



Unipro UGENE Workflow Designer Manual

Version 1.10.4

March 11, 2012



Contents

1	About Unipro	6
1.1	Contacts	6
2	About UGENE	7
2.1	Key Features	7
2.2	User Interface	7
2.3	High Performance Computing	8
2.4	Cooperation	8
3	About the Workflow Designer	9
4	Introduction	10
4.1	Launching Workflow Designer	10
4.2	Workflow Designer Window Components	10
4.3	Workflow Schema Elements and Connections	11
4.4	Managing Parameters	12
4.5	UGENE Components and Workflow Designer	13
4.5.1	Task View, Notifications and Log View	13
4.5.2	Actions Menu	14
4.5.3	Toolbar	14
4.5.4	Context Menus	15
4.5.5	Application Settings	15
4.6	How to Create and Run Schema	16
5	Manipulating Element	18
5.1	Adding Element	18
5.2	Copying Element	18
5.3	Pasting Element	18
5.4	Cutting Element	19
5.5	Deleting Element	19
5.6	Selecting All Elements on Scene	19
6	Manipulating Schema	20
6.1	Creating New Schema	20
6.2	Loading Schema	20
6.3	Saving Schema	20

6.4	Exporting Schema	21
6.5	Validating Schema	21
6.6	Running Schema	21
6.7	Stopping Schema	22
7	Changing Appearance	23
8	Custom Elements with Scripts	24
8.1	Functions supported for <i>Multiple alignment</i> data	25
8.2	Functions supported for <i>Sequence</i> data	26
8.3	Functions supported for <i>Set of annotations</i> data	26
8.4	Common function	26
9	Custom Elements with Command Line Tools	27
9.1	Creating Element	27
9.2	Editing Element	29
9.3	Adding Existent Element	29
9.4	Removing Element	30
10	Using Script to Set Parameter Value	31
11	Using Iterations	32
12	Running Schema Locally or Remotely	33
13	Running Schema from the Command Line	34
14	Workflow Schema File Format	36
14.1	Header	36
14.2	Body	36
14.2.1	Elements	36
14.2.2	Dataflow	37
14.2.3	Iterations	37
14.2.4	Metainformation	38
15	Workflow Elements	41
15.1	Data Readers	42
15.1.1	Read Alignment Element	42
15.1.2	Read From Remote Database Element	42
15.1.3	Read Plain Text Element	43

15.1.4	Read Sequence Element	44
15.2	Data Writers	46
15.2.1	Write Alignment Element	46
15.2.2	Write ClustalW Element	47
15.2.3	Write FASTA Element	48
15.2.4	Write FASTQ Element	49
15.2.5	Write Genbank Element	50
15.2.6	Write Plain Text Element	51
15.2.7	Write Sequence Element	52
15.2.8	Write Stockholm Element	53
15.3	Basic Analysis	55
15.3.1	Analyze with UQL Element	55
15.3.2	Collocation Search Element	56
15.3.3	Extract Annotated Regions Element	57
15.3.4	Find Repeats Element	59
15.3.5	Find Substrings Element	60
15.3.6	Import PHRED Qualities Element	61
15.3.7	Local BLAST Search Element	62
15.3.8	Local BLAST+ Search Element	64
15.3.9	ORF Marker Element	66
15.3.10	Remote BLAST Element	67
15.3.11	Smith-Waterman Search Element	69
15.4	Data Converters	72
15.4.1	Convert Text to Sequence Element	72
15.4.2	Reverse Complement Element	73
15.5	HMMER2 Tools	75
15.5.1	HMM Build Element	75
15.5.2	HMM Search Element	76
15.5.3	Read HMM Profile Element	78
15.5.4	Write HMM Profile Element	79
15.6	Multiple Sequence Alignment	80
15.6.1	Align with ClustalW Element	80
15.6.2	Align with Kalign Element	82
15.6.3	Align with MAFFT Element	83
15.6.4	Align with MUSCLE Element	84
15.6.5	Align with T-Coffee Element	85
15.6.6	Join Sequences into Alignment Element	86
15.6.7	Split Alignment into Sequences Element	87
15.7	Transcription Factor	88
15.7.1	Build Frequency Matrix Element	88
15.7.2	Read Frequency Matrix Element	89
15.7.3	Write Frequency Matrix Element	89
15.7.4	Build Weight Matrix Element	90
15.7.5	Read Weight Matrix Element	92
15.7.6	Write Weight Matrix Element	92
15.7.7	Convert Frequency Matrix Element	93
15.7.8	Search for TFBS with Weight Matrix Element	95
15.7.9	Build SITECON Model Element	96
15.7.10	Read SITECON Model Element	97

15.7.11	Write SITECON Model Element	98
15.7.12	Search for TFBS with SITECON Element	99
15.8	Utils	101
15.8.1	Generate DNA Element	101
15.9	Custom Elements Examples	103
15.9.1	Create Alignment Element	103
15.9.2	Dump Sequence Info Element	104
15.9.3	Quality Filter Element	104
15.10	Data Flow	106
15.10.1	Filter Element	106
15.10.2	Marker Element	106
15.11	DNA Assembly	109
15.11.1	Bowtie Aligner Element	109
15.11.2	Bowtie Build Indexer Element	112
15.11.3	Bowtie Index Reader Element	112
16	Workflow Samples	114
16.1	Data Marking	114
16.1.1	Marking Sequences by Annotation Number	114
16.1.2	Marking Sequences by Length	114

1 About Unipro

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

1.1 Contacts

Company website: <http://unipro.ru>

Address: UniPro, 6/1 Lavrentiev Avenue
630090, Novosibirsk, Russia

Marketing department: Tel: +7 (383) 3326061
Fax: +7 (383) 3302960
Email: marketing@unipro.ru

UGENE website: <http://ugene.unipro.ru>

UGENE technical support: Email: ugene@unipro.ru

2 About UGENE

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

To learn more about UGENE visit [UGENE website](#).

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

2.1 Key Features

- Creating, editing and annotating nucleic acid and protein sequences
- Search through online databases: NCBI, PDB, UniProtKB/Swiss-Prot, UniProtKB/TrEMBL
- Multiple sequence alignment: Clustal, MUSCLE, Kalign, MAFFT, T-Coffee
- Online and local BLAST search
- Restriction analysis with integrated REBASE restriction enzyme database
- Integrated Primer3 package for PCR primers design
- Search for direct, inverted and tandem repeats in DNA sequences
- Constructing dotplots for nucleic acid sequences
- Search for transcription factor binding sites (TFBS) with weight matrix and SITECON algorithms
- Aligning short reads with Bowtie and UGENE genome aligner
- Search for ORFs
- Cloning in silico
- 3D structure viewer for files in PDB and MMDB formats, anaglyph view support
- Protein secondary structure prediction with GOR IV and PSIPRED algorithms
- HMMER2 and HMMER3 packages integration
- Building (using integrated PHYLIP package) and viewing phylogenetic trees
- Local sequence alignment with optimized Smith-Waterman algorithm
- Combining various algorithms into custom workflows with UGENE Workflow Designer
- Search for a pattern of various algorithms' results in a nucleic acid sequence with UGENE Query Designer

2.2 User Interface

- Visual and interactive genome browsing including circular plasmid view
- Multiple alignment editor
- Chromatograms visualization
- 3D viewer for files in PDB and MMDB formats with anaglyph stereo mode support
- Phylogenetic tree viewer

- Easy to use workflow designer for custom computational workflows

2.3 High Performance Computing

- Complete support of modern multicore processors and SSE instructions
- Out of the box support of modern GPUs using NVIDIA CUDA and ATI Stream
- Integrated solutions for Cell Broadband Engine
- Supercomputers and distributed computing support
- Amazon EC2 cloud computing support

2.4 Cooperation

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

3 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 workflow schemas even if he or she is not familiar with any programming language.

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

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

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

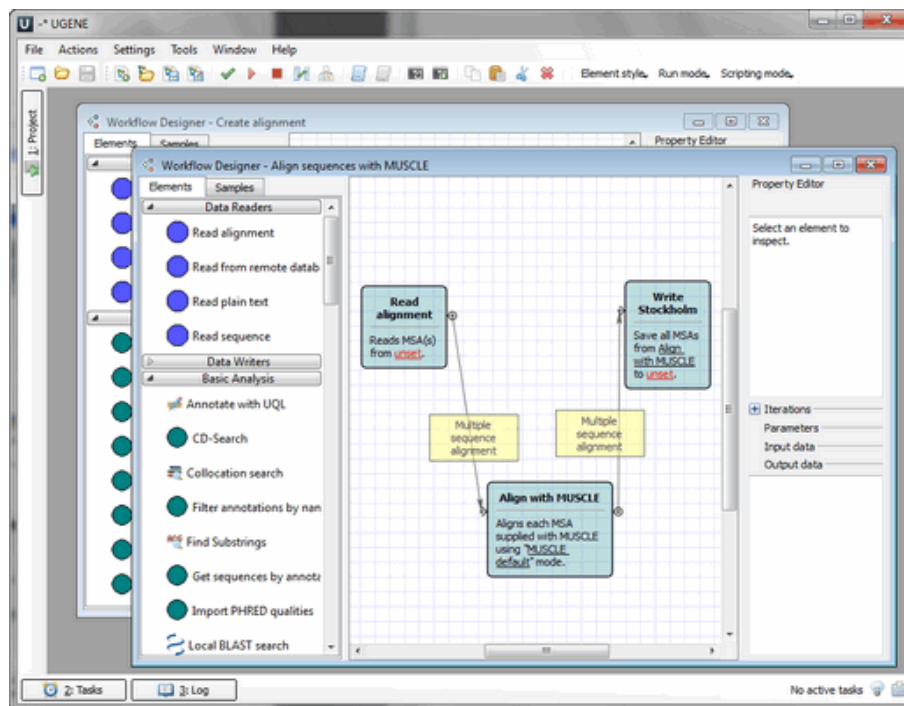
4 Introduction

This chapter describes the Workflow Designer key elements and provides an example on how to create and run a simple *schema*.

4.1 Launching Workflow Designer

To launch the Workflow Designer select the *Tools* → *Workflow Designer* item in the UGENE main menu.

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.



4.2 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 *schemas*.

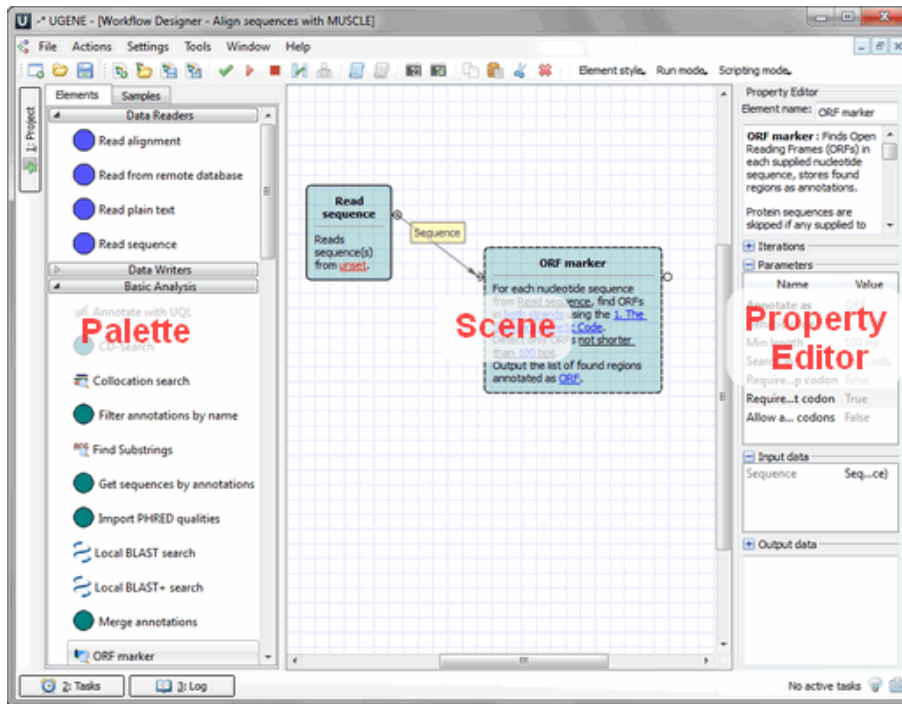
Scene

The main drawing scene is the place where the *workflow elements* are constructed into a *schema*.

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.

4.3 Workflow Schema Elements and Connections

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

workflow schema

A schema is a visual representation of the dataflow. It consists of *workflow elements* and their *connections*.

workflow element

An element of a workflow *schema*. 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. On the *Scene* it is displayed as prominent knob on an element with opened bubble.

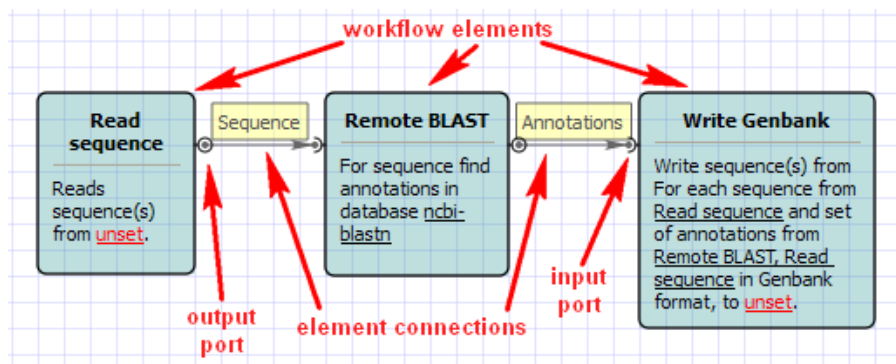
output port

An output port of an *element* is used to provide data to another element. On the *Scene* it is displayed as prominent knob on an element with closed bubble.

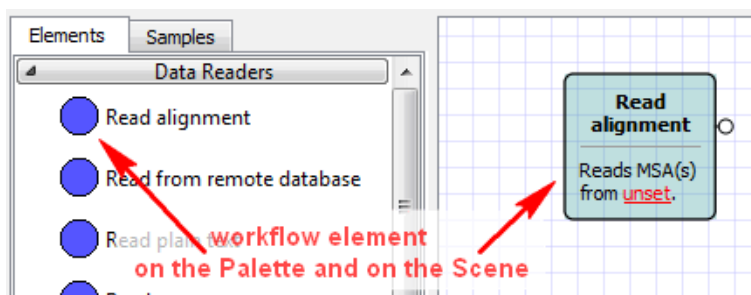
slot

Each *input* or *output* port contains one or several slots. A slot specifies the kind of data that can be passed through it (for example *Sequence*, *Set of annotations*, etc.)

See an example of a *schema* on the image below:

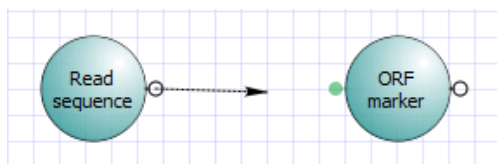


Your first step is to *add* necessary *workflow elements*, for example, by dragging them from the *Palette* to the *Scene*:



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 schema*.

4.4 Managing Parameters

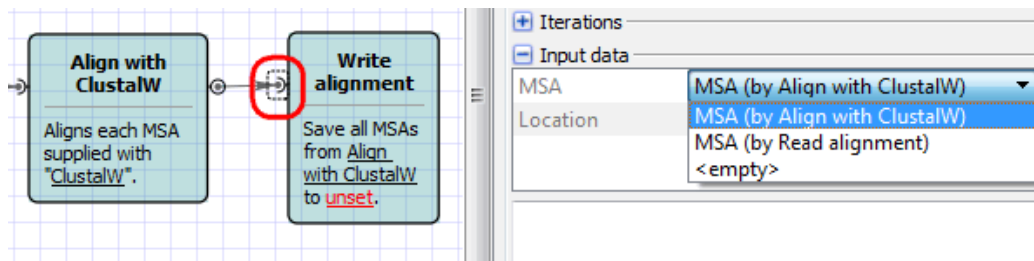
When you select an *element* on the *Scene* the *Property Editor* displays detailed information about it: its name, description, parameters, *input* and *output* ports, etc.

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, its 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 or select a value from a drop-down list and so on. 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.

When you select an *input* or *output* port on the *Scene* the *Property Editor* shows the *slots* of the port. You can configure slots of a connected input port by selecting different (matching) data available through the dataflow. For example on the image the *Write alignment* element can access MSA data from both previous elements:



However, it doesn't make sense to select *MSA (by Read alignment)* value in this case, as the data read from the input file would be written to the output file instead of the aligned MSA data, i.e. *MSA (by Align with ClustalW)*.

There is one more area in the *Property Editor* that has not yet been described — the *Iterations* area. Iterations can be used for executions of the same schema with varied parameters. To learn more about the iteration read *this chapter*.

4.5 UGENE Components and Workflow Designer

This paragraph provides an overview of UGENE components that affect your work with the Workflow Designer.

4.5.1 Task View, Notifications and Log View

When a *schema* is executed in the Workflow Designer a **task** is created.

Task View

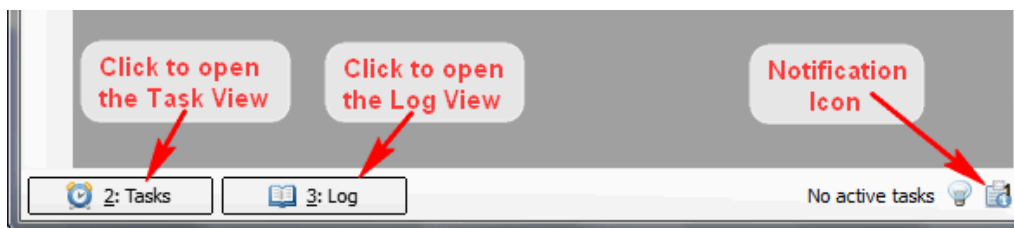
Here you can see the tasks currently executed in UGENE.

Notification Icon

When a task has finished its execution, a notification is pop up. At any time you can watch the last notifications by clicking the *Notification Icon*.

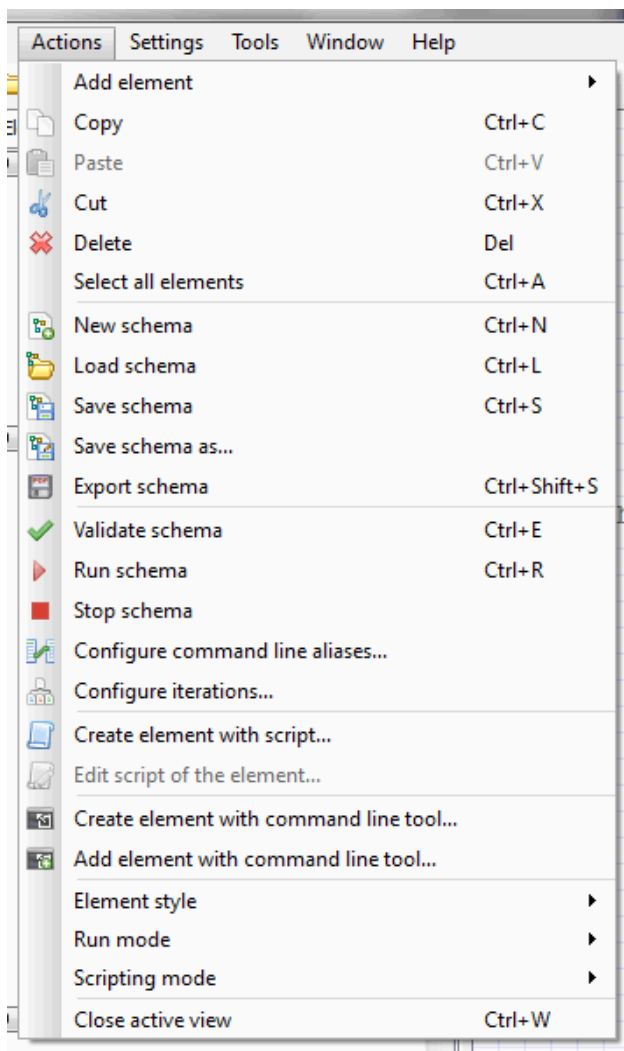
Log View

Here you can see UGENE logs.



4.5.2 Actions Menu

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



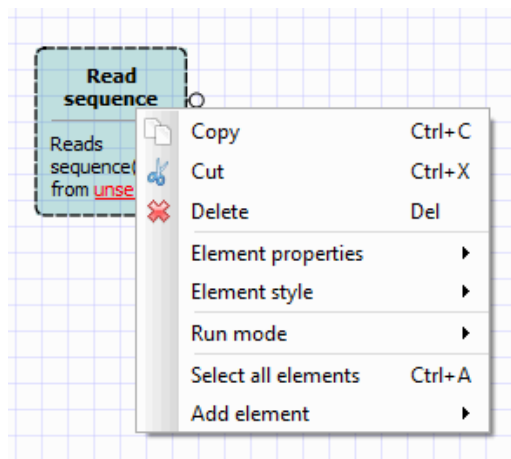
4.5.3 Toolbar

Most common actions are available on the main toolbar:



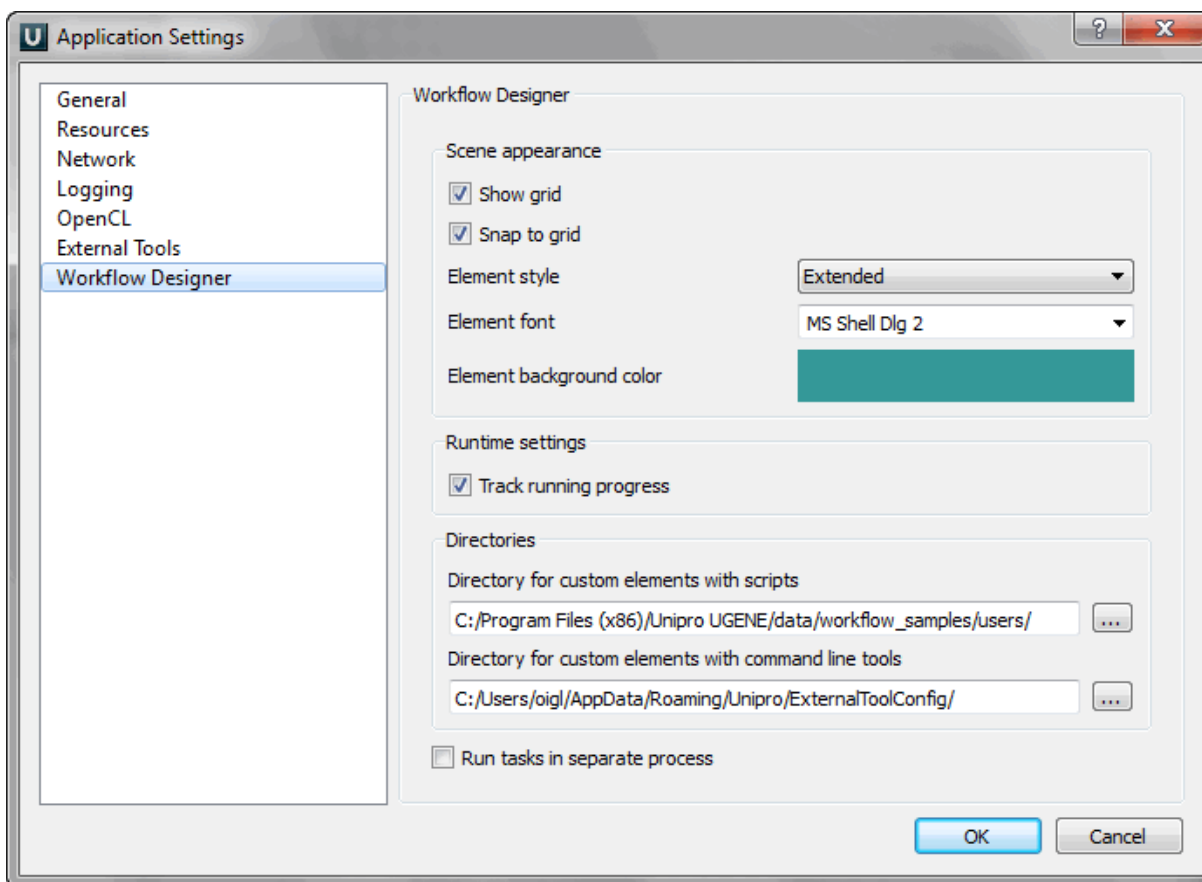
4.5.4 Context Menus

Some features are also available through context menus over corresponding areas, e.g.:



4.5.5 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.



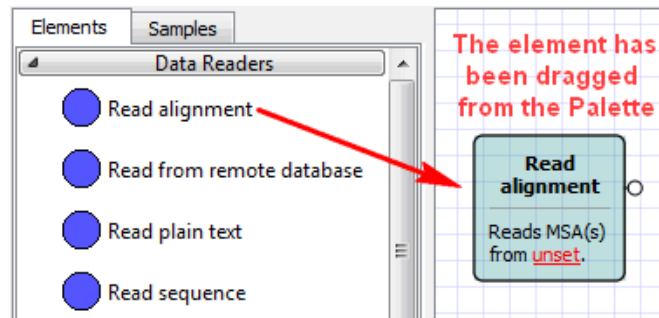
4.6 How to Create and Run Schema

1. Select *Tools* → *Workflow Designer* in the main menu.

Result: The Workflow Designer window appears.

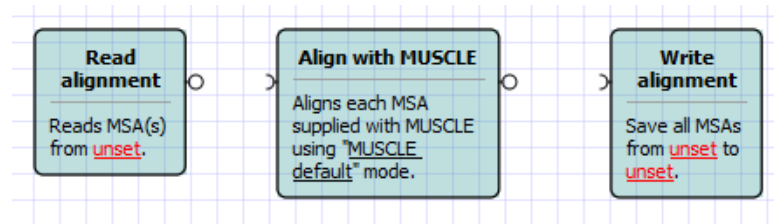
2. On the *Elements* tab of the *Palette* find the *Read alignment* element. It is located in the *Data sources* group and drag it to the *Scene*.

Result: The element is shown on the Scene.



3. 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.

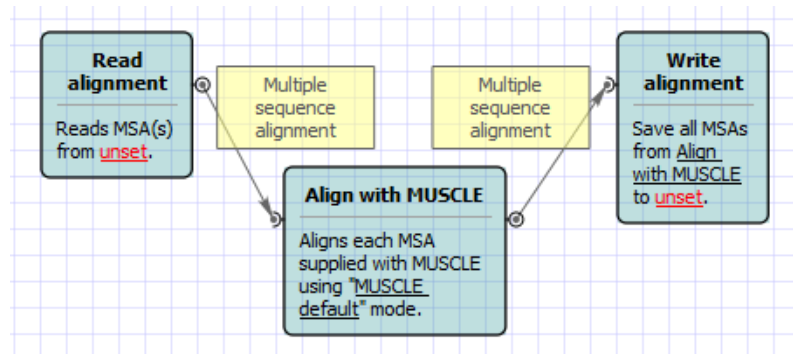
Result: All three elements are on the Scene.



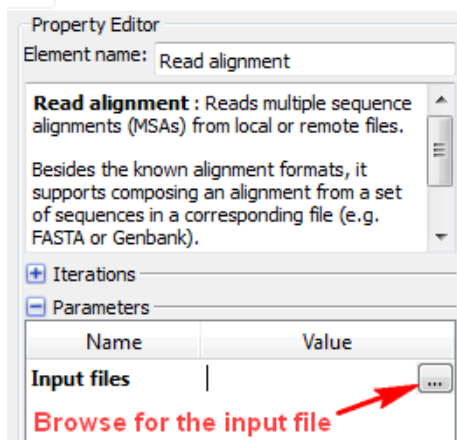
4. Connect the elements:

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

Result: The elements are connected with arrows.



- Select the *Read alignment* element. In the *Parameters* area of the *Property Editor* click on the *Value* column of the *Input files* parameter:



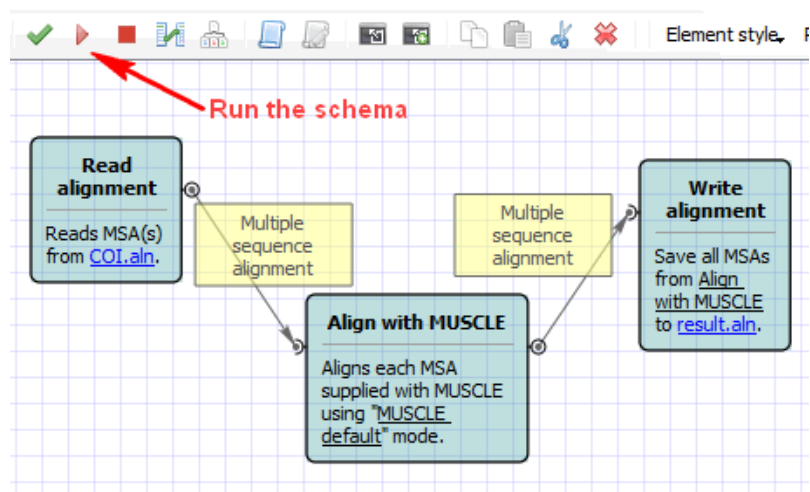
And browse for an input file, e.g. Select the `$UGENE\data\samples\CLUSTALW\COI.aln` file.

Result: The *Input files* value is set to the file's path.

- Select the *Write alignment* element and set the *Output file*, e.g. you can just enter `result.aln`.

Result: All required schema parameters are set.

- Click the *Run schema* button on the toolbar.



Result: After the schema 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.

5 Manipulating Element

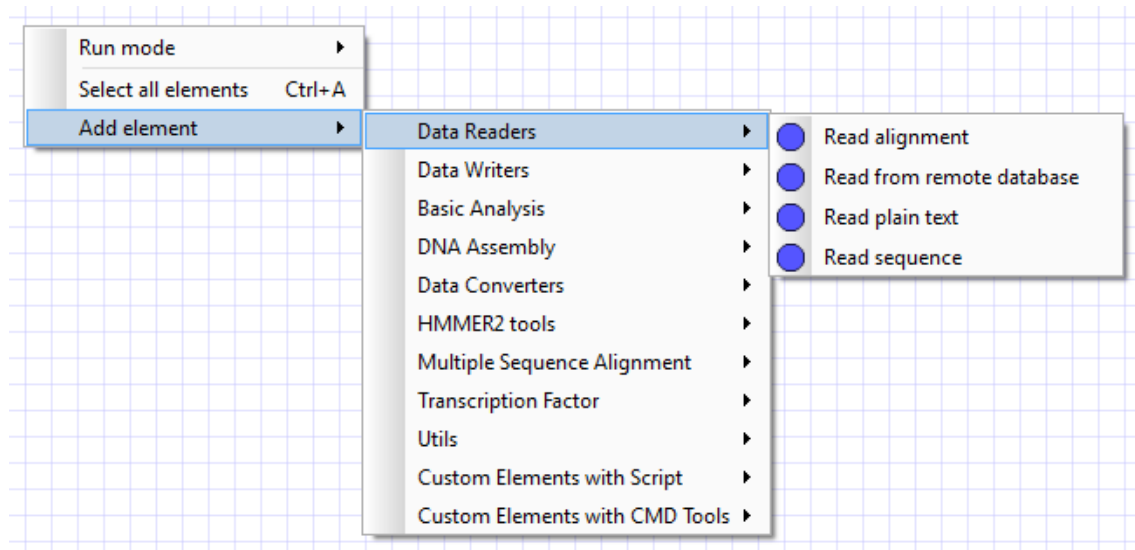
You can add new *workflow element* to the *Scene*, copy, cut, paste or delete it. Also you can select all elements currently presented on the *Scene*.

5.1 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.

5.2 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*.

5.3 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.

5.4 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.

5.5 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`.

5.6 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.

6 Manipulating Schema

You can create a new *schema*, save it and then load it again.

The designed schema can be displayed in a neat self-describing layout and exported to PDF document, raster or vector image with publication-ready quality.

You can validate created or modified schema before running it.

If you need, you can stop a schema execution.

6.1 Creating New Schema

To create a new *schema* select the *Actions* → *New schema* item in the main menu or *New schema* toolbar button.

Or press `Ctrl+N`.

6.2 Loading Schema

To load a *schema* select the *Actions* → *Load schema* item in the main menu or *Load schema* toolbar button.

Or press `Ctrl+L`.

Hint: You can load a *schema* by dragging the schema file (e.g. with `.uwl` extension) to the UGENE window.

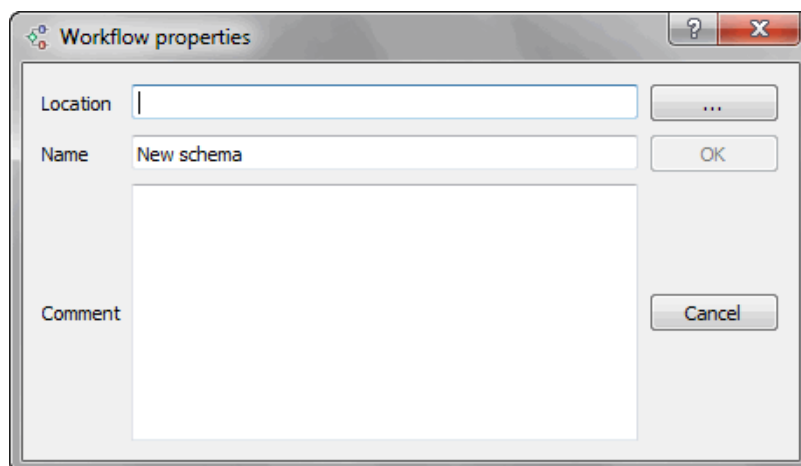
6.3 Saving Schema

Choose *Actions* → *Save schema* item in the main menu or *Save schema* toolbar button to save a workflow *schema*. The schema is saved to a file of native UGENE format (with `.uwl` extension).

The format is human-readable, you can find its description in chapter *Workflow Schema File Format*.

There is `Ctrl+S` keyboard shortcut for this action.

If you save a schema for the first time the *Workflow properties* dialog appears:



Here you can browse for the schema file *Location* and specify the schema *Name* and *Comment*.

Once a schema has been saved, it can be *loaded*. If you modify the loaded schema and save changes, then corresponding `.uwl` file is modified.

To save the schema with different properties choose the *Actions* → *Save schema as* item in the main menu and specify the required settings in the *Workflow properties* dialog.

6.4 Exporting Schema

Workflow *schema* can be exported as:

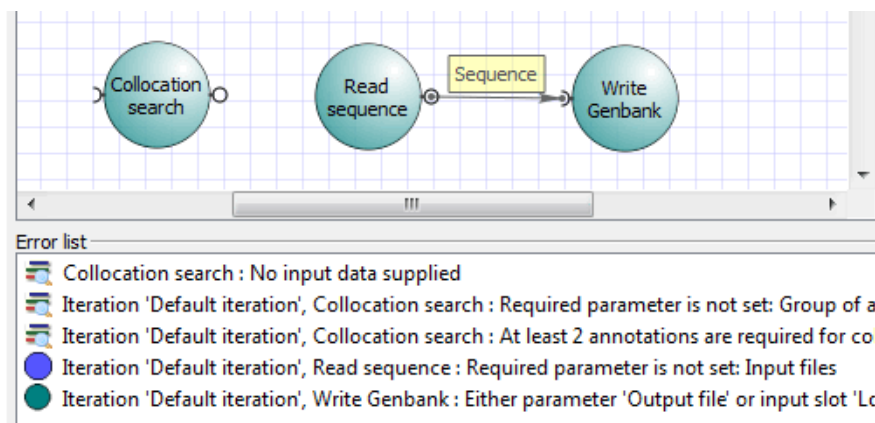
- Raster image (*.png, *.bmp, *.jpg, *.jpeg, *.ppm, *.xbm, *.xpm)
- Vector image (*.svg)
- Portable document (*.pdf, *.ps)

To export a *schema* select the *Actions* → *Export schema* item in the main menu or use the `Ctrl+Shift+S` keyboard shortcut. *Export workflow schema to image* save file dialog appears. Enter a file name and choose the file type.

6.5 Validating Schema

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 schema* item in the main menu or *Validate schema* 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.

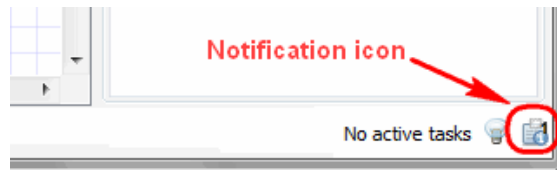


Double-clicking on items in the list selects the faulty element/iteration.

6.6 Running Schema

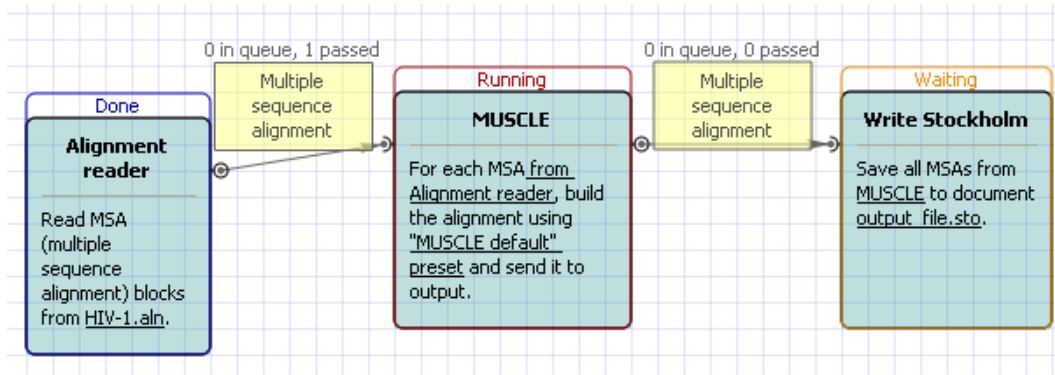
Once you are satisfied with the designed workflow *schema* and have it configured, click the *Run schema* button on the toolbar (alternatively, you can select the *Actions* → *Run schema* item in the main menu or launch it by pressing `Ctrl+R`). The schema gets verified and scheduled for background execution. 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 summary report. To view it, click the notification button at the bottom right corner of the UGENE window and select an appropriate notification in the appeared *Notifications* pop-up window.



The report displays status of each iteration execution and provides other details.

Note, that you can see the progress of a schema execution in a Workflow Designer window by checking the *Track running progress on diagram* option in the *Application Settings* dialog:

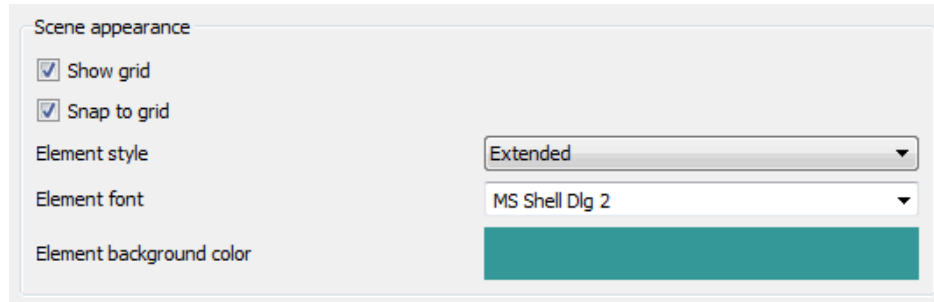


6.7 Stopping Schema

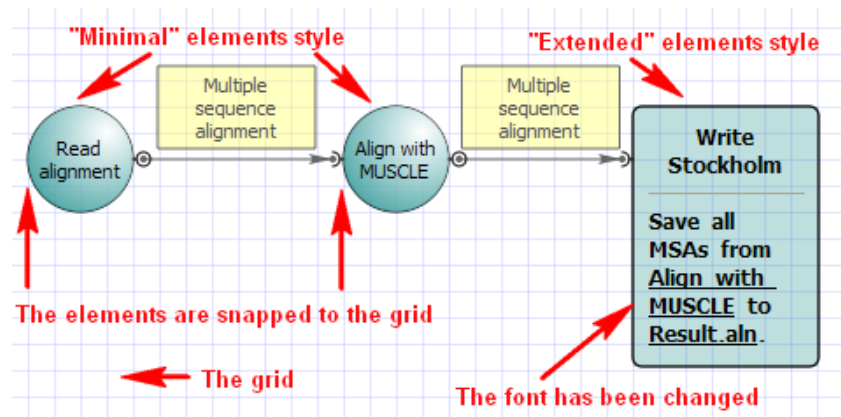
A *schema* execution can be stopped by selecting the *Actions* → *Stop schema* item in the main menu or *Stop schema* toolbar button.

7 Changing Appearance

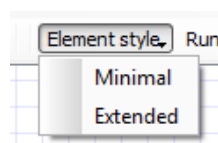
Default setting that influence the Workflow Designer appearance can be set in the *Application Settings* dialog. The parameters are shown on the image below:



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



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.



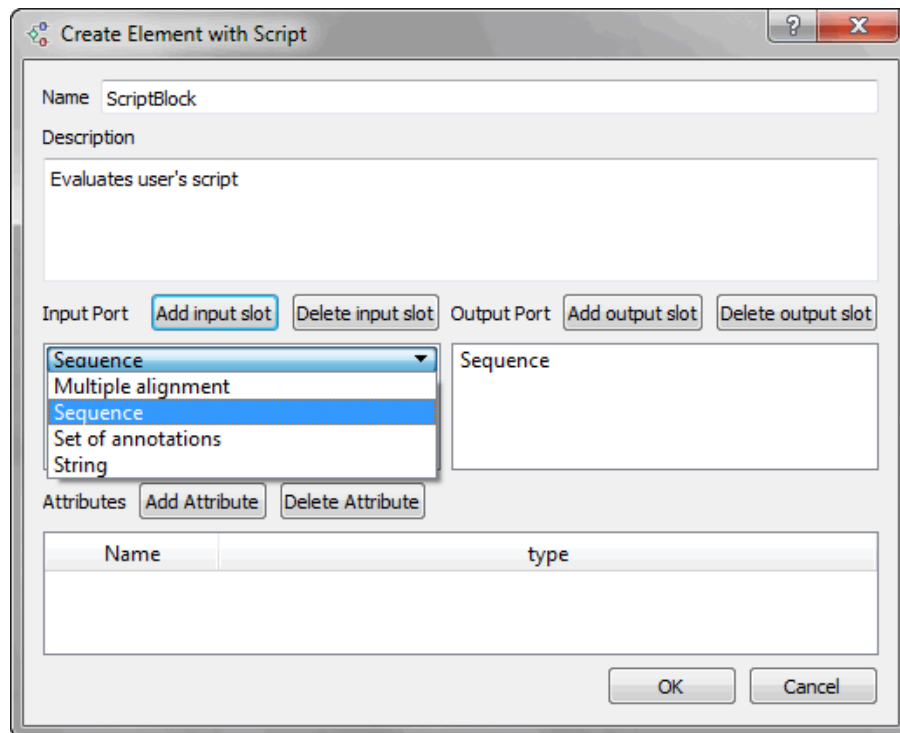
8 Custom Elements with Scripts

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 3 types of data for a slot available:

- Multiple alignment
- Sequence
- Set of annotations
- String

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.

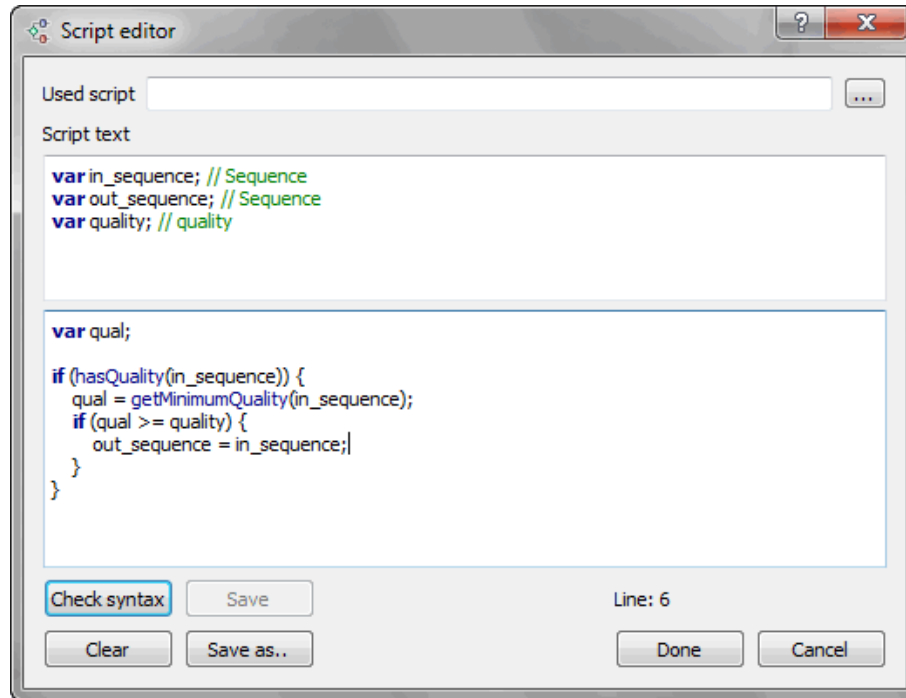
The element also becomes available in the *Custom Elements with Scripts* group on the *Palette*.

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.

8.1 Functions supported for *Multiple alignment data*

- **createAlignment** (Sequence seq1, Sequence seq2, ...) — returns the alignment created from the sequences.
- **addToAlignment** (MAAlignment 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** (MAAlignment aln, int row) — returns the sequence from the specified row of the alignment.
- **findInAlignment** (MAAlignment 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** (MAAlignment aln, QString name) — searches the alignment for a sequence with the specified name.
- **removeFromAlignment** (MAAlignment 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.

8.2 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.

8.3 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.

8.4 Common function

- **printToLog** (parameter) - prints the results to the *Log View*.

9 Custom Elements with Command Line Tools

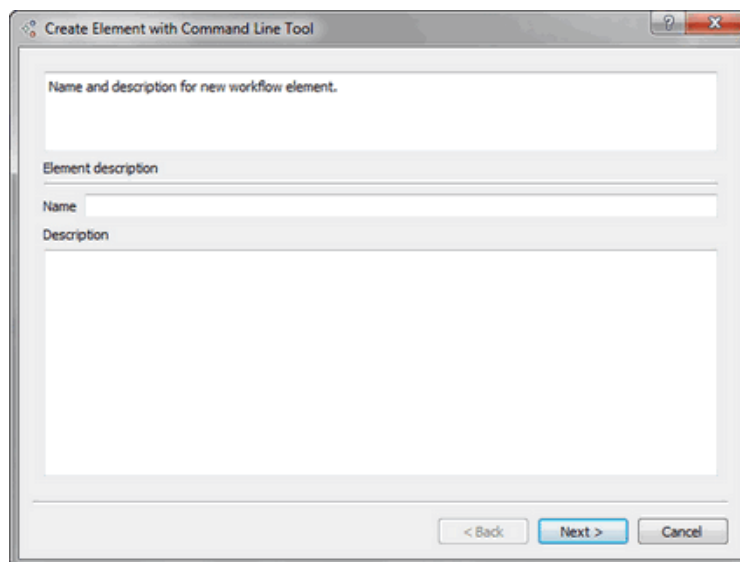
In UGENE you can create a custom workflow *element* that would launch any command line tool.

9.1 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:

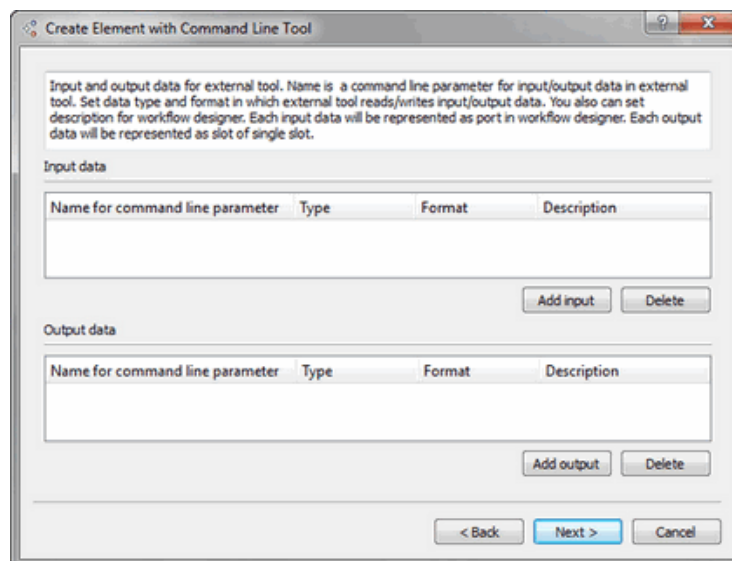


The *Create Element with Command Line Tool* wizard appears. On the first page of the wizard input a name and a description of the element in the *Property Editor*. Letters, numbers and underscores are allowed in the name.



The screenshot shows the first page of the wizard. The title bar reads "Create Element with Command Line Tool". The main area contains a text box for "Name and description for new workflow element." Below this is a section titled "Element description" with two sub-sections: "Name" and "Description", each with a corresponding text input field. At the bottom right, there are three buttons: "< Back", "Next >" (highlighted in blue), and "Cancel".

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

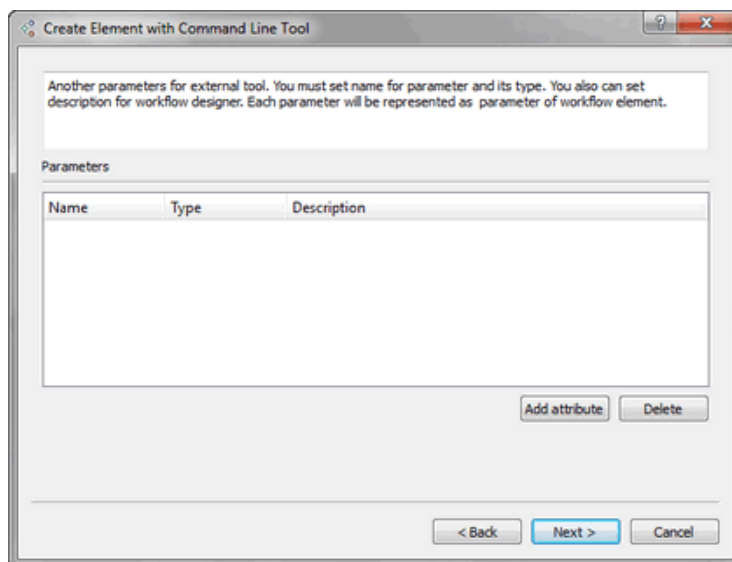


The screenshot shows the second page of the wizard. The title bar reads "Create Element with Command Line Tool". The main area contains a text box with instructions: "Input and output data for external tool. Name is a command line parameter for input/output data in external tool. Set data type and format in which external tool reads/writes input/output data. You also can set description for workflow designer. Each input data will be represented as port in workflow designer. Each output data will be represented as slot of single slot." Below this are two sections: "Input data" and "Output data". Each section contains a table with columns for "Name for command line parameter", "Type", "Format", and "Description". Below each table are "Add input" and "Delete" buttons. At the bottom right, there are three buttons: "< Back", "Next >" (highlighted in blue), and "Cancel".

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 or a set of annotations.
- Select a format required.
- Optionally input a description.

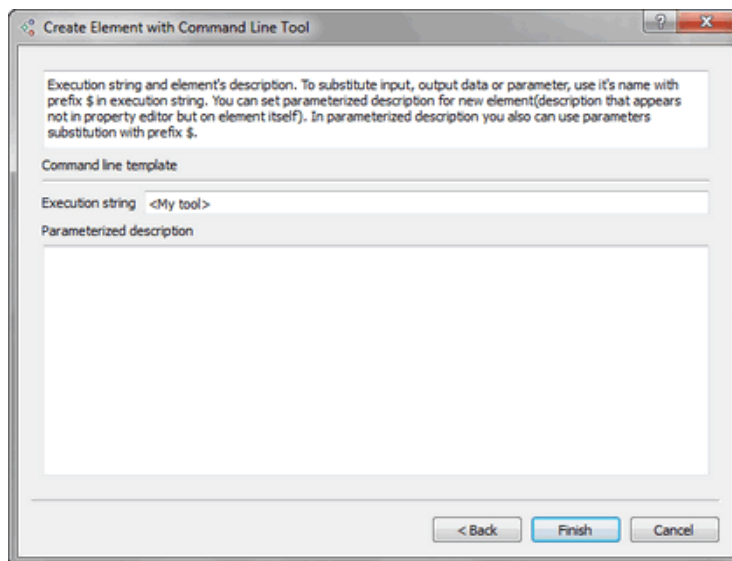
On the third page of the wizard you can add attributes for the command line tool. Later you would be able to set values for the attributes in the Property Editor, i.e. the attributes are actually the parameters of the new element.



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.



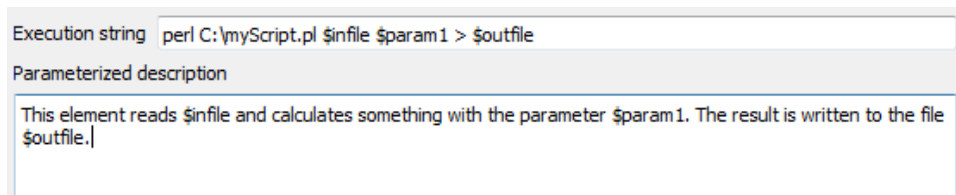
The signature of the execution string depends on the command that is launched. But the general rule is that input/output data and attributes have prefix \$. For example let there be some perl script "myScript.pl" that accepts an input file as the first attribute and accepts the second attribute denoted as "param1". The command may look as follows:

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

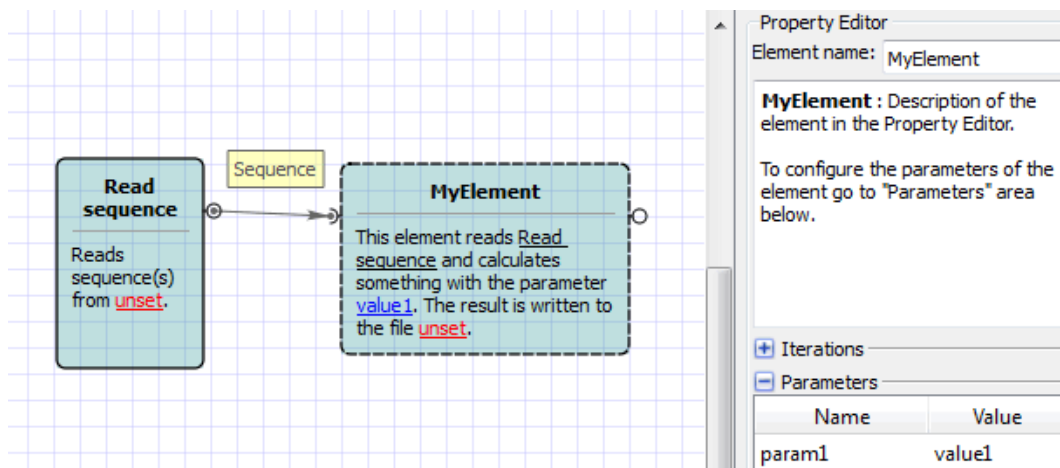
Here *infile* and *outfile* are input and output data set on the step 2, *param1* is an attribute set on the step 3.

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:



9.2 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. The same wizard as during creation of the element would appear.

9.3 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:



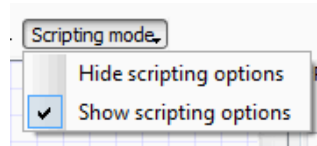
In the appeared dialog select the required `.etc` file. The element is added to the group on the Palette and appears on the Scene.

9.4 Removing Element

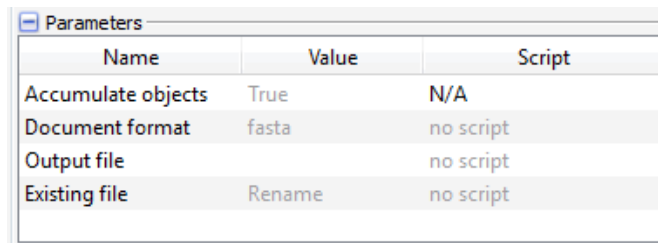
To remove an element right-click on it and select the *Remove* item in the element's context menu. The corresponding `.etc` file is also removed in this case.

10 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:

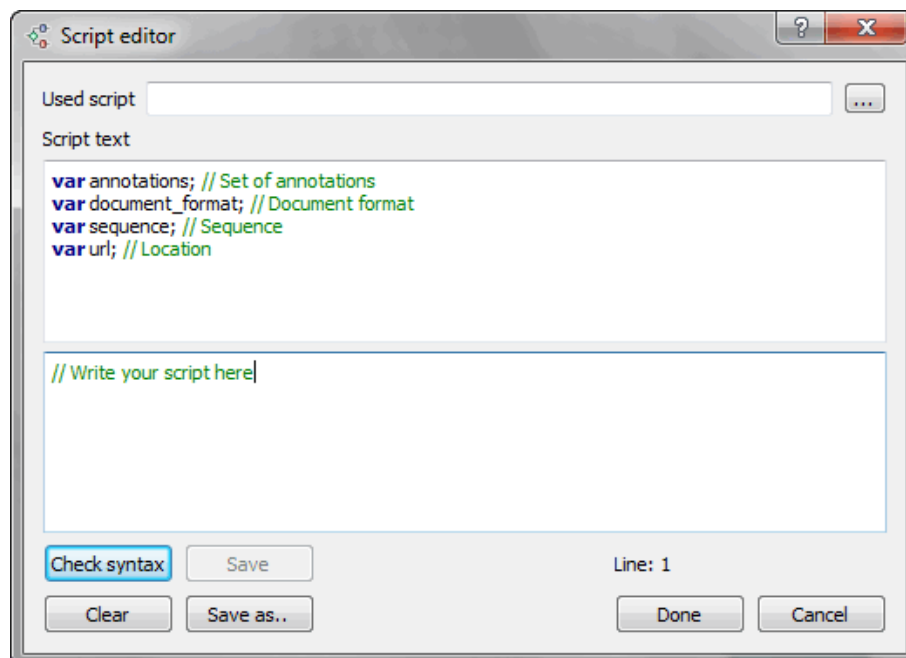


Name	Value	Script
Accumulate objects	True	N/A
Document format	fasta	no script
Output file		no script
Existing file	Rename	no script

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:

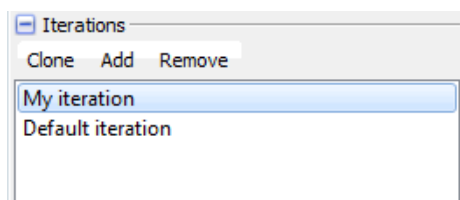


Here you can see the variables available from the dataflow and can write your script. Supported languages for the script are languages based on the ECMAScript (Javascript, QtScript).

11 Using Iterations

Iterations are used to input various sets of parameters values to the same *schema*. All iterations are executed in parallel when the schema is executed.

When a schema is just created the *Default iteration* is also created. It can be seen in the *Iterations* area in the *Property Editor*. In the same area there are *Clone*, *Add* and *Remove* buttons:



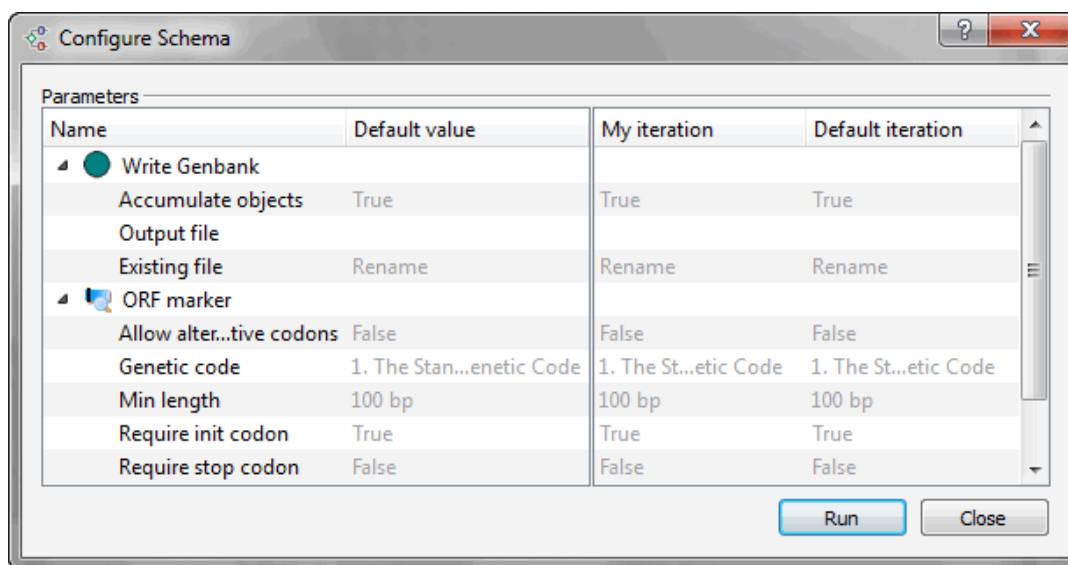
When you add an iteration, a new iteration is created with unset parameters values. An iteration can be also created by cloning it from some other iteration, all values of the parameters are remained in this case. It is possible to rename iterations, directly in the list.

You can also delete an iteration, but there always should be at least one iteration.

If you have several iterations you can switch between them by selecting the required iteration in the iterations list. Now when you set a parameter value for an *element* the value is modified only in the selected iteration.

Note: The list and selection of current iteration are global to the *workflow schema*. Usually it is reasonable to build the schema topology first and then start configuring iterations.

There is a dedicated interface to overview and configure all parameters of the schema and watch variance between iterations side by side. It is invoked by selecting a *Configure iterations* item in the *Actions* main menu or by clicking on the same button on the toolbar.



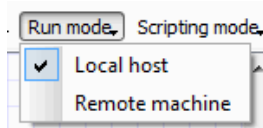
On the left side of the dialog you can see a hierarchical list of all the configurable parameters of the schema, along with their default values. The right side displays parameter values for all defined iterations. The default values are applied to all iterations unless explicitly overridden in a particular iteration, they are indicated by gray labels.

Upon tuning the overall configuration, you can immediately launch execution of the schema or return to editing it.

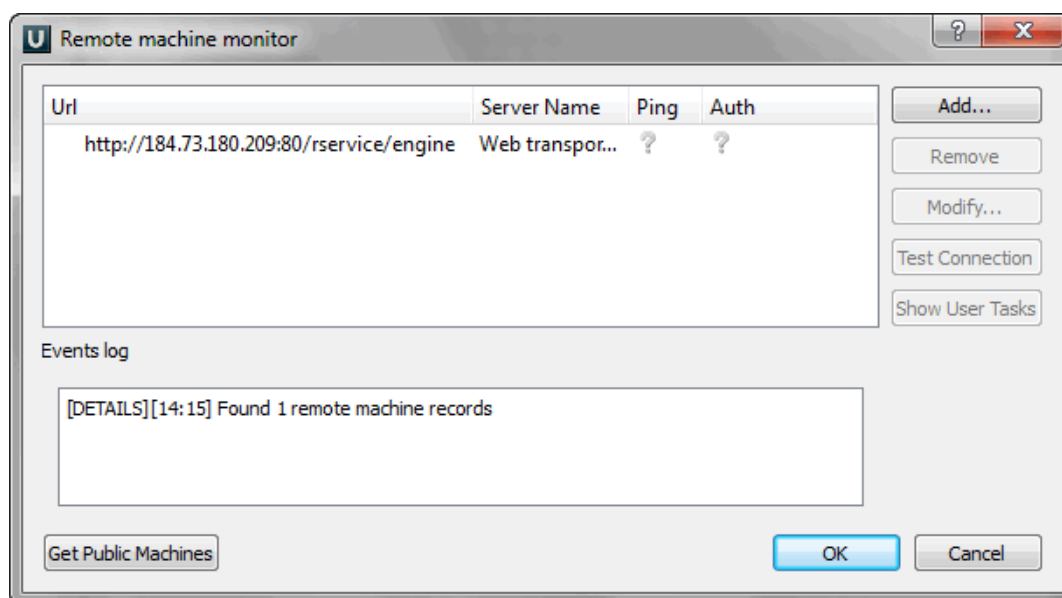
12 Running Schema Locally or Remotely

By default a *schema* is executed locally on your computer. You can also use a remote machine resources to calculate the task and execute the schema on the **Amazon EC2 cloud**. Currently, this service is in test mode and is **free to use**.

To run a *schema* remotely select the *Remote machine* item in the *Run mode* menu. The *Run mode* menu is available on the Workflow Designer toolbar, in the *Actions* main menu and in the *Scene* context menu.



The next time the schema is *executed* (on condition that it is *valid*) the following dialog appears:



Select the *Url* of the remote machine and press *Run*.

To run schemas on your local computer again check the *Local host* item in the *Run mode* menu.

To learn more about distributed computing in UGENE, please, read [main UGENE User Manual](#).

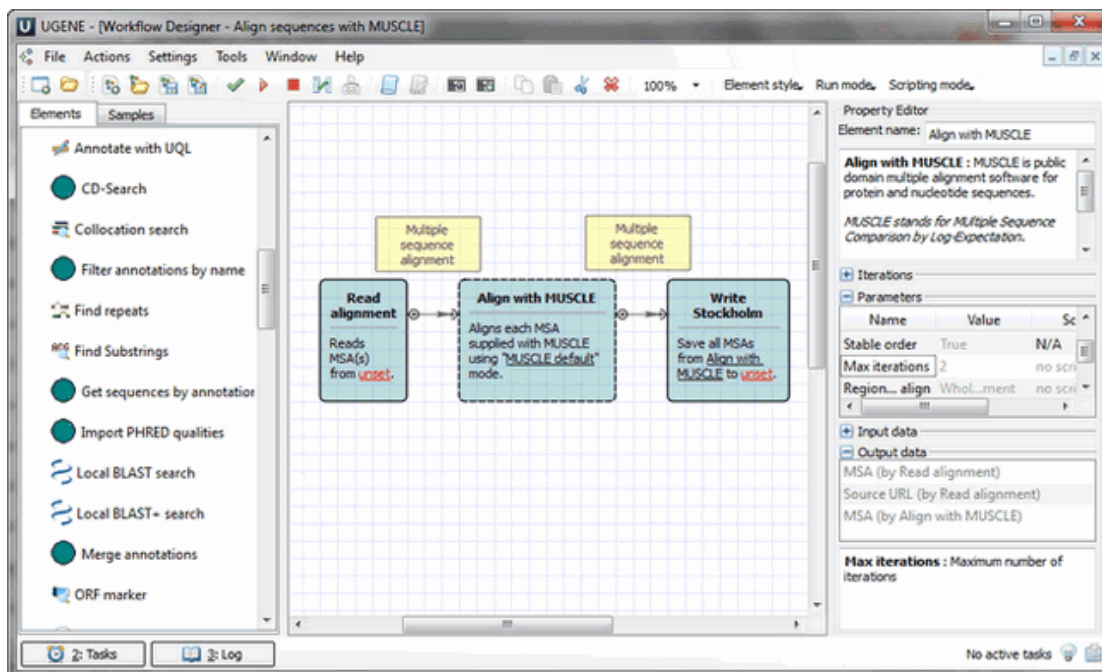
13 Running Schema from the Command Line

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

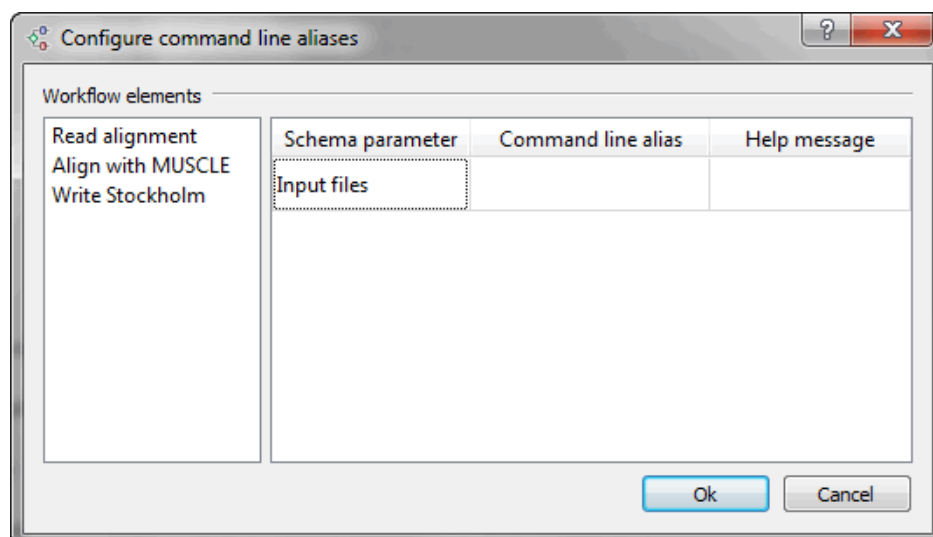
This chapter describes how you can create a new command using a *schema*.

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

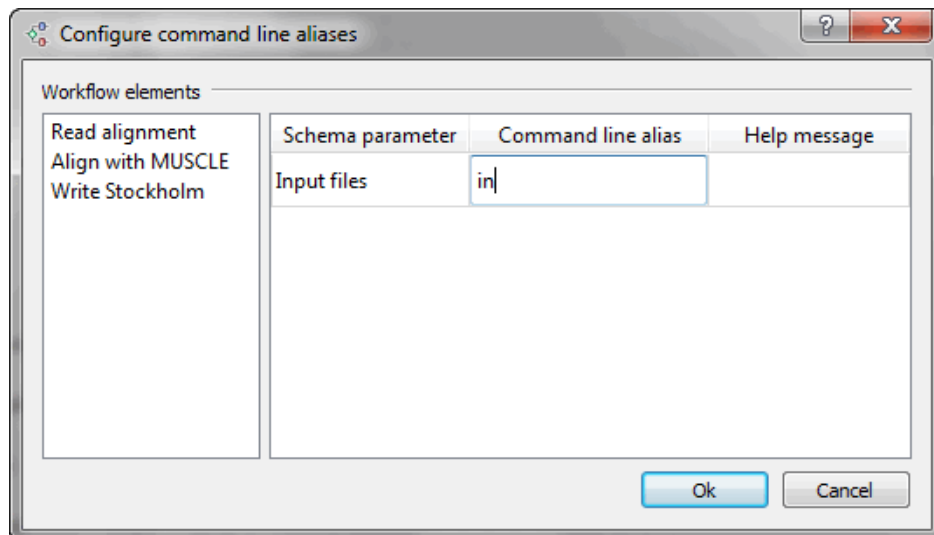
1. Create the schema in the Workflow Designer. For example on the image below the *Align sequences with MUSCLE* sample schema is used:



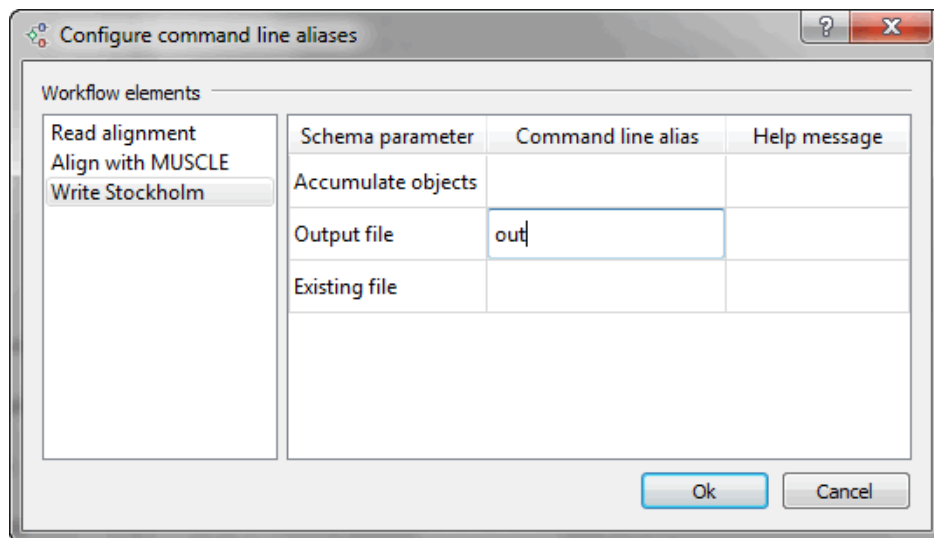
2. Now you should configure aliases for those parameters that you are going to use from the command line. To do it select the *Actions* → *Configure command line aliases* item in the main menu or the *Configure command line aliases* toolbar button. The following dialog appears:



It contains the list of objects that corresponds to the *elements* of the schema. 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).



Press the *Ok* button to save aliases and close the dialog.

3. *Save the schema* to a file: if you follow the example, choose the *Actions* → *Save schema as...* item in the main menu, browse for the file location and enter **mySchema** as the schema name. This name will be used to launch the schema from the command line.
4. Launch the schema from the command line:

```
'[path_to_ugene]\ugene --task={schema_name} [--{parameter1}={value1}
[--{parameter2}={value2} ...]'
```

For example on Windows the command can look as follows:

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

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

14 Workflow Schema File Format

Using the GUI is not the only way to create / edit a *workflow schema*. A schema 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 schema file using a text editor.

The best way to learn workflow schema 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.

14.1 Header

The header consists of the following key string:

```
#!/UGENE_WORKFLOW
```

And multiline description of the schema:

```
# Write here the description  
# of your schema.
```

14.2 Body

The body begins with the **workflow** keyword followed by the name of the schema and curly braces:

```
workflow schema_name {  
  
    # Description of the elements  
    # Description of the dataflow  
    # Description of the iterations  
    # Metainformation (aliases and visual information)  
  
}
```

14.2.1 Elements

Each *element* used in the *schema* 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.

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
};
```

14.2.2 Dataflow

The description of the elements is followed by the description of their connections to each other, i.e. the dataflow.

It consists of a list of elements pairs, separated by the new line. The following format is used for a pair:

```
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 schema, that aligns an input MSA and writes the result to a file in ClustalW format.

```
read-msa.msa->muscle.in-msa.msa
muscle.msa->write-clustalw.in-msa.msa
```

14.2.3 Iterations

An *iteration* description starts with the **.iteration** keyword and has the following format:

```
.iteration iteration-name {
  id: id-value
  # The iteration data for element1
  # The iteration data for element2
  # ...
}
```

The *iteration-name* corresponds to the name of the iteration in the GUI.

The *id-value* must be a unique number which identifies the iteration, i.e. for different iteration there must be different IDs.

Then the description of the the parameters values for the iteration follows. It consists of a list of the following blocks:

```
element_name {
  parameter1:value1;
  parameter2:value2;
  ...
}
```

The name of the block corresponds to an existing element name.

See, for example, the description of *Default iteration* data of two elements:

```
.iteration "Default iteration"{
  id:1

  read-msa {
    url-in:/home/user/pkinase.aln;
  }
  write-msa {
    url-out:"/home/user/my data/pkinase_copy.aln";
  }
}
```

14.2.4 Metainformation

A metainformation block sets visual parameters of the schema 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 schema aliases
  }
  visual {
    # Visual data for element1
    # Visual data for element2
    # ...
  }
}
```

Aliases

The block starts with the **aliases** keyword and has the following format:

```
aliases {
  element_name.parameter_name:value;
  ...
}
```

The value specified for an element parameter is used as the alias for this parameter when the schema is *executed from the command line*.

See an example of setting schema aliases:

```
.meta {
  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 schema in a Workflow Designer window, i.e. appearance of the schema *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";
  }
}
```

15 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 schema 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

15.1 Data Readers

Data Readers *elements* read data (from files, remote databases, etc.) and provide them to other elements in a workflow *schema*.

15.1.1 Read Alignment Element

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

Parameters in GUI

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

Parameters in Schema File

Type: read-msa

Parameter	Parameter in the GUI	Type
url-in	Input files	<i>string</i>

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *Multiple sequence alignment*

Name in Schema File: out-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>
Source URL	url	<i>string</i>

15.1.2 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

Parameters in Schema File

Type: fetch-sequence

Parameter	Parameter in the GUI	Type
resource-id	Resource IDs	<i>string</i>
database	Database	<i>string</i> Available values are: <ul style="list-style-type: none"> • NCBI GenBank (DNA sequence) • NCBI protein sequence database • PDB • SWISS-PROT • UniProtKB/Swiss-Prot • UniProtKB/TrEMBL
save-dir	Save file to directory	<i>string</i>

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *Sequence*

Name in Schema File: out-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>
Set of annotations	annotations	<i>annotation-table</i>

15.1.3 Read Plain Text Element

Reads text from local or remote files.

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 Schema File

Type: read-text

Parameter	Parameter in the GUI	Type
url-in	Input files	<i>string</i>
read-by-lines	Read by lines	<i>boolean</i>

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *Plain text*

Name in Schema File: out-text

Slots:

Slot In GUI	Slot in Schema File	Type
Plain text	text	<i>string</i>
Source URL	url	<i>string</i>

15.1.4 Read Sequence Element

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

Parameters in GUI

Parameter	Description	Default value
Input files (required)	Semicolon-separated list of paths to the input files.	
Mode (required)	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
Accession filter	Only reports a sequence with the specified accession (id).	

Parameters in Schema File

Type: read-sequence

Parameter	Parameter in the GUI	Type
url-in	Input files	<i>string</i>
mode	Mode	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 0 - for split mode • 1 - for merge mode
merge-gap	Merging gap	<i>numeric</i>
accept-accession	Accession filter	<i>string</i>

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *Sequence*

Name in Schema File: out-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>
Set of annotations	annotations	<i>annotation-table</i>
Source URL	url	<i>string</i>

15.2 Data Writers

Data Writers *elements* write data supplied from other elements in a workflow *schema* to a file or files.

15.2.1 Write Alignment Element

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

Parameters in GUI

Parameter	Description	Default value
Output file (required)	Location of the output data file. If this parameter 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
Document format	Format of the output file.	clustal

Parameters in Schema File

Type: write-msa

Parameter	Parameter in the GUI	Type
url-out	Output file	<i>string</i>
write-mode	Existing file	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 0 - for overwrite • 1 - for append • 2 - for rename
document-format	Document format	<i>string</i> Available values are: <ul style="list-style-type: none"> • clustal • mega • msf • sam • srfasta • stockholm

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Multiple sequence alignment*

Name in Schema File: in-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>
Location	url	<i>string</i>

15.2.2 Write ClustalW Element

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

Parameters in GUI

Parameter	Description	Default value
Output file (required)	Location of the output data file. If this parameter 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

Parameters in Schema File

Type: write-clustalw

Parameter	Parameter in the GUI	Type
url-out	Output file	<i>string</i>
write-mode	Existing file	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 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 Schema File: in-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>
Location	url	<i>string</i>

15.2.3 Write FASTA Element

Writes all supplied sequences to file(s) in FASTA 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 Schema File

Type: write-fasta

Parameter	Parameter in the GUI	Type
url-out	Output file	<i>string</i>
write-mode	Existing file	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 0 - for overwrite • 1 - for append • 2 - for rename
accumulate	Accumulate objects	<i>boolean</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Sequence*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>
Location	url	<i>string</i>
FASTA header	fasta-header	<i>string</i>

15.2.4 Write FASTQ 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 Schema File

Type: write-fastq

Parameter	Parameter in the GUI	Type
url-out	Output file	<i>string</i>
write-mode	Existing file	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 0 - for overwrite • 1 - for append • 2 - for rename
accumulate	Accumulate objects	<i>boolean</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Sequence*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>
Location	url	<i>string</i>

15.2.5 Write Genbank Element

Writes all supplied sequences and related annotations to file(s) in Genbank 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 Schema File

Type: write-genbank

Parameter	Parameter in the GUI	Type
url-out	Output file	<i>string</i>
write-mode	Existing file	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 0 - for overwrite • 1 - for append • 2 - for rename

accumulate	Accumulate objects	<i>boolean</i>
------------	--------------------	----------------

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Sequence*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>
Location	url	<i>string</i>
Set of annotations	annotations	<i>annotation-table-list</i>

15.2.6 Write Plain Text Element

Writes strings to a file.

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 Schema File

Type: write-text

Parameter	Parameter in the GUI	Type
url-out	Output file	<i>string</i>
write-mode	Existing file	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 0 - for overwrite • 1 - for append • 2 - for rename
accumulate	Accumulate objects	<i>boolean</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Plain text*

Name in Schema File: in-text

Slots:

Slot In GUI	Slot in Schema File	Type
Plain text	text	<i>string</i>
Location	url	<i>string</i>

15.2.7 Write Sequence Element

Writes all supplied sequences to file(s) in selected 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
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

Parameters in Schema File

Type: write-sequence

Parameter	Parameter in the GUI	Type
url-out	Output file	<i>string</i>
write-mode	Existing file	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 0 - for overwrite • 1 - for append • 2 - for rename
document-format	Document format	<i>string</i> Available values are: <ul style="list-style-type: none"> • fasta • fastq • genbank • raw
accumulate	Accumulate objects	<i>boolean</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Sequence*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>
Location	url	<i>string</i>
Set of annotations	annotations	<i>annotation-table-list</i>

15.2.8 Write Stockholm Element

Writes all supplied alignments to file(s) in Stockholm 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 Schema File

Type: write-stockholm

Parameter	Parameter in the GUI	Type
url-out	Output file	<i>string</i>
write-mode	Existing file	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 0 - for overwrite • 1 - for append • 2 - for rename
accumulate	Accumulate objects	<i>boolean</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Multiple sequence alignment*

Name in Schema File: in-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>
Location	url	<i>string</i>

15.3 Basic Analysis

15.3.1 Analyze with UQL Element

Analyzes a nucleotide sequence with a UGENE Query Language (UQL) schema. The schema specifies a set of features to search for and their positional relationship.

To learn more about UQL schemas read [UGENE Query Designer Manual](#).

Parameters in GUI

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

Parameters in Schema File

Type: query

Parameter	Parameter in the GUI	Type
schema	Schema	<i>string</i>
merge	Merge	<i>boolean</i>
offset	Offset	<i>numeric</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input sequences*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

And 1 *output port*:

Name in GUI: *Result annotations*

Name in Schema File: out-annotations

Slots:

Slot In GUI	Slot in Schema File	Type
Set of annotations	annotations	<i>annotation-table</i>

15.3.2 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 annotation (required)	Name of the result annotation to mark found collocations.	misc_feature
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 Schema File

Type: collocated-annotation-search

Parameter	Parameter in the GUI	Type
result-name	Result annotation	<i>string</i>
annotations	Group of annotations	<i>string</i>
region-size	Region size	<i>numeric</i>
must-fit	Must fit into region	<i>boolean</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input data*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>
Set of annotations	annotations	<i>annotation-table-list</i>

And 1 *output port*:

Name in GUI: *Group annotations*

Name in Schema File: out-annotations

Slots:

Slot In GUI	Slot in Schema File	Type
Set of annotations	annotations	<i>annotation-table</i>

15.3.3 Extract Annotated Regions Element

Extracts annotated regions from input sequence.

Parameters in GUI

Parameter	Description	Default value
Annotation names (required)	List of annotation names which will be accepted or filtered. Use space as the separator.	
Accept of filter	Selects the name filter: accept specified names or accept all except specified.	Accept
Complement	Complements the annotated regions if the corresponding annotation is located on the complement strand.	True
Translate	Translates the annotated regions if the corresponding annotation marks a protein subsequence.	True
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.	1

Parameters in Schema File

Type: extract-annotated-sequence

Parameter	Parameter in the GUI	Type
annotation-names	Annotation names	<i>string</i>
accept-or-filter	Accept or filter	<i>boolean</i> Available values are: <ul style="list-style-type: none"> • true - for accept • false - for filter
complement	Complement	<i>boolean</i>
translate	Translate	<i>boolean</i>
extend-left	Extend left	<i>numeric</i>
extend-right	Extend right	<i>numeric</i>
merge-gap-length	Gap length	<i>numeric</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input sequence*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>
Set of annotations	annotations	<i>annotation-table</i>

And 1 *output port*:

Name in GUI: *Annotated regions*

Name in Schema File: out-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>
Set of annotations	annotations	<i>annotation-table</i>

15.3.4 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 Schema File

Type: repeats-search

Parameter	Parameter in the GUI	Type
result-name	Annotate as	<i>string</i>
algorithm	Algorithm	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 0 - algorithm choosed automatically • 1 - for diagonal algorithm • 2 - for suffix index algorithm
filter-nested	Filter nested	<i>boolean</i>
identity	Identity	<i>numeric</i>
max-distance	Max distance	<i>numeric</i>
min-distance	Min distance	<i>numeric</i>
min-length	Min length	<i>numeric</i>
threads	Parallel threads	<i>numeric</i> 0 - for using autodetected threads number

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input sequence*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

And 1 *output port*:

Name in GUI: *Repeat annotations*

Name in Schema File: out-annotations

Slots:

Slot In GUI	Slot in Schema File	Type
Set of annotations	annotations	<i>annotation-table</i>

15.3.5 Find Substrings 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 (required)	Name of the result annotation.	misc_feature
Pattern(s) (required)	Semicolon-separated list of patterns to search for.	
Max Mismatches	Maximum number of mismatches between a substring and a pattern.	0
Allow Insertions/Deletions	Takes into account possibility of insertions/deletions when searching. By default substitutions are only considered.	False
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

Parameters in Schema File

Type: search

Parameter	Parameter in the GUI	Type
result-name	Annotate as	<i>string</i>
pattern	Pattern(s)	<i>string</i>
max-mismatches-num	Max Mismatches	<i>numeric</i>
allow-ins-del	Allow Insertions/Deletions	<i>boolean</i>
strand	Search in	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 0 - for searching in both strands • 1 - for searching in direct strand • 2 - for searching in complement strand
amino	Search in Translation	<i>boolean</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input data*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

And 1 *output port*:

Name in GUI: *Pattern annotations*

Name in Schema File: out-annotations

Slots:

Slot In GUI	Slot in Schema File	Type
Set of annotations	annotations	<i>annotation-table</i>

15.3.6 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 Schema File

Type: import-phred-qualities

Parameter	Parameter in the GUI	Type
url-in	PHRED input	<i>string</i>
quality-format	Quality format	<i>string</i> Available values are: <ul style="list-style-type: none"> • Sanger • Illumina 1.3+ • Solexa/Illumina 1.0

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *DNA sequences*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

And 1 *output port*:

Name in GUI: *DNA sequences with imported qualities*

Name in Schema File: out-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

15.3.7 Local BLAST Search Element

Finds annotations for the supplied DNA sequence in local BLAST database.

Warning: 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
Database name (required)	Base name for BLAST DB files.	
Database path (required)	Path to the database files.	
Tool path (required)	Path to the BLAST executable.	default
Search type (required)	Selects the type of the BLAST searches.	blastn
Annotate as	Name of the result annotations.	misc_feature
Expected value	Expectation threshold value.	10
Temporary directory (required)	Directory for temporary files.	default

Parameters in Schema File

Type: blast

Parameter	Parameter in the GUI	Type
db-name	Database name	<i>string</i>
db-path	Database path	<i>string</i>
tool-path	Tool path	<i>string</i>
blast-type	Search type	<i>string</i> Available values are: <ul style="list-style-type: none"> • blastn • blastp • blastx • tblastn • tblastx
result-name	Annotate as	<i>string</i>
e-val	Expected value	<i>numeric</i>
temp-dir	Temporary directory	<i>string</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input sequence*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

And 1 *output port*:

Name in GUI: Annotations

Name in Schema File: out-annotations

Slots:

Slot In GUI	Slot in Schema File	Type
Set of annotations	annotations	<i>annotation-table</i>

15.3.8 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.

Warning: 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
Database name (required)	Base name for BLAST DB files.	
Database path (required)	Path to the database files.	
Tool path (required)	Path to a BLAST+ executable.	default
Search type (required)	Selects the type of BLAST searches.	blastn
Annotate as	Name of the result annotations.	misc_feature
Expected value	Expectation threshold value.	10
Temporary directory (required)	Directory for temporary files.	default

Parameters in Schema File

Type: blast-plus

Parameter	Parameter in the GUI	Type
db-name	Database name	<i>string</i>
db-path	Database path	<i>string</i>

tool-path	Tool path	<i>string</i>
blast-type	Search type	<i>string</i> Available values are: <ul style="list-style-type: none"> • blastn • blastp • blastx • tblastn • tblastx
result-name	Annotate as	<i>string</i>
e-val	Expected value	<i>numeric</i>
temp-dir	Temporary directory	<i>string</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input sequence*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

And 1 *output port*:

Name in GUI: *Annotations*

Name in Schema File: out-annotations

Slots:

Slot In GUI	Slot in Schema File	Type
Set of annotations	annotations	<i>annotation-table</i>

15.3.9 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 beyond 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 Schema File

Type: orf-search

Parameter	Parameter in the GUI	Type
result-name	Annotate as	<i>string</i>
strand	Search in	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 0 - for searching in both strands • 1 - for searching in direct strand • 2 - for searching in complement strand

min-length	Min length	<i>numeric</i>
genetic-code	Genetic code	<i>string</i> Available values are: <ul style="list-style-type: none"> • NCBI-GenBank #1 • NCBI-GenBank #2 • etc.
require-init-codon	Require init codon	<i>boolean</i>
require-stop-codon	Require stop codon	<i>boolean</i>
allow-alternative-codons	Allow alternative codons	<i>boolean</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input sequence*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

And 1 *output port*:

Name in GUI: *ORF annotations*

Name in Schema File: out-annotations

Slots:

Slot In GUI	Slot in Schema File	Type
Set of annotations	annotations	<i>annotation-table</i>

15.3.10 Remote BLAST Element

Finds annotations for the supplied DNA sequence in the NCBI remote database.

Parameters in GUI

Parameter	Description	Default value
Database (required)	Selects the database to search through. Available databases are blastn, blastp and cdd.	ncbi-blastn
Annotate as	Name of the result annotations.	
Expected value	This parameter specifies the statistical significance threshold of reporting matches against the database sequences.	10
Max hits	Maximum number of hits.	10
Short sequence	Optimizes search for short sequences.	False
BLAST output	Location of the BLAST output file. This parameter insignificant for cdd search.	

Parameters in Schema File

Type: blast-ncbi

Parameter	Parameter in the GUI	Type
db	Database	<i>string</i> Available values are: <ul style="list-style-type: none"> • ncbi-blastn • ncbi-blastp • ncbi-cdd
result-name	Annotate as	<i>string</i>
e-val	Expected value	<i>string</i>
max-hits	Max hits	<i>numeric</i>
short-sequence	Short sequence	<i>boolean</i>
blast-output	BLAST output	<i>string</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input sequence*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

And 1 *output port*:

Name in GUI: *Annotations*

Name in Schema File: out-annotations

Slots:

Slot In GUI	Slot in Schema File	Type
Set of annotations	annotations	<i>annotation-table</i>

15.3.11 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
Annotate as (required)	Name of the result annotations.	misc_feature
Pattern(s) (required)	Semicolon-separated list of patterns to search for.	
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.	Classic2
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

Parameters in Schema File

Type: ssearch

Parameter	Parameter in the GUI	Type
result-name	Annotate as	<i>string</i>
pattern	Pattern(s)	<i>string</i>
matrix	Substitution Matrix	<i>string</i> Available values are: <ul style="list-style-type: none"> • Auto - for auto detecting matrix • blosum60 • dna • rna • ...
algorithm	Algorithm	<i>string</i> Available values are: <ul style="list-style-type: none"> • Classic 2 • SSE2 • OpenCL • CUDA
filter-strategy	Filter Results	<i>string</i> Available values are: <ul style="list-style-type: none"> • filter-intersections • none
min-score	Min Score	<i>numeric</i>
strand	Search in	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 0 - for searching in both strands • 1 - for searching in direct strand • 2 - for searching in complement strand

amino	Search in Translation	<i>boolean</i>
gap-open-score	Gap Open Score	<i>numeric</i>
gap-ext-score	Gap Extension Score	<i>numeric</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input data*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

And 1 *output port*:

Name in GUI: *Pattern annotations*

Name in Schema File: out-annotations

Slots:

Slot In GUI	Slot in Schema File	Type
Set of annotations	annotations	<i>annotation-table</i>

15.4 Data Converters

15.4.1 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.	<i>Sequence</i>
Sequence alphabet	Alphabet of the sequence. Choose <i>Auto</i> to auto-detect the alphabet or one of the following values: <ul style="list-style-type: none"> • <i>All symbols</i> • <i>Extended DNA</i> • <i>Extended RNA</i> • <i>Standard DNA</i> • <i>Standard RNA</i> • <i>Standard amino</i> 	<i>Auto</i>
Skip unknown symbols	If <i>True</i> , ignores all symbols that are not presented in the sequence alphabet selected.	<i>True</i>
Replace unknown symbols with	Replaces all unknown symbols with the specified symbol.	<i>N</i>

Parameters in Schema File

Type: convert-text-to-sequence

Parameter	Parameter in the GUI	Type
sequence-name	Sequence name	<i>string</i>
alphabet	Alphabet	<i>string</i>
skip-unknown	Skip unknown symbols	<i>boolean</i>
replace-unknown-with	Replace unknown symbols with	<i>string</i> (1 character)

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input text*

Name in Schema File: in-text

Slots:

Slot In GUI	Slot in Schema File	Type
Plain text	text	<i>string</i>

And 1 *output port*:

Name in GUI: *Output sequence*

Name in Schema File: out-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

15.4.2 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 Schema File

Type: reverse-complement

Parameter	Parameter in the GUI	Type
op-type	Operation type	<i>string</i> Available values are: <ul style="list-style-type: none"> • reverse-complement • complement • reverse

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input sequence*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

And 1 *output port*:

Name in GUI: *Output sequence*

Name in Schema File: out-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

15.5 HMMER2 Tools

15.5.1 HMM 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 hmmlcalibrate on the same HMM will give slightly different results. You can use this option to generate reproducible results for different hmmlcalibrate runs on the same HMM.	0

Parameters in Schema File

Type: hmm2-build

Parameter	Parameter in the GUI	Type
profile-name	Profile name	<i>string</i>
strategy	HMM strategy	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 0 - for hmms • 1 - for hmmls • 2 - for hmmsfs • 3 - for hmmsw
calibrate	Calibrate profile	<i>boolean</i>
calibration-threads	Parallel calibration	<i>numeric</i>
deviation	Standard deviation	<i>numeric</i>
fix-samples-length	Fixed length of samples	<i>numeric</i>
mean-samples-length	Mean length of samples	<i>numeric</i>
samples-num	Number of samples	<i>numeric</i>
seed	Random seed	<i>numeric</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input MSA*

Name in Schema File: in-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>

And 1 *output port*:

Name in GUI: *HMM profile*

Name in Schema File: out-hmm2

Slots:

Slot In GUI	Slot in Schema File	Type
HMM profile	hmm2-profile	<i>hmm2-profile</i>

15.5.2 HMM 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 Schema File

Type: hmm2-search

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

Input/Output Ports

The element has 2 *input port*. The first gets the input sequence:

Name in GUI: *Input sequence*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

The second input port gets the HMM profile:

Name in GUI: *HMM profile*

Name in Schema File: in-hmm2

Slots:

Slot In GUI	Slot in Schema File	Type
-------------	---------------------	------

HMM profile	hmm2-profile	<i>hmm2-profile</i>
-------------	--------------	---------------------

And 1 *output port*:

Name in GUI: *HMM annotations*

Name in Schema File: out-annotations

Slots:

Slot In GUI	Slot in Schema File	Type
Set of annotations	annotations	<i>annotation-table</i>

15.5.3 Read HMM 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 Schema File

Type: hmm2-read-profile

Parameter	Parameter in the GUI	Type
url-in	Input files	<i>string</i>

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *HMM profile*

Name in Schema File: out-hmm2

Slots:

Slot In GUI	Slot in Schema File	Type
HMM profile	hmm2-profile	<i>hmm2-profile</i>

15.5.4 Write HMM 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 Schema File

Type: hmm2-write-profile

Parameter	Parameter in the GUI	Type
url-out	Output file	<i>string</i>
write-mode	Existing file	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 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 Schema File: in-hmm2

Slots:

Slot In GUI	Slot in Schema File	Type
HMM profile	hmm2-profile	<i>hmm2-profile</i>
Location	url	<i>string</i>

15.6 Multiple Sequence Alignment

15.6.1 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.

Warning: 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 Schema File

Type: clustalw

Parameter	Parameter in the GUI	Type
matrix	Weight matrix	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 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	<i>boolean</i>
gap-distance	Gap distance	<i>numeric</i>
gap-ext-penalty	Gap extension penalty	<i>numeric</i>
gap-open-penalty	Gap open penalty	<i>numeric</i>
no-hydrophilic-gaps	Hydrophilic gaps off	<i>boolean</i>
no-residue-specific-gaps	Residue-specific gaps off	<i>boolean</i>
iteration-type	Iteration type	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 0 - for None • 1 - for Tree • 2 - for Alignment
iterations-max-num	Number of iterations	<i>numeric</i>
path	Tool path	<i>string</i>
temp-dir	Temporary directory	<i>string</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input MSA*

Name in Schema File: in-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>

And 1 *output port*:

Name in GUI: *ClustalW result MSA*

Name in Schema File: out-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>

15.6.2 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 on <http://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 Schema File

Type: kalign

Parameter	Parameter in the GUI	Type
gap-ext-penalty	Gap extension penalty	<i>numeric</i>
gap-open-penalty	Gap open penalty	<i>numeric</i>
terminal-gap-penalty	Terminal gap penalty	<i>numeric</i>
bonus-score	Bonus score	<i>numeric</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input MSA*

Name in Schema File: in-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>

And 1 *output port*:

Name in GUI: *Kalign result MSA*

Name in Schema File: out-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>

15.6.3 Align with MAFFT Element

Originally, MAFFT is a multiple sequence alignment program for unix-like operating systems. Currently, Windows version is also available.

Warning: 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 Schema File

Type: mafft

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

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input MSA*

Name in Schema File: in-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>

And 1 *output port*:Name in GUI: *Multiple sequence alignment*

Name in Schema File: out-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>

15.6.4 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 rearranges sequences so that similar sequences are adjacent in the output file. This makes the alignment easier to evaluate by eye.	True

Parameters in Schema File

Type: muscle

Parameter	Parameter in the GUI	Type
mode	Mode	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 0 - for MUSCLE default • 1 - for Large alignment • 2 - for Refine only

stable	Stable order	<i>boolean</i>
--------	--------------	----------------

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input MSA*

Name in Schema File: in-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>

And 1 *output port*:

Name in GUI: *Multiple sequence alignment*

Name in Schema File: out-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>

15.6.5 Align with T-Coffee Element

T-Coffee is a multiple sequence alignment package.

Warning: 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 Schema File

Type: tcoffee

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

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input MSA*

Name in Schema File: in-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>

And 1 *output port*:

Name in GUI: *Multiple sequence alignment*

Name in Schema File: out-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>

15.6.6 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 Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

And 1 *output port*:

Name in GUI: *Result alignment*

Name in Schema File: out-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>

15.6.7 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 Schema File: in-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>

And 1 *output port*:

Name in GUI: *Output sequences*

Name in Schema File:

Slots: out-sequence

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

15.7 Transcription Factor

15.7.1 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 Schema File

Type: fmatrix-build

Parameter	Parameter in the GUI	Type
type	Matrix type	<i>boolean</i> Available values are: <ul style="list-style-type: none"> • true - for Dinucleic • false - for Mononucleic

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input alignment*

Name in Schema File: in-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>

And 1 *output port*:

Name in GUI: *Frequency matrix*

Name in Schema File: out-fmatrix

Slots:

Slot In GUI	Slot in Schema File	Type
-------------	---------------------	------

Frequency matrix	fmatrix	<i>fmatrix</i>
------------------	---------	----------------

15.7.2 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 Schema File

Type: fmatrix-read

Parameter	Parameter in the GUI	Type
url-in	Input files	<i>string</i>

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *Frequency matrix*

Name in Schema File: out-fmatrix

Slots:

Slot In GUI	Slot in Schema File	Type
Frequency matrix	fmatrix	<i>fmatrix</i>

15.7.3 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 Schema File

Type: fmatrix-write

Parameter	Parameter in the GUI	Type
url-out	Output file	<i>string</i>
write-mode	Existing file	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 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 Schema File: in-fmatrix

Slots:

Slot In GUI	Slot in Schema File	Type
Frequency matrix	fmatrix	<i>fmatrix</i>
Source URL	url	<i>string</i>

15.7.4 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 Schema File

Type: wmatrix-build

Parameter	Parameter in the GUI	Type
type	Matrix type	<i>boolean</i> Available values are: <ul style="list-style-type: none"> • true - for Dinucleic • false - for Mononucleic
weight-algorithm	Weight algorithm	<i>string</i> Available values are: <ul style="list-style-type: none"> • Berg and von Hippel • Log-odds • Match • NLG

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input alignment*

Name in Schema File: in-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>

And 1 *output port*:

Name in GUI: *Weight matrix*

Name in Schema File: out-wmatrix

Slots:

Slot In GUI	Slot in Schema File	Type
Weight matrix	wmatrix	<i>wmatrix</i>

15.7.5 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 Schema File

Type: wmatrix-read

Parameter	Parameter in the GUI	Type
url-in	Input files	<i>string</i>

Input/Output Ports

And 1 *output port*:

Name in GUI: *Weight matrix*

Name in Schema File: out-wmatrix

Slots:

Slot In GUI	Slot in Schema File	Type
Weight matrix	wmatrix	<i>wmatrix</i>

15.7.6 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 Schema File

Type: wmatrix-write

Parameter	Parameter in the GUI	Type
url-out	Output file	<i>string</i>
write-mode	Existing file	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 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 Schema File: in-wmatrix

Slots:

Slot In GUI	Slot in Schema File	Type
Weight matrix	wmatrix	<i>wmatrix</i>
Source URL	url	<i>string</i>

15.7.7 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 Schema File

Type: fmatrix-to-wmatrix

Parameter	Parameter in the GUI	Type
type	Matrix type	<i>boolean</i> Available values are: <ul style="list-style-type: none"> • true - for Dinucleic • false - for Mononucleic
weight-algorithm	Weight algorithm	<i>string</i> Available values are: <ul style="list-style-type: none"> • Berg and von Hippel • Log-odds • Match • NLG

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Frequency matrix*

Name in Schema File: in-fmatrix

Slots:

Slot In GUI	Slot in Schema File	Type
Frequency matrix	fmatrix	<i>fmatrix</i>

And 1 *output port*:

Name in GUI: *Weight matrix*

Name in Schema File: out-wmatrix

Slots:

Slot In GUI	Slot in Schema File	Type
Weight matrix	wmatrix	<i>wmatrix</i>

15.7.8 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 Schema File

Type: wmatrix-search

Parameter	Parameter in the GUI	Type
result-name	Result annotation	<i>string</i>
strand	Search in	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 0 - for searching in both strands • 1 - for searching in direct strand • 2 - for searching in complement strand
min-score	Min score	<i>numeric</i>

Input/Output Ports

The element has 2 *input ports*. The first port:

Name in GUI: *Sequence*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

The second input port gets the SITECON model:

Name in GUI: *Weight matrix*

Name in Schema File: in-wmatrix

Slots:

Slot In GUI	Slot in Schema File	Type
Weight matrix	wmatrix	wmatrix

And there is 1 *output port*:

Name in GUI: *Weight matrix annotations*

Name in Schema File: out-annotations

Slots:

Slot In GUI	Slot in Schema File	Type
Set of annotations	annotations	annotation-table

15.7.9 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 Schema File

Type: sitecon-build

Parameter	Parameter in the GUI	Type
weight-algorithm	Weight algorithm	<i>boolean</i> Available values are: <ul style="list-style-type: none"> • 0 - for None • 1 - for Algorithm2
window-size	Window size, bp	<i>numeric</i>
calibrate-length	Calibration length	<i>numeric</i>
seed	Random seed	<i>numeric</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input alignment*

Name in Schema File: in-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>
Origin	url	<i>string</i>

And 1 *output port*:

Name in GUI: *Sitecon model*

Name in Schema File: out-sitecon

Slots:

Slot In GUI	Slot in Schema File	Type
Sitecon model	sitecon-model	<i>sitecon-model</i>

15.7.10 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 Schema File

Type: sitecon-read

Parameter	Parameter in the GUI	Type
url-in	Input files	<i>string</i>

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *Sitecon model*

Name in Schema File: out-sitecon

Slots:

Slot In GUI	Slot in Schema File	Type
Sitecon model	sitecon-model	<i>sitecon-model</i>

15.7.11 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 Schema File

Type: sitecon-write

Parameter	Parameter in the GUI	Type
url-out	Output file	<i>string</i>
write-mode	Existing file	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 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 Schema File: in-sitecon

Slots:

Slot In GUI	Slot in Schema File	Type
Sitecon model	sitecon-model	<i>sitecon-model</i>
Source URL	url	<i>string</i>

15.7.12 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 Schema File

Type: sitecon-search

Parameter	Parameter in the GUI	Type
result-name	Result annotation	<i>string</i>
strand	Search in	<i>numeric</i> Available values are: <ul style="list-style-type: none"> • 0 - for searching in both strands • 1 - for searching in direct strand • 2 - for searching in complement strand
min-score	Min score	<i>numeric</i>
err1	Min err1	<i>numeric</i>
err2	Max err2	<i>numeric</i>

Input/Output Ports

The element has 2 *input ports*. The first port:

Name in GUI: *Sequence*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

The second input port gets the SITECON model:

Name in GUI: *Sitecon model*

Name in Schema File: in-sitecon

Slots:

Slot In GUI	Slot in Schema File	Type
Sitecon model	sitecon-model	<i>sitecon-model</i>

And there is 1 *output port*:

Name in GUI: *Sitecon annotations*

Name in Schema File: out-annotations

Slots:

Slot In GUI	Slot in Schema File	Type
Set of annotations	annotations	<i>annotation-table</i>

15.8 Utils

15.8.1 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 <i>Reference</i> 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 <i>A</i> , <i>C</i> , <i>G</i> , <i>T</i> are used to generate the sequence. Otherwise, the <i>GC Skew</i> 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 <i>G</i> is the number of G's in the window, and <i>C</i> is the number of C's.	0.25

Parameters in Schema File

Type: generate-dna

Parameter	Parameter in the GUI	Type
length	Lenght	<i>numeric</i>
count	Count	<i>numeric</i>
seed	Seed	<i>numeric</i>
content	Content	<i>string</i>
algorithm	Algorithm	<i>string</i> Available values are: <ul style="list-style-type: none"> • gc-content • gc-skew
window-size	Window size	<i>numeric</i>
reference-url	Reference	<i>string</i> Available values are: <ul style="list-style-type: none"> • manual • reference
percent-a	A	<i>numeric</i>
percent-c	C	<i>numeric</i>
percent-g	G	<i>numeric</i>
percent-t	T	<i>numeric</i>
gc-skew	GC Skew	<i>numeric</i>

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *Sequences*

Name in Schema File: out-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

15.9 Custom Elements Examples

15.9.1 Create Alignment Element

Creates an alignment from sequences with length less than **len**.

The element works on the basis of the following script:

```
var length;
length = size(in_sequence);
if (length < len) {
    out_msa = addToAlignment(out_msa, in_sequence);
}
```

Parameters in GUI

Parameter	Description	Default value
len	Ignores sequences shorter than the specified value.	

Parameters in Schema File

Type: "Script-Create alignment example"

Parameter	Parameter in the GUI	Type
len	len	<i>numeric</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input data*

Name in Schema File: in

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

And 1 *output port*:

Name in GUI: *Output data*

Name in Schema File: out

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>

15.9.2 Dump Sequence Info Element

For each incoming sequences, dumps to output the sequence name and the sequence size.

The element works on the basis of the following script:

```
out_text=getName(in_sequence) + ": " + size(in_sequence);
```

Parameters in GUI

The element has no parameters.

Parameters in Schema File

Type: "Script-Dump sequence info"

The element has no parameters.

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input data*

Name in Schema File: in

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

And 1 *output port*:

Name in GUI: *Output data*

Name in Schema File: out

Slots:

Slot In GUI	Slot in Schema File	Type
Plain text	text	<i>string</i>

15.9.3 Quality Filter Element

Filters sequences by their qualities.

The element works on the basis of the following script:

```

var qual;

if(hasQuality(in_sequence)) {
    qual = getMinimumQuality(in_sequence);
    if(qual >= quality) {
        out_sequence = in_sequence;
    }
}

```

Parameters in GUI

Parameter	Description	Default value
quality	Quality used to filter.	

Parameters in Schema File

Type: "Script-Quality filter example"

Parameter	Parameter in the GUI	Type
quality	quality	<i>numeric</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input data*

Name in Schema File: in

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

And 1 *output port*:

Name in GUI: *Output data*

Name in Schema File: out

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

15.10 Data Flow

15.10.1 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 Schema File

Type: filter-by-values

Parameter	Parameter in the GUI	Type
text	Filter by value(s)	<i>string</i>

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Input values*

Name in Schema File: in-data

Slots:

Slot In GUI	Slot in Schema File	Type
Input values	text	<i>string</i>

The element has 1 *output port*:

Name in GUI: *Passing values (by Filter)*

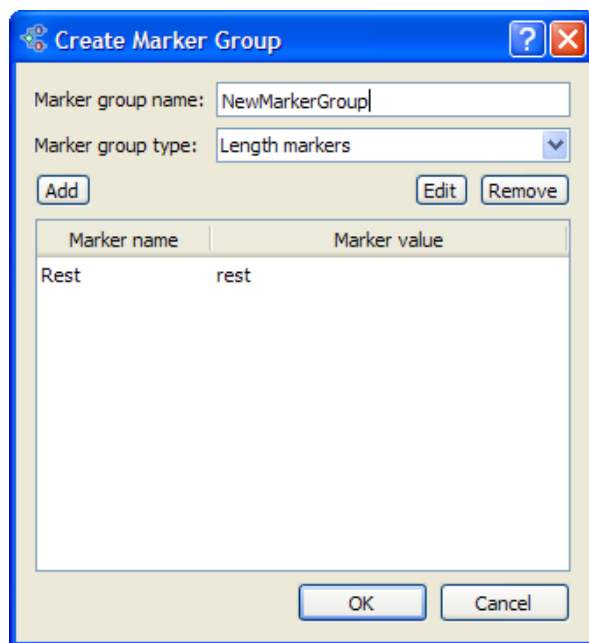
Name in Schema File: filtered-data

15.10.2 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.

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.

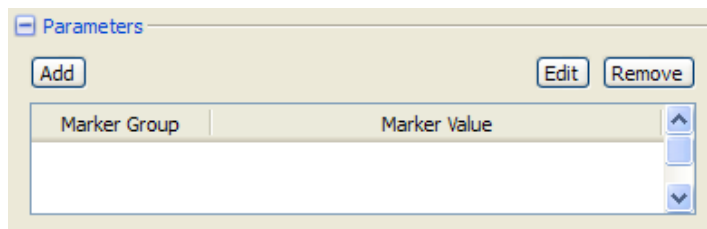
Qualifier float value markers — marks a sequence by the number of float qualifiers.

Text markers — marks a sequence by text. For example, if the sequence:

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 Schema File

Type: mark-sequence

Input/Output Ports

The element has 1 *input port*:

Name in GUI: *Sequence*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>
Location	url	<i>string</i>
Set of annotations	annotations	<i>annotation-table-list</i>

The element has 1 *output port*.

Name in GUI: *Marked sequence*

Name in Schema File: out-marked-seq

Slots:

Each created marker group adds a text slot with the following properties:

Slot In GUI	Slot in Schema File	Type
Name of the marker group	Name of the marker group	<i>string</i>

15.11 DNA Assembly

15.11.1 Bowtie Aligner Element

An ultrafast memory-efficient short read aligner, <http://bowtie-bio.sourceforge.net>

Parameters in GUI

Parameter	Description	Default value
-n alignment mode	Alignments may have no more than N mismatches (where N is a number 0-3, set with -n) in the first L bases (where L is a number 5 or greater, set with -l) on the high-quality (left) end of the read. The first L bases are called the "seed".	2
-v alignment mode	"-1" - use default value. Report alignments with at most mismatches. -e and -l options are ignored and quality values have no effect on what alignments are valid. -v is mutually exclusive with -n.	-1
Do not align against the forward reference strand	Bowtie "--nofw" option. If --nofw is specified, bowtie will not attempt to align against the forward reference strand.	False
Do not align against the reverse-complement reference strand	Bowtie "--norc" option. If --norc is specified, bowtie will not attempt to align against the reverse-complement reference strand.	False
Skip Maq quality rounding	Bowtie "--nomaqround" option. 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
Maximum number of backtracks permitted	Bowtie "--maxbts" option. The maximum number of backtracks permitted when aligning a read in -n 2 or -n 3 mode (default: 125 without --best, 800 with --best). A "backtrack" is the introduction of a speculative substitution into the alignment. Without this limit, the default parameters will sometimes require that bowtie try 100s or 1,000s of backtracks to align a read, especially if the read has many low-quality bases and/or has no valid alignments, slowing bowtie down significantly. However, this limit may cause some valid alignments to be missed. Higher limits yield greater sensitivity at the expense of longer running times.	-1

Parameter	Description	Default value
Maximum permitted total of quality values	Bowtie "-e/-maqerr" option. Maximum permitted total of quality values at all mismatched read positions throughout the entire alignment, not just in the "seed".	70
Chunk Mbs	Bowtie "-chunkmbs" option. The number of megabytes of memory a given thread is given to store path descriptors in -best mode.	64
Pseudo random seed number	Bowtie "-seed" option. Use as the seed for pseudo-random number generator. "-1" - use pseudo random.	-1
Seed length	Bowtie "-l/-seedlen" option. 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. bowtie is faster for larger values of -l.	28
Tryhard	Bowtie "-y/-tryhard" option. Try as hard as possible to find valid alignments when they exist, including paired-end alignments.	False
Align reads with "best" option	Guarantees 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
Report about all valid alignments	Reports all valid alignments per read or pair.	False

Parameters in Schema File

Type: bowtie

Parameter	Parameter in the GUI	Type
mismatches-num	-n alignment mode	<i>numeric</i>
report-with-mismatches	-v alignment mode	<i>numeric</i>
no-forward	Do not align against the forward reference strand	<i>boolean</i>
no-reverse-complemented	Do not align against the reverse-complement reference strand	<i>boolean</i>
no-maq-rounding	Skip Maq quality rounding	<i>boolean</i>

max-backtracks	Maximum number of backtracks permitted	<i>numeric</i>
maq-err	Maximum permitted total of quality values	<i>numeric</i>
chunk-mbs	Chunk Mbs	<i>numeric</i>
seed	Pseudo random seed number	<i>numeric</i>
seed-len	Seed length	<i>numeric</i>
try-hard	Tryhard	<i>boolean</i>
best	Align reads with "best" option	<i>boolean</i>
all	Report about all valid alignments	<i>boolean</i>

Input/Output Ports

The element has 2 *input ports*. The first port supplies the element with the Bowtie index:

Name in GUI: *EBWT index*

Name in Schema File: in-ebwt

Slots:

Slot In GUI	Slot in Schema File	Type
EBWT index	in-ebwt	<i>ebwt-index</i>

The second input port is used to input the short reads:

Name in GUI: *Short read sequences*

Name in Schema File: in-sequence

Slots:

Slot In GUI	Slot in Schema File	Type
Sequence	sequence	<i>sequence</i>

There is an *output port*:

Name in GUI: *Short reads alignment*

Name in Schema File: out-msa

Slots:

Slot In GUI	Slot in Schema File	Type
MSA	msa	<i>msa</i>

15.11.2 Bowtie Build Indexer Element

Bowtie-build builds a Bowtie index from a set of DNA sequences. It outputs a set of 6 files with suffixes .1.ebwt, .2.ebwt, .3.ebwt, .4.ebwt, .rev.1.ebwt, and .rev.2.ebwt. These files together constitute the index: they are all that is needed to align reads to that reference. The original sequence files are no longer used by Bowtie once the index is built.

Parameters in GUI

Parameter	Description	Default value
EBWT (required)	Output index url.	
Reference (required)	Reference sequence url. The short reads will be aligned to this reference genome.	

Parameters in Schema File

Type: bowtie-build-index

Parameter	Parameter in the GUI	Type
url-ebwt	EBWT	<i>string</i>
url-reference	Reference	<i>string</i>

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *EBWT index*

Name in Schema File: out-ebwt

Slots:

Slot In GUI	Slot in Schema File	Type
EBWT index	ebwt-index	<i>ebwt-index</i>

15.11.3 Bowtie Index Reader Element

Reads a set of 6 files with suffixes .1.ebwt, .2.ebwt, .3.ebwt, .4.ebwt, .rev.1.ebwt, and .rev.2.ebwt. These files together constitute the index: they are all that is needed to align reads to that reference.

Parameters in GUI

Parameter	Description	Default value
EBWT (required)	Select one of the index files.	

Parameters in Schema File

Type: bowtie-read-index

Parameter	Parameter in the GUI	Type
url-ebwt	EBWT	<i>string</i>

Input/Output Ports

The element has 1 *output port*:

Name in GUI: *EBWT index*

Name in Schema File: out-ebwt

Slots:

Slot In GUI	Slot in Schema File	Type
EBWT index	ebwt-index	<i>ebwt-index</i>

16 Workflow Samples

This section contains detailed description of workflow samples presented in the Workflow Designer.

16.1 Data Marking

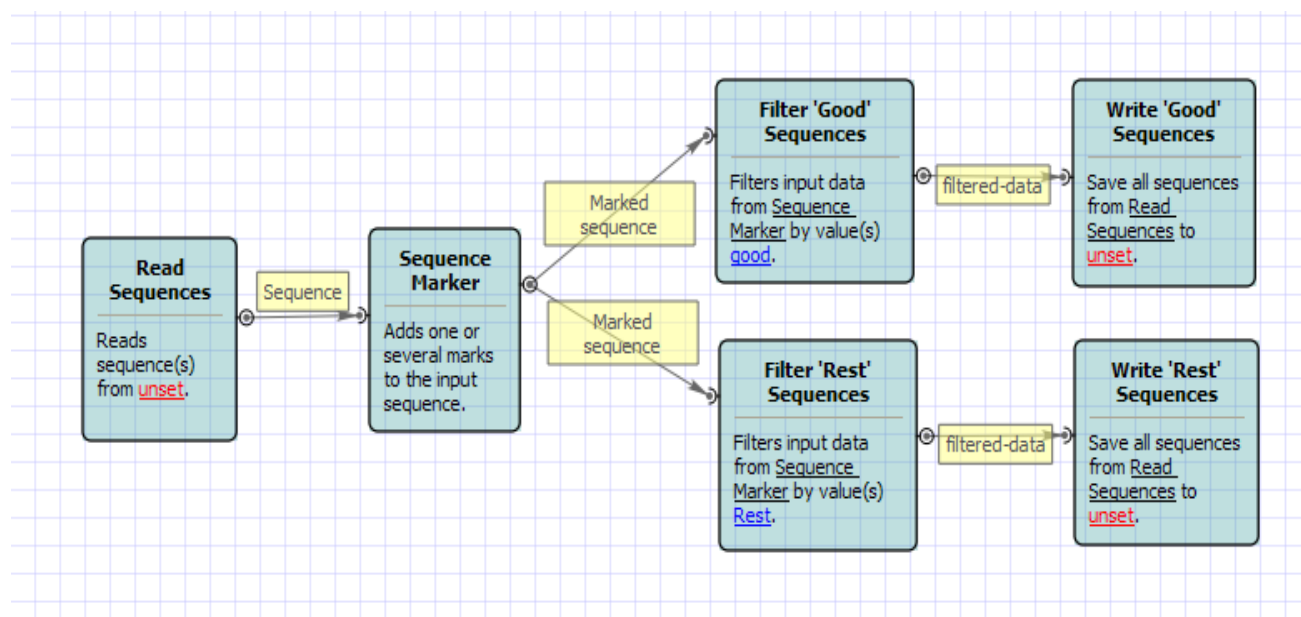
16.1.1 Marking Sequences by Annotation Number

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.

To try out this sample, add the input files to the *Read Sequence* element, select the name and location of the output files in the *Write Sequence* elements (*Write "Good" Sequences* and *Write "Rest" Sequences*) and run the schema.



16.1.2 Marking Sequences by Length

This sample describes how to identify sequences with the specified length.

First, the schema 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.

To try out this sample, add the input files to the *Read Sequence* element, select the name and location of the output files in the *Write Sequence* elements (*Write "Short" Sequences* and *Write "Long" Sequences*) and run the schema.

