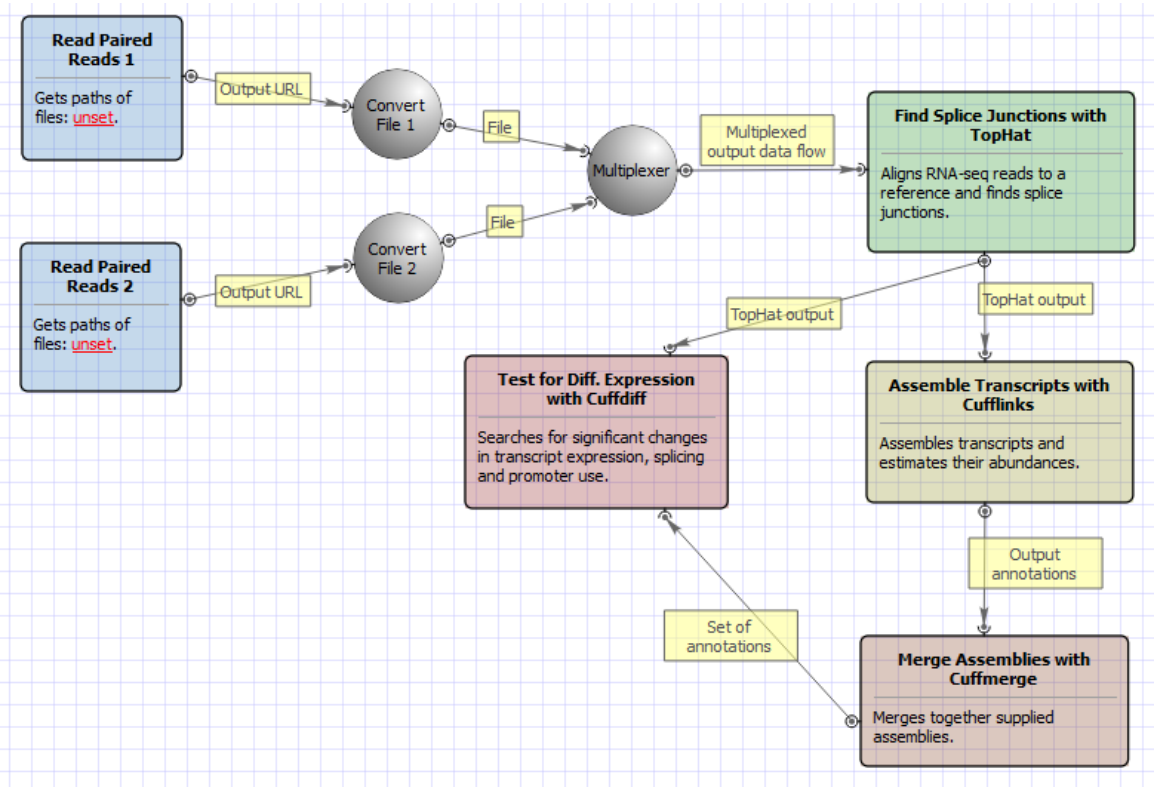
There are two short reads types of workflow: single-end and paired-end reads. For both of them there are three analysis types:

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.

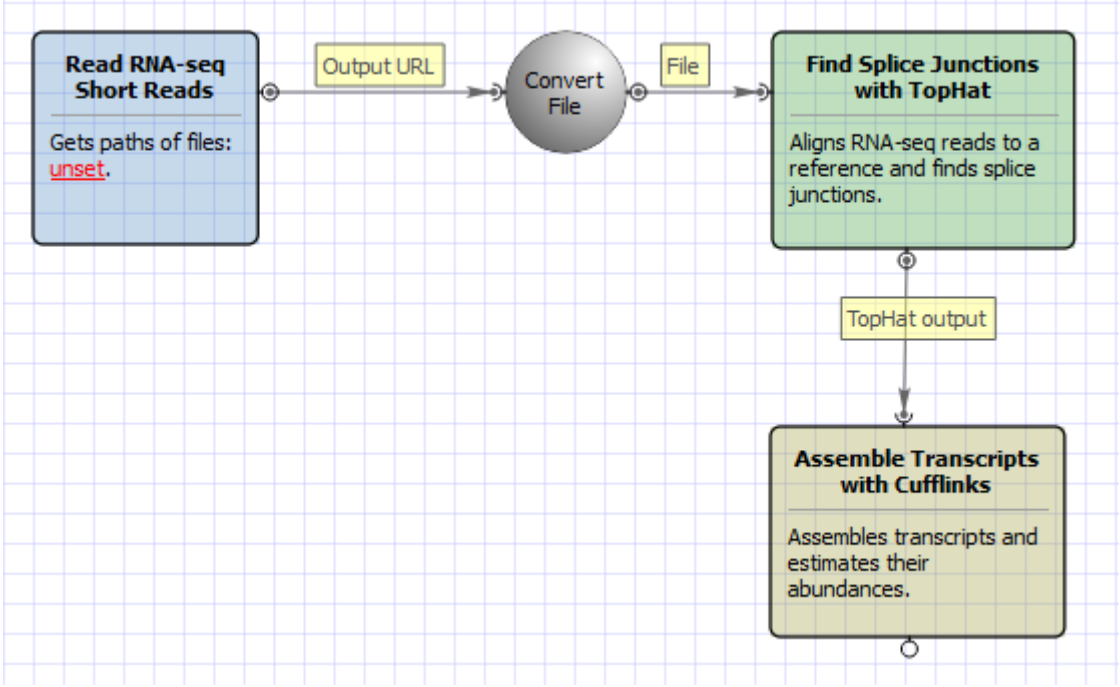
For **Full Tuxedo Pipeline** analysis type and **single-end reads** type the following workflow appears:



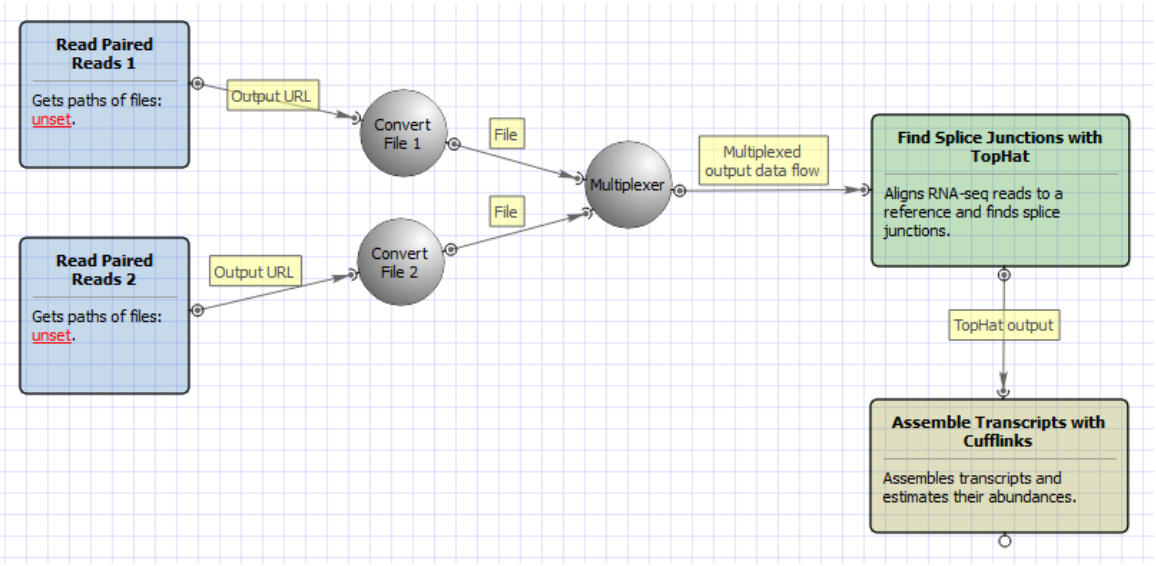
For **Full Tuxedo Pipeline** analysis type and **paired-end reads** type the following workflow appears:



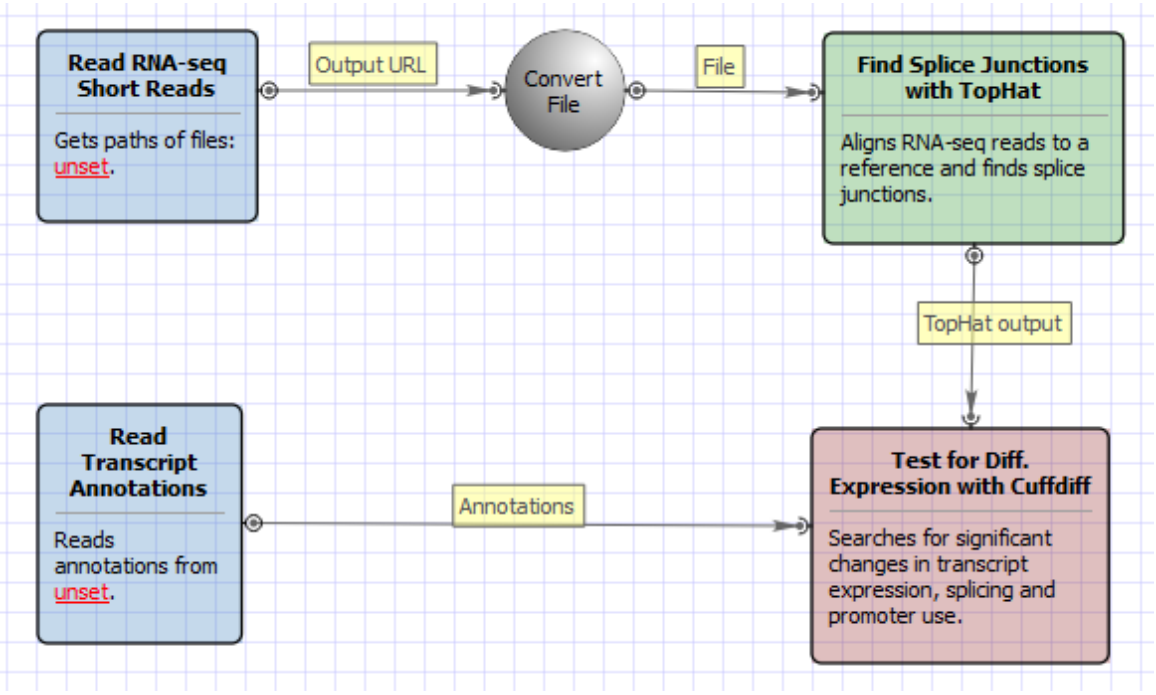
For **Single-sample Tuxedo Pipeline** analysis type and **single-end reads** type the following workflow appears:



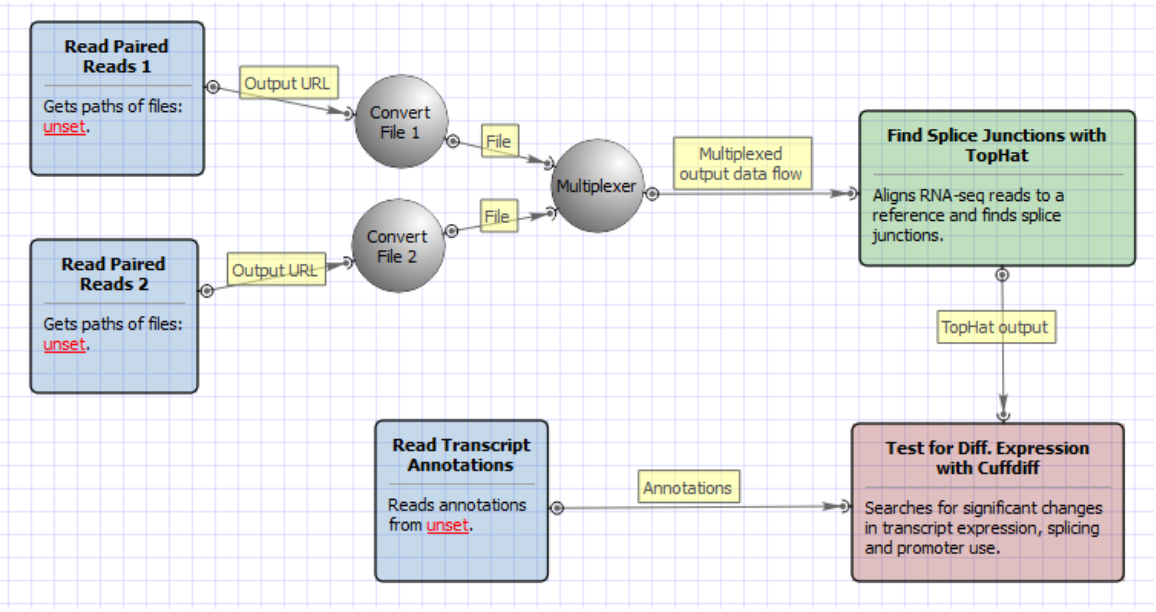
For **Single-sample Tuxedo Pipeline** analysis type and **paired-end reads** type the following workflow appears:



For **No-new-transcripts Tuxedo Pipeline** analysis type and **single-end reads** type the following workflow appears:



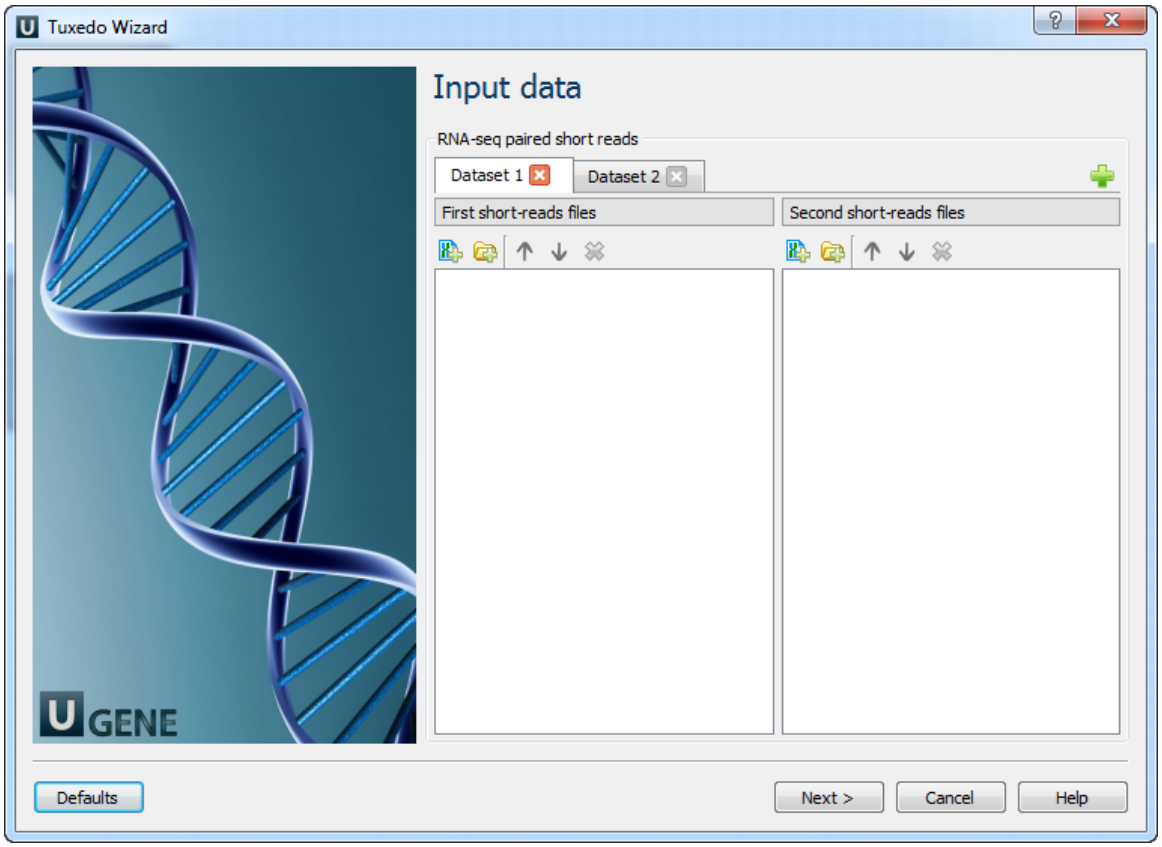
For **No-new-transcripts Tuxedo Pipeline** analysis type and **paired-end reads** type the following workflow appears:



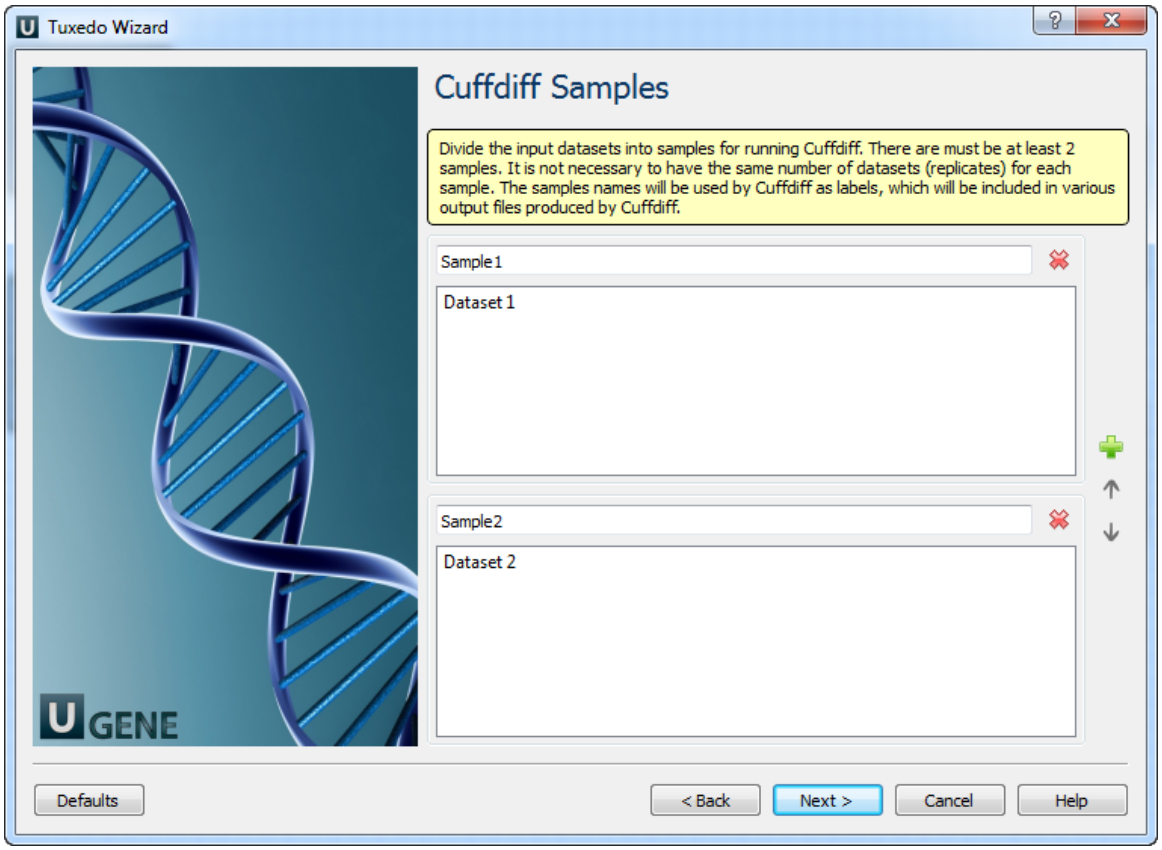
Workflow Wizard

All of these workflows have the similar wizards. For **Full Tuxedo Pipeline** analysis type and **paired-end reads** type wizard has 7 pages.

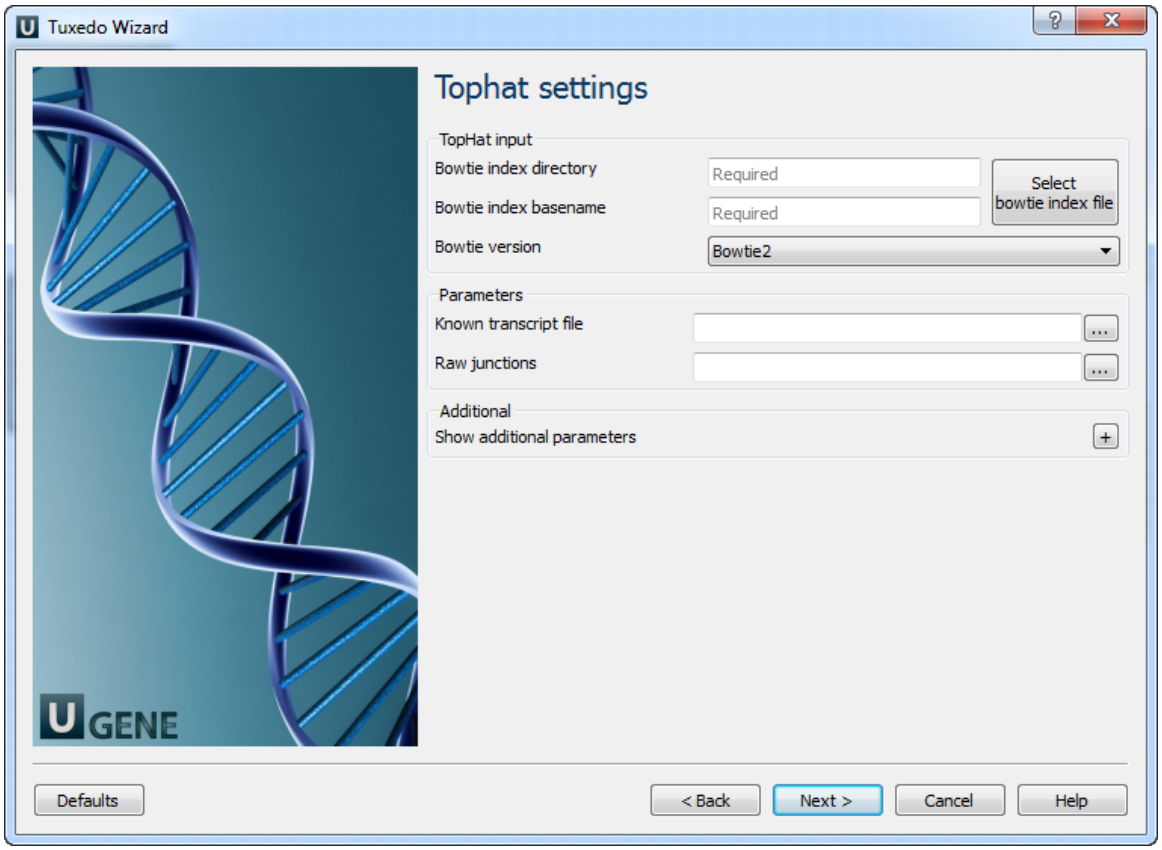
1. Input data: Here you need to input RNA-seq short reads in FASTA or FASTQ formats. Many datasets with different reads can be added.



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 neccessary 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.

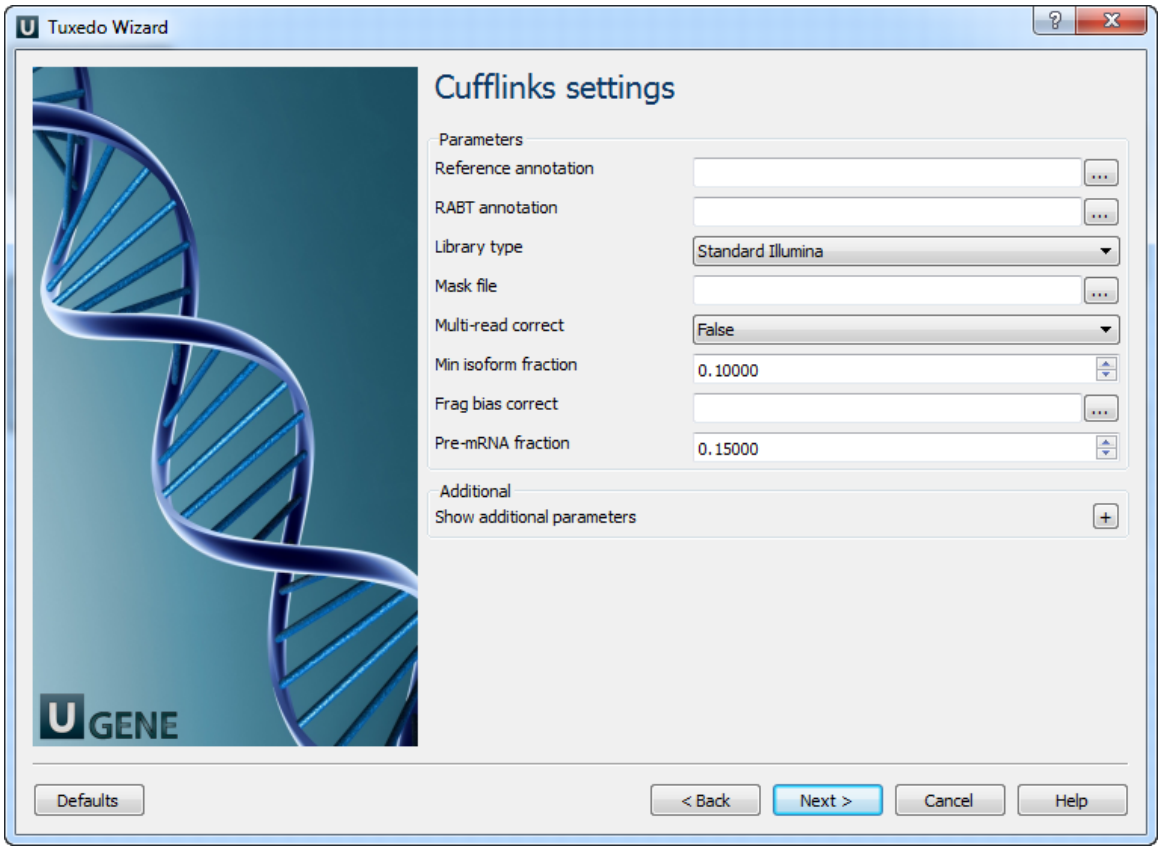


The following parameters are available:

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 multihints	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 multihints	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:

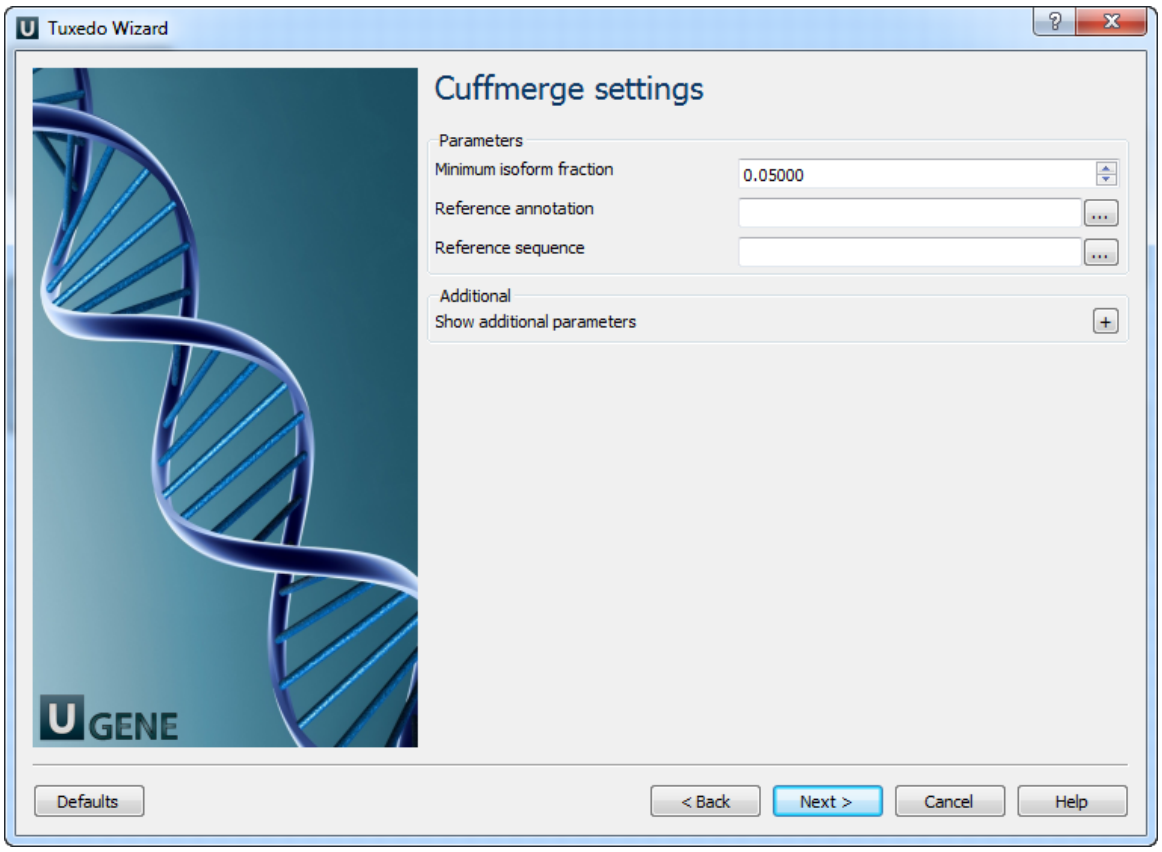


The following parameters are available:

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 assembly. Output will include all reference transcripts as well as any novel genes and isoforms that are assembled.
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.
Multi-read correct	Tells Cufflinks to do an initial estimation procedure to more accurately weight reads mapping to multiple locations in the genome.
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.

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.

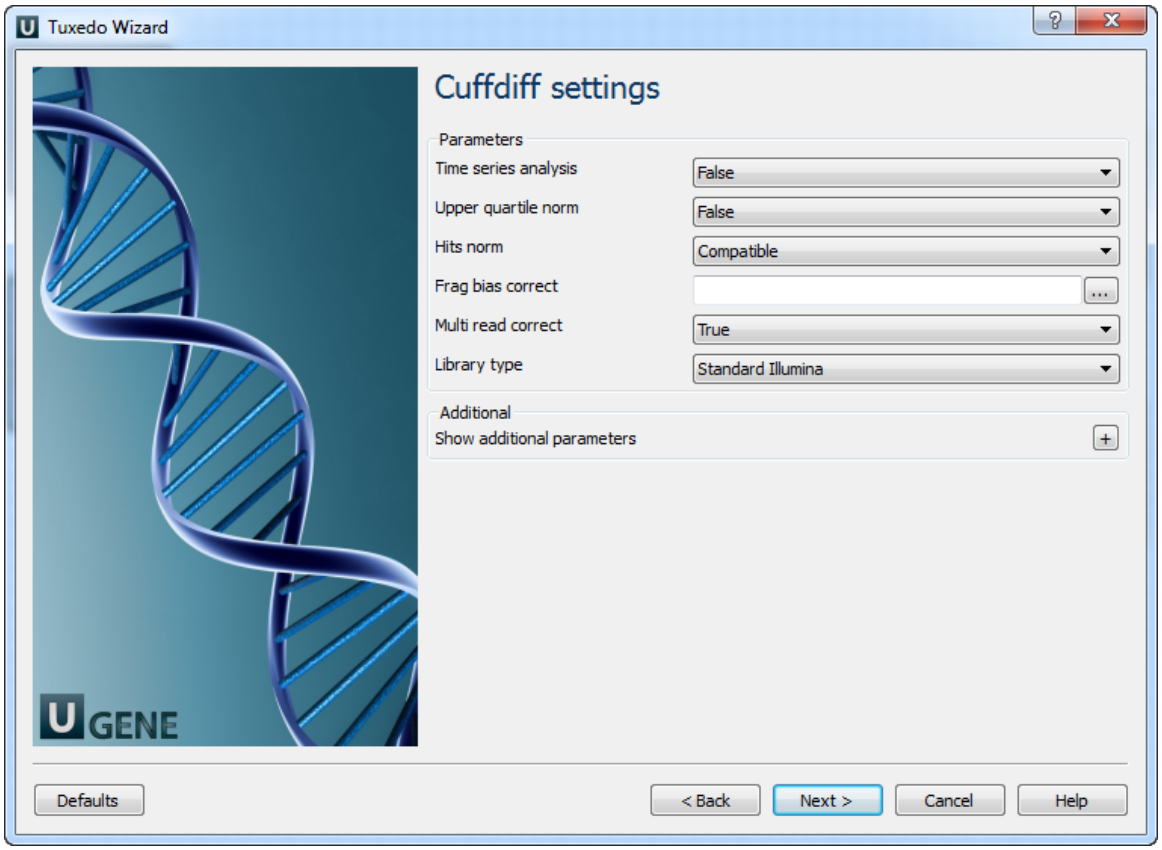
5. Cuffmerge settings: On this page you can modify Cuffmerge parameters.



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:

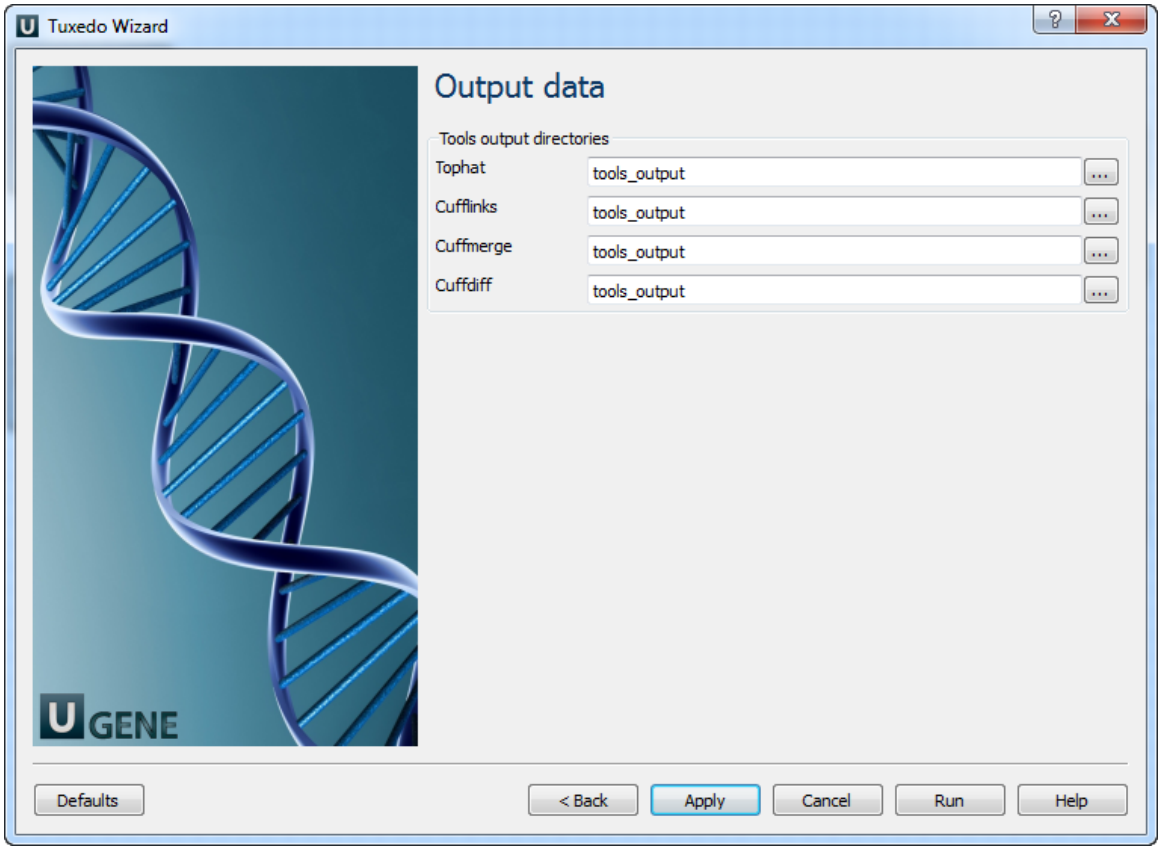


The following parameters are available:

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 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 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 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.
FDR	Allowed false discovery rate used in testing.
Max MLE iterations	Sets the number of iterations allowed during maximum likelihood estimation of abundances.
Emit count tables	Include information about the fragment counts, fragment count variances, and fitted variance model into the report.
Cuffdiff tool path	The path to the Cuffdiff external tool in UGENE.
Temporary directory	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.

Get Unmappet Reads

Use this workflow sample to extract unmapped reads from an input SAM/BAM file.

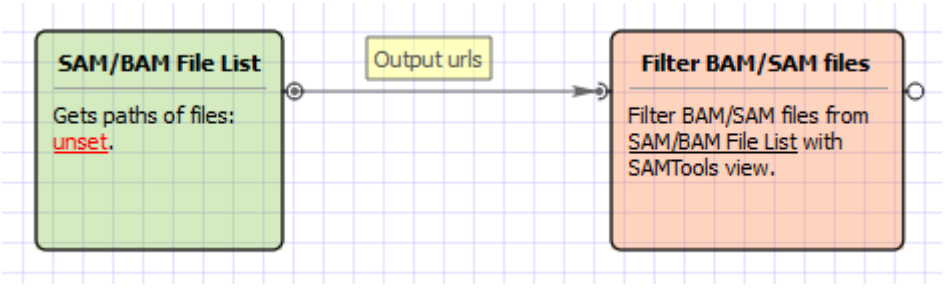
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Get Unmapped Reads" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

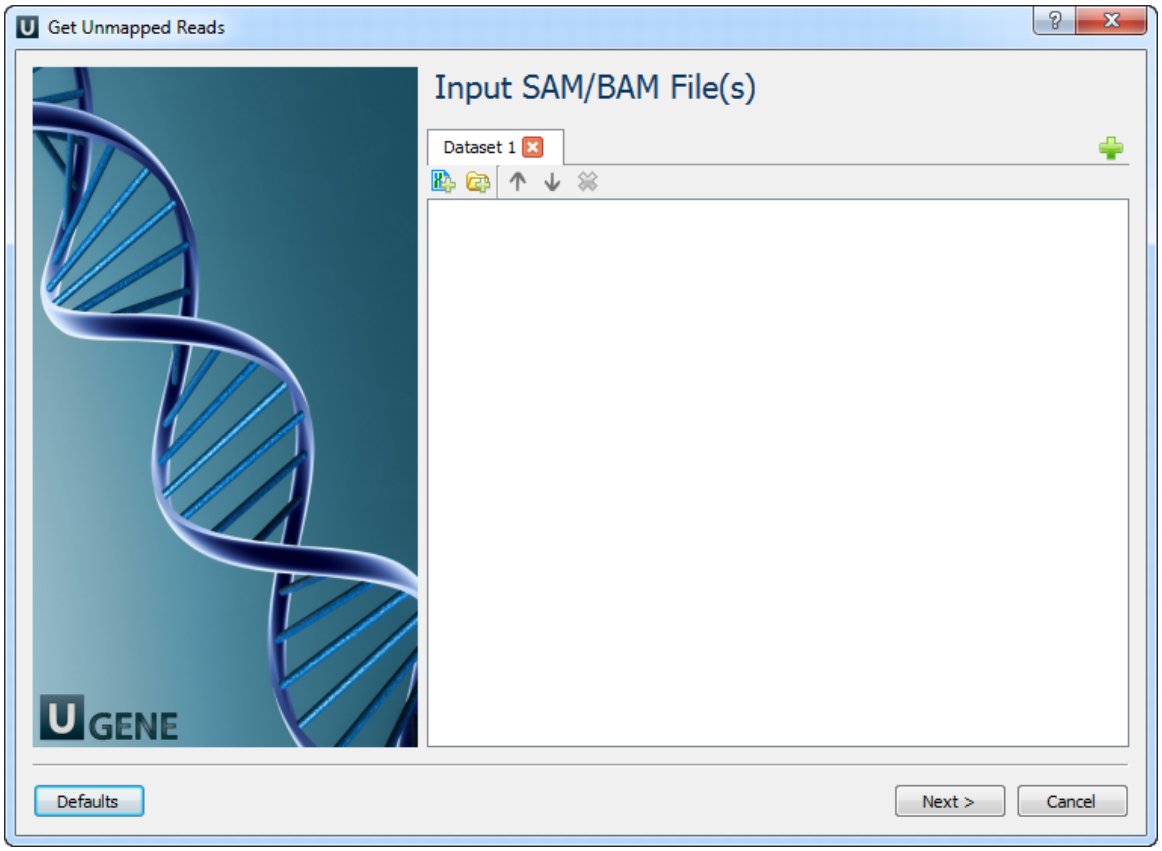
The workflow looks as follows:



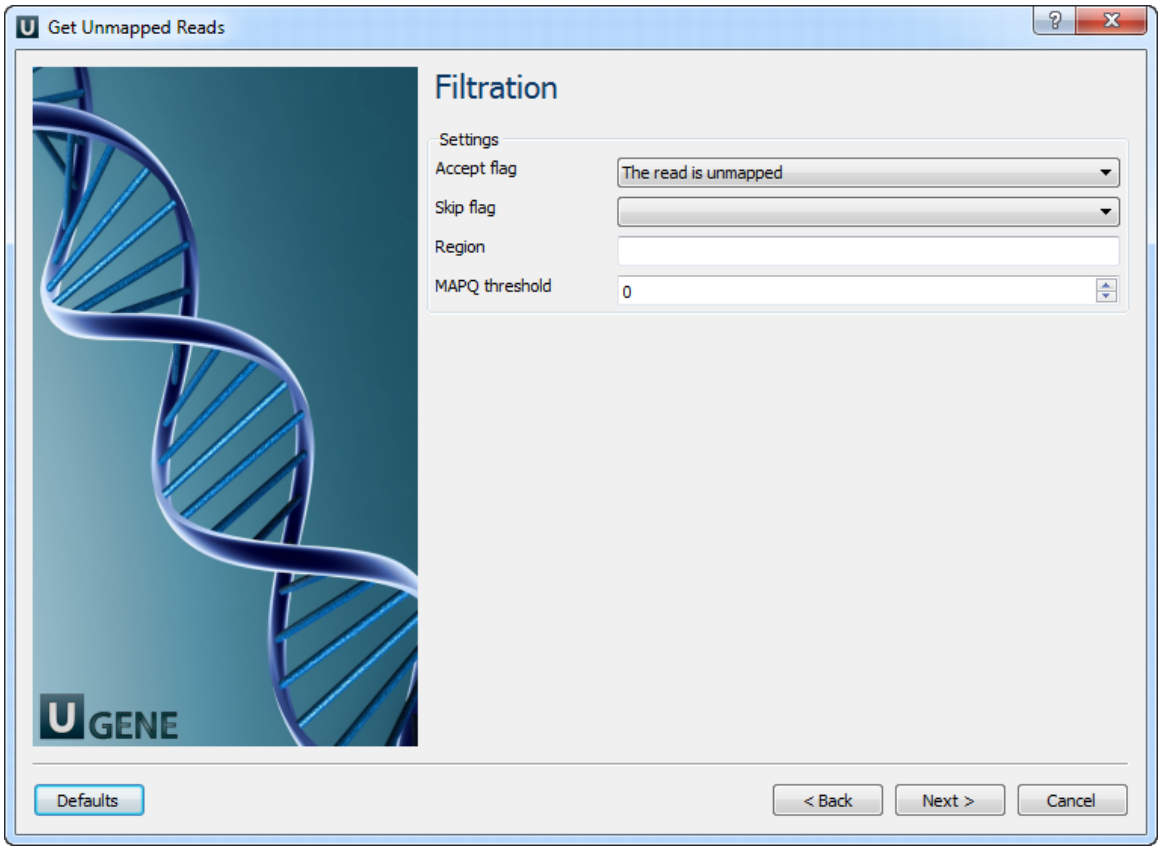
Workflow Wizard

The wizard has 3 page.

1. Input SAM/BAM File(s): On this page you need input SAM/BAM file(s).



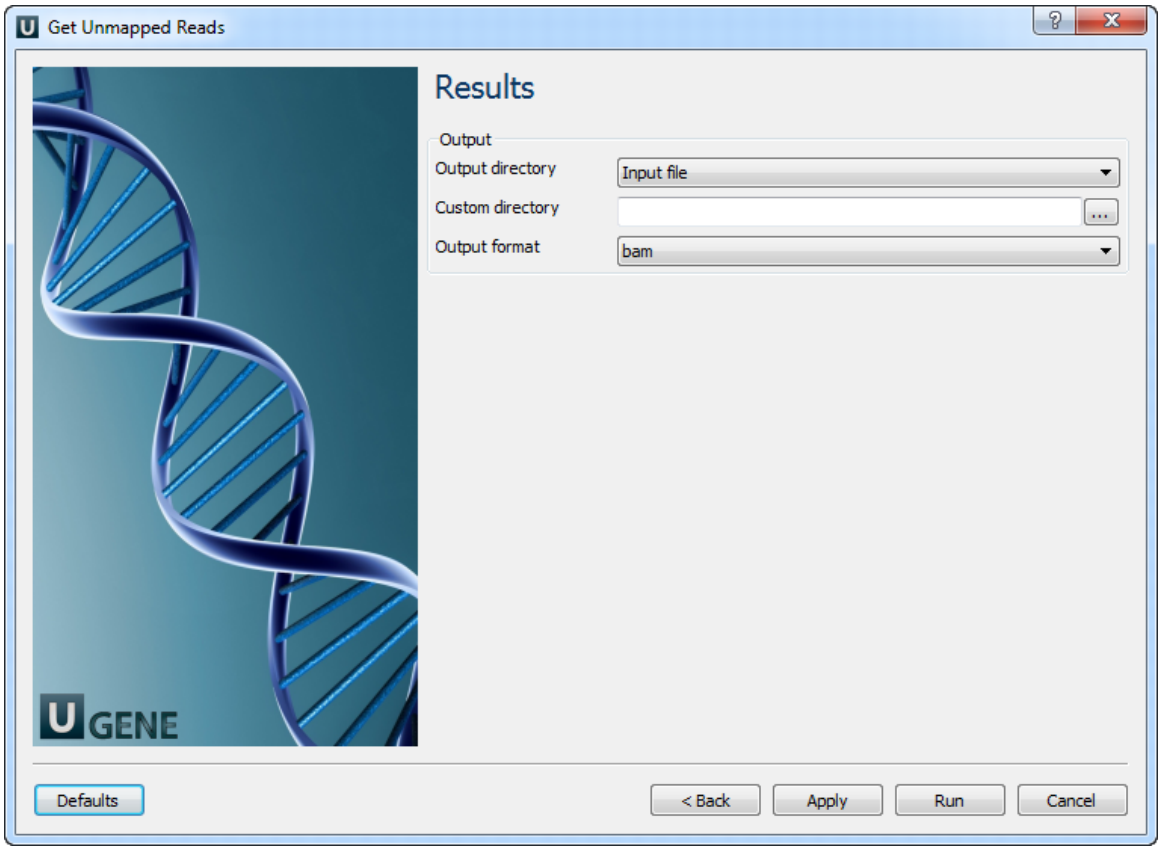
2. Filtration: On this page you can change the filtration parameters.



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.



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).

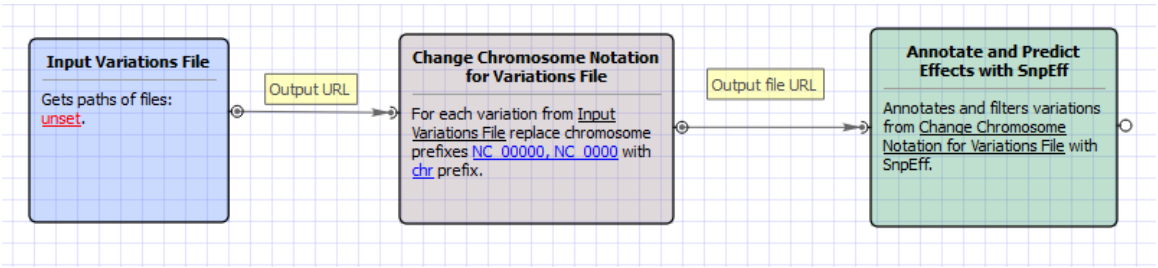
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Variation Annotation with SnpEff" can be found in the "NGS" section of the Workflow Designer samples.

Workflow Image

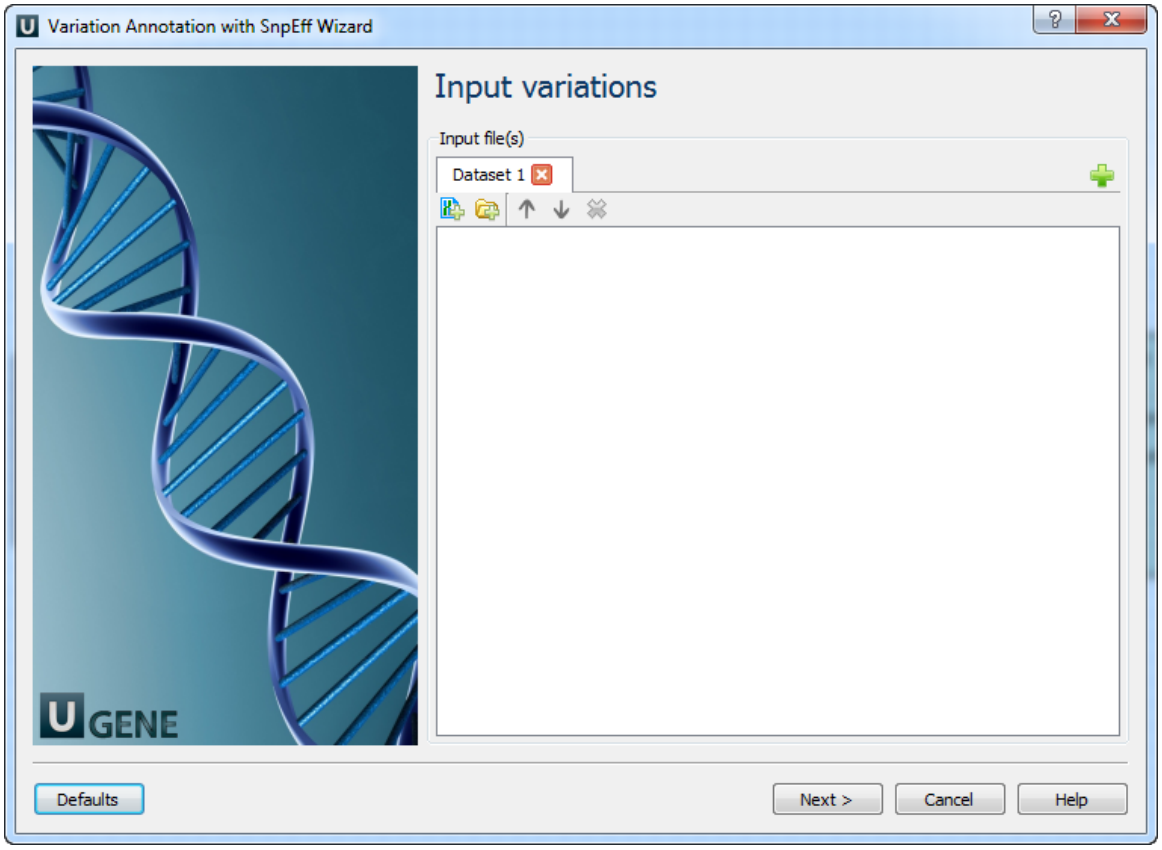
The opened workflow looks as follows:



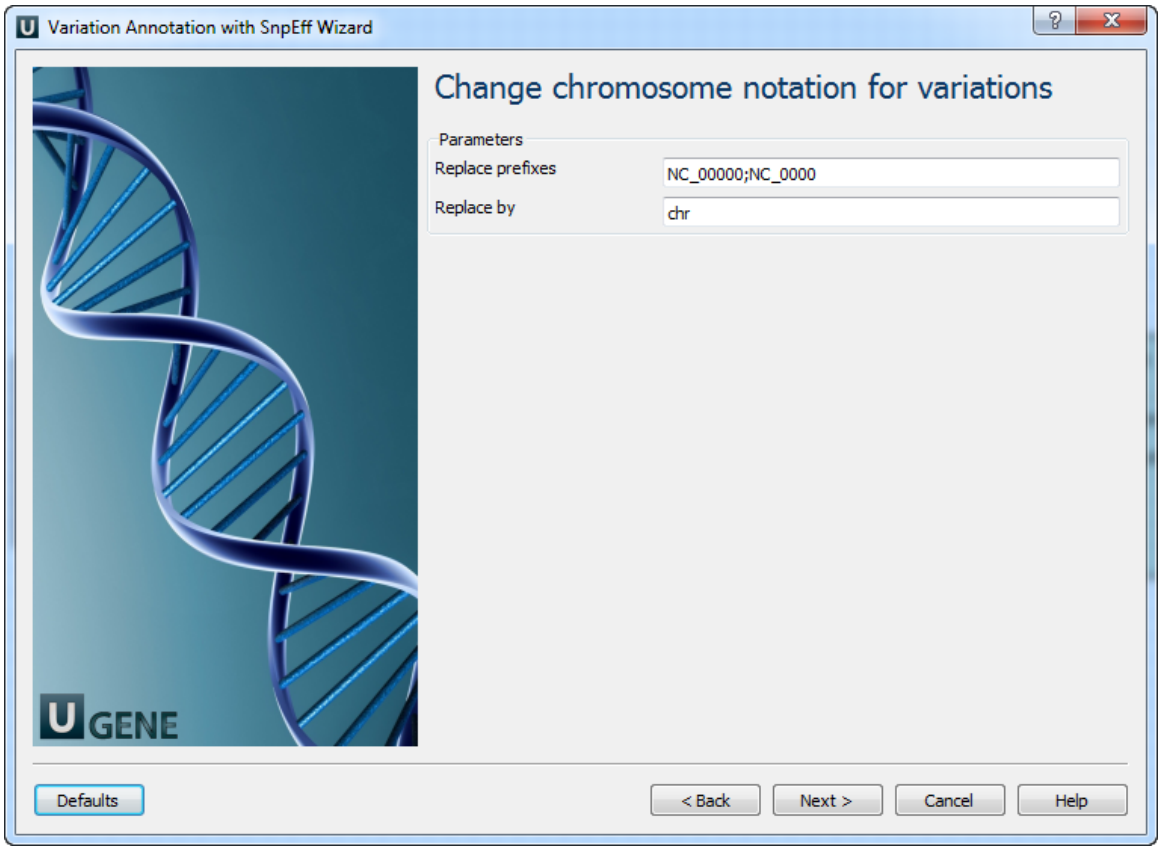
Workflow Wizard

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.

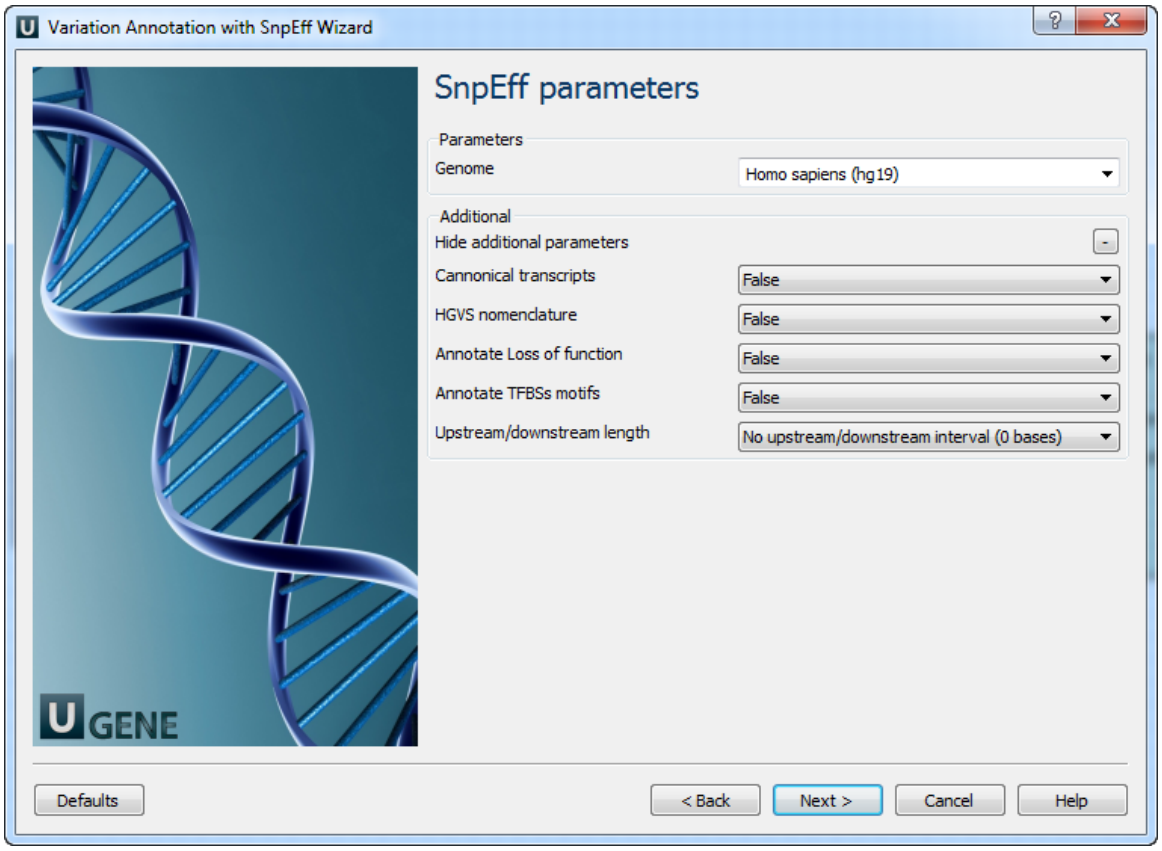


The following parameters are available:

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".

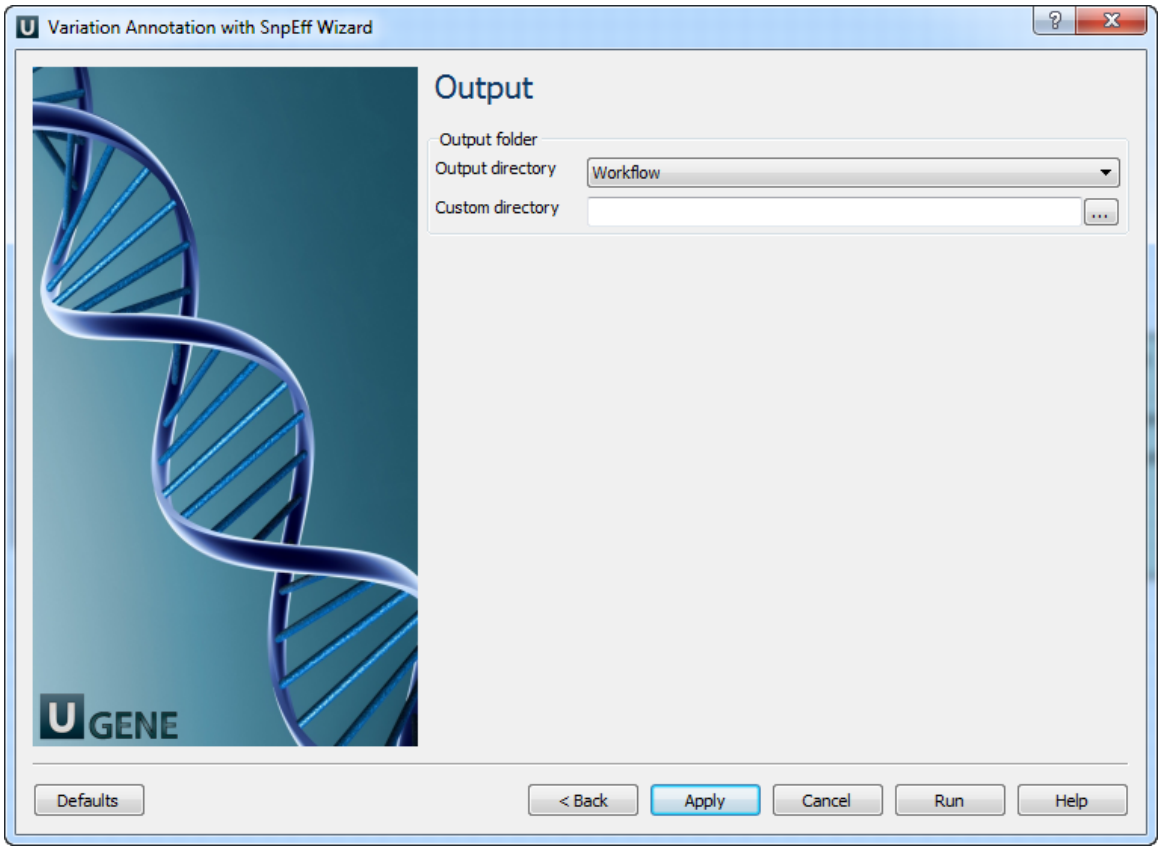
3. SnpEff Parameters: On this page you can modify SnpEff parameters.



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.



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.

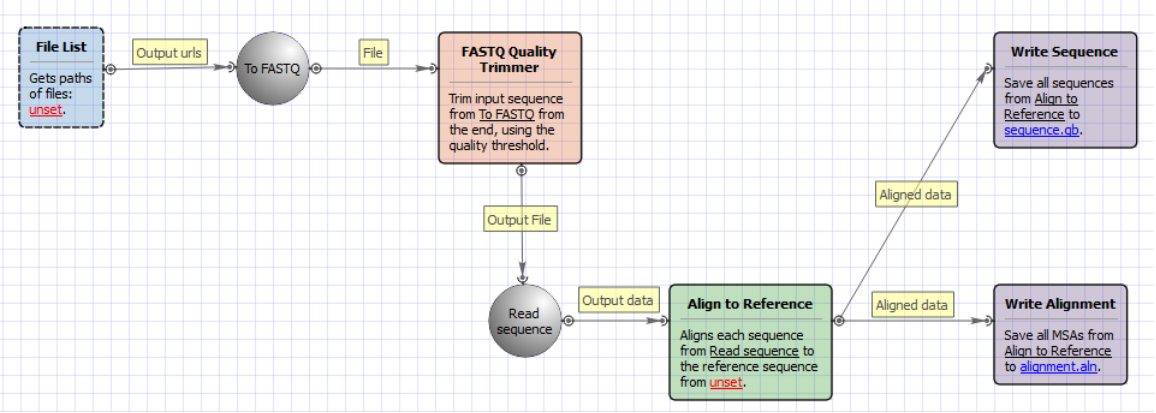
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

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

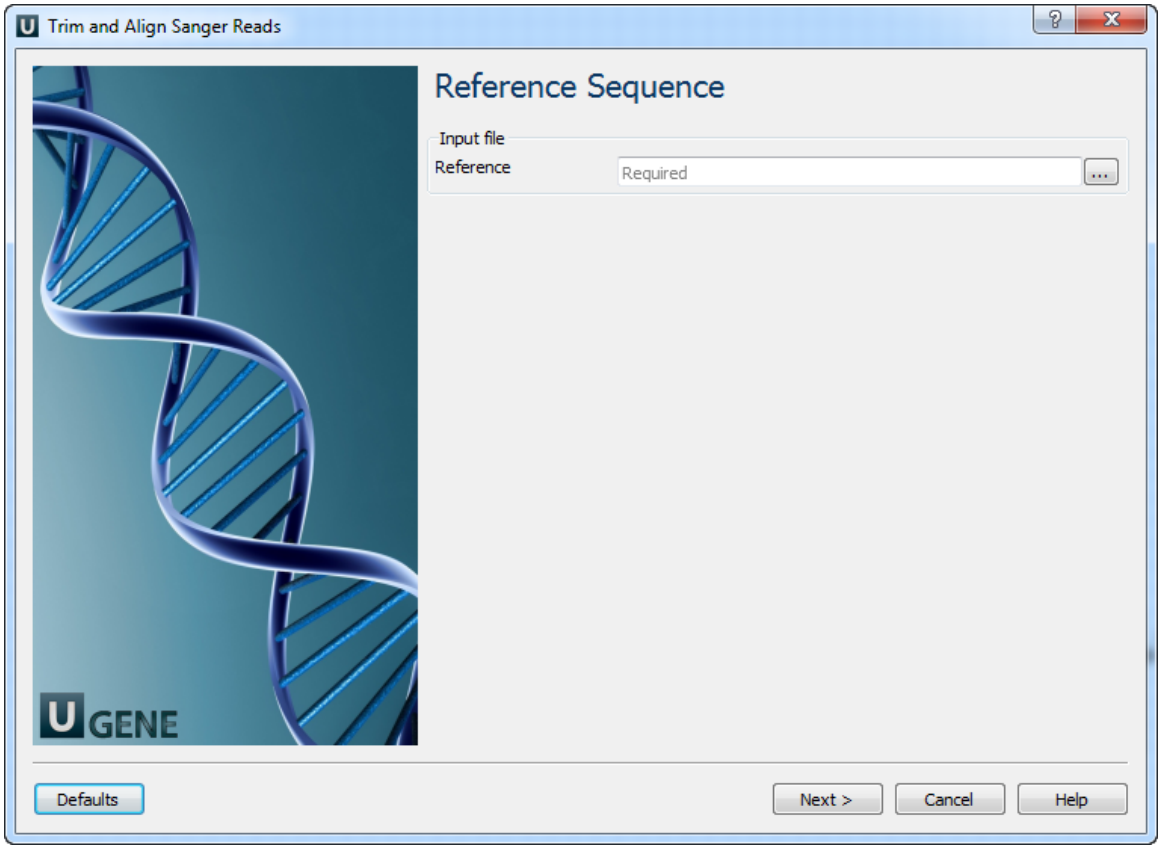
The opened workflow looks as follows:



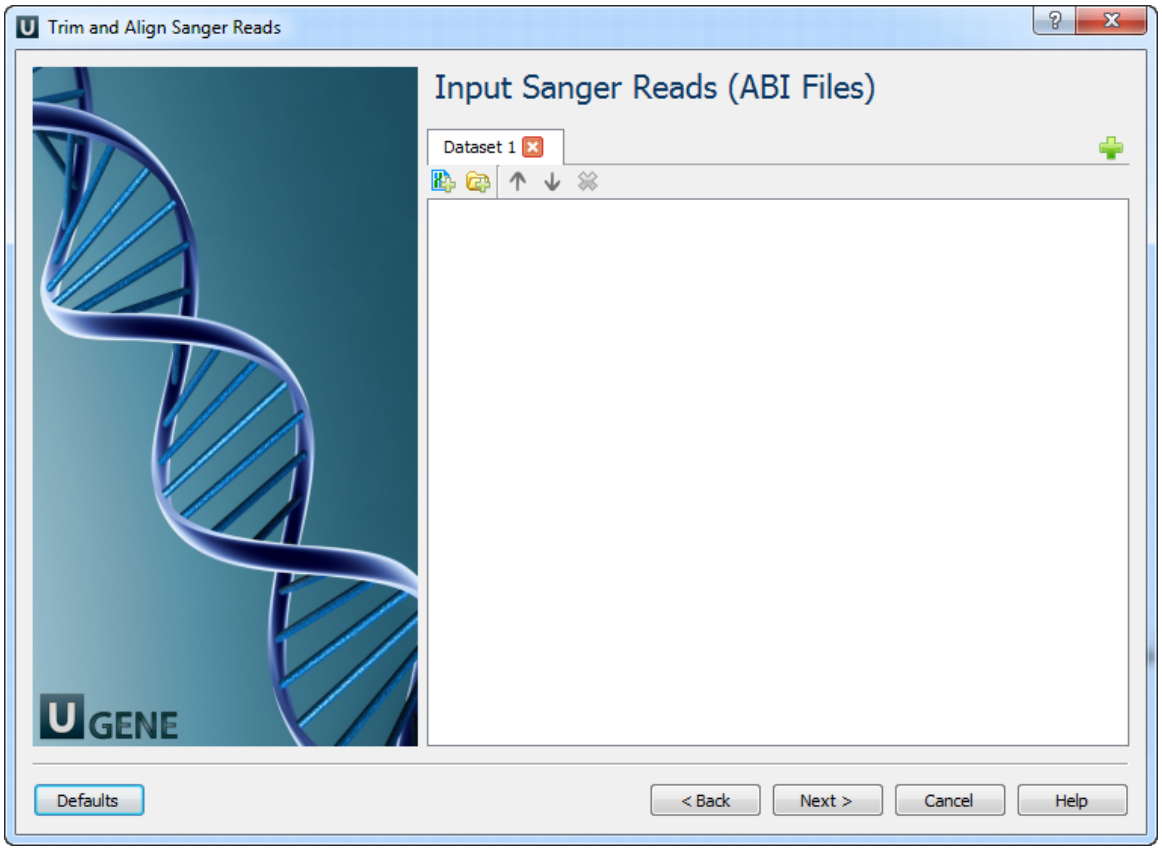
Workflow Wizard

The wizard has 4 pages.

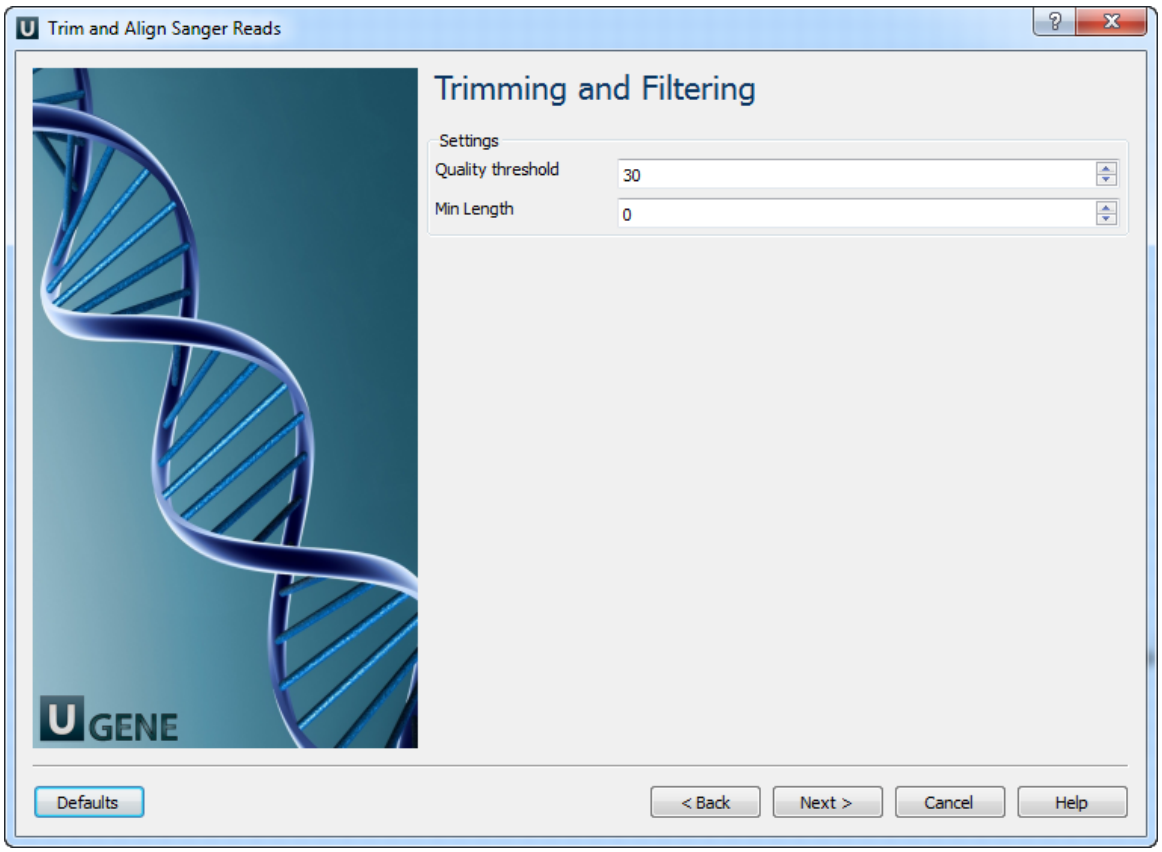
1. Reference Sequence: On this page you must input reference sequence.



2. Input Sanger Reads (ABI Files): On this page you must input ABI file(s).



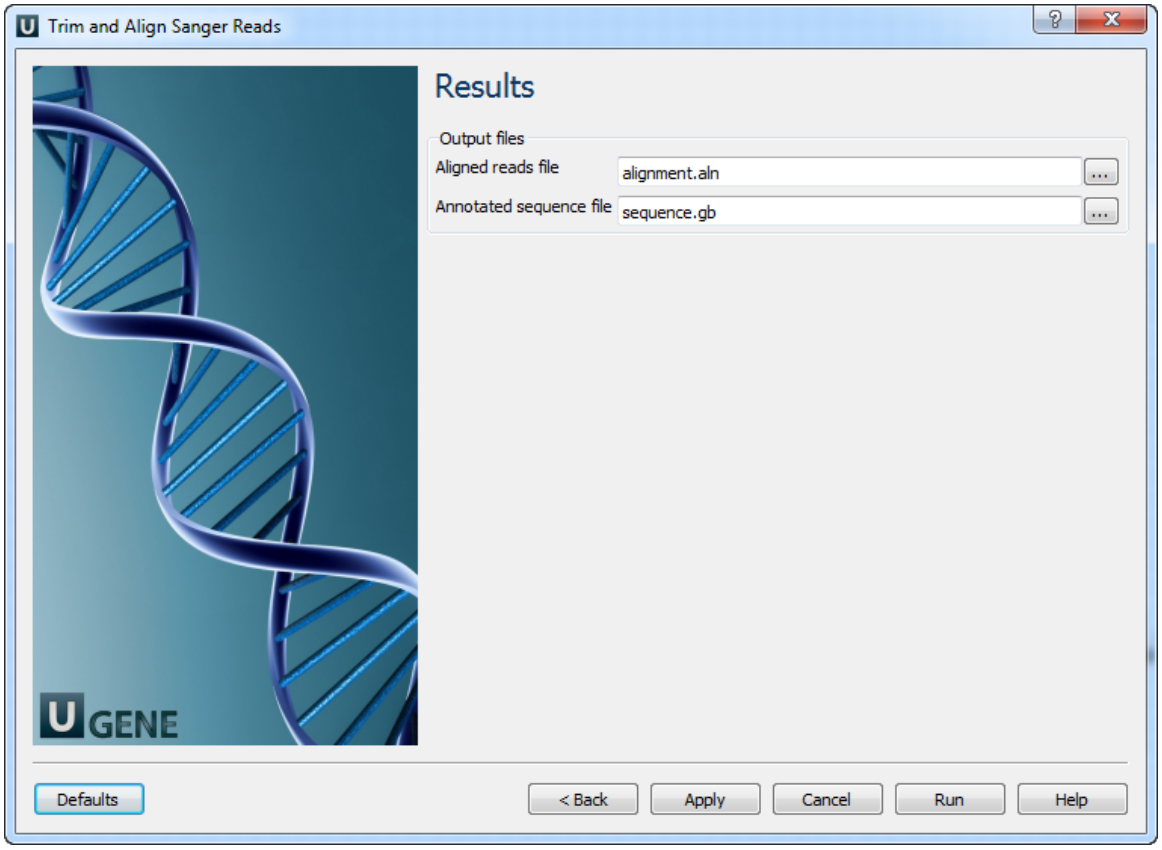
3. Trimming and Filtering: On this page you can modify trimming and filtering settings.



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
- Merge Sequences and Annotations
- In Silico PCR
- 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.

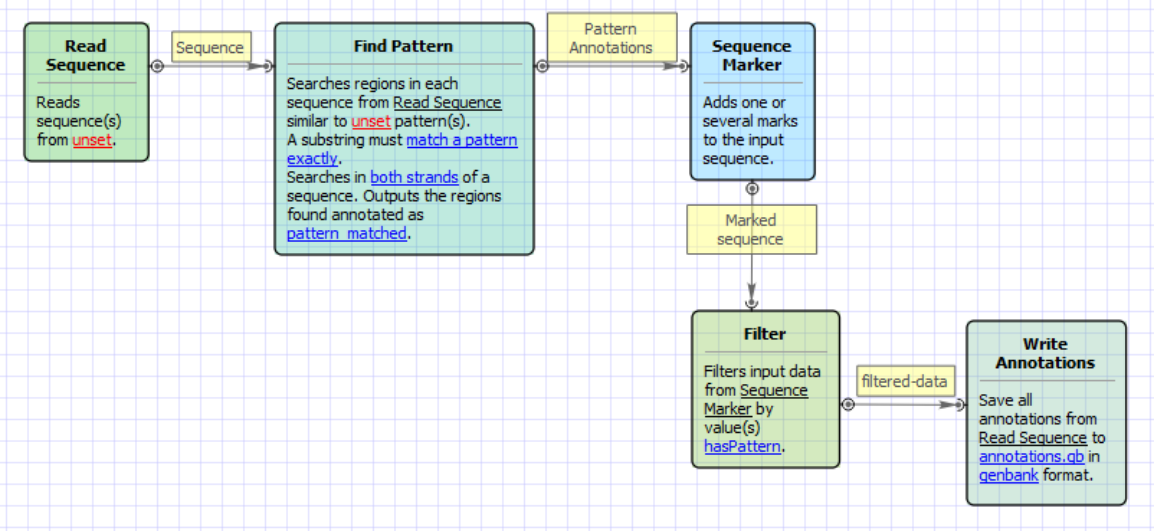
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

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

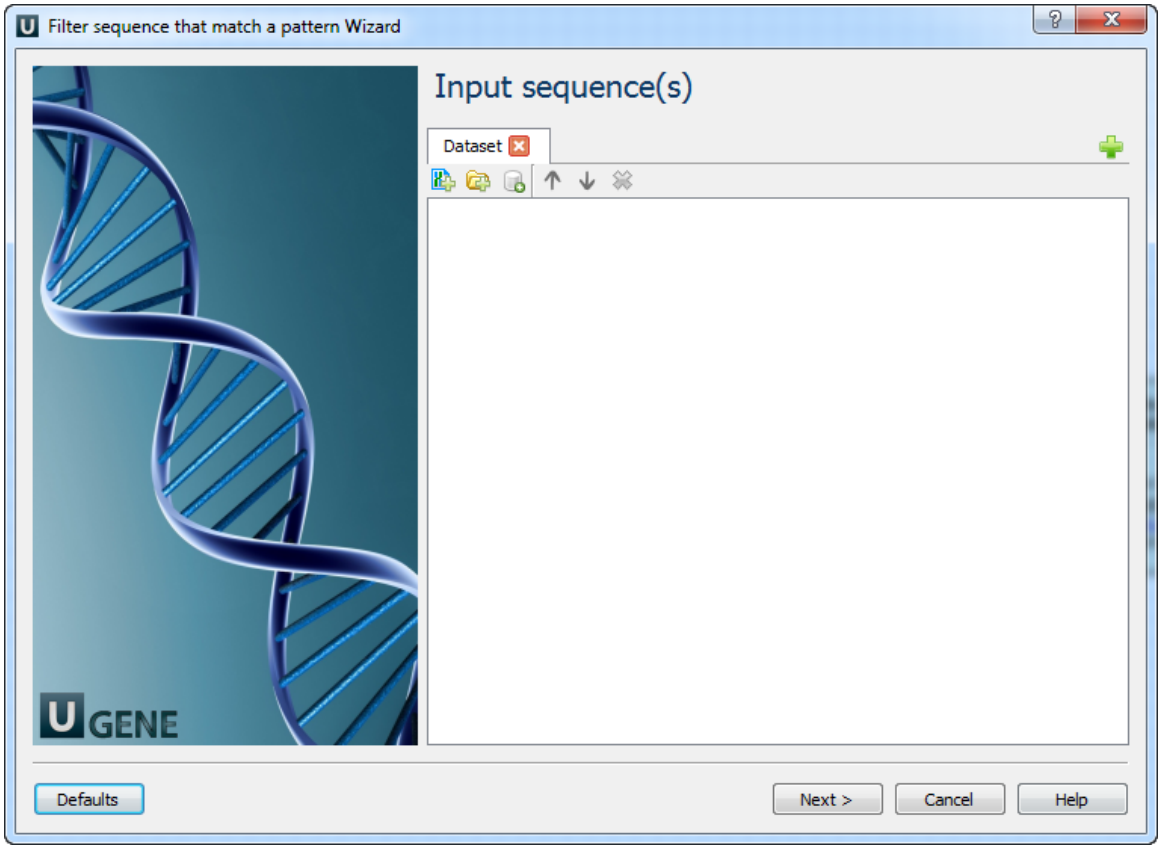
The workflow looks as follows:



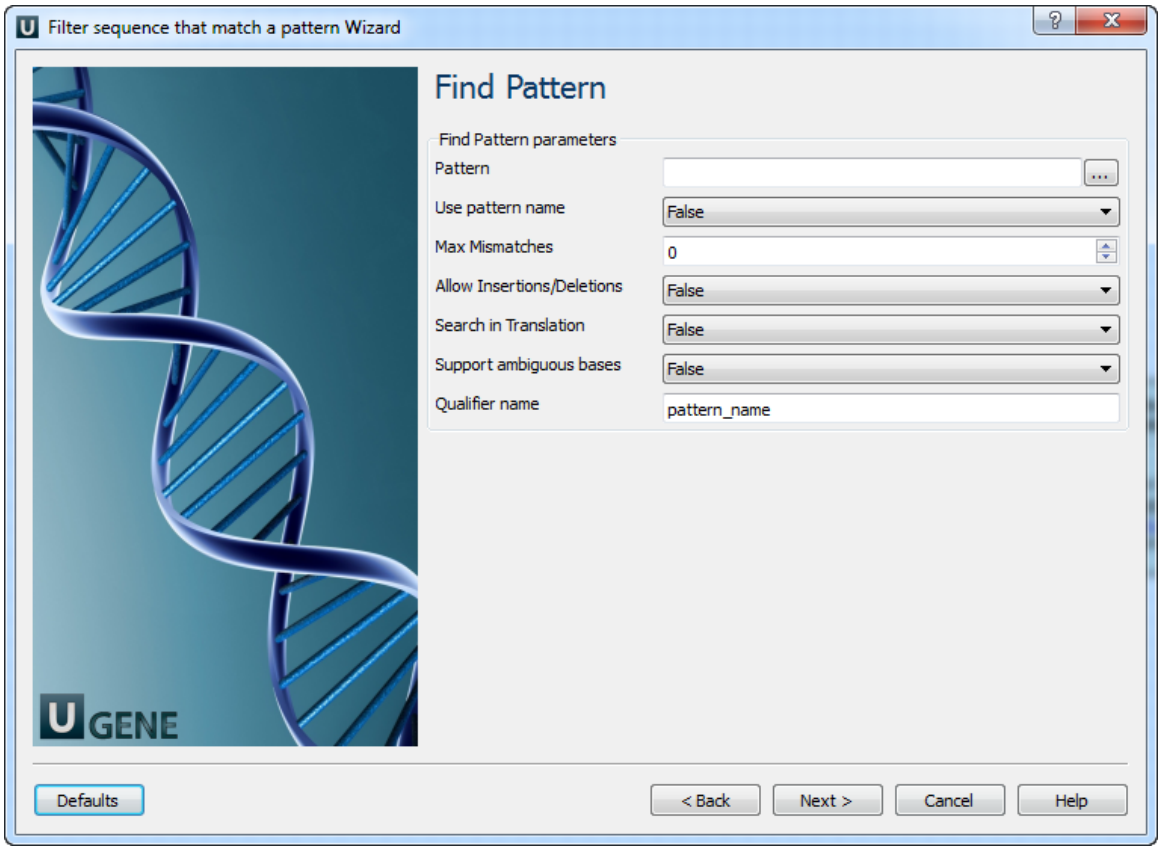
Workflow Wizard

The wizard has 3 pages.

1. Input sequence(s): On this page you must input sequence(s).



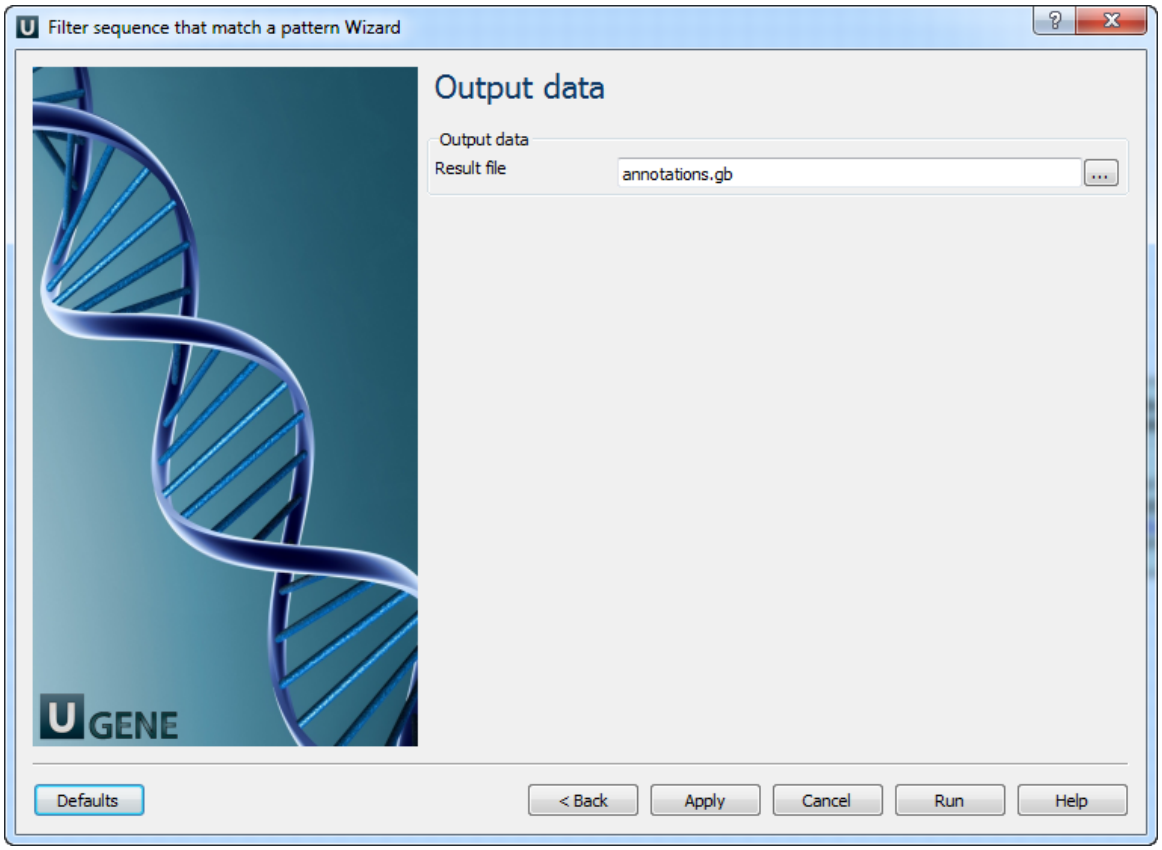
2. Find pattern: On this page you must input pattern(s) and you can modify searching parameters.



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.

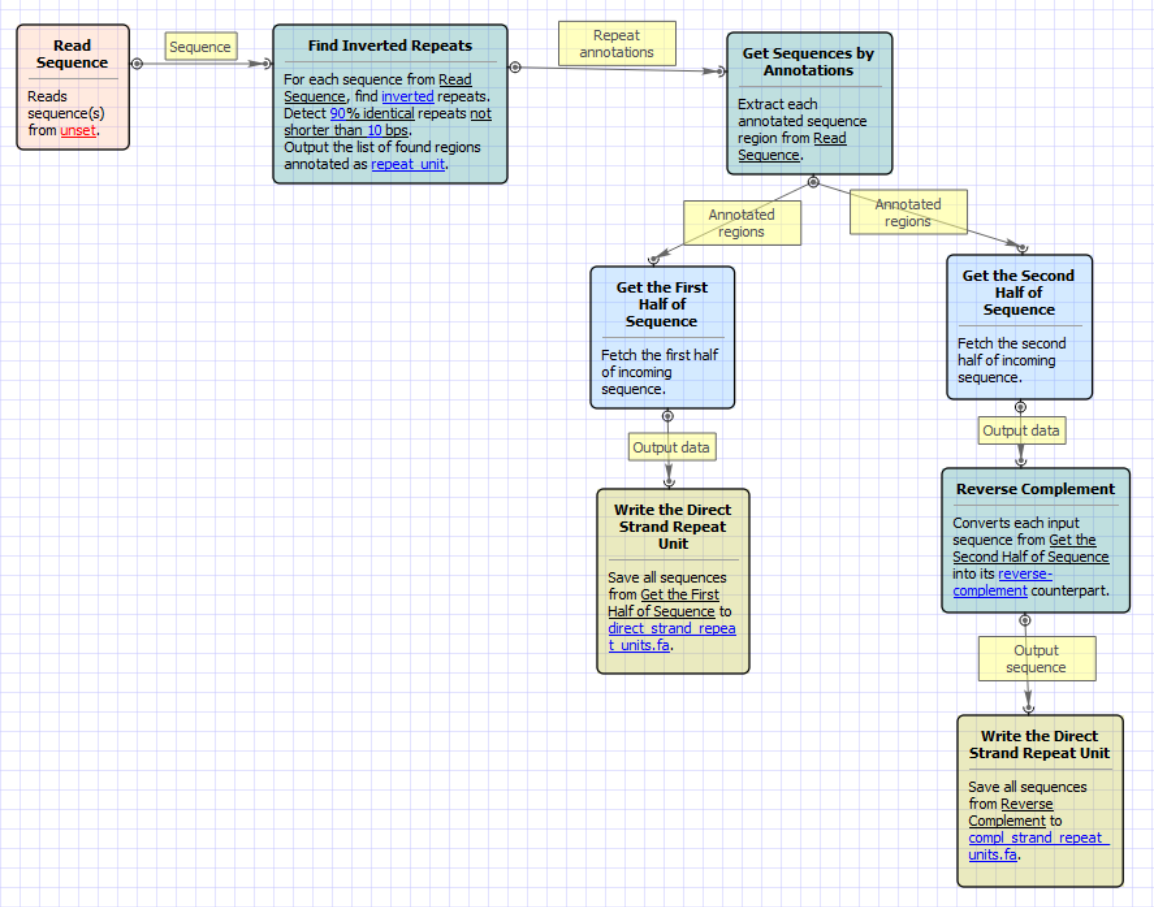
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Search for Inverted Repeats" can be found in the "Scwnarios" section of the Workflow Designer samples.

Workflow Image

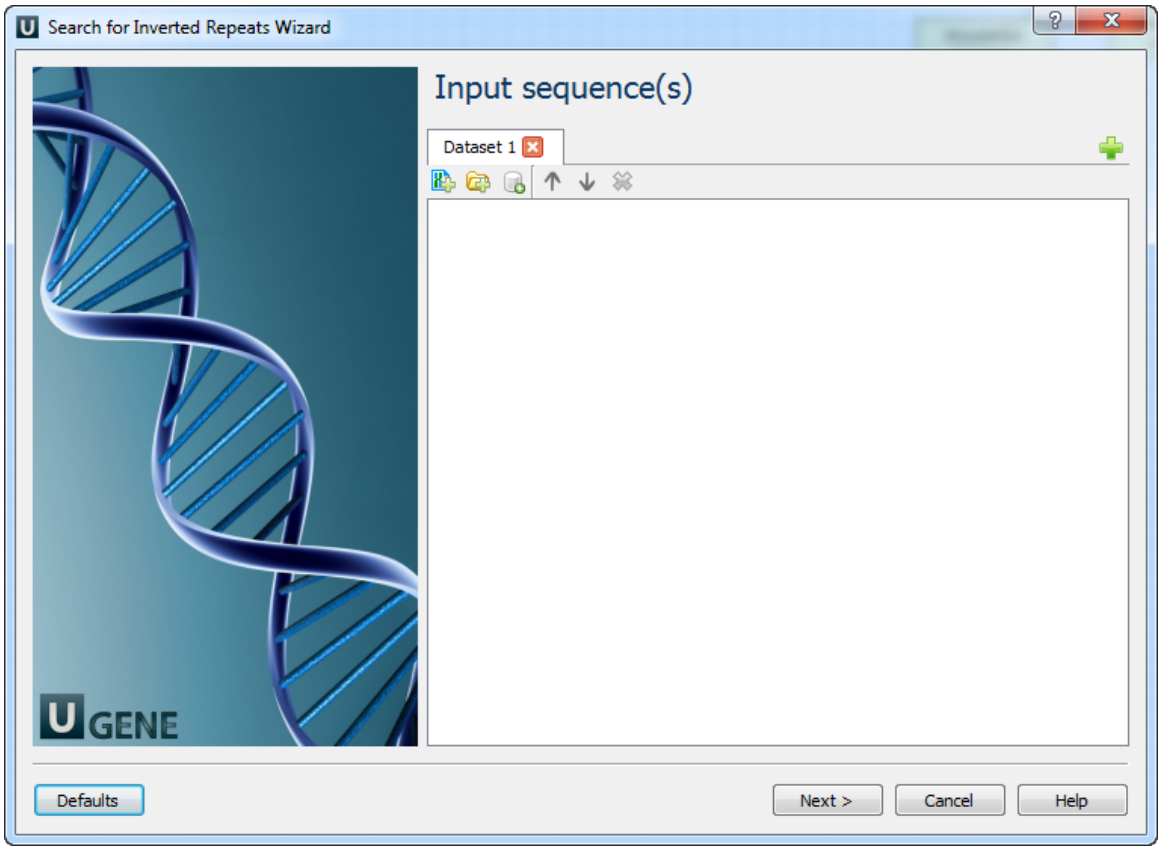
The opened workflow looks as follows:



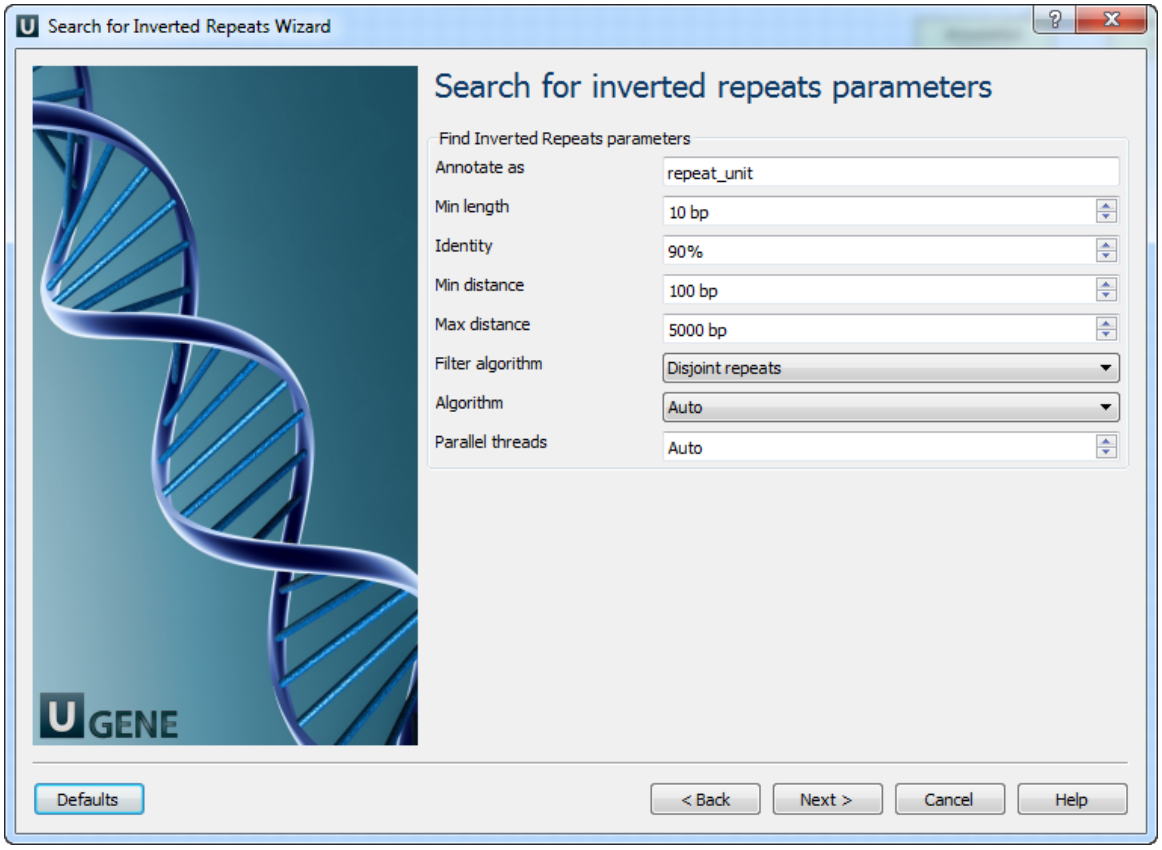
Workflow Wizard

The wizard has 3 pages.

1. Input sequence(s): On this page you must input sequence(s).



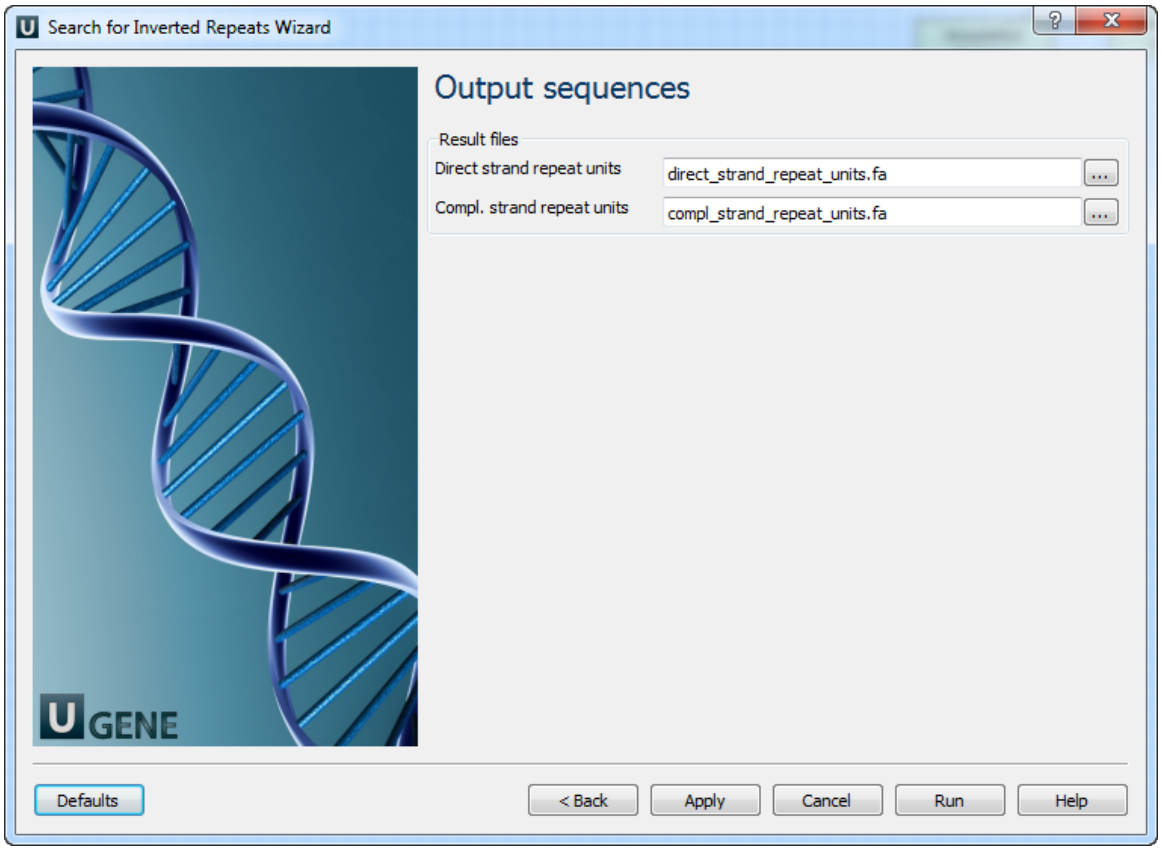
2. Search for inverted repeats parameters: On this page you can modify inverted repeats parameters.



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.

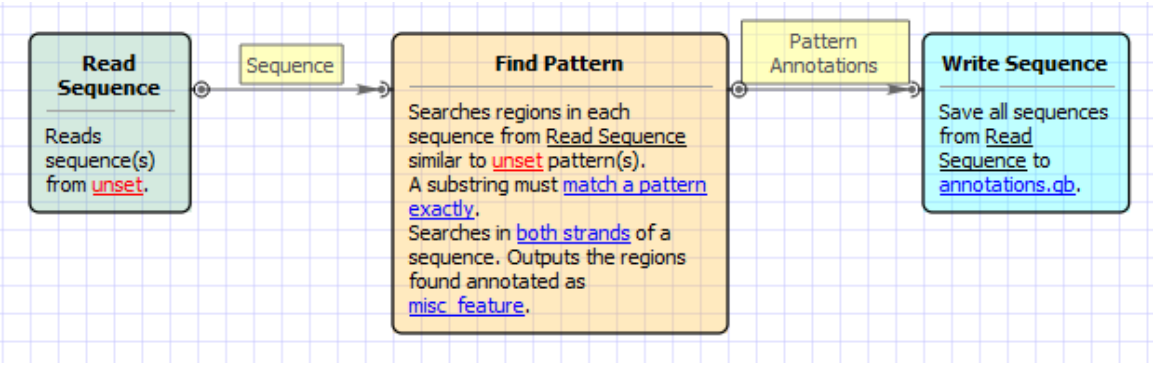
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Find Patterns" can be found in the "Scenarios" section of the Workflow Designer samples.

Workflow Image

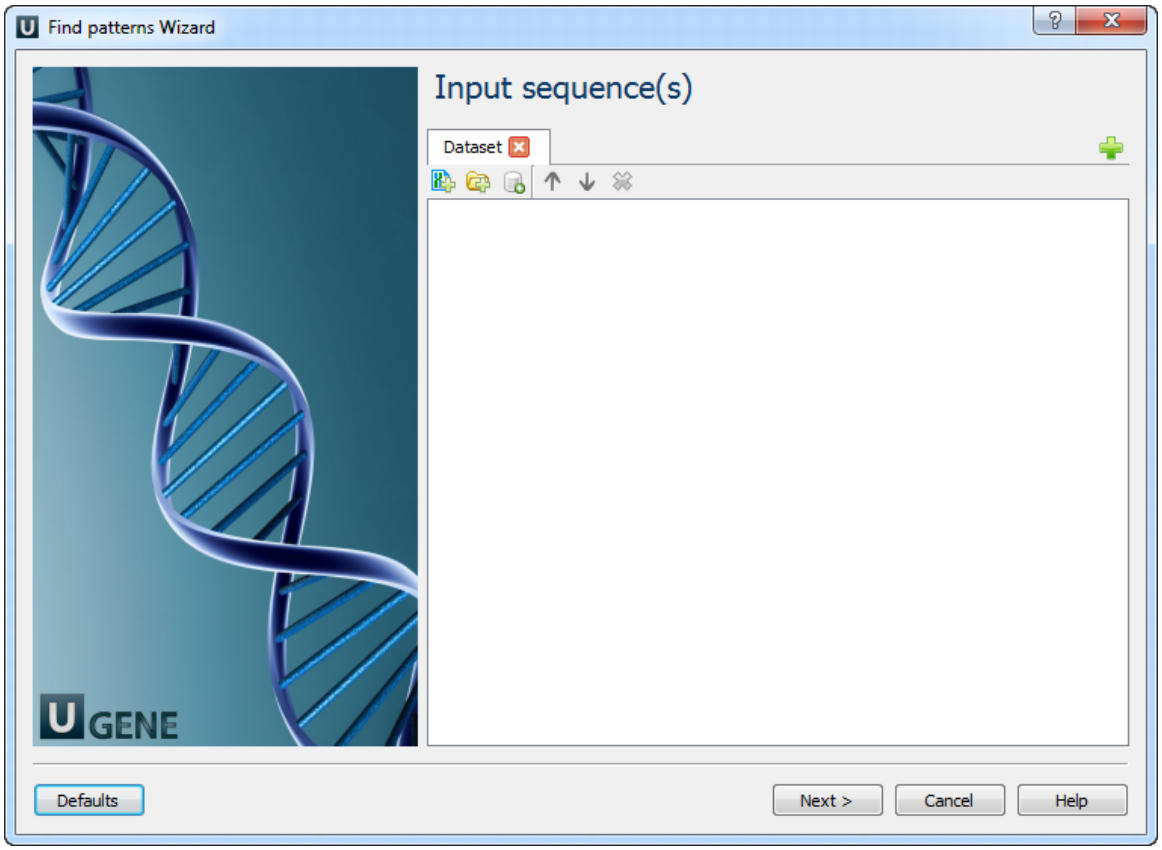
The workflow looks as follows:



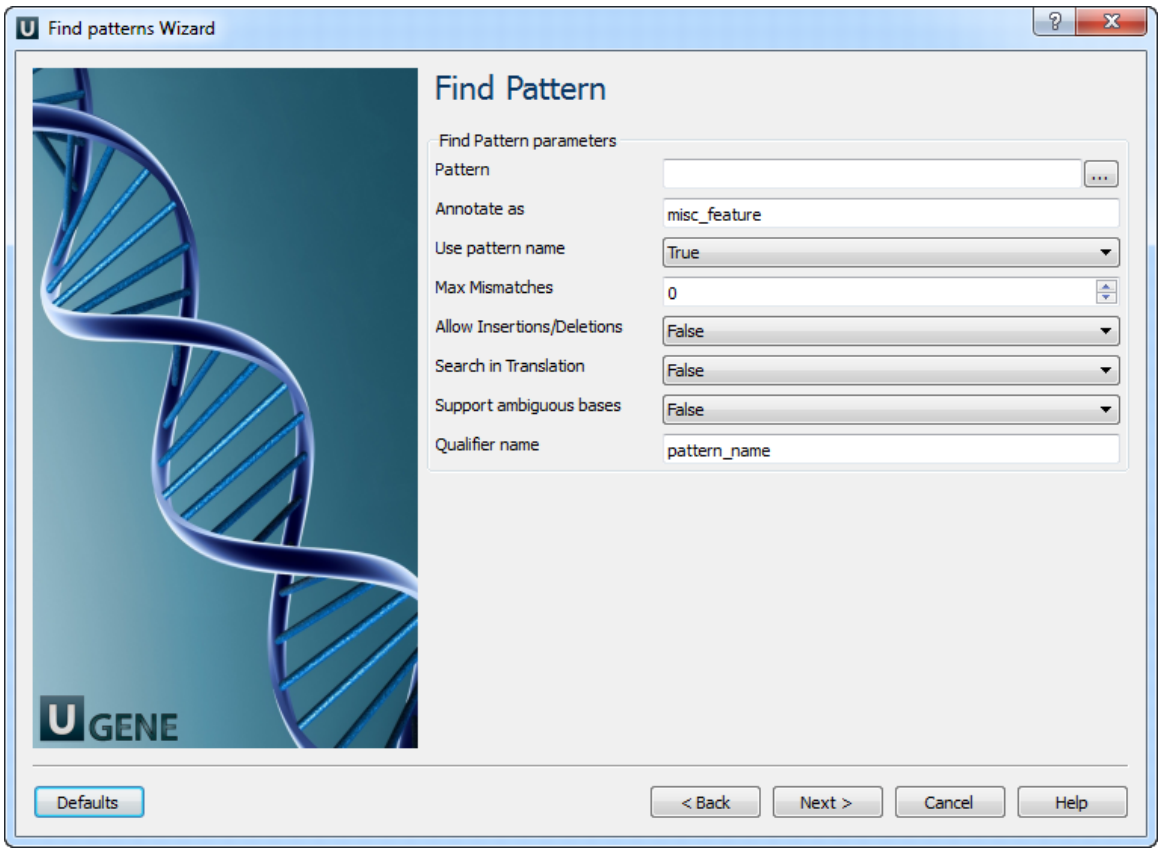
Workflow Wizard

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.

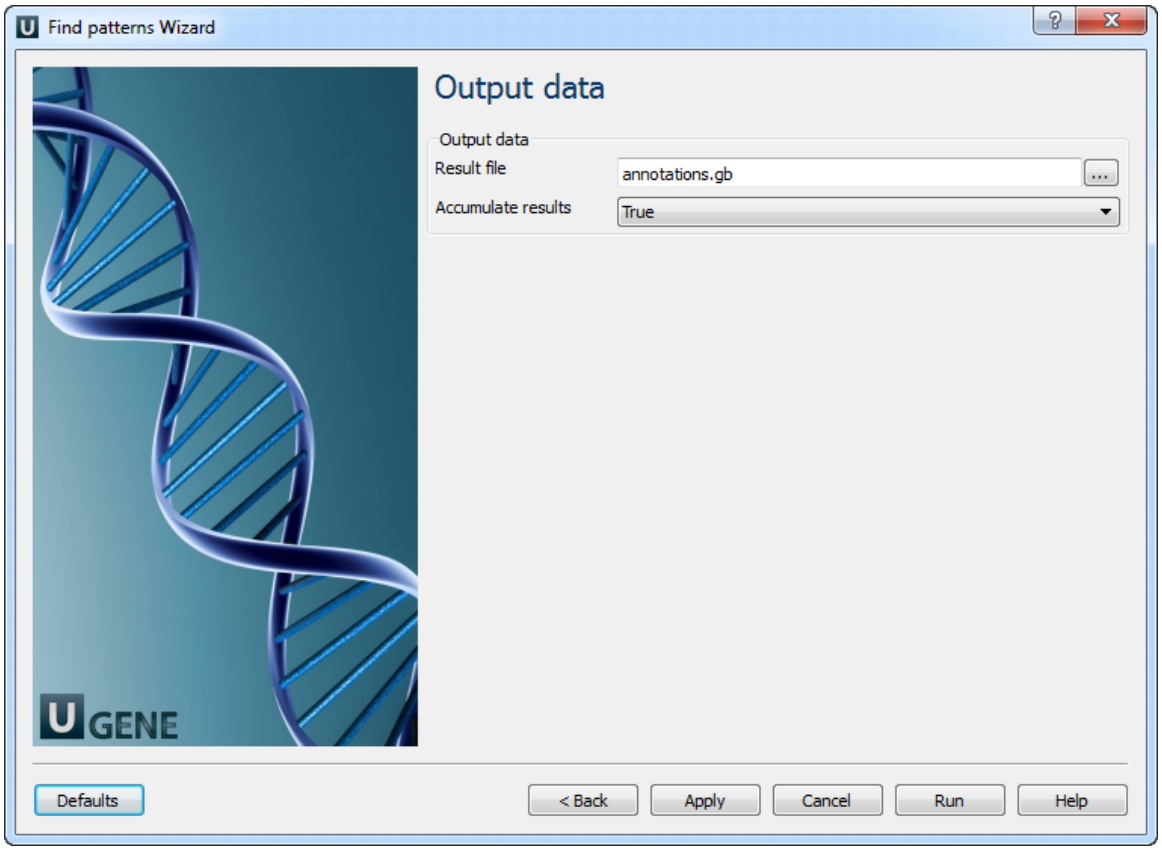


The following parameters are available:

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/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.



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.

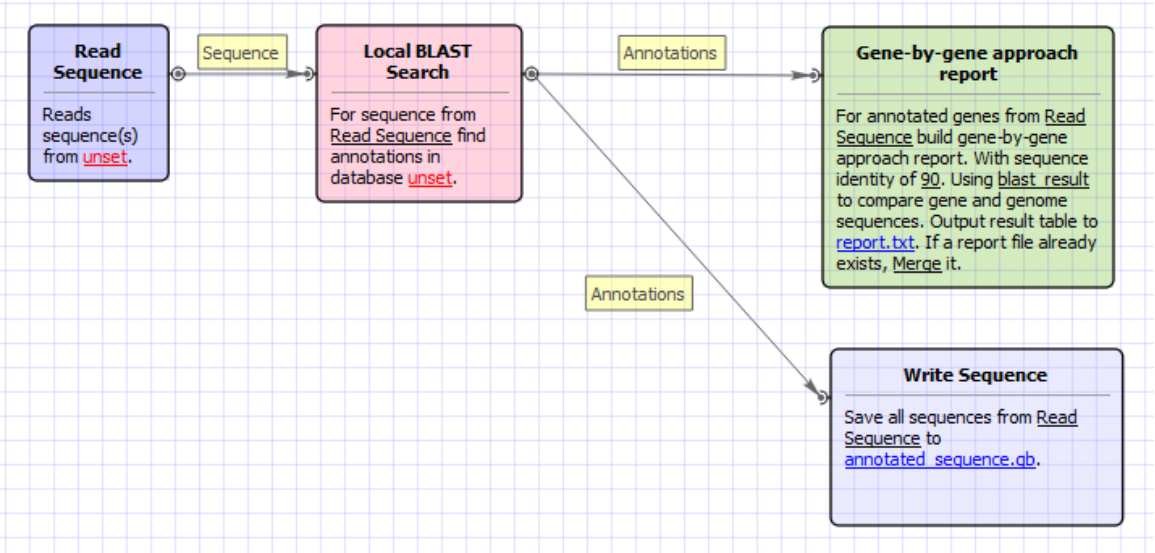
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

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

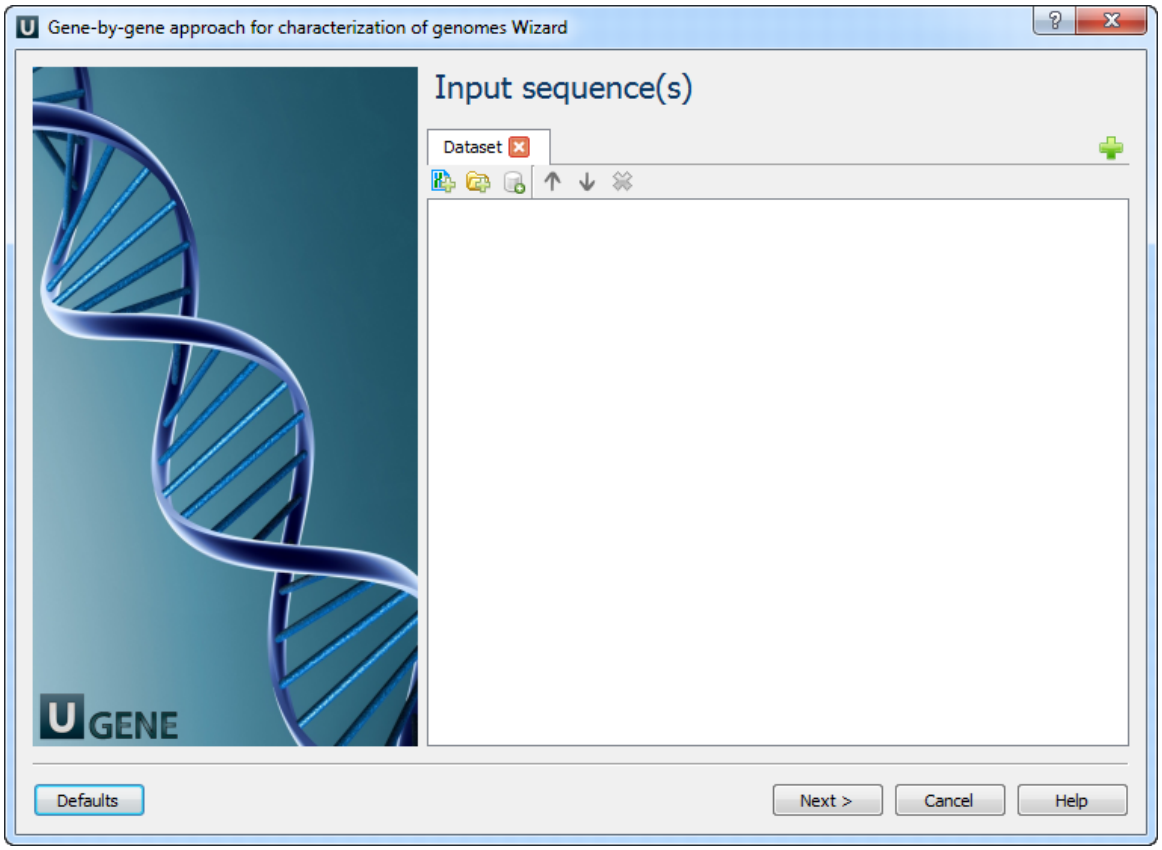
The workflow looks as follows:



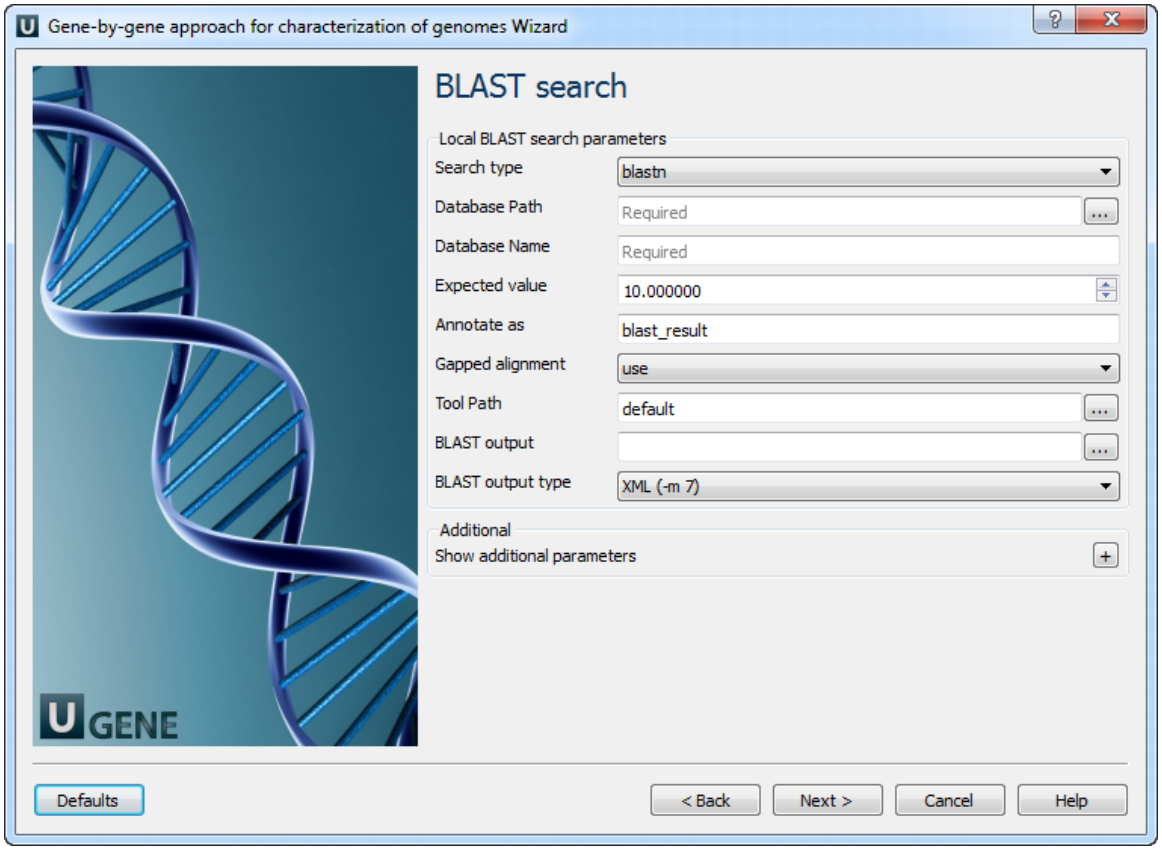
Workflow Wizard

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.

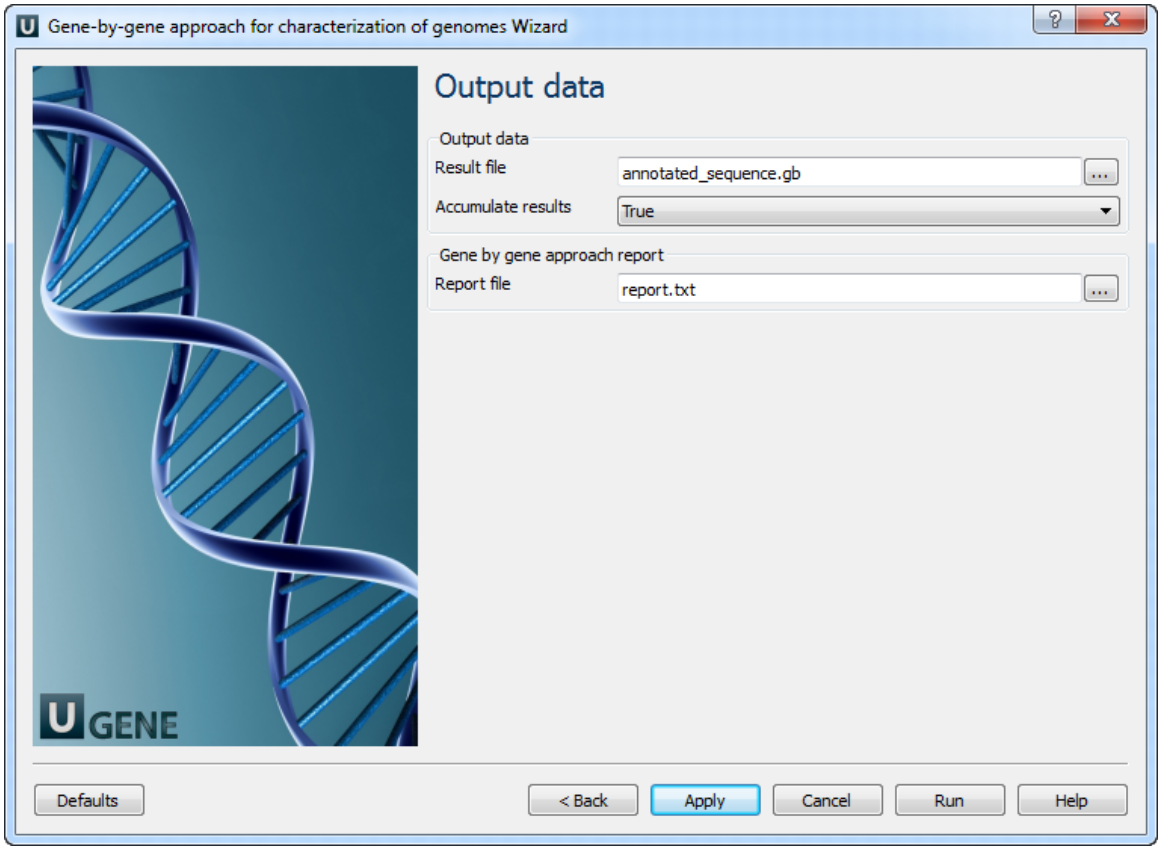


The following parameters are available:

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.

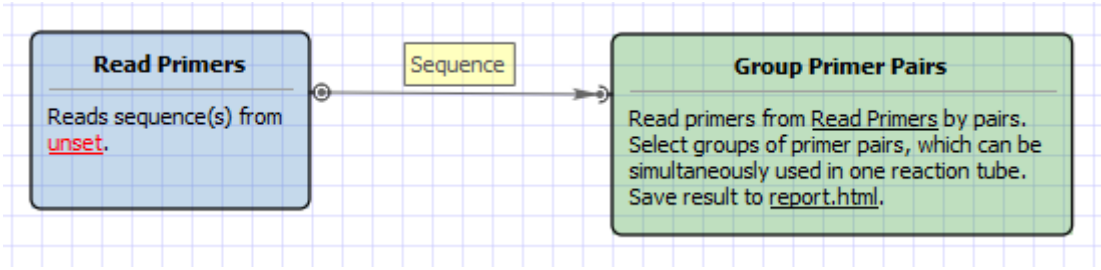
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Group Primer Pairs" can be found in the "Scenarios" section of the Workflow Designer samples.

Workflow Image

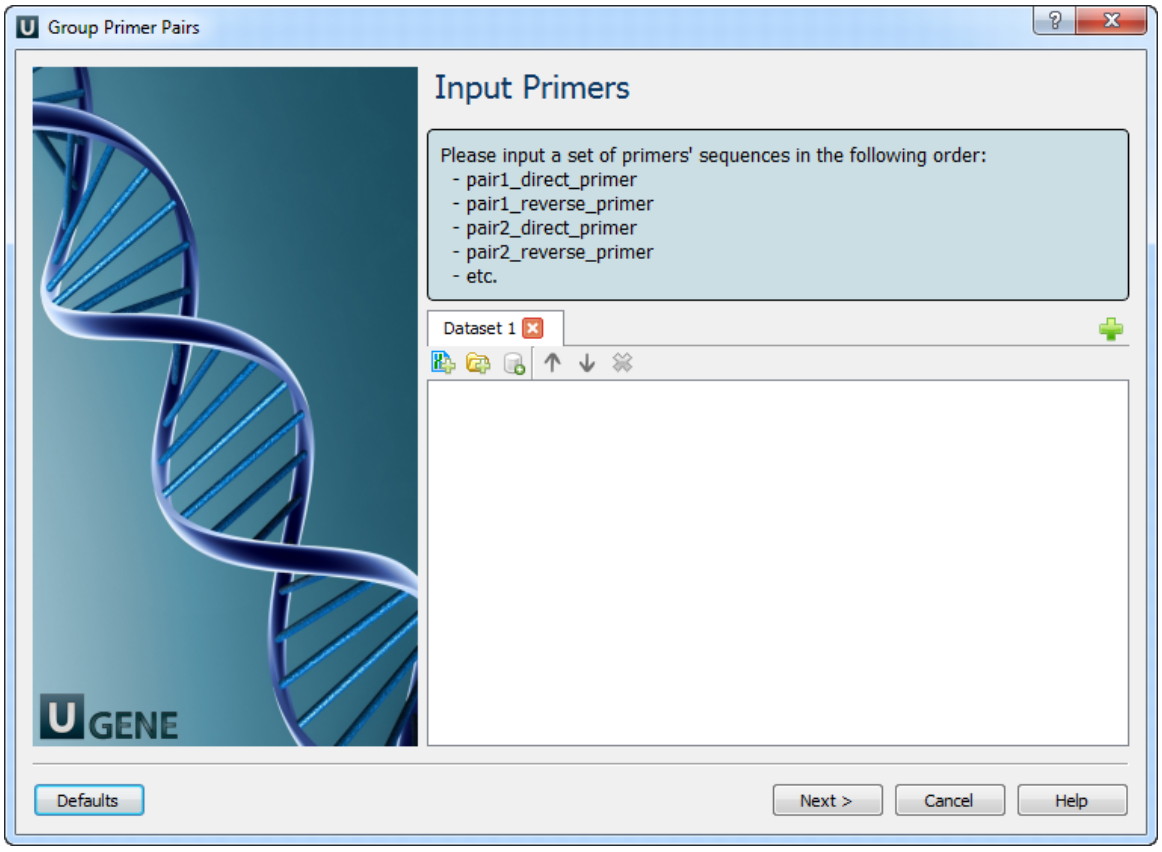
The workflow looks as follows:



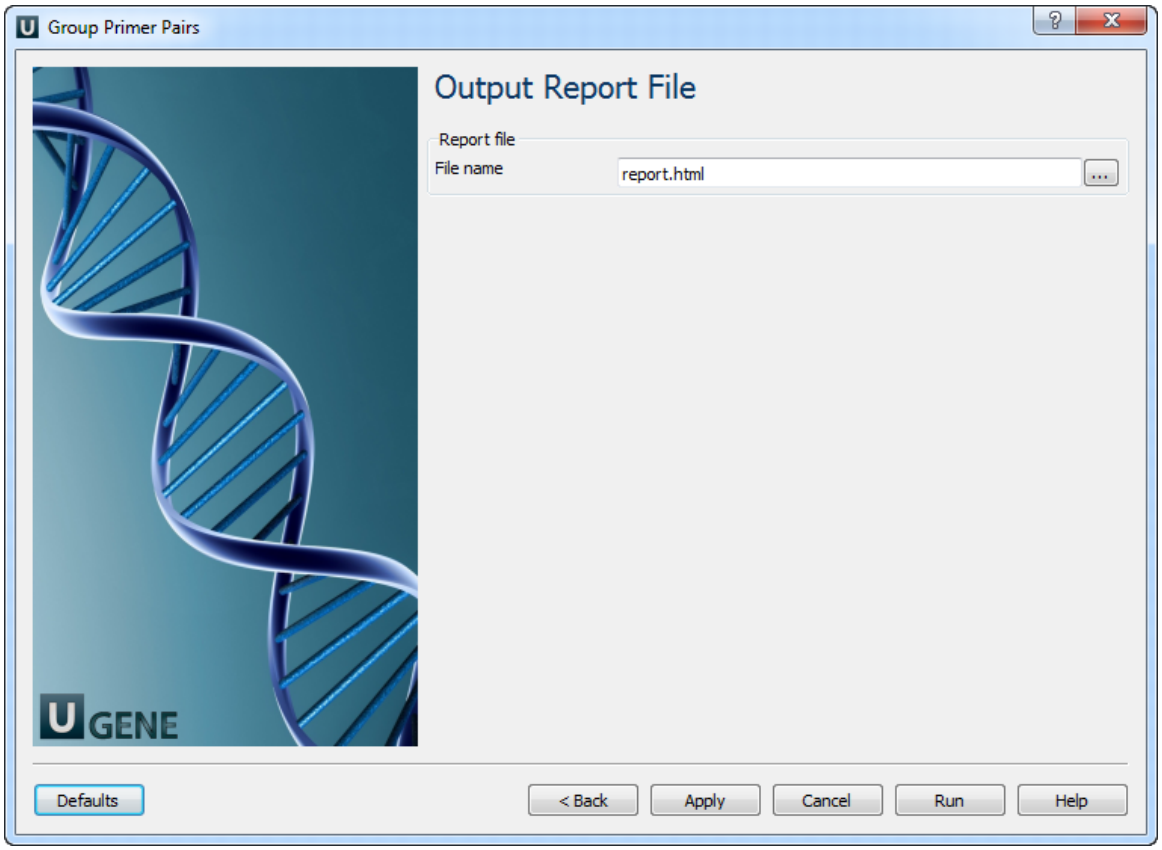
Workflow Wizard

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.

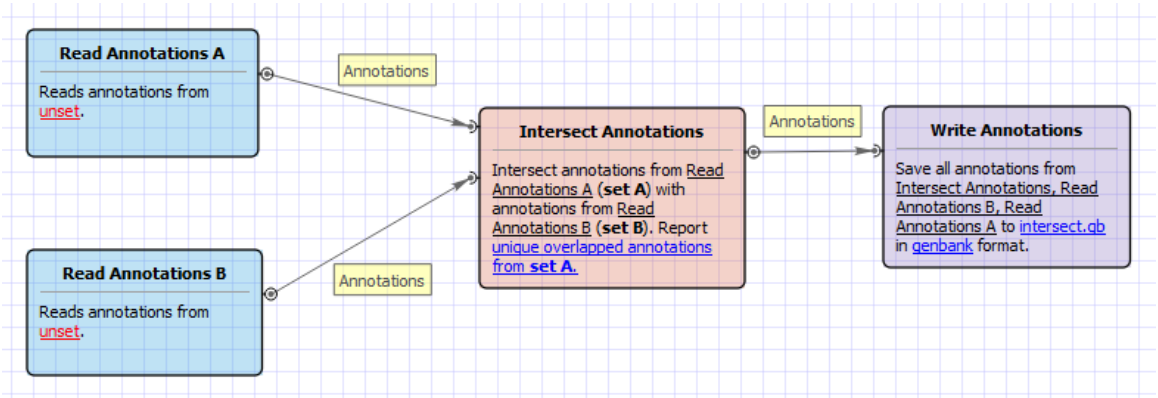
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Intersect Annotations" can be found in the "Scenarios" section of the Workflow Designer samples.

Workflow Image

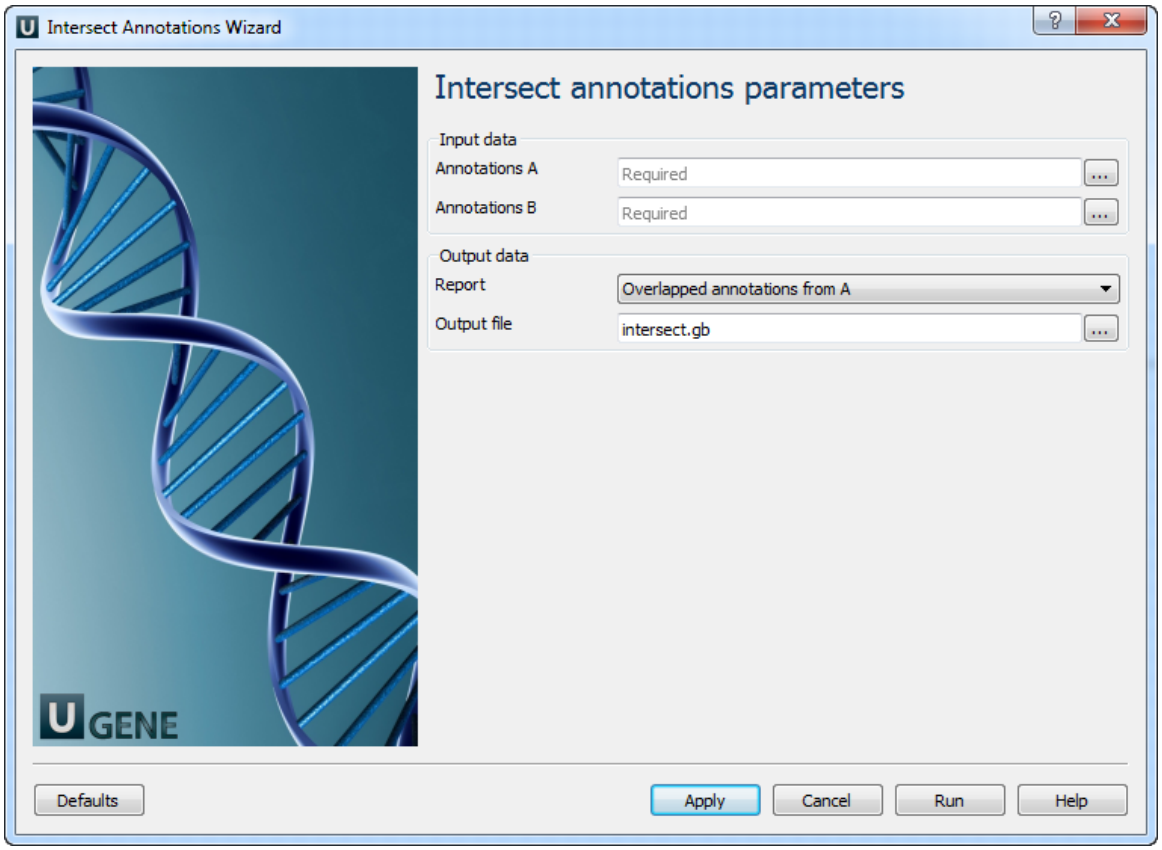
The opened workflow looks as follows:



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.



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.

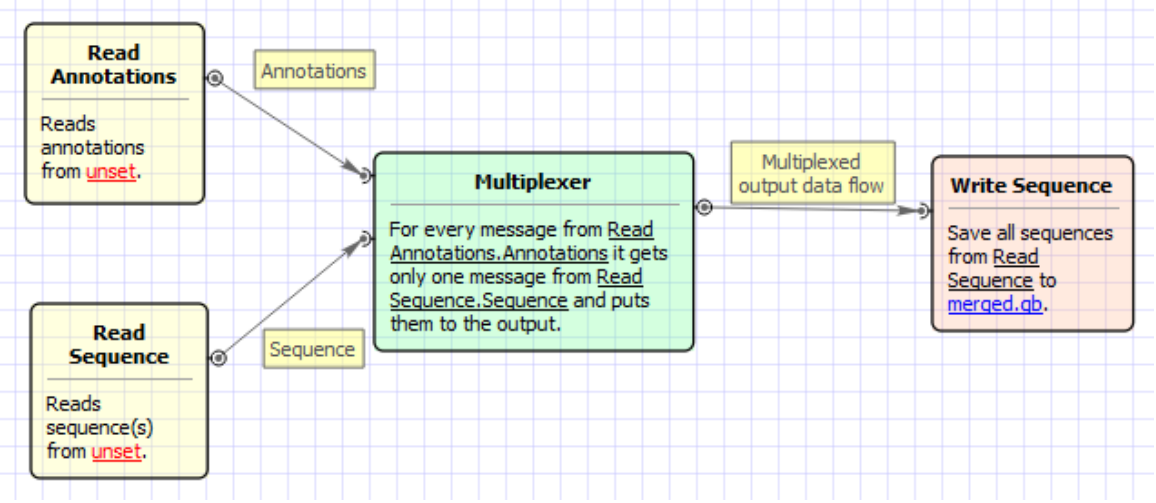
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Merge Sequences and Annotations" can be found in the "Scenarios" section of the Workflow Designer samples.

Workflow Image

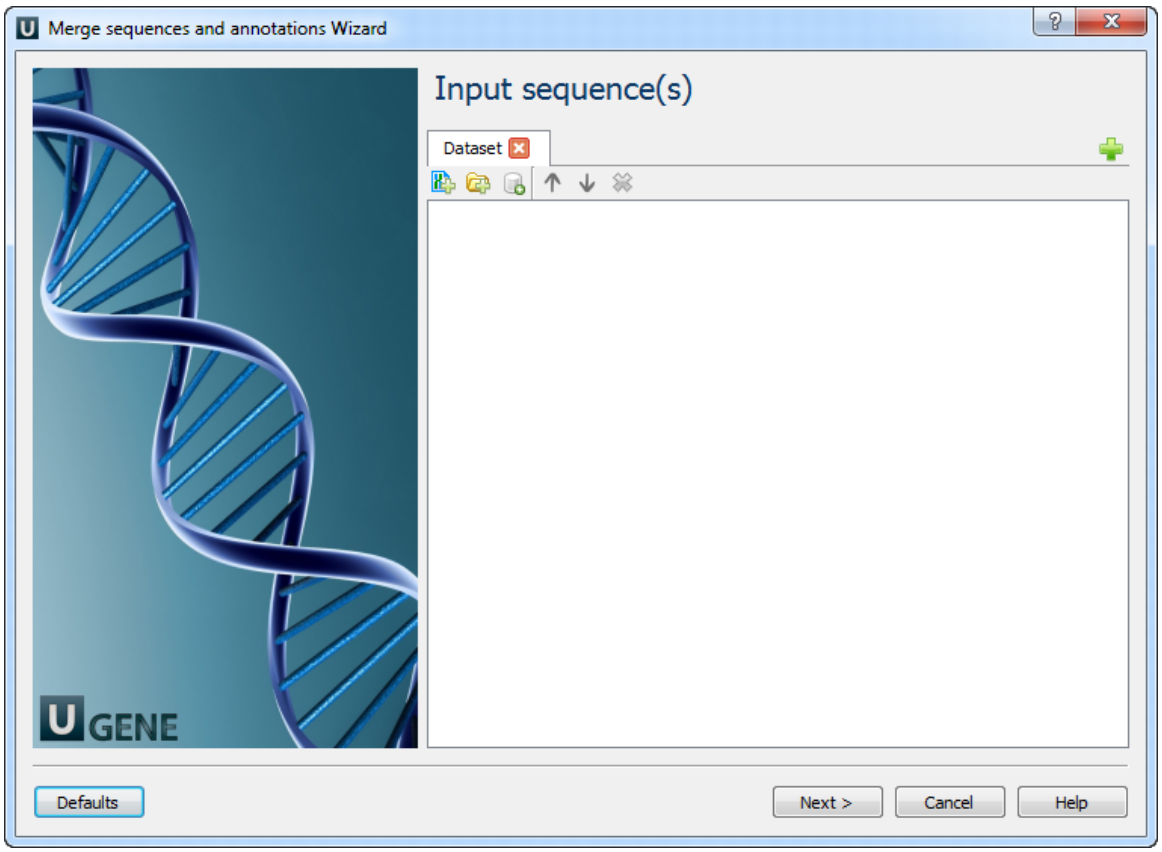
The workflow looks as follows:



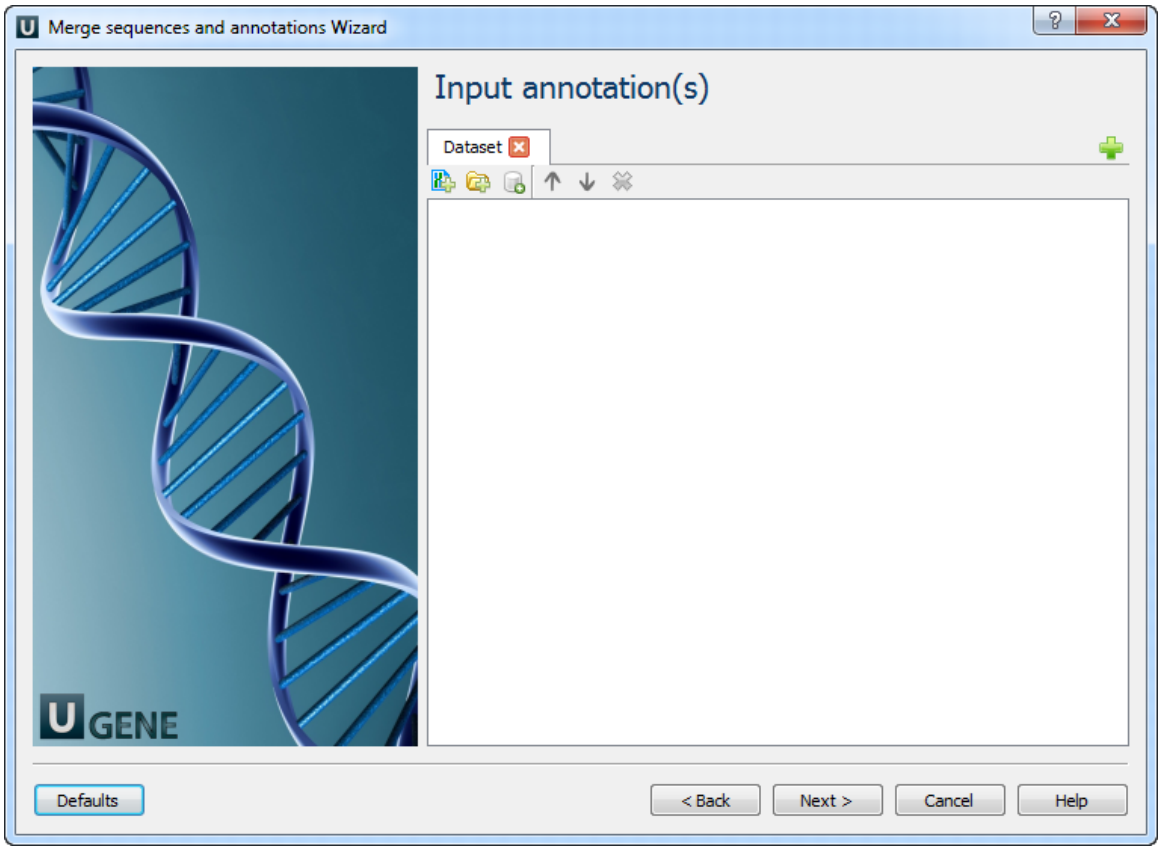
Workflow Wizard

The wizard has 3 pages.

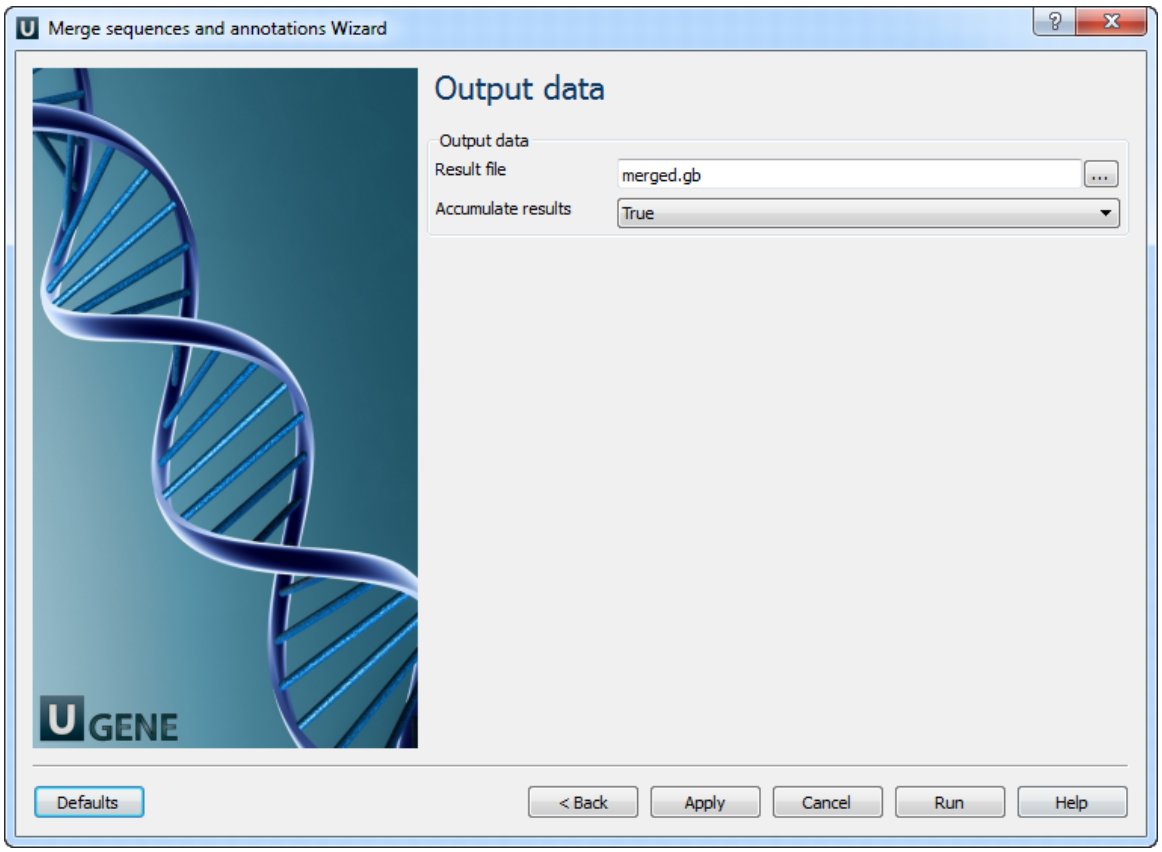
1. Input sequence(s): On this page you must input sequence(s).



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

This workflow simulates the PCR process.

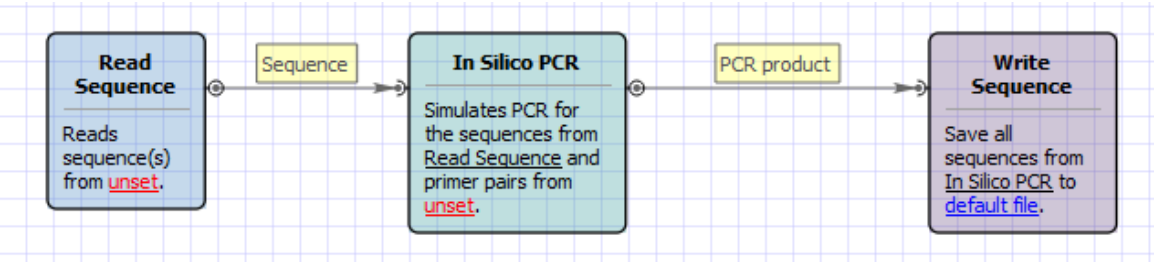
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "In Silico PCR" can be found in the "Scenarios" section of the Workflow Designer samples.

Workflow Image

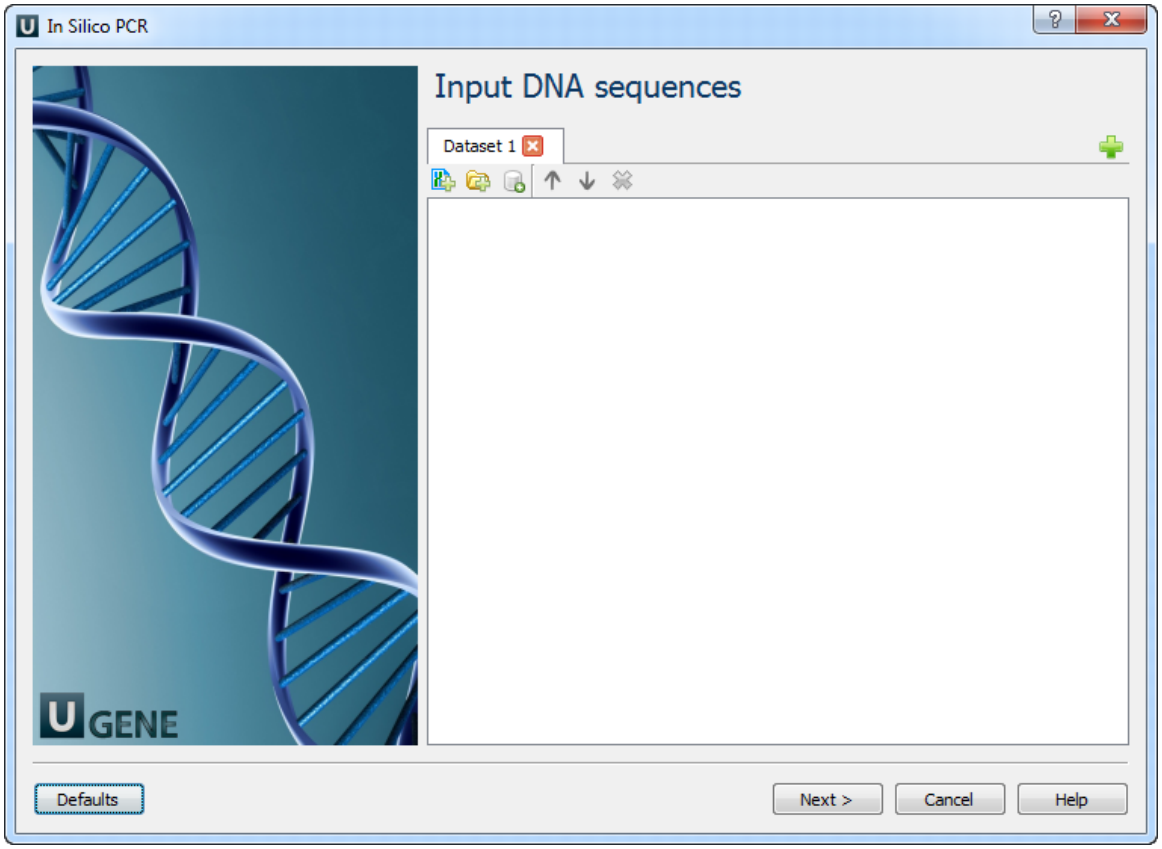
The opened workflow looks as follows:



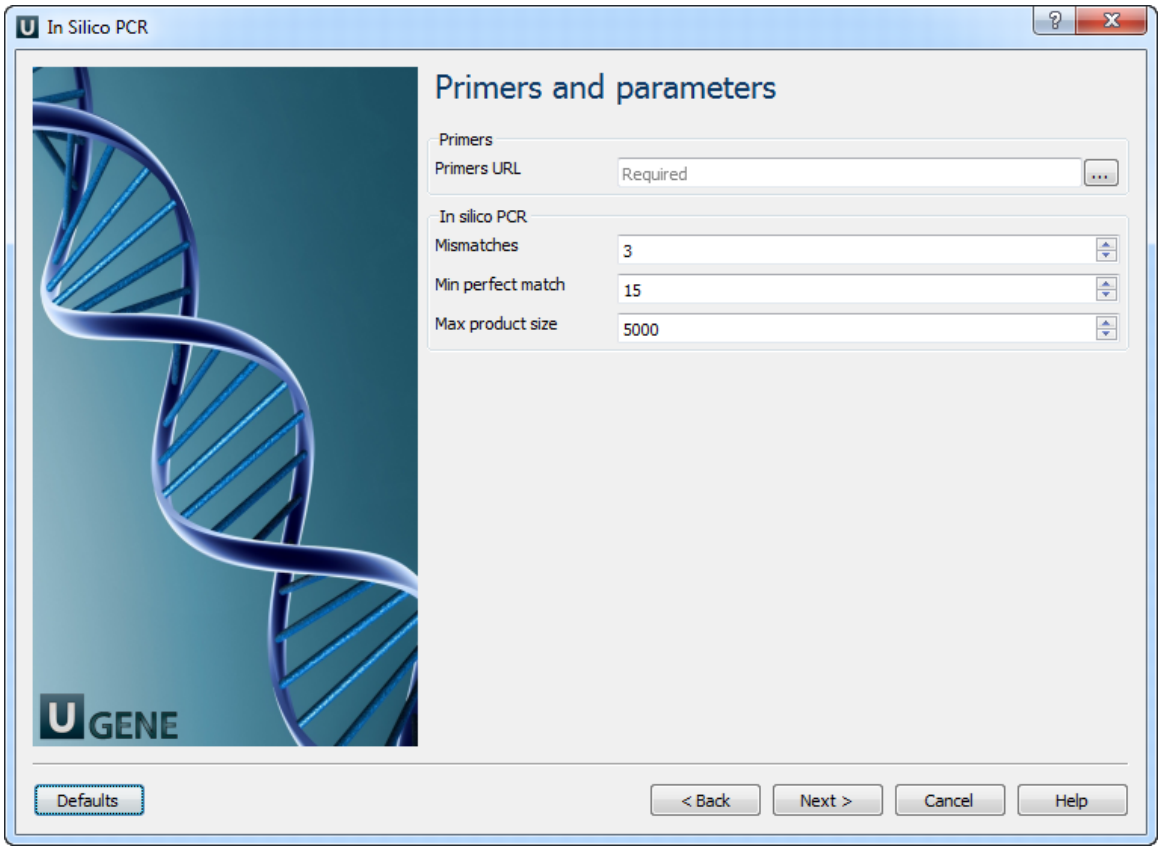
Workflow Wizard

The wizard has 3 pages.

1. Input DNA Sequences: On this page you must input DNA sequences.



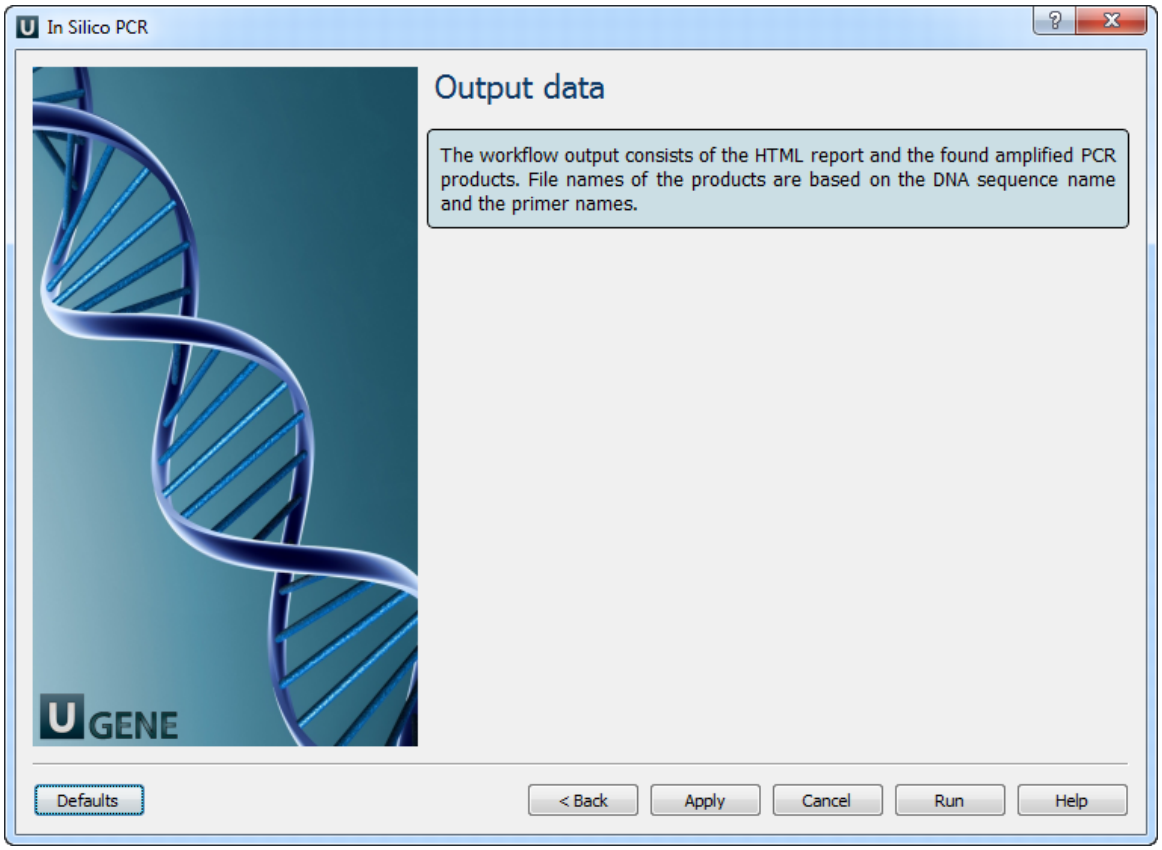
2. Primers and Parameters: Here you must input *Primers* and you can optionally modify *In Silico PCR* parameters.



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.

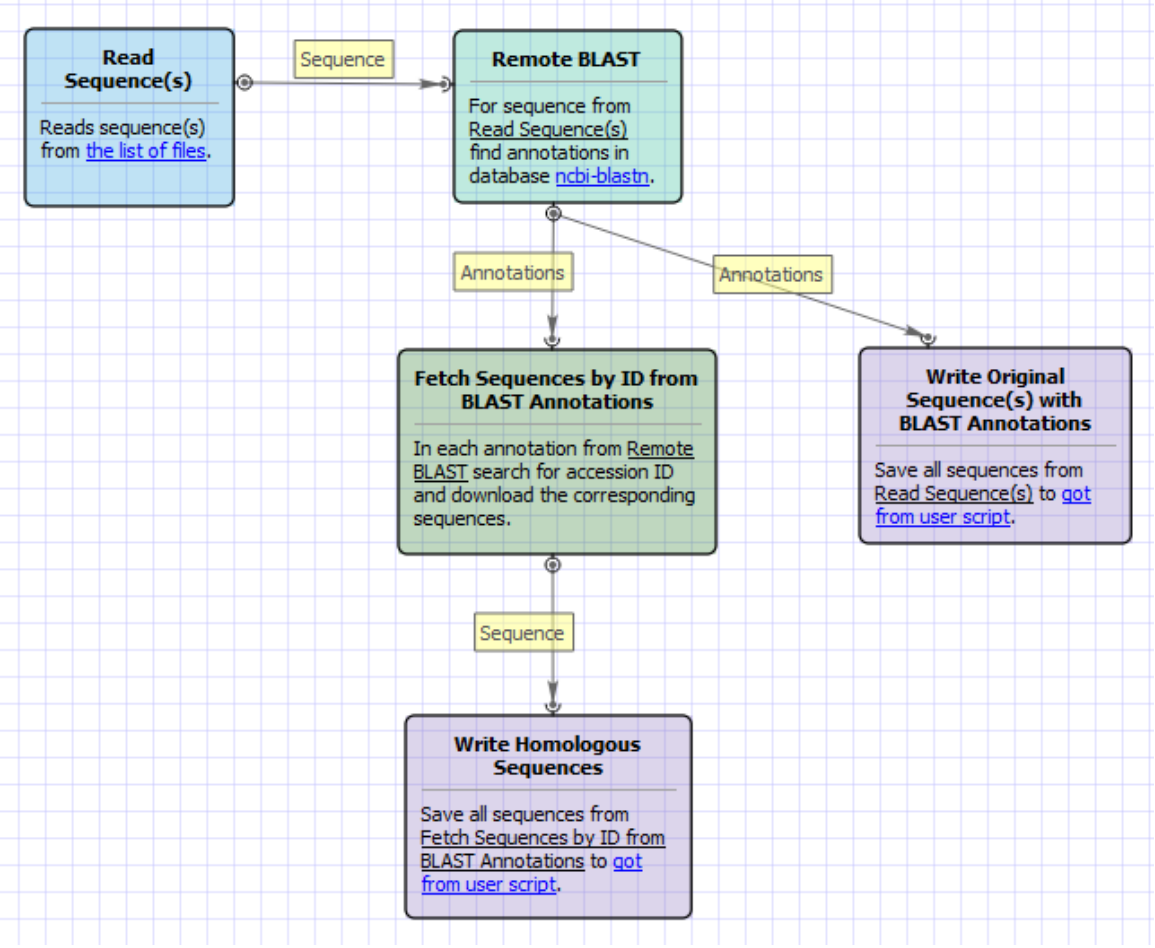
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

Workflow Sample Location

The workflow sample "Remote BLASTing" can be found in the "Scenarios" section of the Workflow Designer samples.

Workflow Image

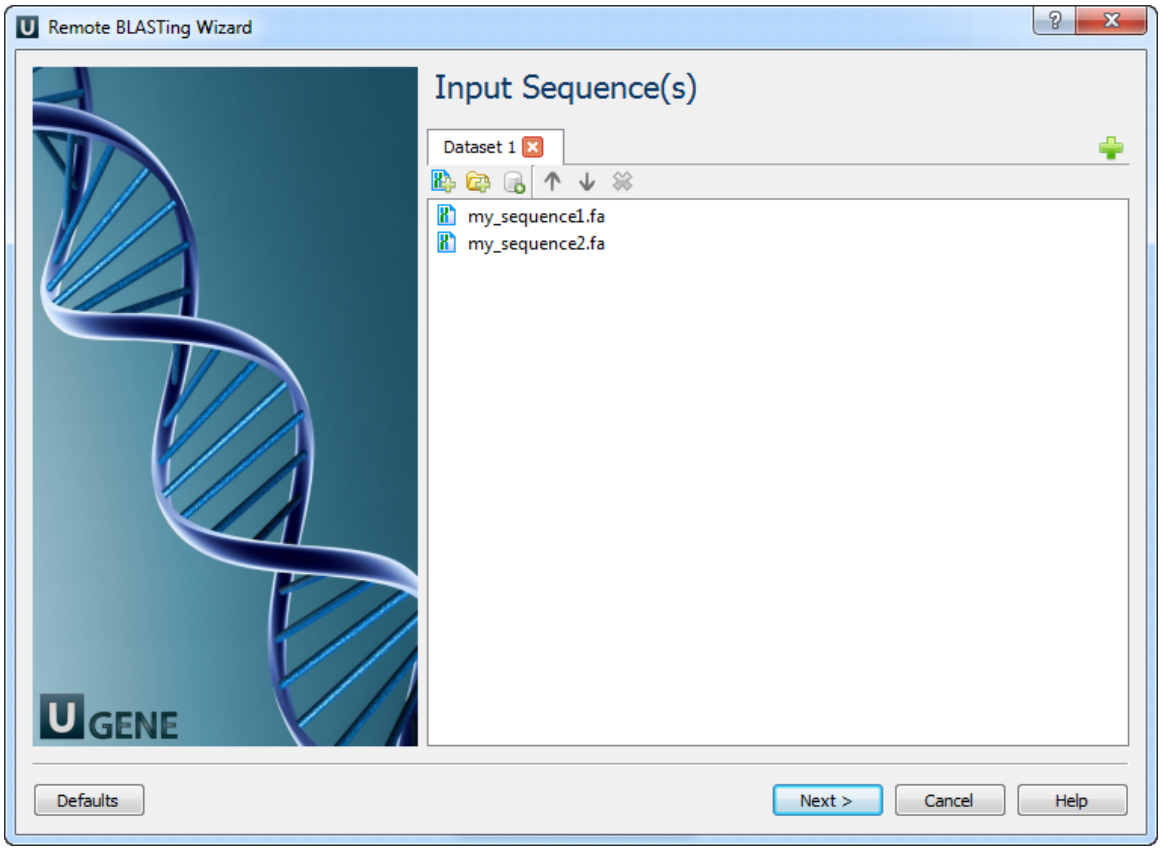
The opened workflow looks as follows:



Workflow Wizard

The wizard has 3 pages.

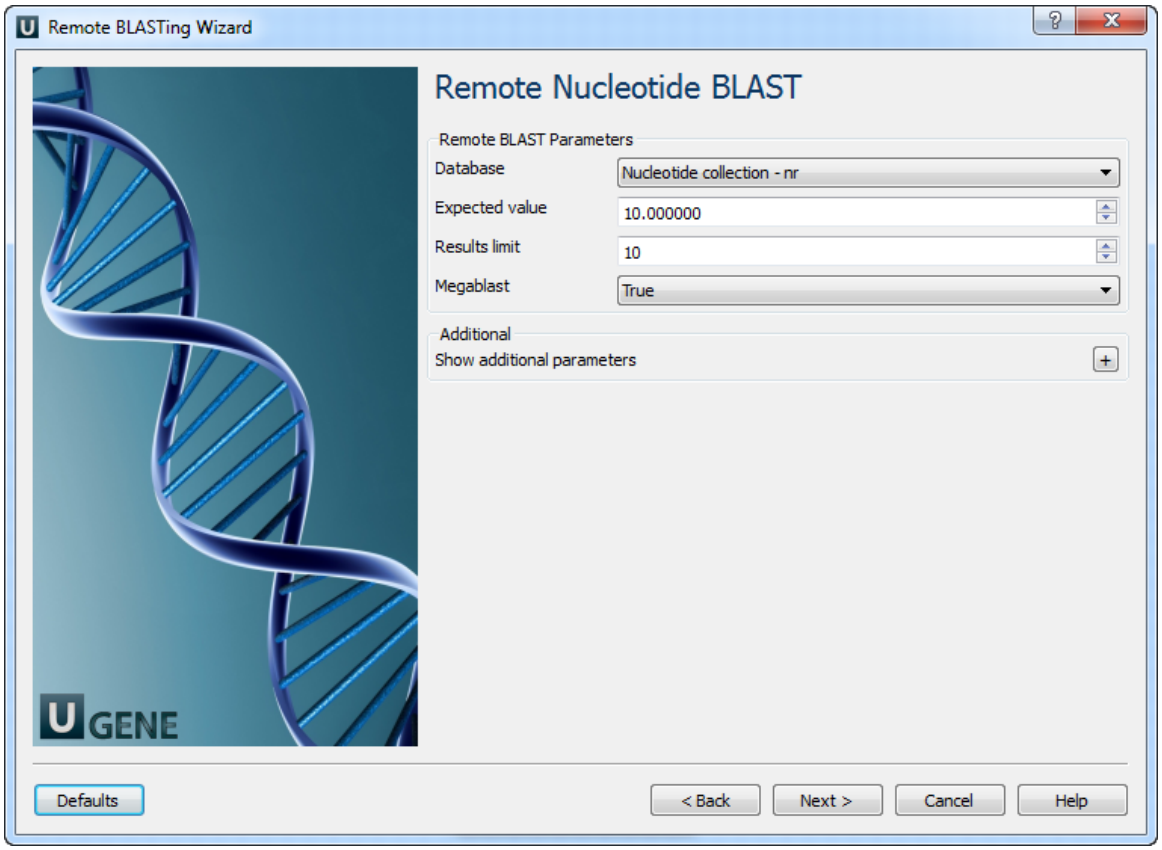
1. Input Sequence(s) Page: On this page you must input at least one nucleotide sequence.



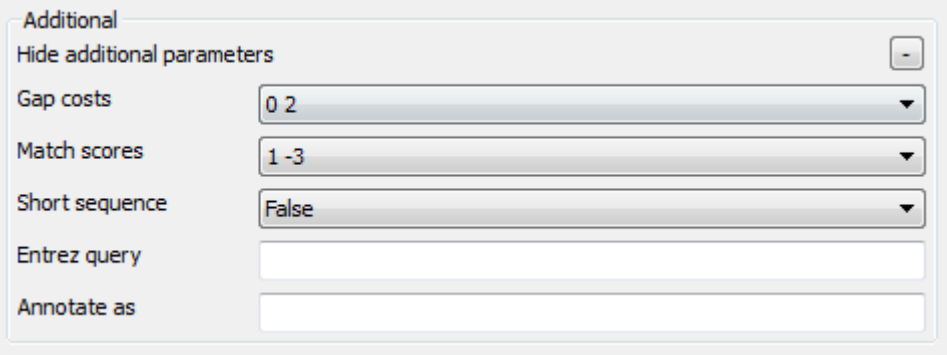
Example Input Data
For example, you can use the following two files as an input to the workflow:

- 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.



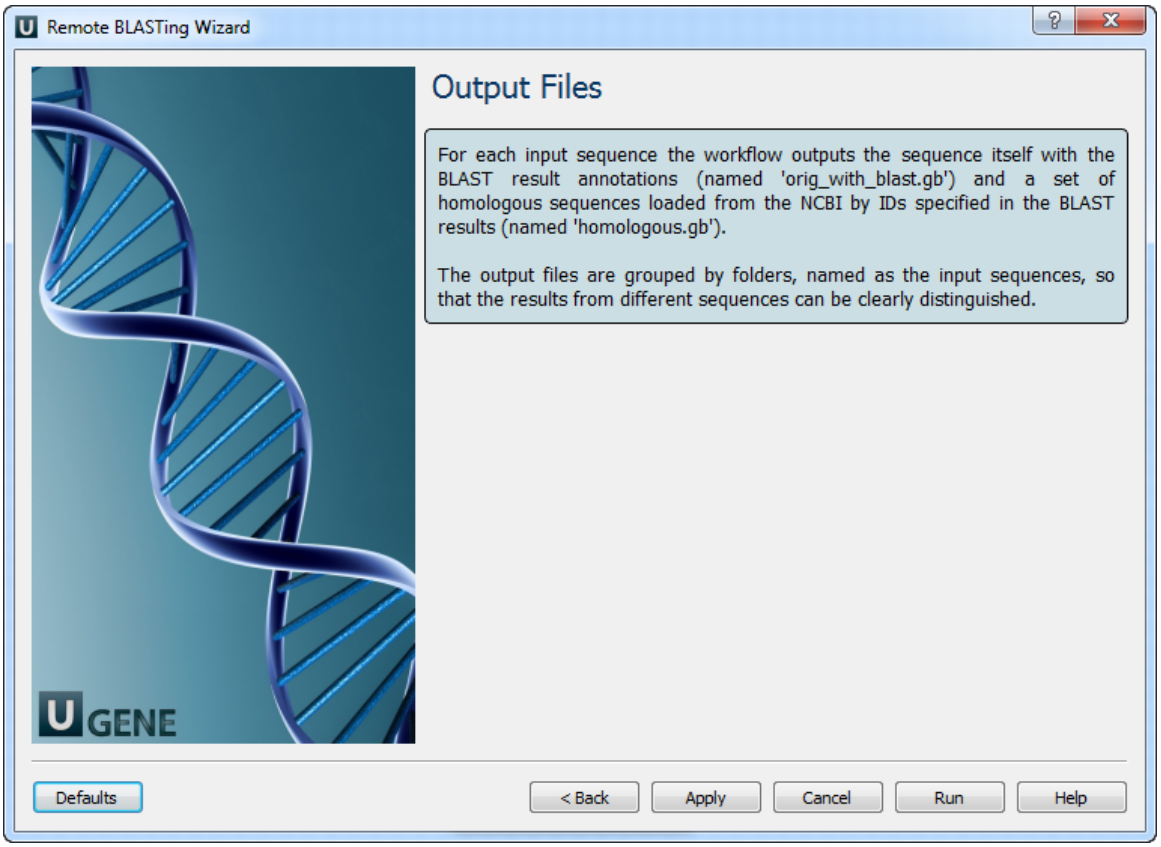
There are also some additional parameters. Description of them can be found in the Remote BLAST workflow element chapter of the documentation.



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.

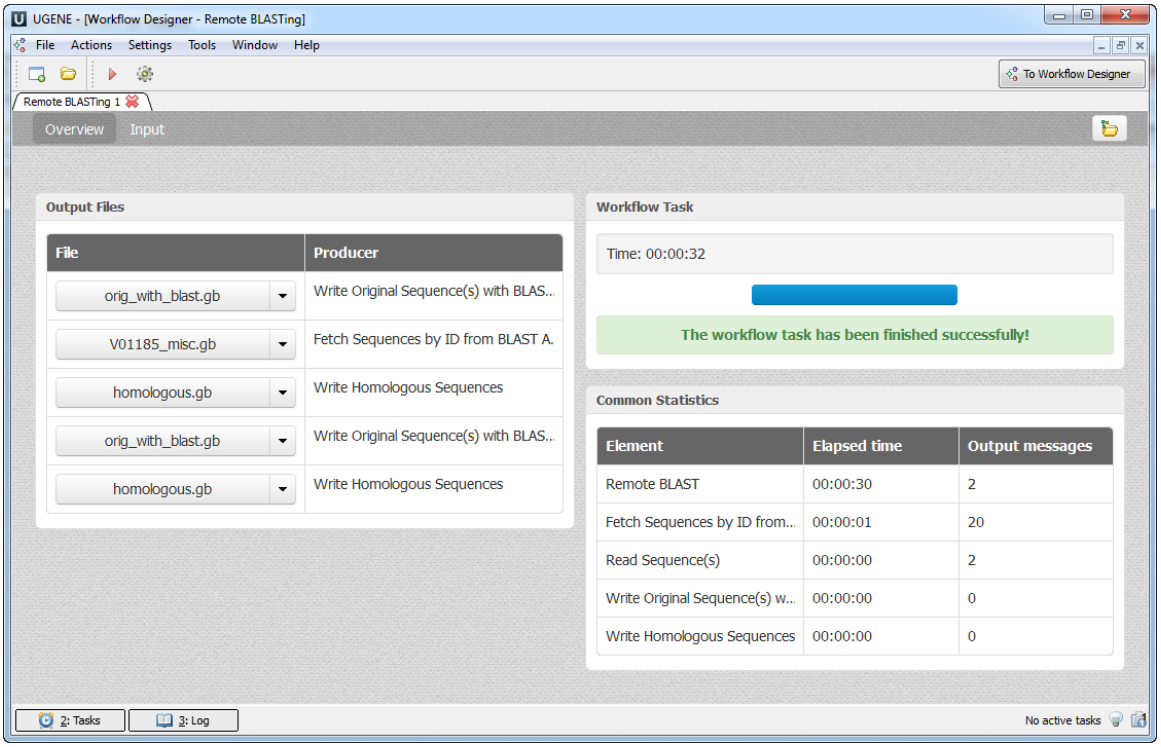
The results on the hard drive are grouped by folders (see below).

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.

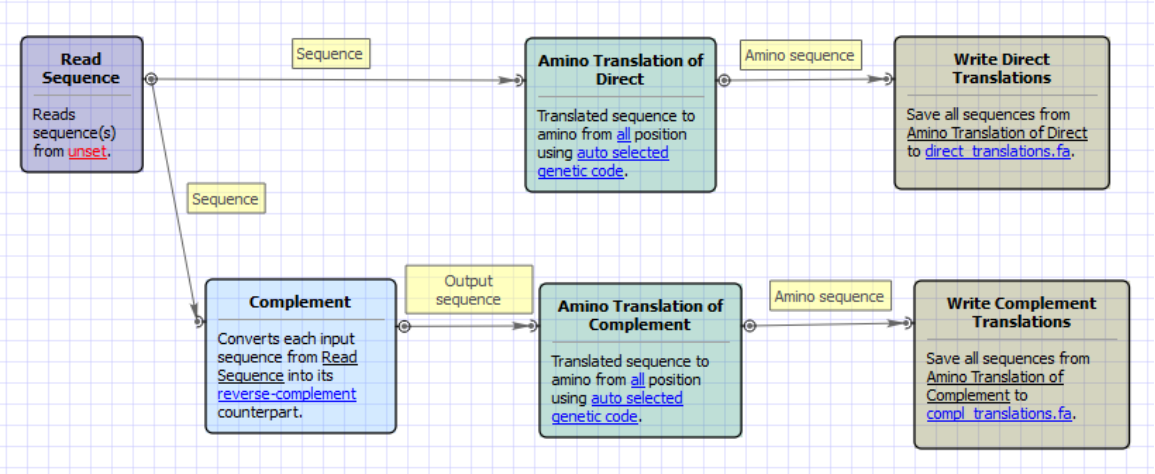
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

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

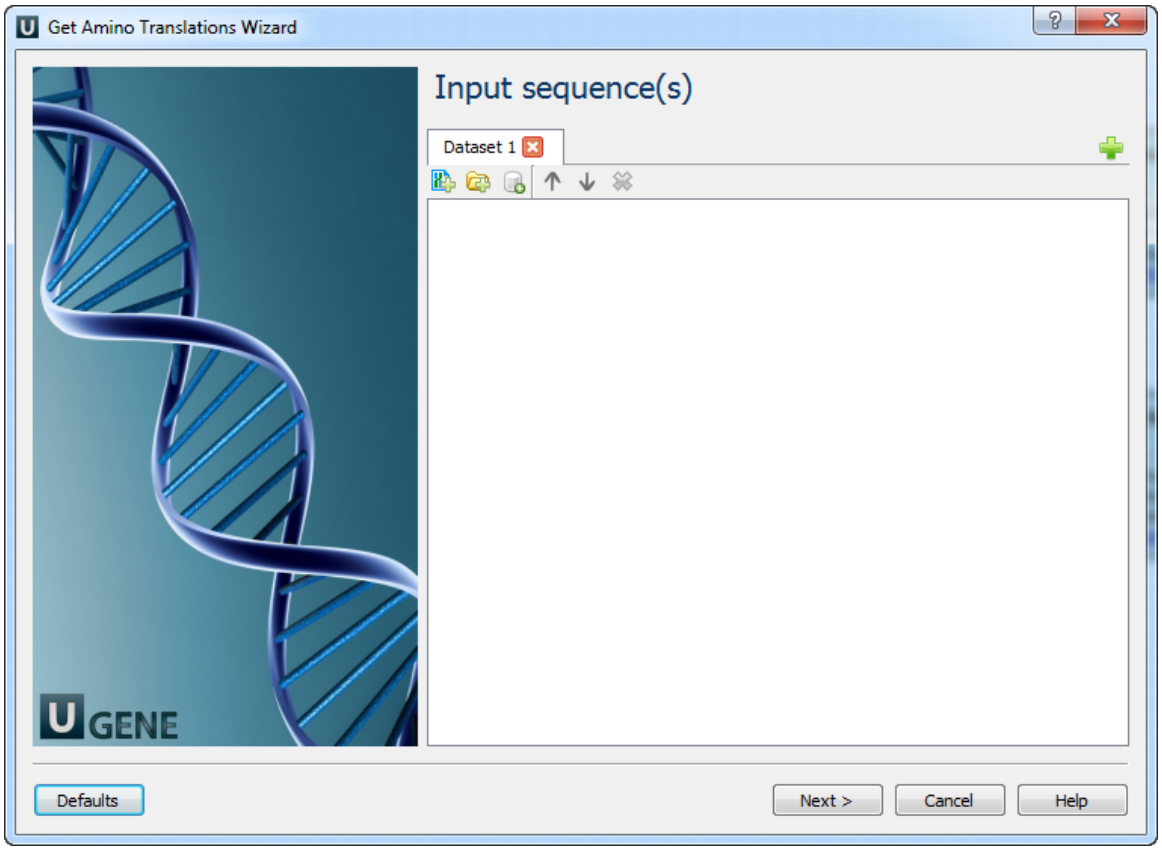
The opened workflow looks as follows:



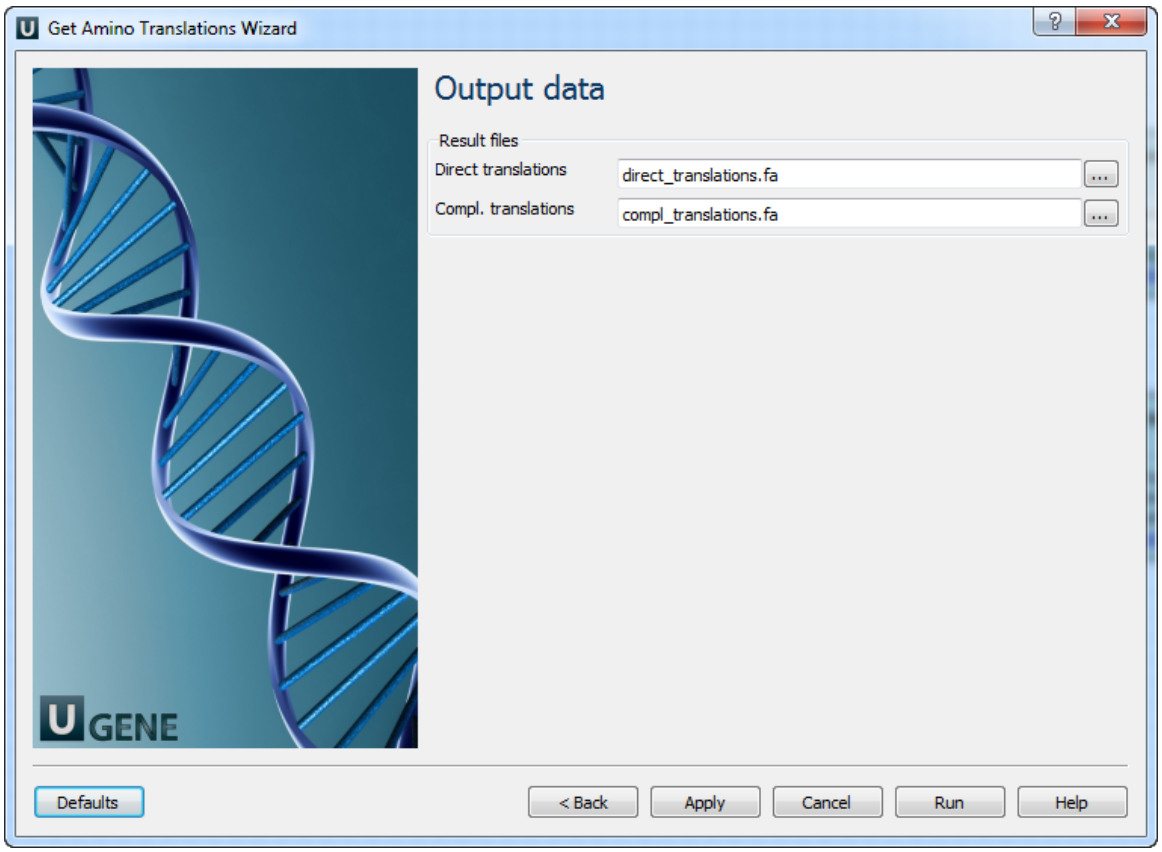
Workflow Wizard

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.

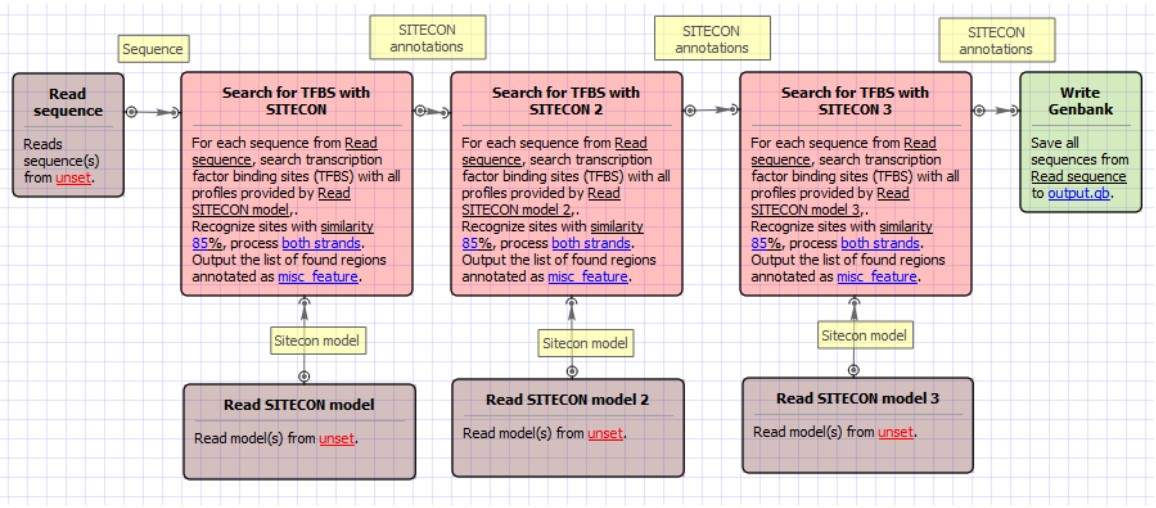
How to Use This Sample
If you haven't used the workflow samples in UGENE before, look at the "How to Use Sample Workflows" section of the documentation.

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

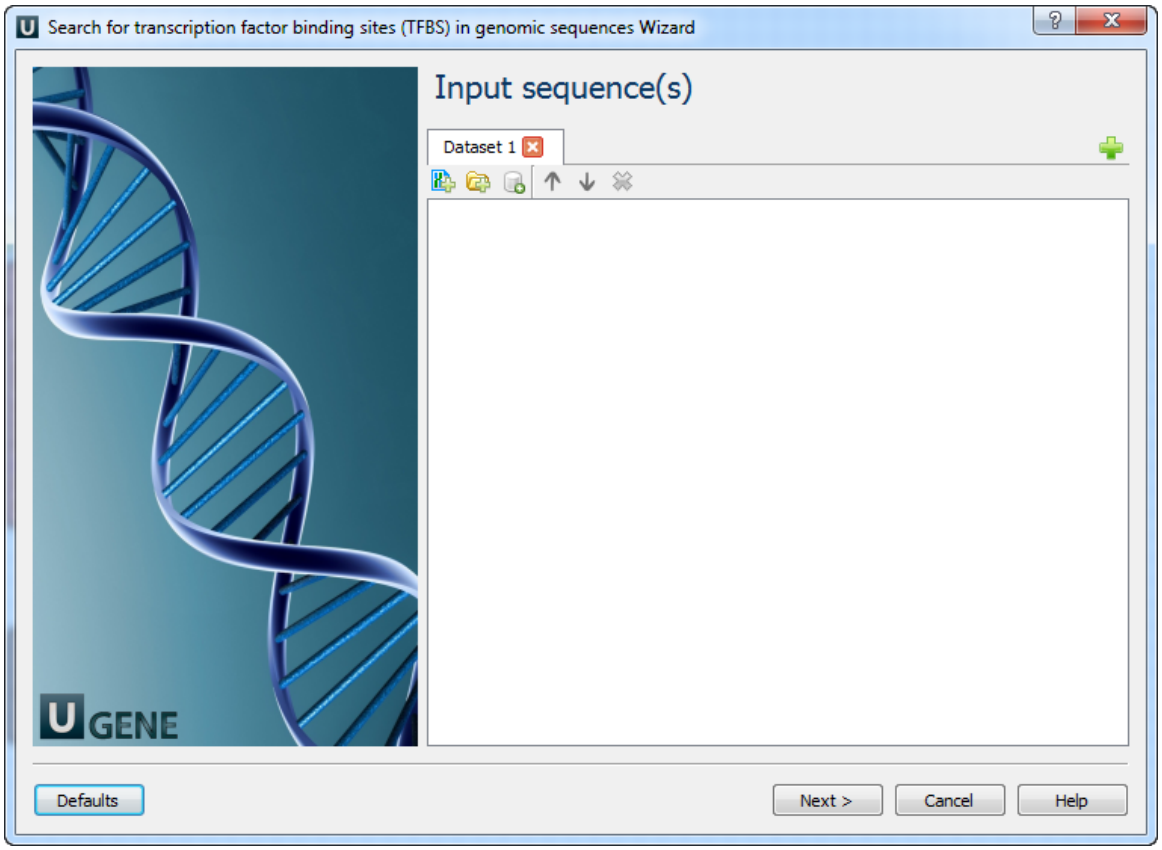
The workflow looks as follows:



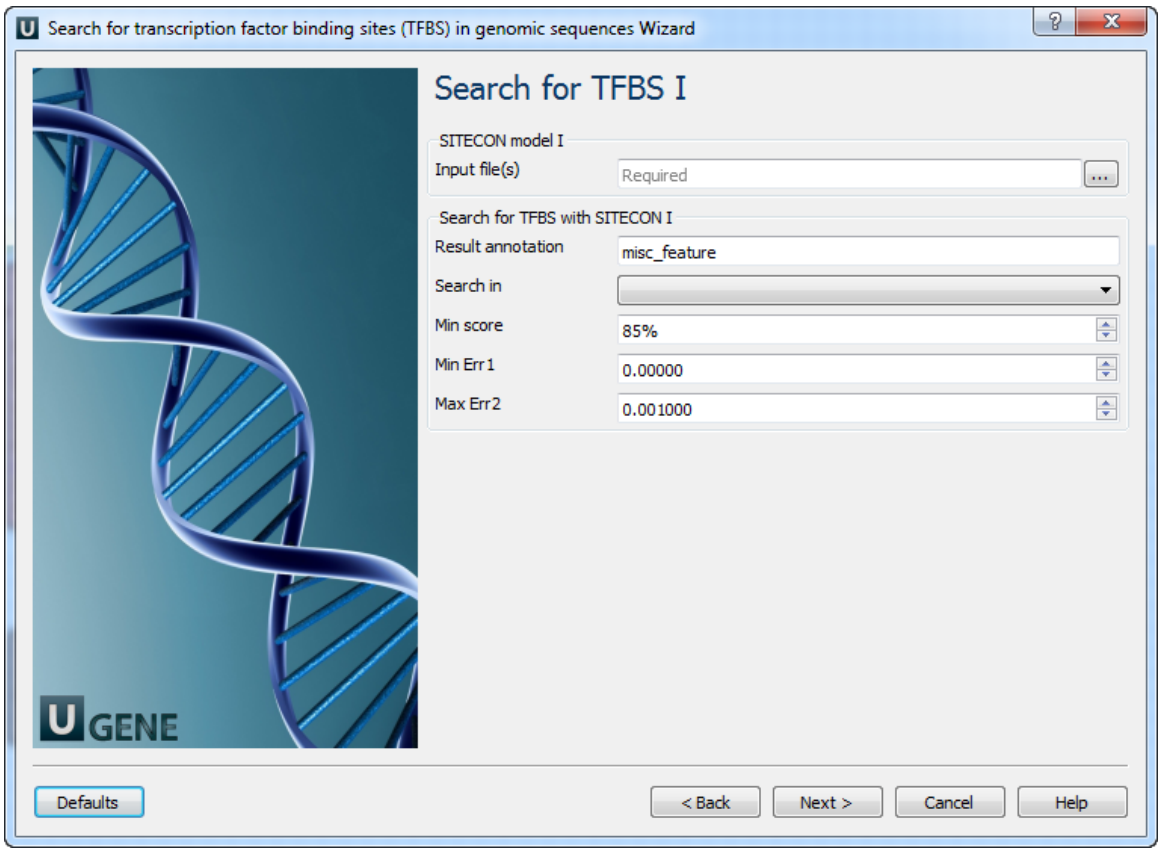
Workflow Wizard

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.

