Tutorial

Using CoEd
The EGene's Configuration Editor

Tutorial: using CoEd

The goal of this tutorial is to explain how to use the graphical tool CoEd to configure and run an EGene pipeline. We will:

- explain how to start CoEd and the various parts of the graphical window;
- configure a pipeline with four steps;
- run the pipeline directly from CoEd;
- produce an EGene Configuration file and run the same pipeline from command line;
- configure a new pipeline using parts of the old one.

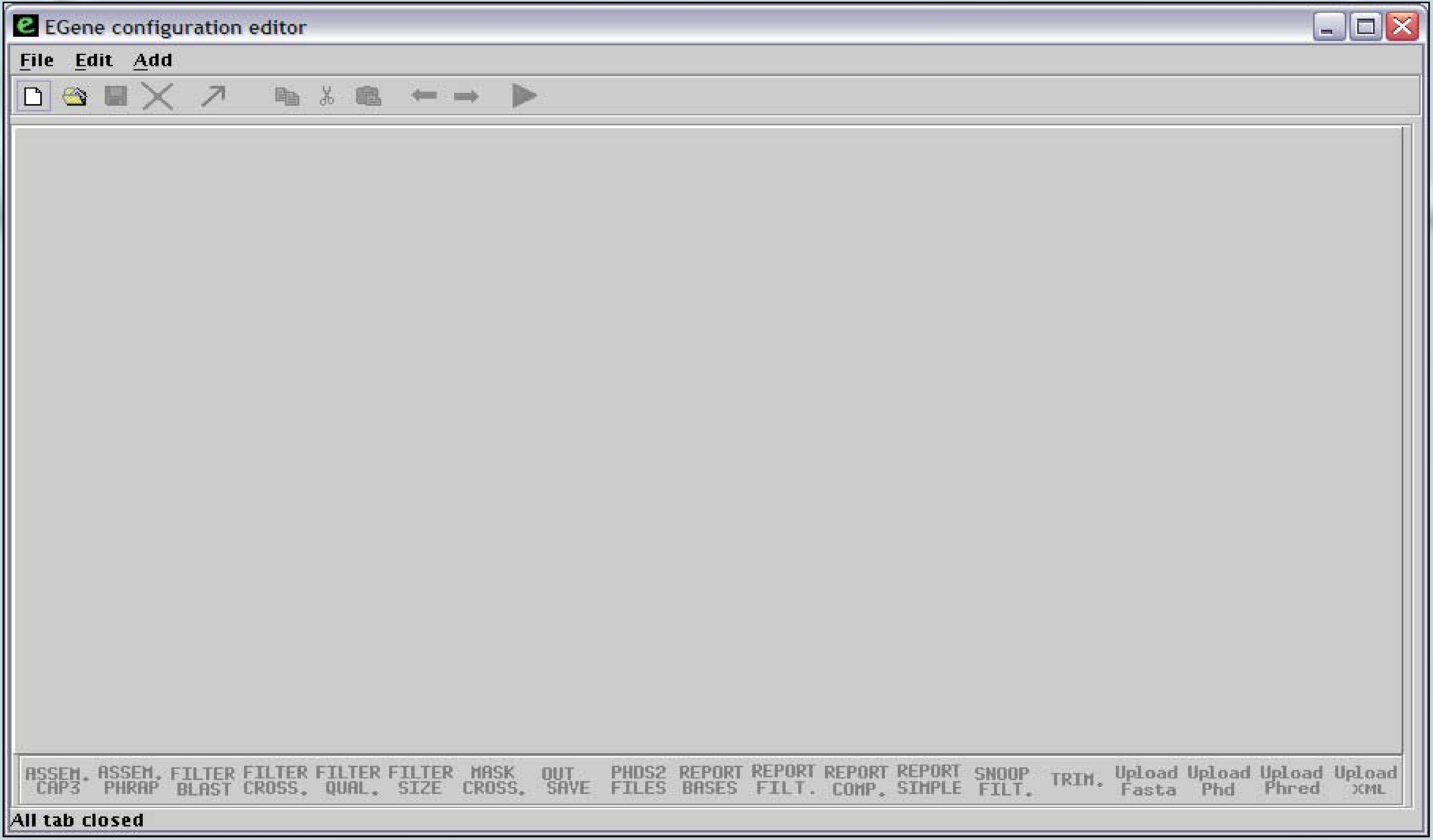
Starting CoEd

- Initially, we need to start CoEd with the command:

```
coed.pl
```

- This is a Perl script that will start the Java application. The graphic window of CoEd will appear:

Starting CoEd

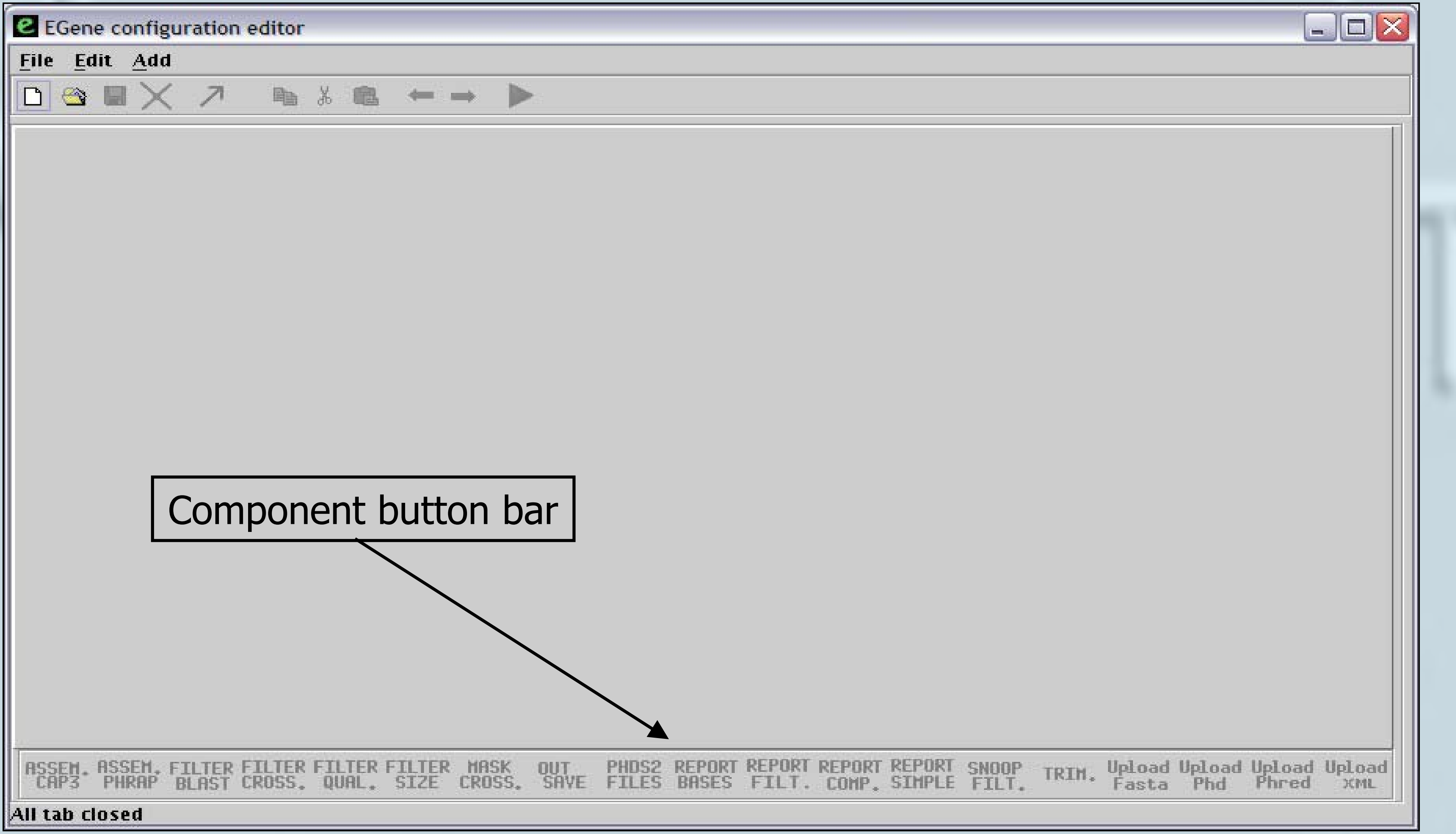


All tab closed

Explaining CoEd: component buttons

- The bottom part of the window contains a bar with buttons representing the components that can be used for the pipeline steps. Here CoEd is configured to show 19 components:

Explaining CoEd: component buttons



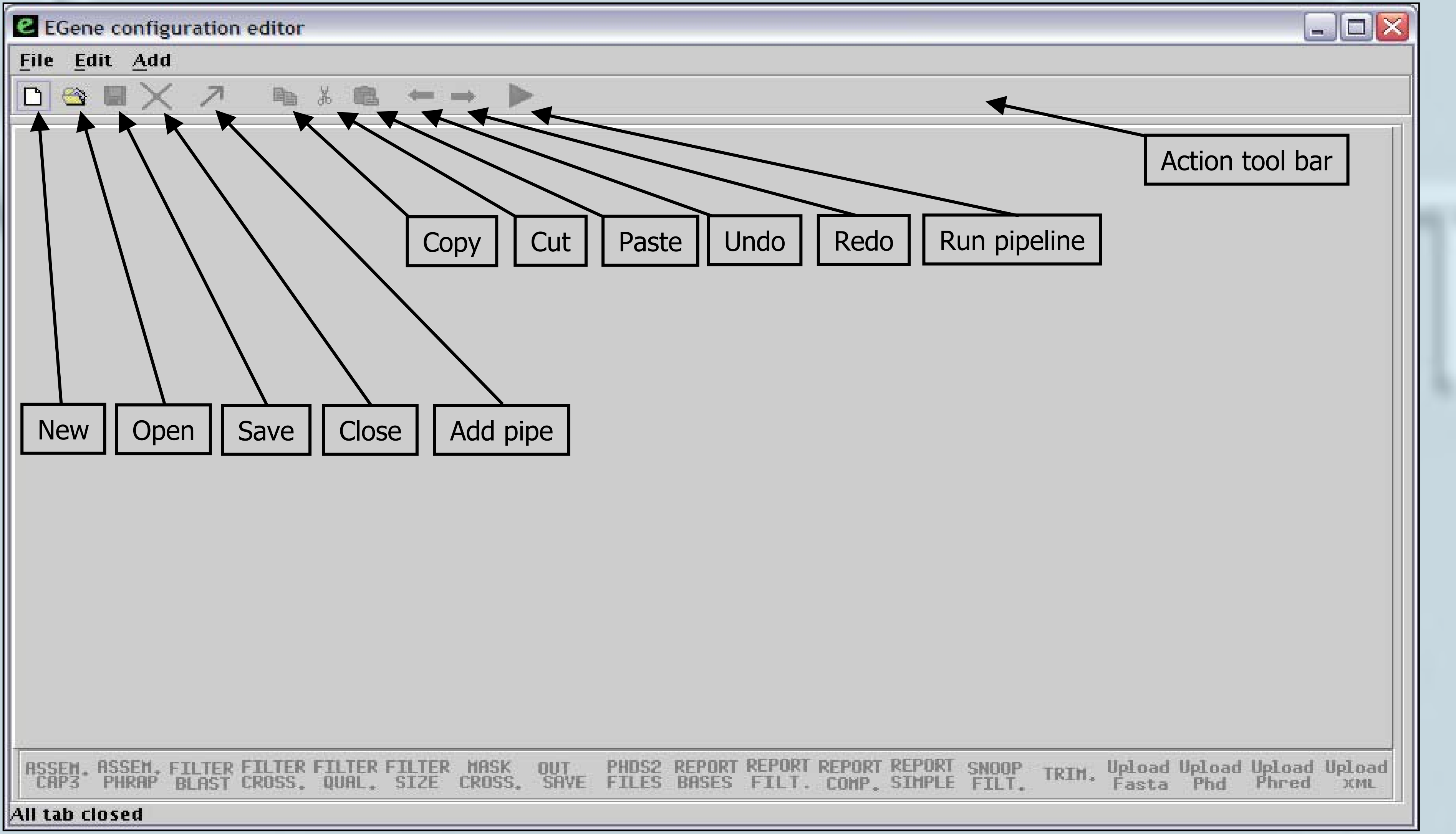
All tab closed

Component button bar

Explaining CoEd: the tool bar

- On the top of the canvas we have a tool bar with 10 action icons: new, open, save, close, add pipe, copy, cut, paste, undo, redo, and run pipeline.

Explaining CoEd: the tool bar



All tab closed

Action tool bar

Copy

Cut

Paste

Undo

Redo

Run pipeline

New

Open

Save

Close

Add pipe

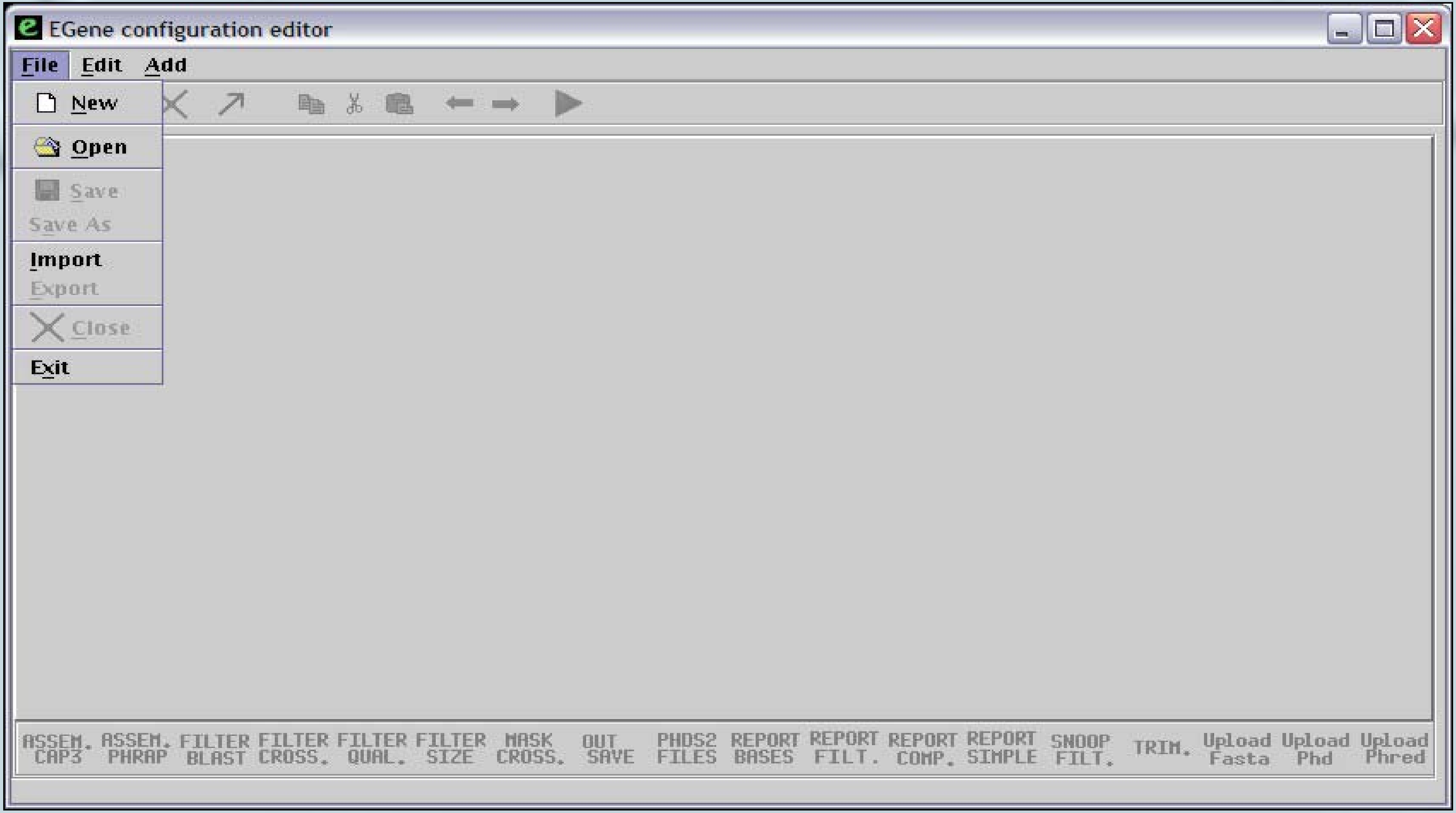
Explaining CoEd: pull down menus

The top part of the window presents 3 pull down menus, offering all the functionality available with the buttons and some extra ones:

- **File**: new, open, save, save as, import, export, close, exit
- **Edit**: copy, cut, paste, undo, redo, clear selection, select all
- **Add**: list all components available for placing on the canvas

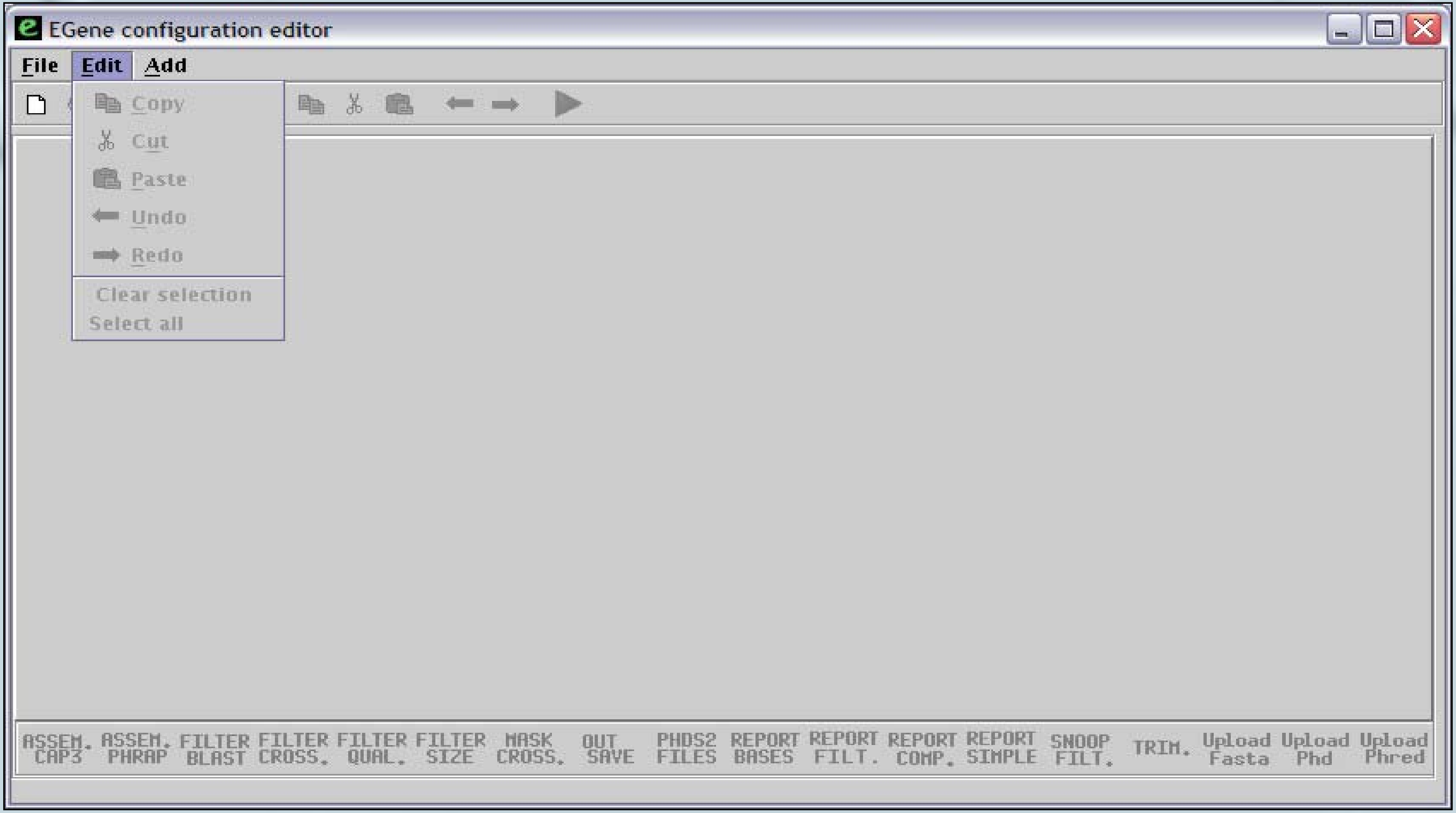
Explaining CoEd: pull down menus

File: new, open, save, save as, import, export, close, exit



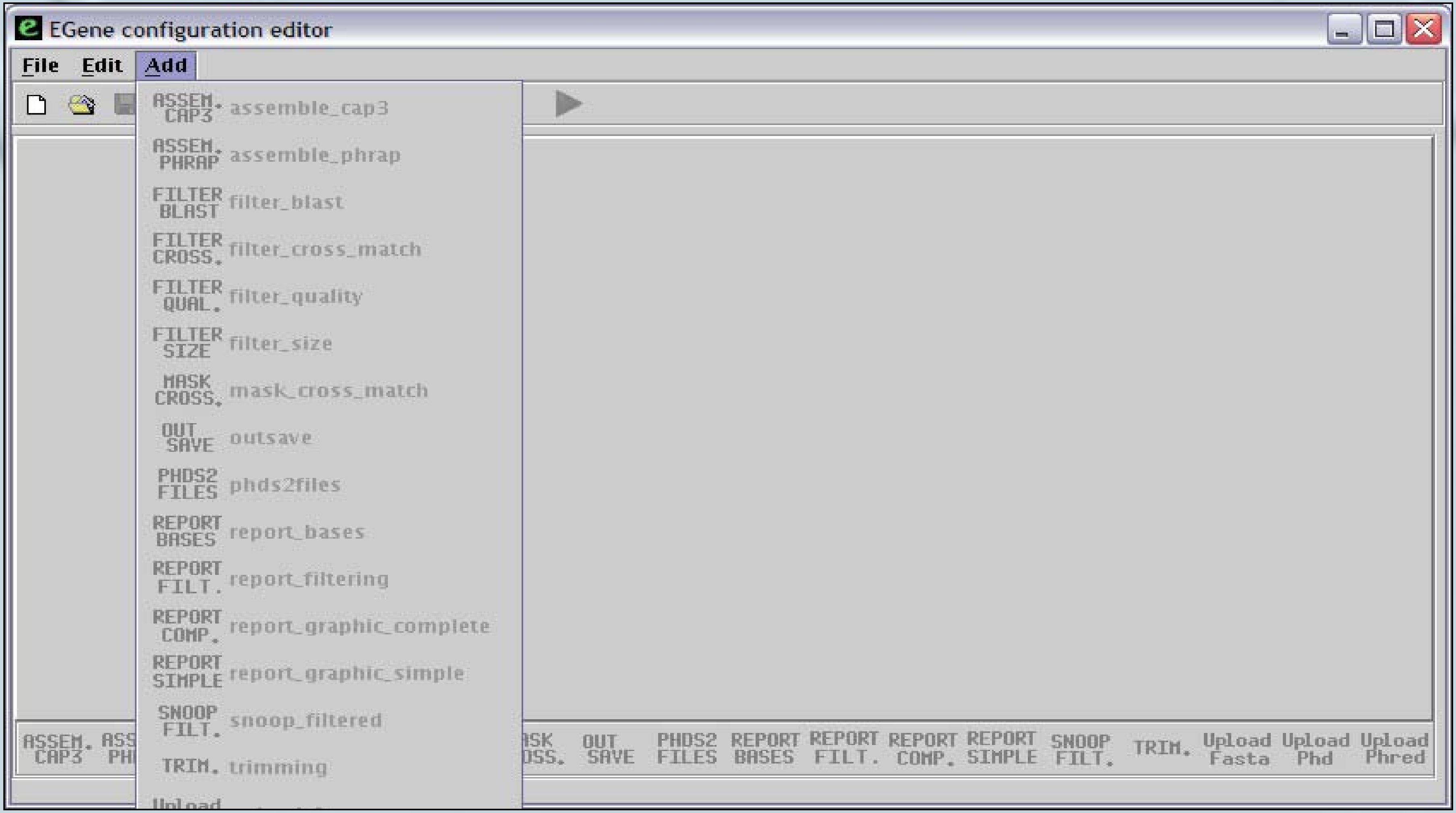
Explaining CoEd: pull down menus

Edit: copy, cut, paste, undo, redo, clear selection, select all



Explaining CoEd: pull down menus

Add: list all components available for placing on the canvas



Creating a Pipeline

Now we will configure a pipeline with 4 steps:

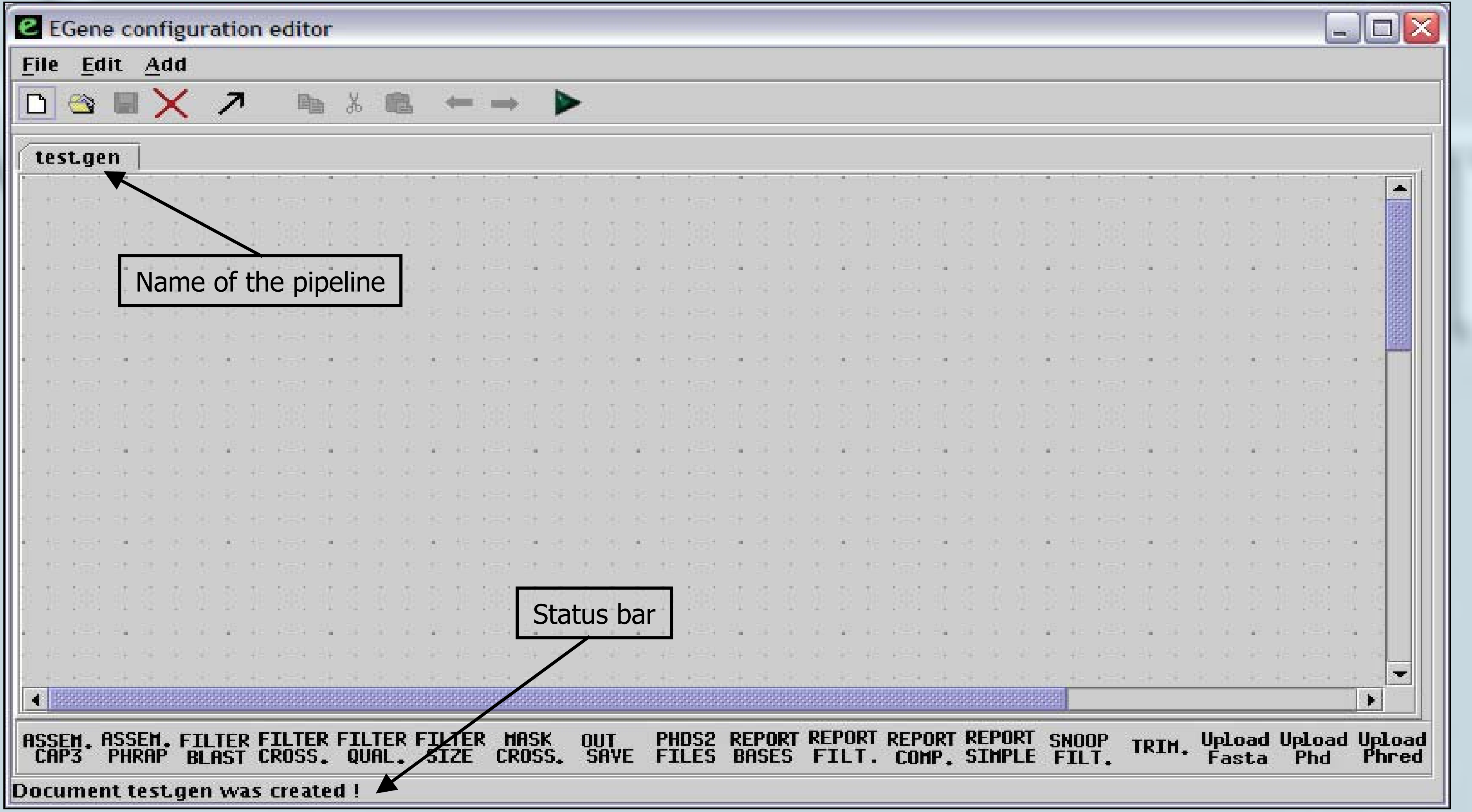
- Upload sequences from chromatogram files to the pipeline;
- Trim the low quality regions of the sequences;
- Look for sequences that match a database with ribosomal sequences, eliminating them from the reads being processed (in EGene, invalidating them);
- Generate a graphic report showing the quality of the remaining sequences.

Creating a Pipeline

First of all: creating a new pipeline

- Click on the New icon. 🗋
- … or on the File New option from the pull down menu.
- Enter a name for the pipeline (i.e. test). This name will appear at the upper left corner tab.
- The bottom status bar will display a message warning that a new pipeline has been created.

Creating a Pipeline



Document test.gen was created !

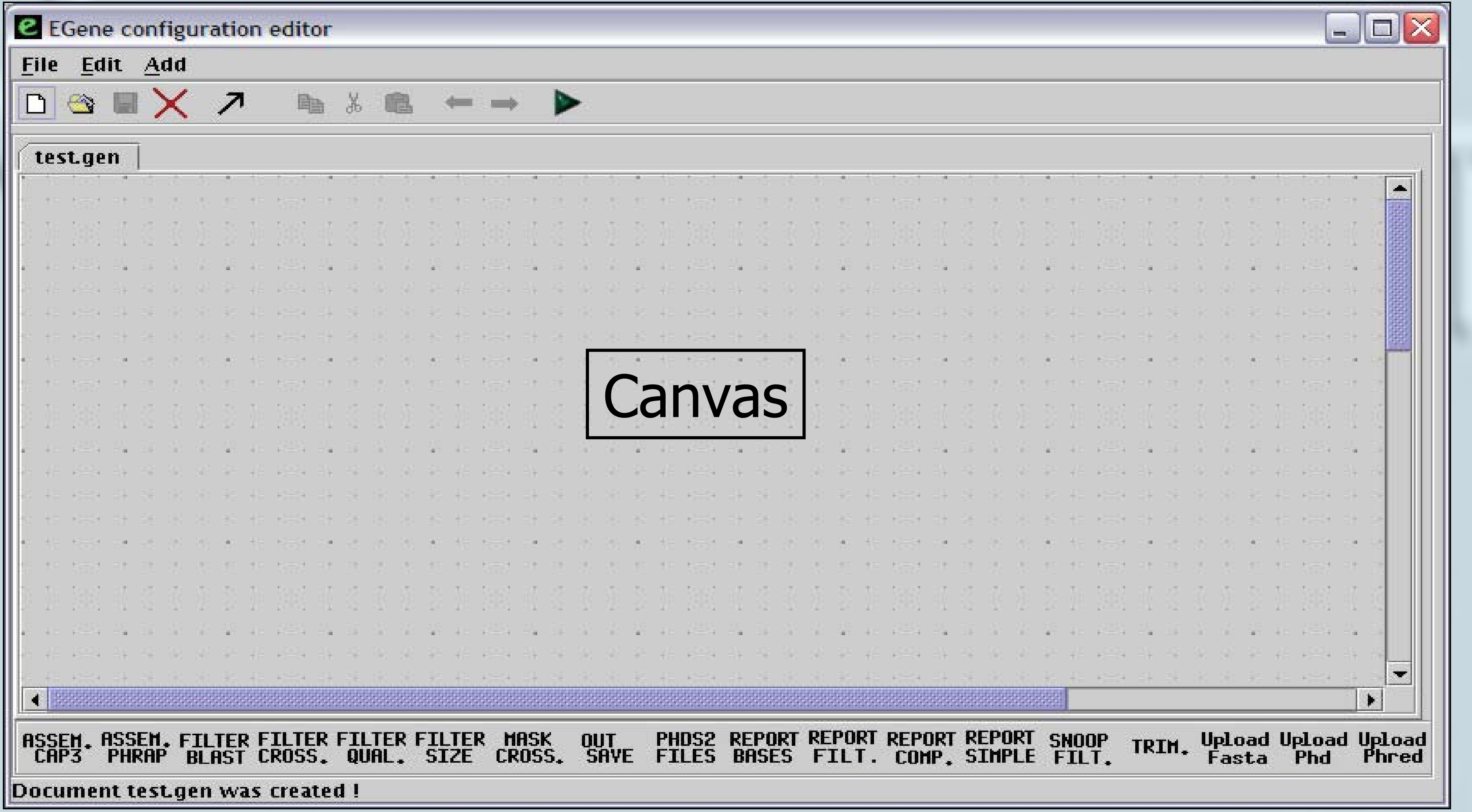
Name of the pipeline

Status bar

Explaining CoEd: the canvas

- The canvas is the place where icons representing the pipeline steps (nodes) will be placed and manipulated.

Explaining CoEd: the canvas



Document test.gen was created !

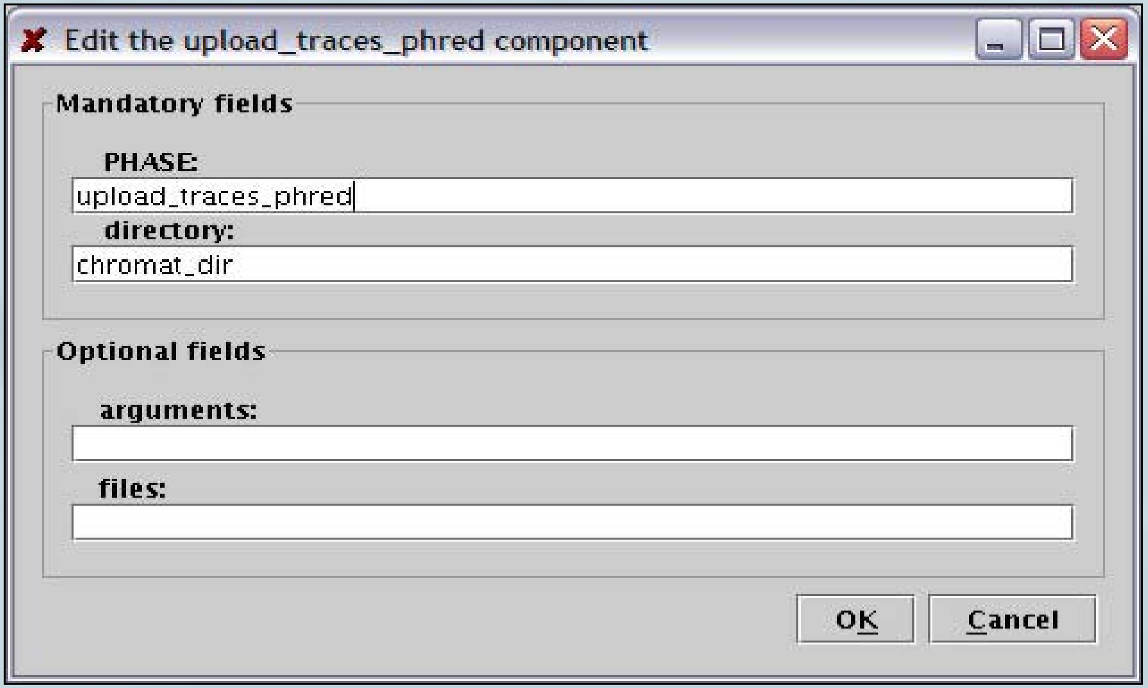
Configuring the Pipeline: upload traces

- For each step we should select the component from the component button bar or from the pull-down menu.
- After selecting the component, click the left mouse button on the place in the canvas you want the component to be placed.
- A window with the configuration parameters will open up.

Configuring the Pipeline: upload traces

Upload_traces only has two mandatory parameters:

- Phase name (the text that will show above the icon)
- The name of the directory containing the chromatograms to be uploaded



Edit the upload_traces_phred component

Mandatory fields

PHASE: upload_traces_phred

directory: chromat_dir

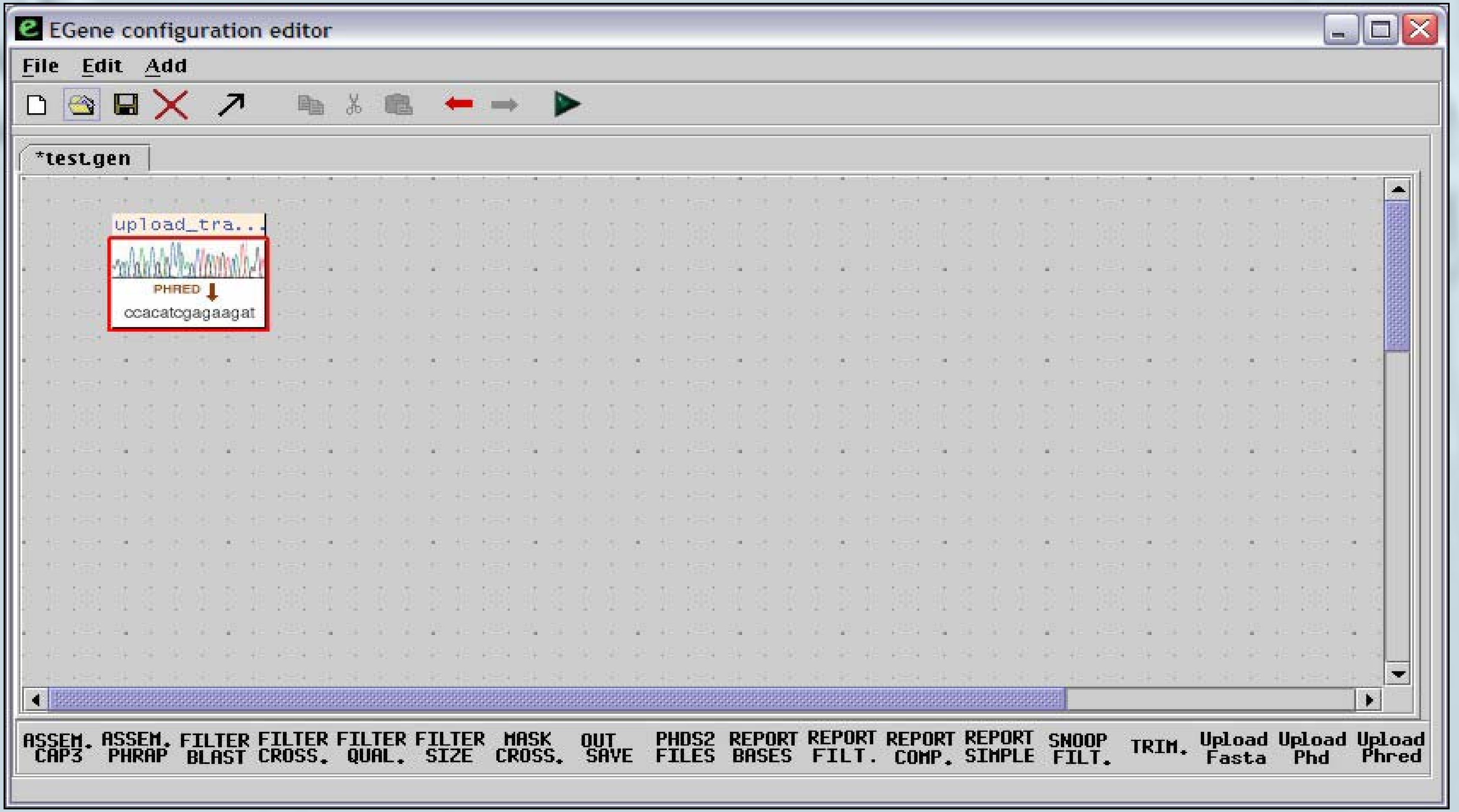
Optional fields

arguments:

files:

Configuring the Pipeline:upload

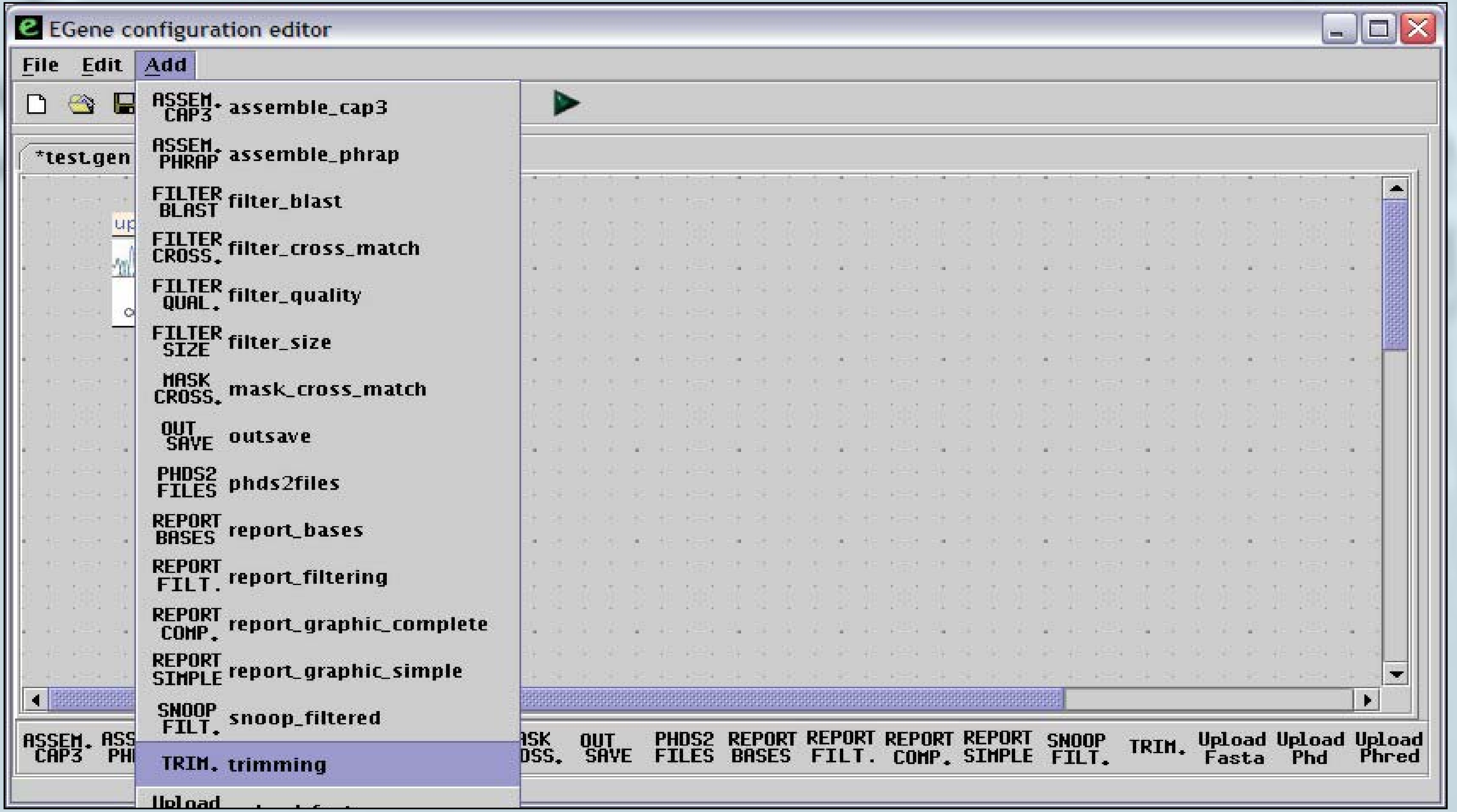
After clicking OK, the canvas will display the new component:



Configuring the Pipeline: trimming

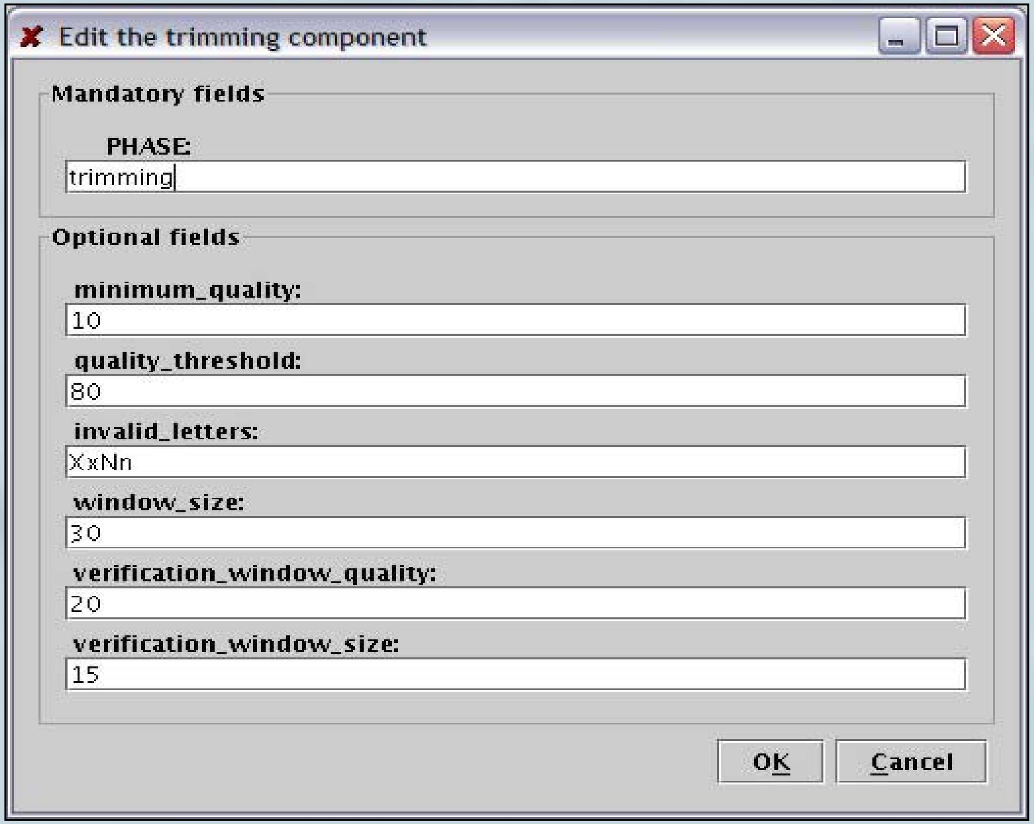
- The next step is to trim the low quality parts of the sequence.
- We use the component `trimming.pl`.
- You can this time select the component from the pull-down menu.

Selecting a component from the pull-down menu



Configuring the Pipeline: trimming

After clicking the left mouse button on the canvas, you get the configuration window:



Edit the trimming component

Mandatory fields

PHASE:
trimming

Optional fields

minimum_quality:
10

quality_threshold:
80

invalid_letters:
XxNn

window_size:
30

verification_window_quality:
20

verification_window_size:
15

Configuring the Pipeline: trimming

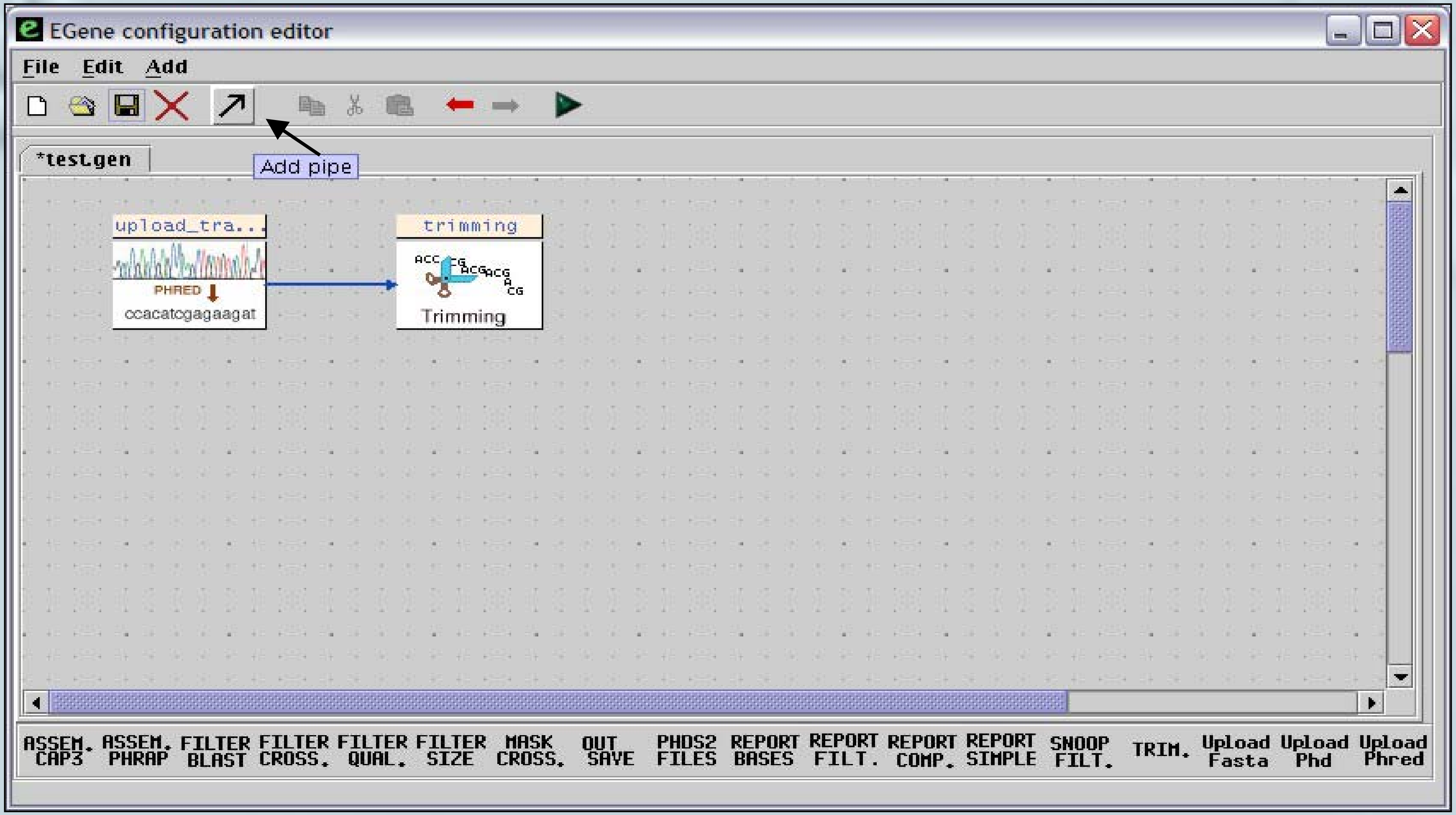
- Only one argument is mandatory: phase name
- There are many optional arguments, all with default values. We will use the default values. For an explanation about the arguments, please see the tutorial on building pipes and the component documentation.

Configuring the Pipeline: trimming

- After the configuration we now have two components on the canvas. We can use the “add pipe” ↗ icon to connect them.
- Click on the arrow (add pipe) icon. Now click first on the component that should process information before (upload), and then on the next one (trimming).

Configuring the Pipeline: the first two components

Now we have a pipeline with two components:



Configuring the Pipeline: filtering against ribosomal sequences

- The next step is to filter out ribosomal sequences.
- This will be performed using BLAST.
- We will use the `filter-blast.pl` component.

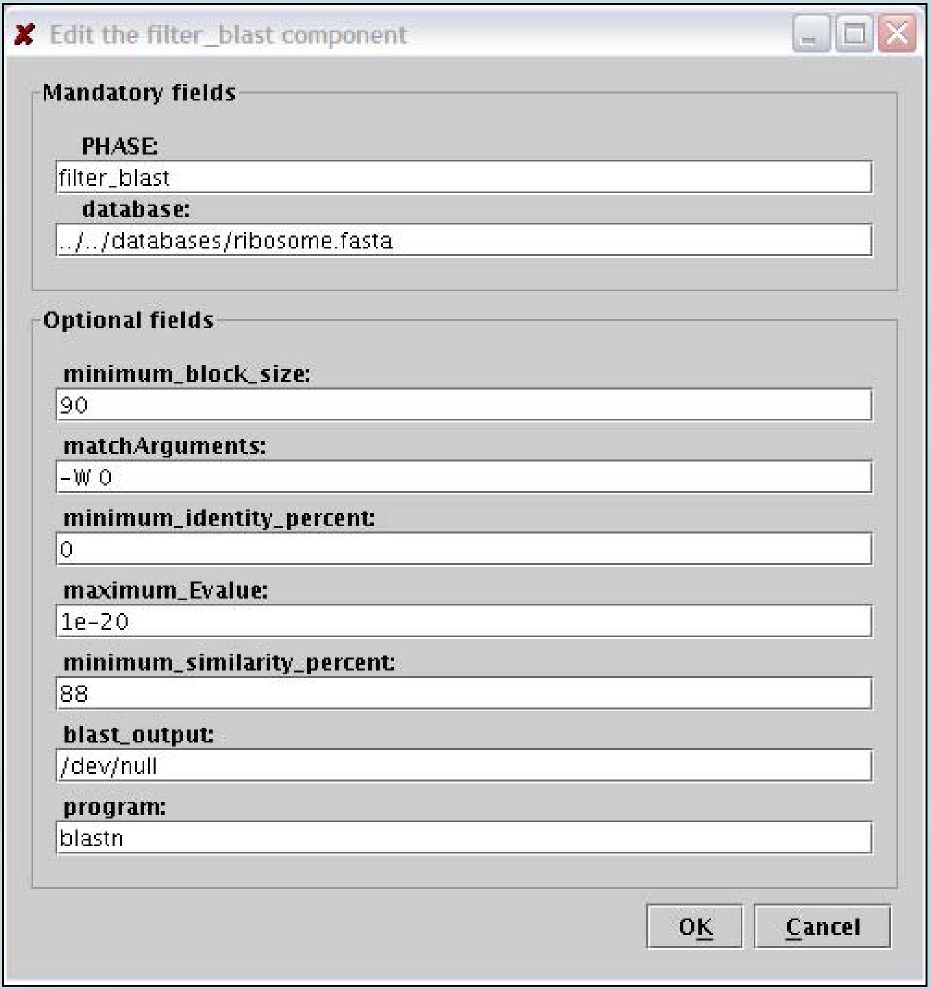
Configuring the Pipeline: filtering against ribosomal sequences

- `filter-blast.pl` has two mandatory arguments: phase name and database to be searched.
- There are also seven optional arguments, all with default values.
- We will accept the default values and assume that the database is located at:

  `../../databases/ribosome.fasta`

- For more details, check the other tutorials or the component's documentation.

Configuring the Pipeline: filtering against ribosomal sequences



Mandatory fields

PHASE:
filter_blast

database:
../../databases/ribosome.fasta

Optional fields

minimum_block_size:
90

matchArguments:
-W 0

minimum_identity_percent:
0

maximum_Evalue:
1e-20

minimum_similarity_percent:
88

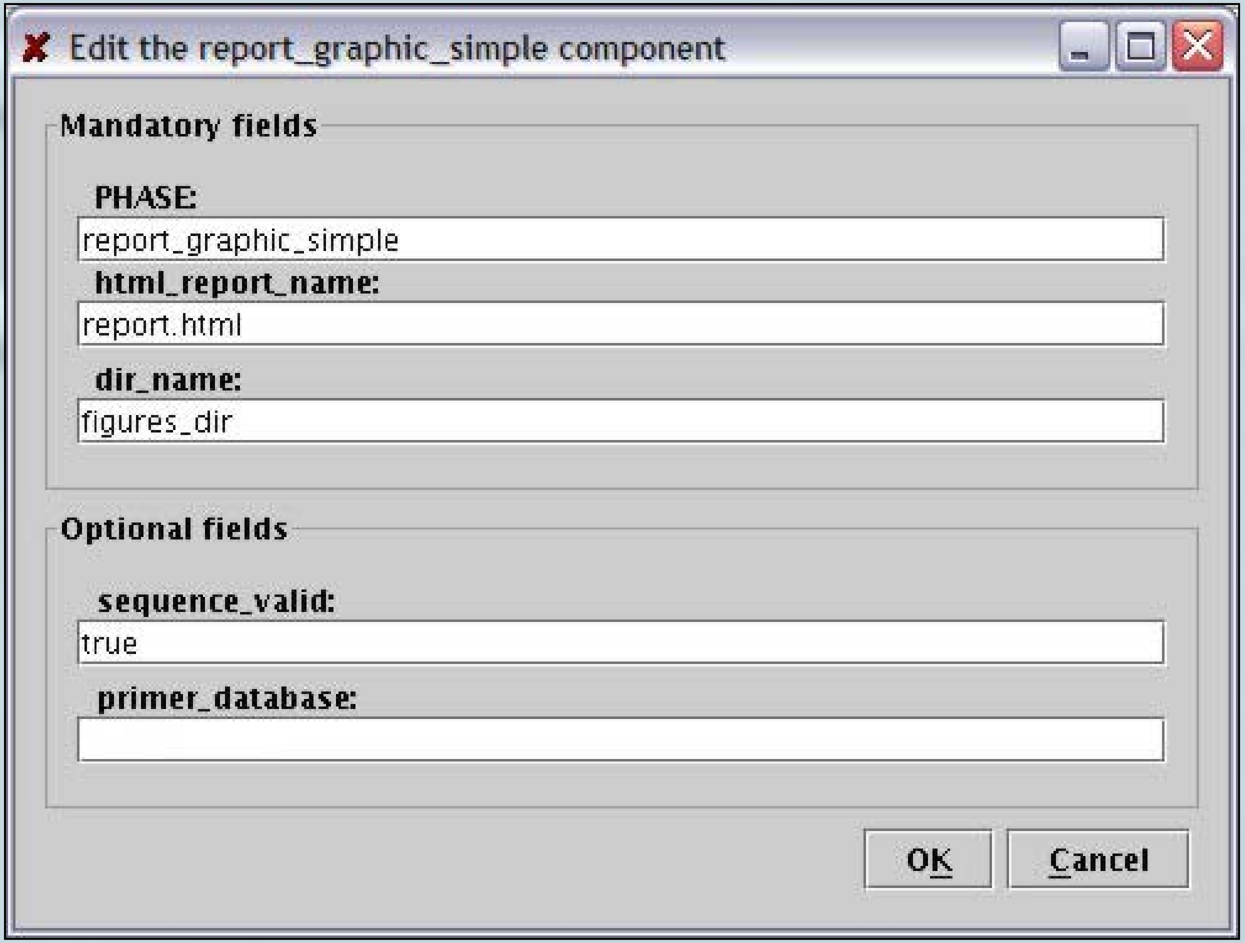
blast_output:
/dev/null

program:
blastn

Configuring the Pipeline: generating a graphical report

- We now will configure the last component: the simple graphical report
- We repeat the procedure of selecting the component from either the buttons or the pull down menu.
- The configuration window now shows five parameters
  - Three mandatory parameters: phase name, report file name, directory to place the graphics files (for more details check the other tutorials)
  - Two optional parameters: `sequence_valid` and `primer_database`

Configuring the Pipeline: generating a graphical report



Mandatory fields

PHASE: report_graphic_simple

html_report_name: report.html

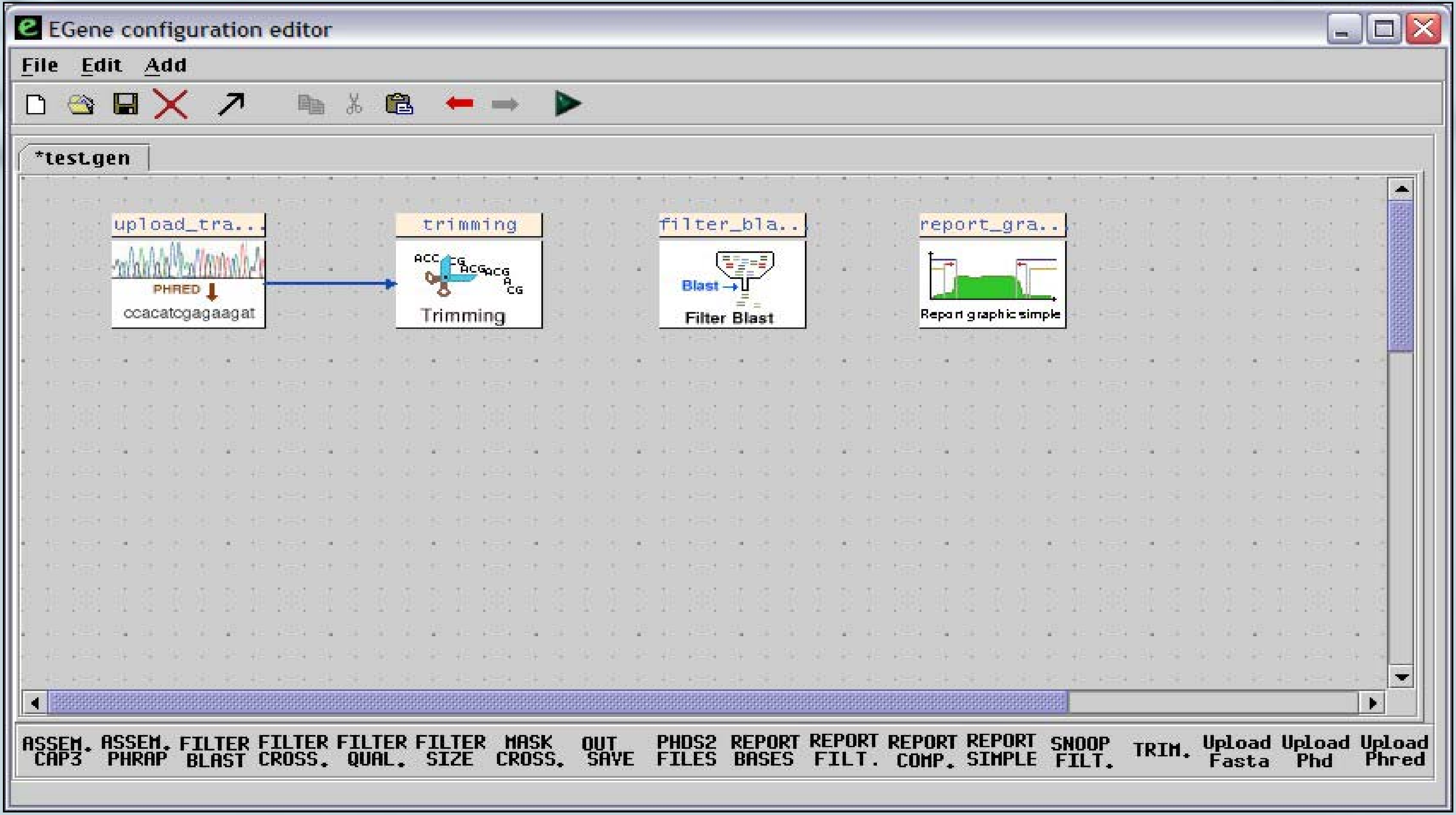
dir_name: figures_dir

Optional fields

sequence_valid: true

primer_database:

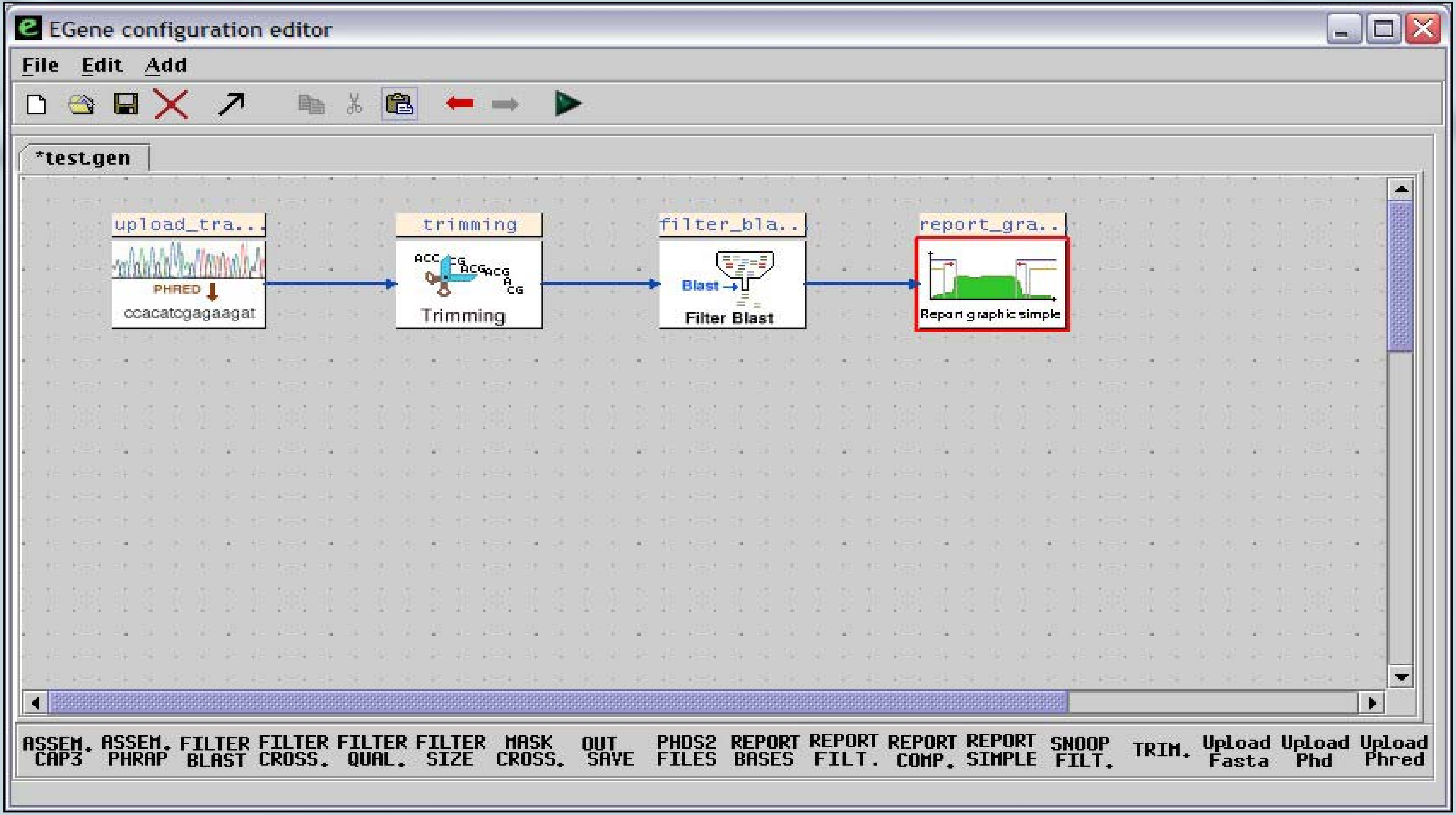
Configuring the Pipeline: generating a graphical report



Configuring the Pipeline: connecting the two last components

- We now use the arrow (add pipe) icon to connect the last two components to the pipeline.
- Remember:
  - click the add pipe icon: ↗
  - connect two components, clicking each one with the right button, in the order in which they should be placed in the pipe
- We now have the complete pipe.

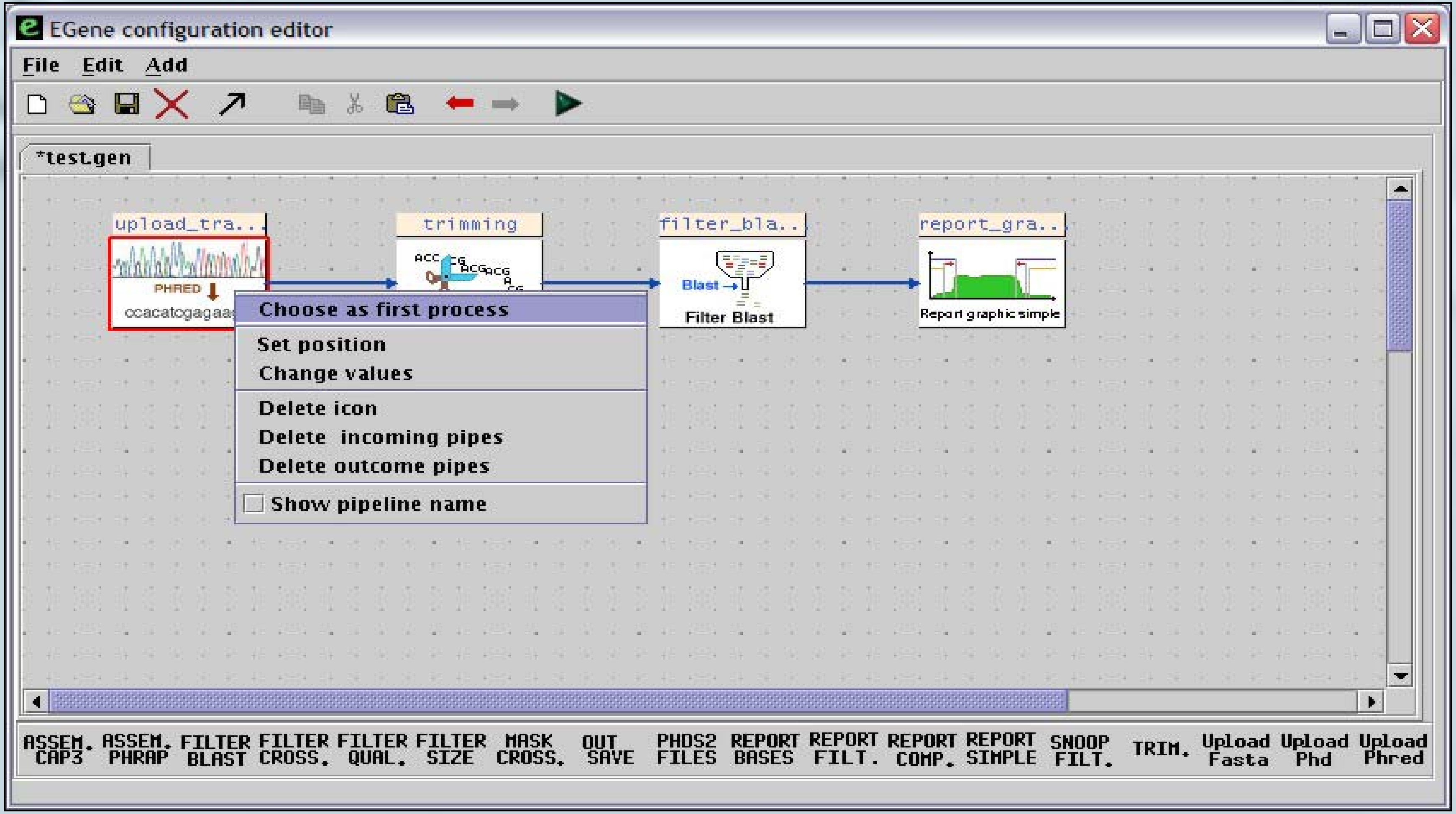
Configuring the Pipeline: connecting the two last components



Configuring the Pipeline: setting the first component

- The next step is to define what is the first component to be executed in the pipeline.
- To set a component as the first one, place the mouse over it and use the right button menu, selecting the “first process values” option.
- We can now run the pipe.

Configuring the Pipeline: setting the first component



Running the pipeline: the "run pipeline" icon

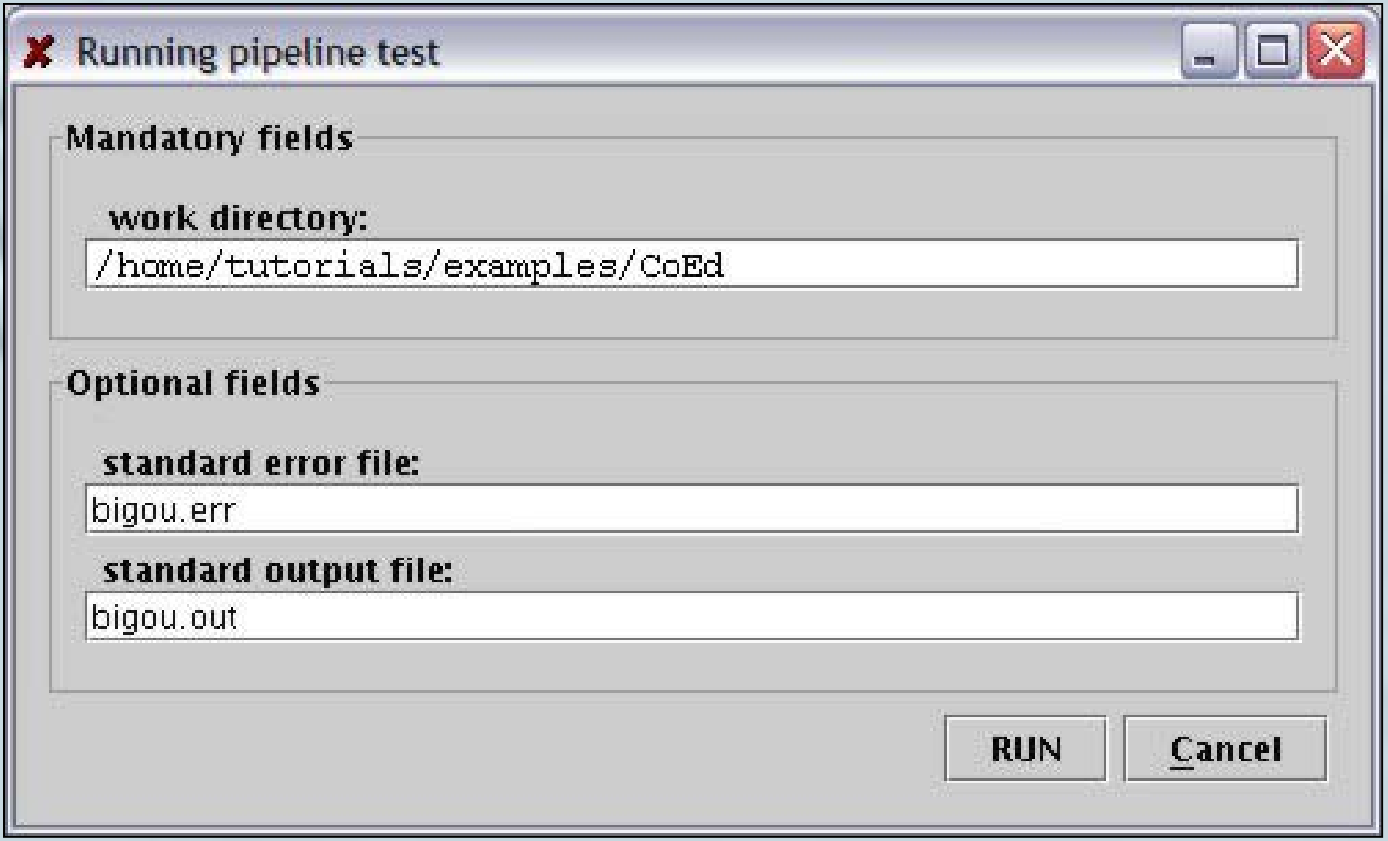
- To run the pipeline, click on the "run pipeline" icon:

▶

- CoEd will ask you were should the pipeline be executed (remember that file names can be relative).
- You can click on the mouse right button and navigate to find the directory of the CoEd tutorial.
- Let's assume we are at

`/home/tutorials/examples/CoEd/`

Running the pipeline: setting the work directory



Mandatory fields

work directory:

/home/tutorials/examples/CoEd

Optional fields

standard error file:

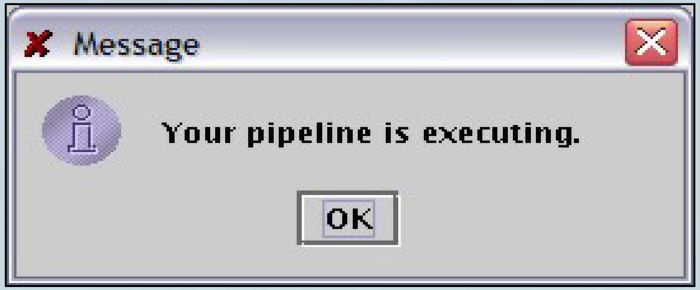
bigou.err

standard output file:

bigou.out

Running the pipeline: results

- CoEd notifies you that the pipe is running:



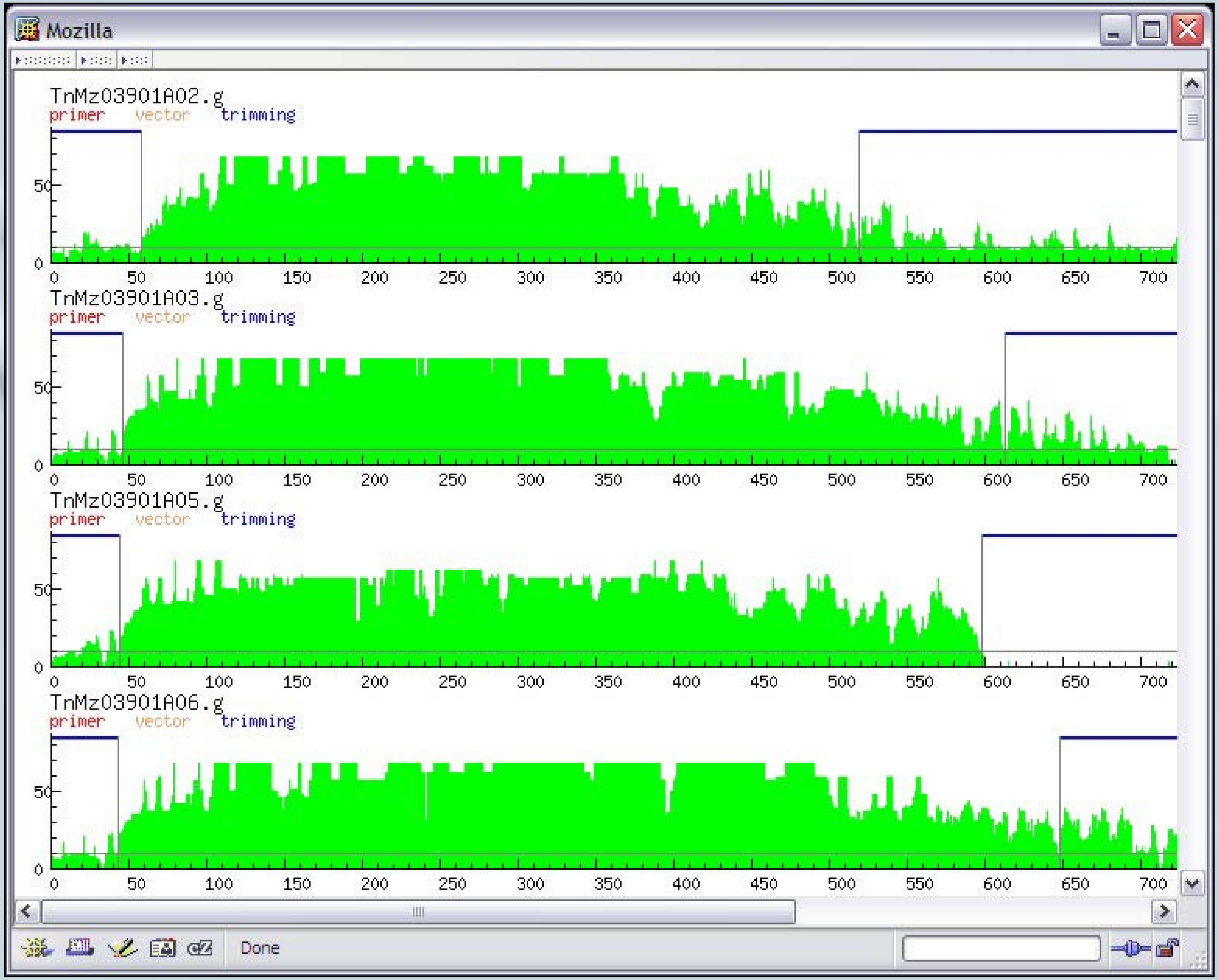
Your pipeline is executing.

- Pipelines run in the background, so you should check your directories to see when the pipe has finished.

Running the pipe: results

- After completing the pipeline, your directory should contain the file `report.html` and the directory `figures_dir`, which contains the graphical reports. Use a browser to look at the `report.html` file.

Pipe results: the report file



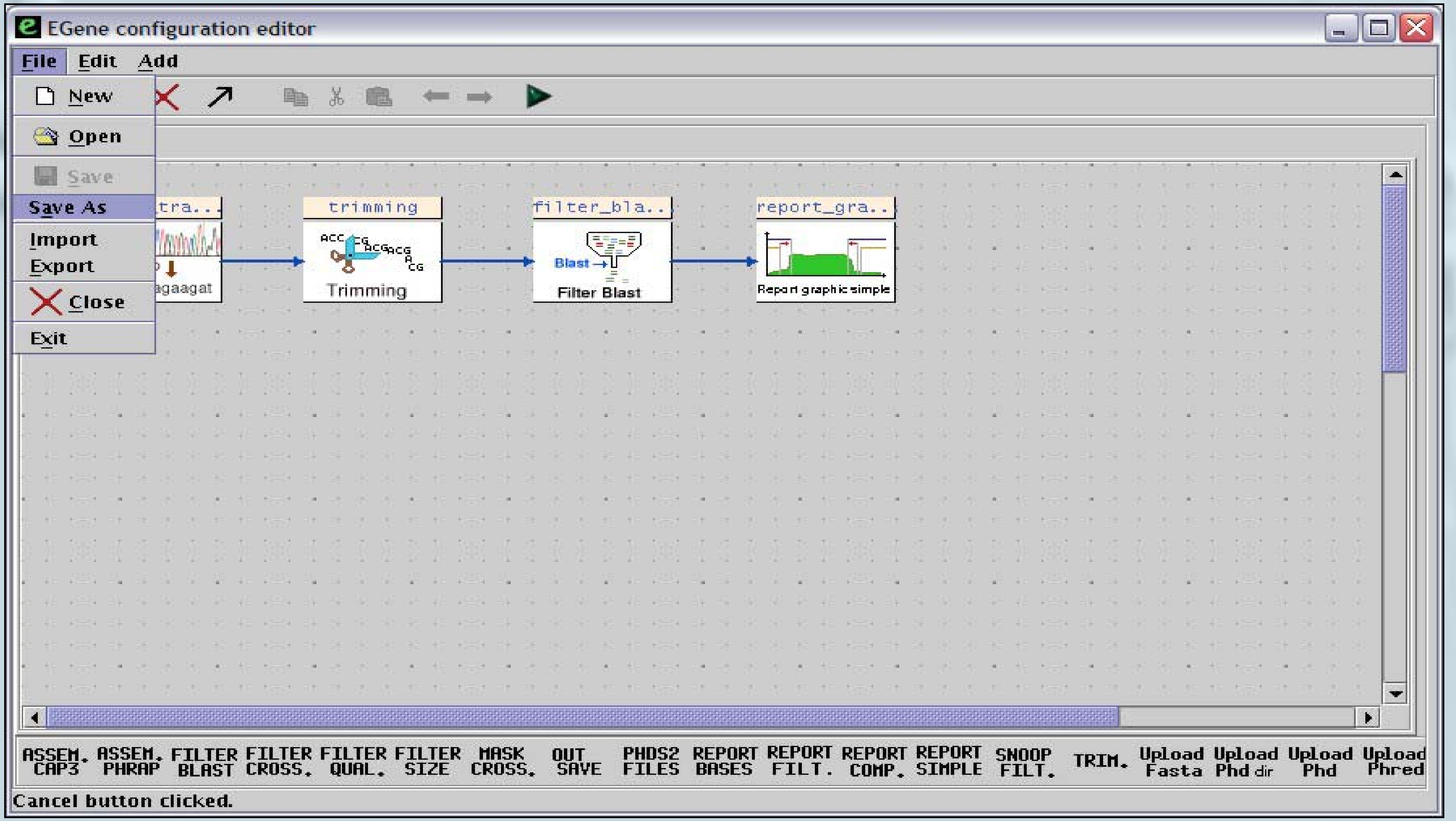
Running the pipe: final comments

- **Important note:** CoEd is a handy tool for configuring the pipe and testing the appropriate parameters.
- However, for massive processing it is best to use `bigou.pl` once the pipe is configured. That way, pipeline processing can be included into a Unix script.

Saving your work

- Now use the save button to save your work: 💾
- CoEd’s standard file extension is `*.gen`.
- You can also generate an EGene file (`*.cnf`) using “save as” from the pull-down menu. EGene files can be used by `bigou.pl`, but they do not keep information about the position of the icons.
- Save your pipeline in EGene format as `test.cnf`

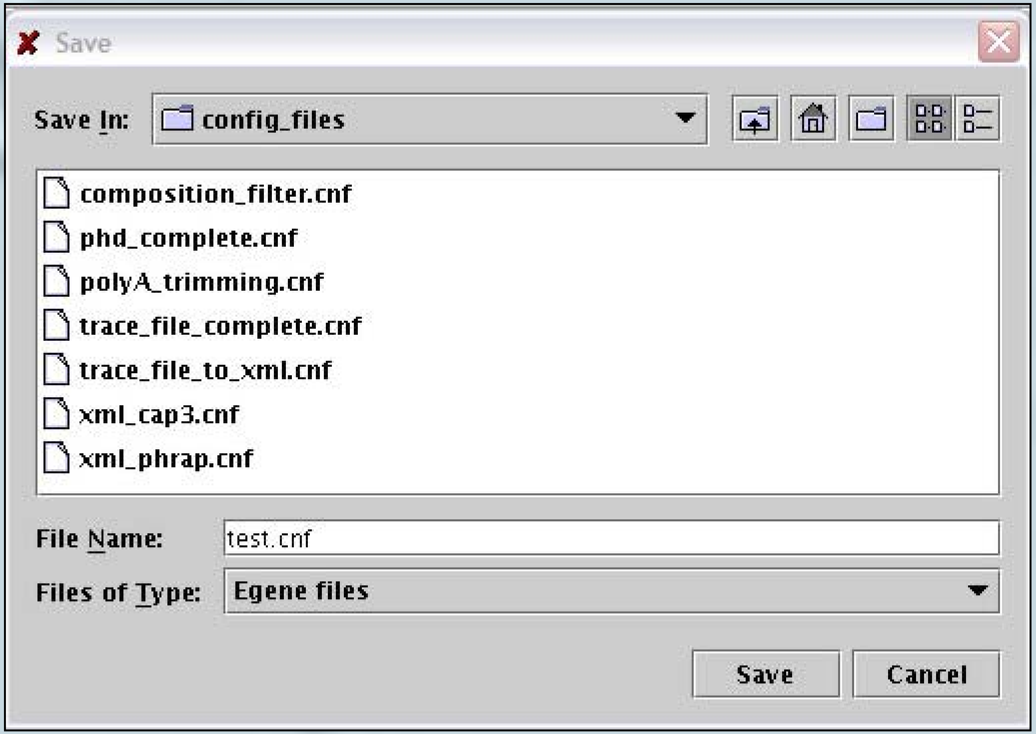
Saving your work in EGene format



Cancel button clicked.

Saving your work in EGene format

- Save your work.



Save In: config_files

composition_filter.cnf
phd_complete.cnf
polyA_trimming.cnf
trace_file_complete.cnf
trace_file_to_xml.cnf
xml_cap3.cnf
xml_phrap.cnf

File Name: test.cnf
Files of Type: Egene files

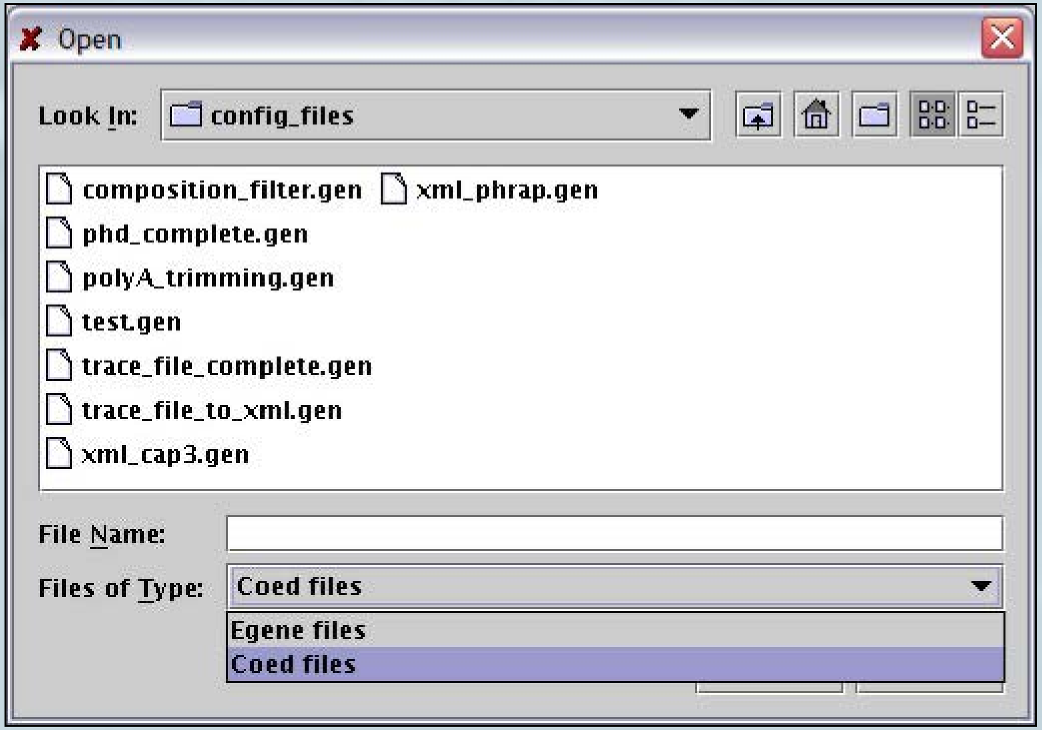
- Now use the close button. ✕

Loading an existing file into EGene

- To edit an existing pipeline, open its EGene or CoEd description.
- Use the open button: 📂

Loading an existing file into EGene

Select the format your pipeline file (EGene or CoEd):



Look In: config_files

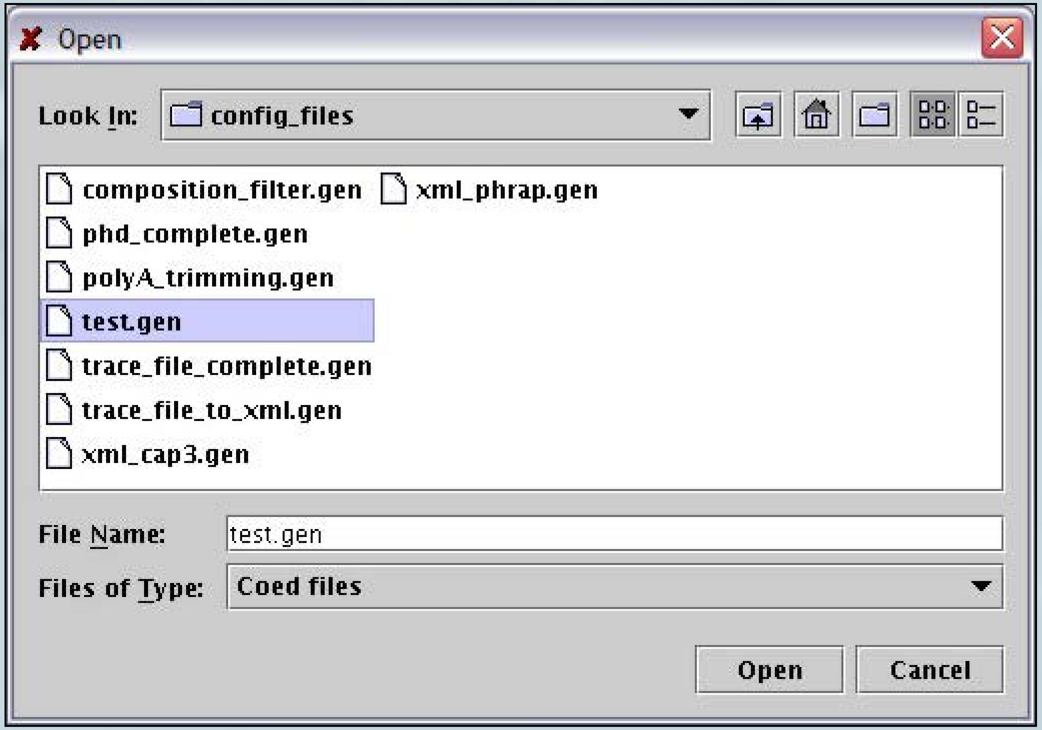
composition_filter.gen
xml_phrap.gen
phd_complete.gen
polyA_trimming.gen
test.gen
trace_file_complete.gen
trace_file_to_xml.gen
xml_cap3.gen

File Name:

Files of Type: Coed files

Loading an existing file into EGene

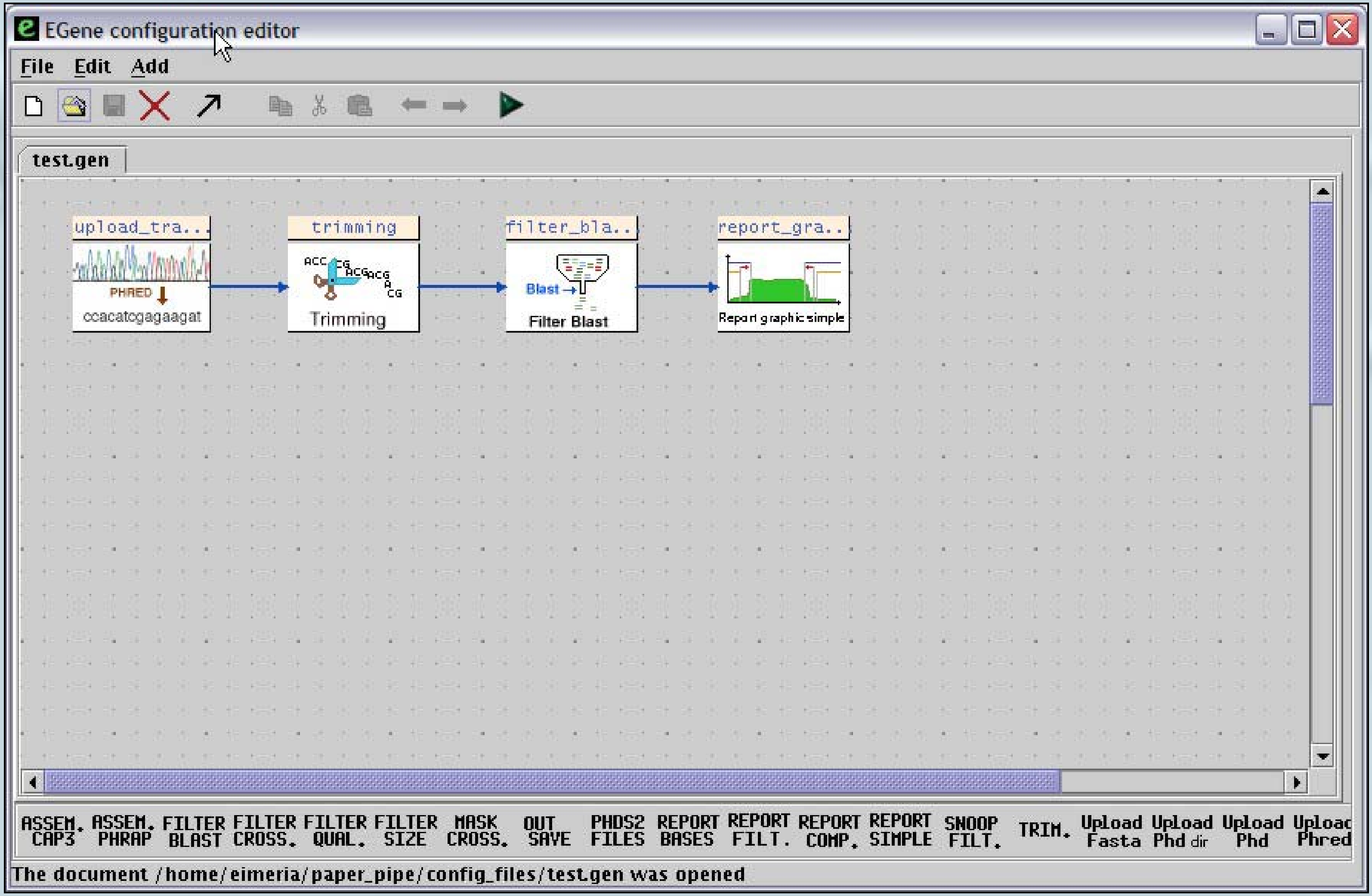
Select the file to upload:



File Name: test.gen

Files of Type: Coed files

`Test.gen` is now back to CoEd



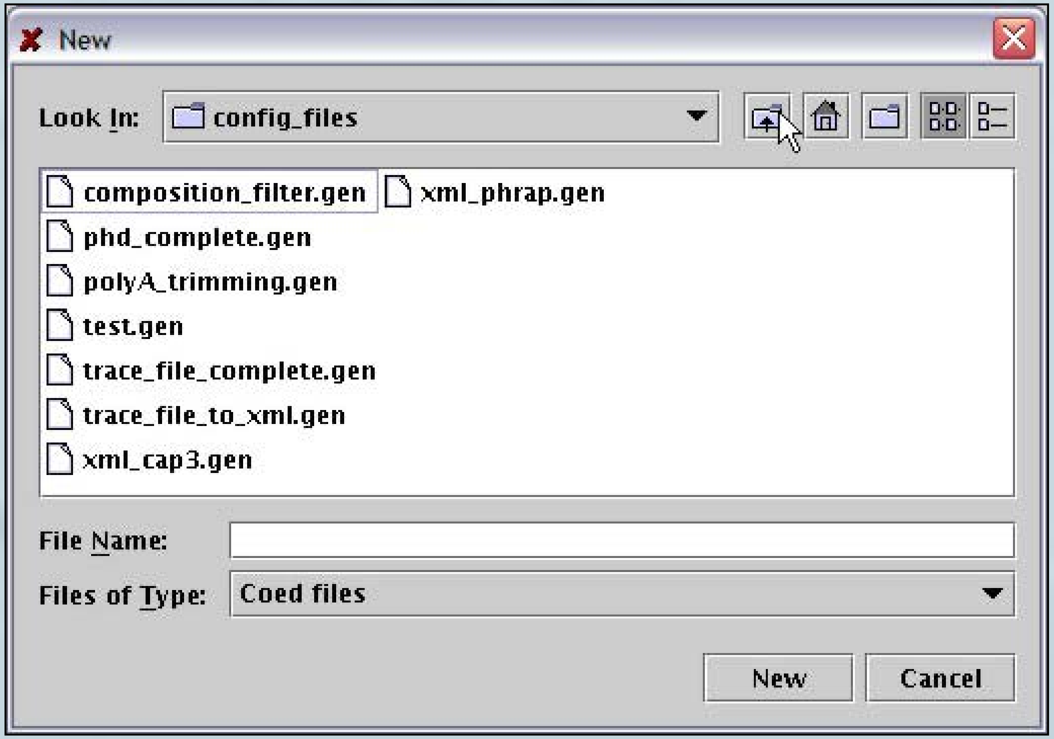
The document /home/eimeria/paper_pipe/config_files/test.gen was opened

Editing pipelines: using the editing operations

- You can use the editing buttons to edit your pipeline.
- You can copy ⎘ , cut ✂ , and paste 📋 all or some of the components of your pipeline.
- Editing can be undone or redone using…
- …the undo ← and redo → buttons.
- Try them!

Creating a new pipe

- First, click on the "new" 🗋 icon in the tool bar.
- You will be asked for the name of the new pipeline.
- Fill the File Name form with `new.gen`.

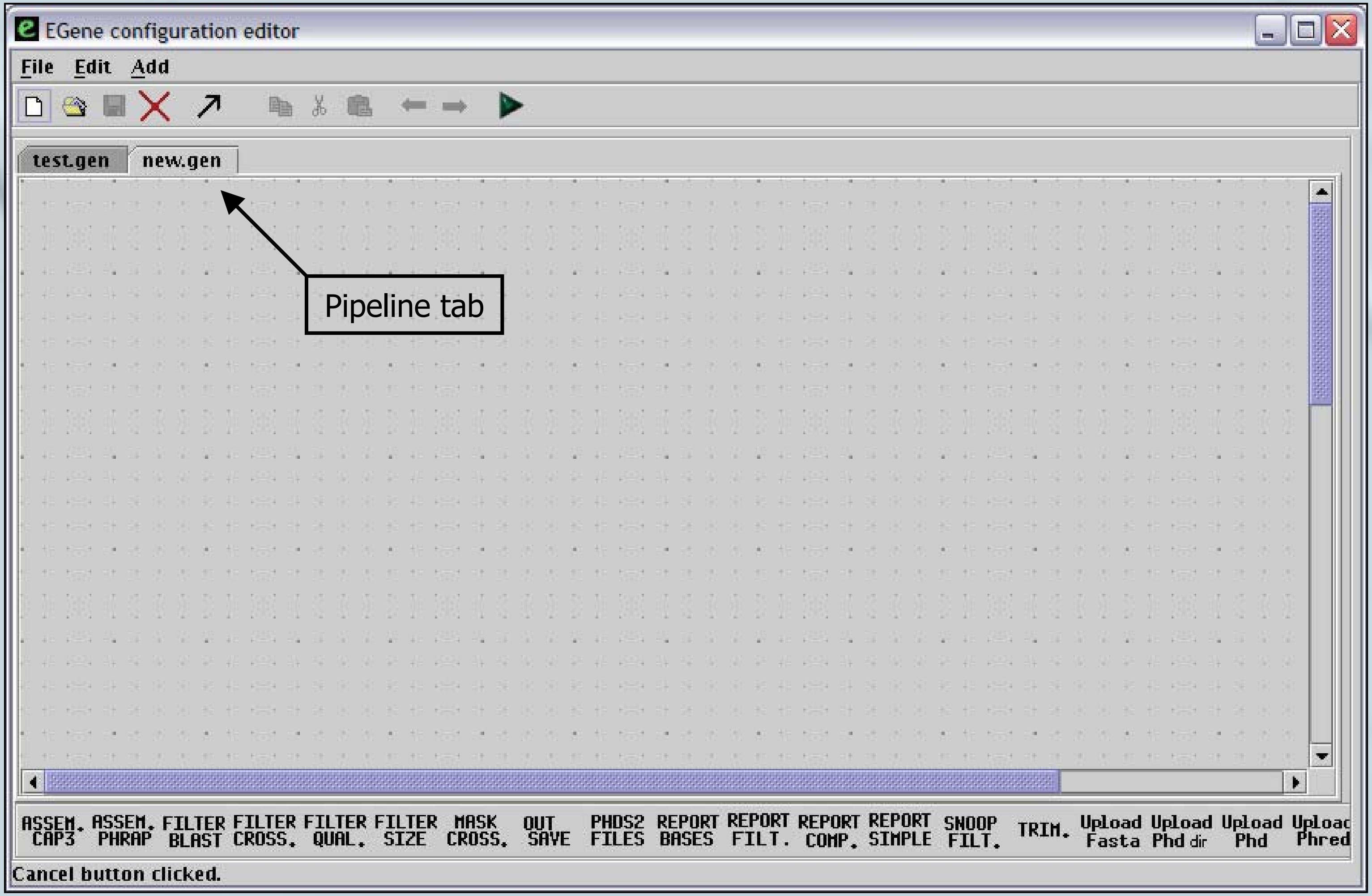


composition_filter.gen
xml_phrap.gen
phd_complete.gen
polyA_trimming.gen
test.gen
trace_file_complete.gen
trace_file_to_xml.gen
xml_cap3.gen

File Name:
Files of Type: Coed files

Creating a new pipe

- A tab `new.gen` will be created on the canvas.

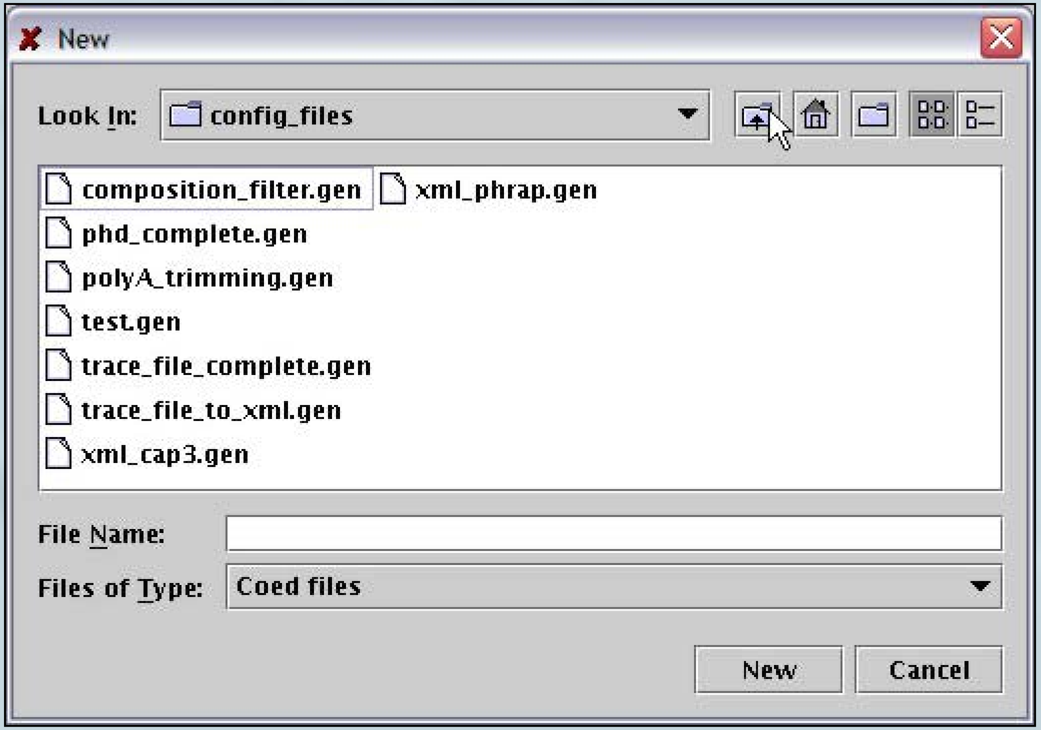


Pipeline tab

Cancel button clicked.

Creating a new pipe

- You will be asked for the name of the new pipeline.
- Fill the File Name form with `new.gen`.



Look In: config_files

composition_filter.gen
phd_complete.gen
polyA_trimming.gen
test.gen
trace_file_complete.gen
trace_file_to_xml.gen
xml_cap3.gen
xml_phrap.gen

File Name:

Files of Type: Coed files

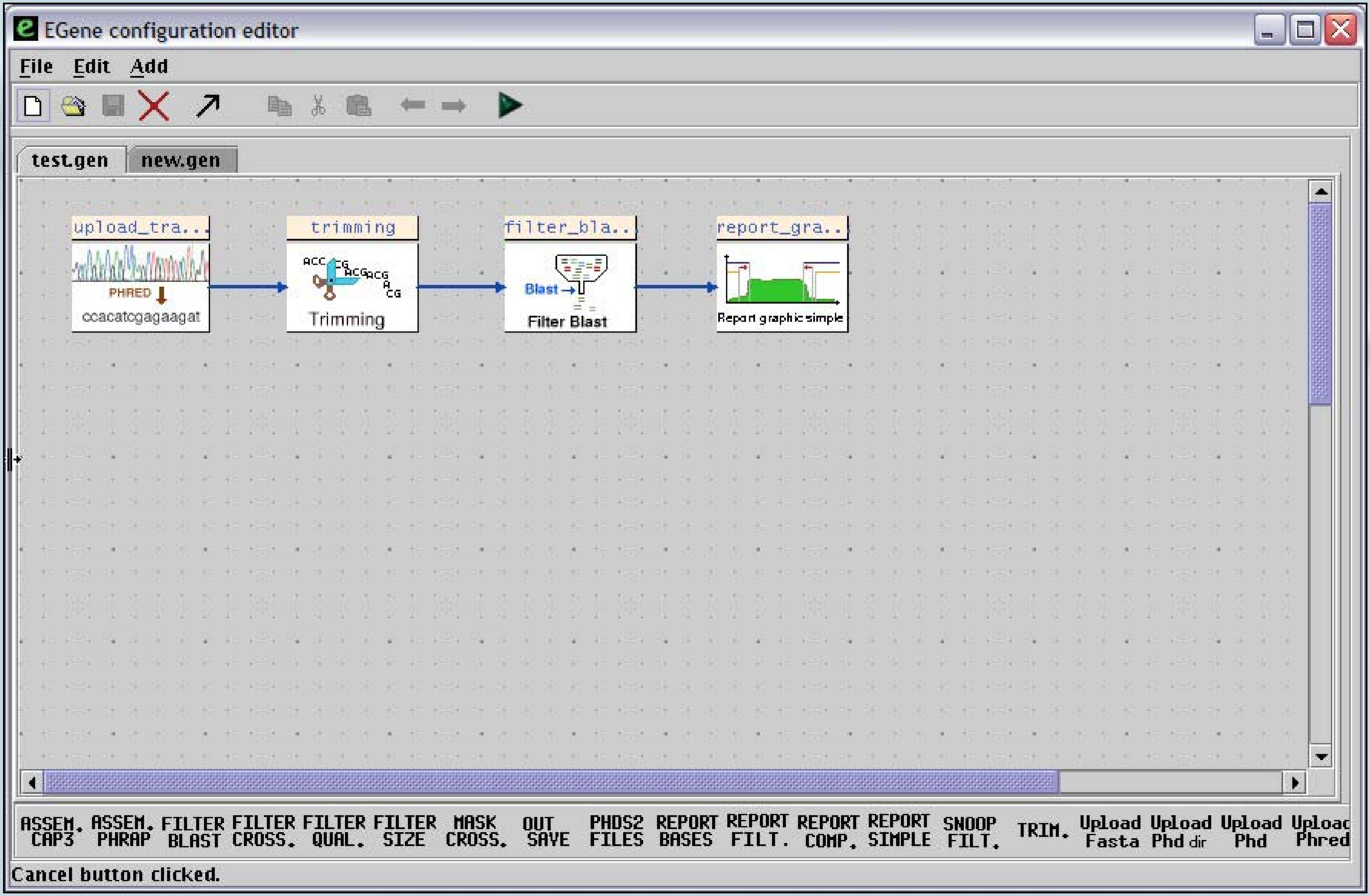
Creating a new pipe from an existing one

- You can create a new pipeline from an existing one by loading a pipe into CoEd, modifying it, and using the “File Save As” option in the pull-down menu.

Switching pipelines in CoEd

- You can switch among different pipeline by selecting the respective tab.
- Select the `test.gen` pipe by clicking on the respective tab.

Creating a new pipe from an existing one

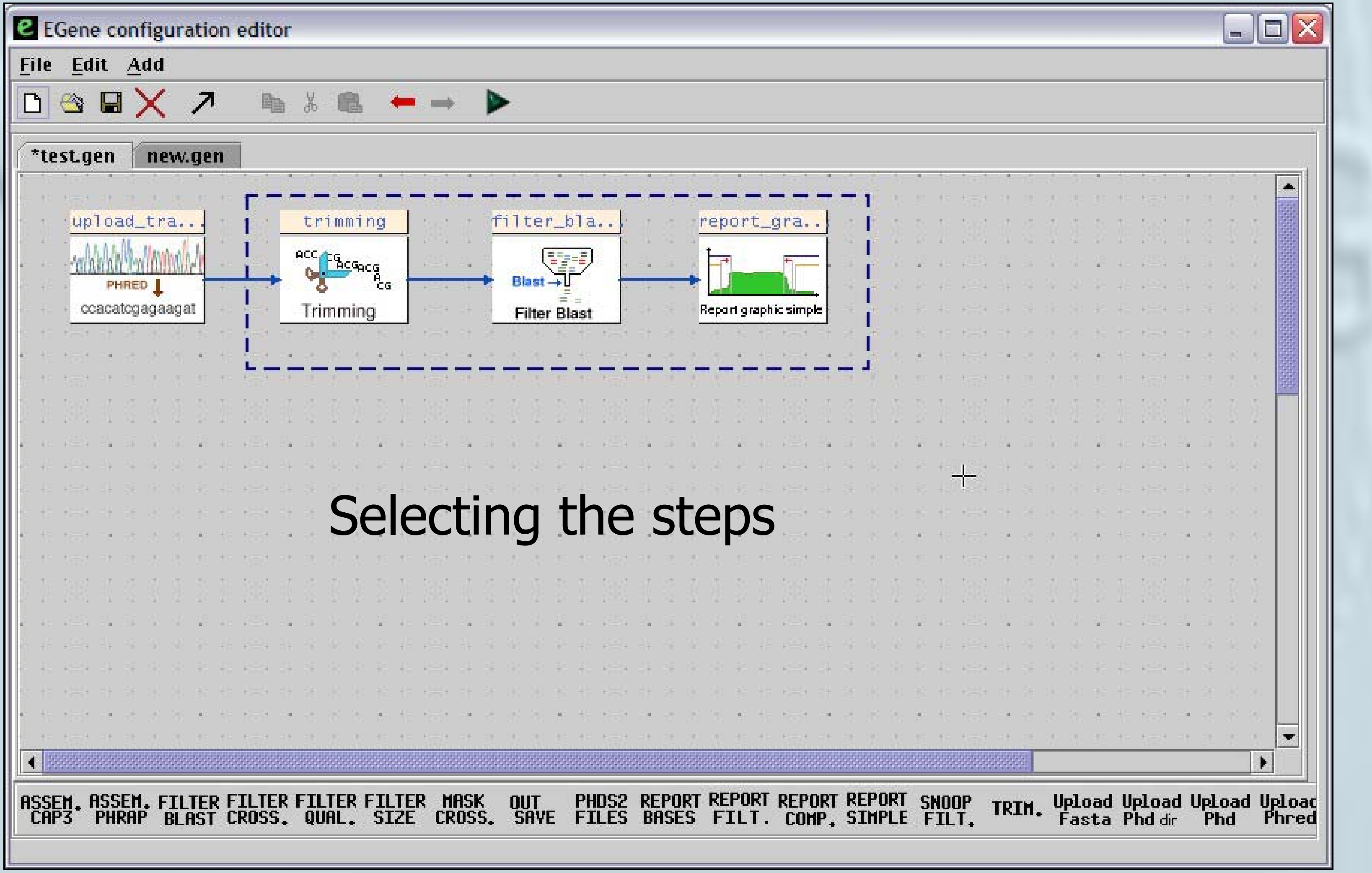


Cancel button clicked.

Copying part of a pipeline into another pipeline

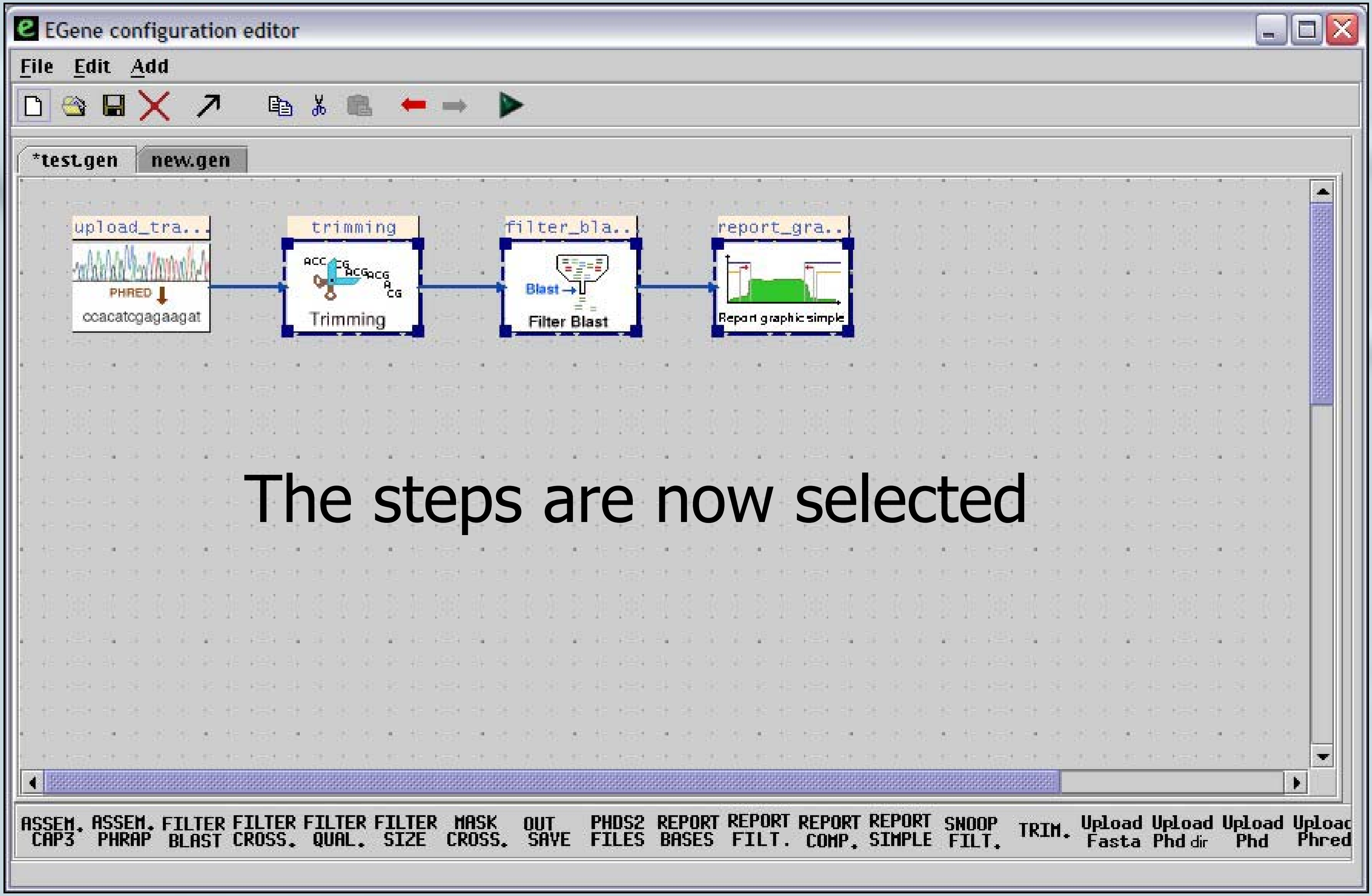
- You can copy parts of a pipeline into another pipeline.
- Select the part of the pipeline you want:
  - We will select all components but the first one.
  - To do this, click the left button of the mouse on the canvas and drag it to delimit a region including the three last components.
  - Alternatively, you can click on each component.
  - To deselect all components click on the canvas.

Copying part of a pipeline into another pipeline



Selecting the steps

Copying part of a pipeline into another pipeline

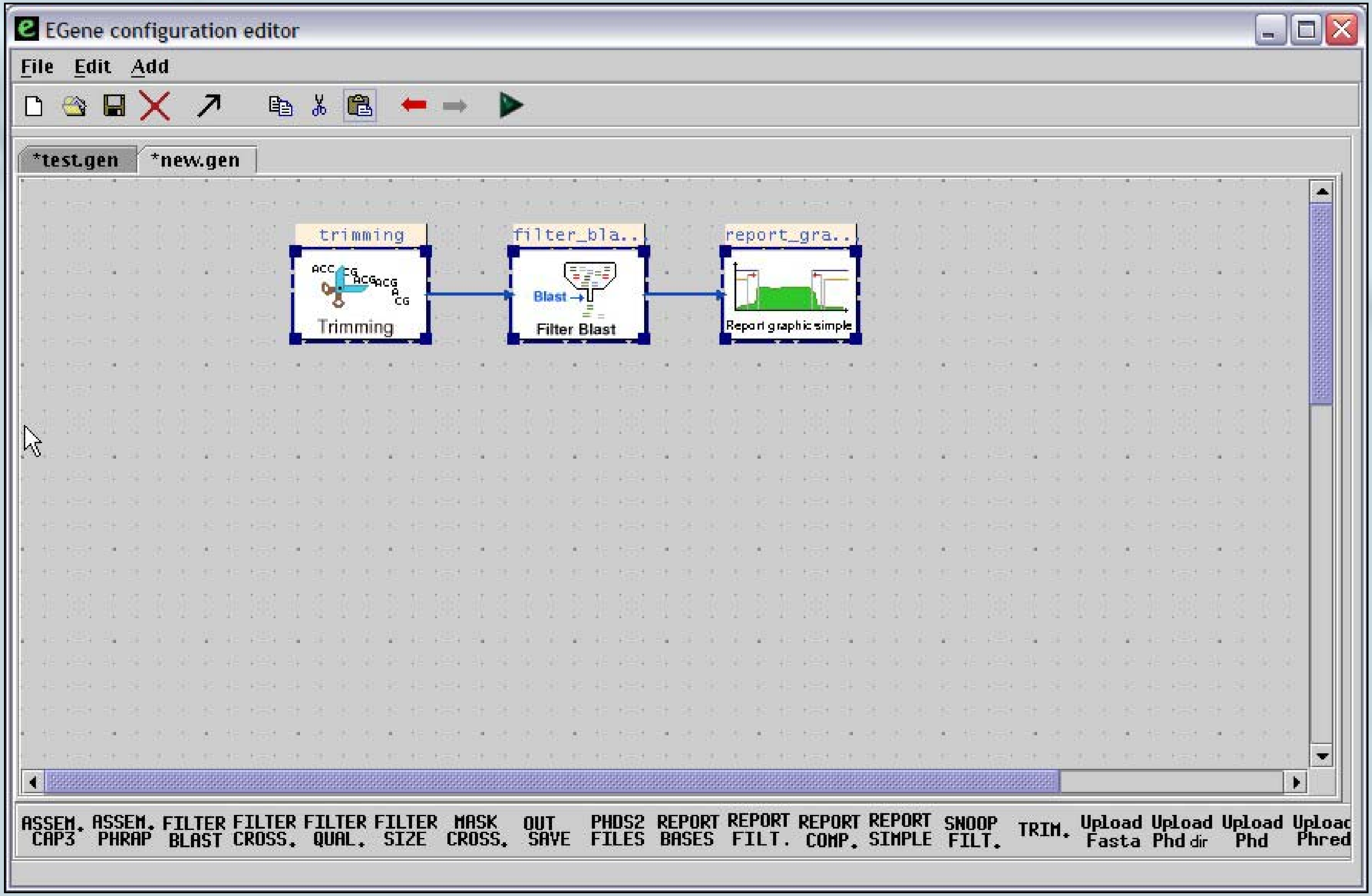


The steps are now selected

Copying and pasting the selected steps into the new pipeline

- Click on the "copy" icon ⎘ in the tool bar.
- Now click on the the `new.gen` tab.
- Use the "paste" icon 📋 to copy the previous components into the new pipeline.
- You should now have the following pipeline:

New pipeline added of pasted components



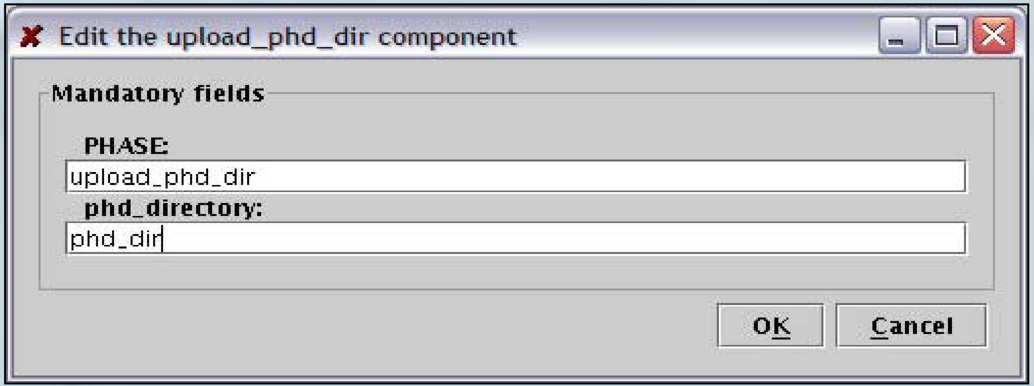
Finishing the new pipeline

- We will now add a new upload component, this time for uploading Phred files from a `phd_dir` directory
- And… we will generate a graphical report with a new name, changing the specification of the last component.

Adding a new upload component

- We should use the `upload_phd_dir` component.

- This component has only two parameters, both mandatory: `PHASE` name and `phd_directory`:

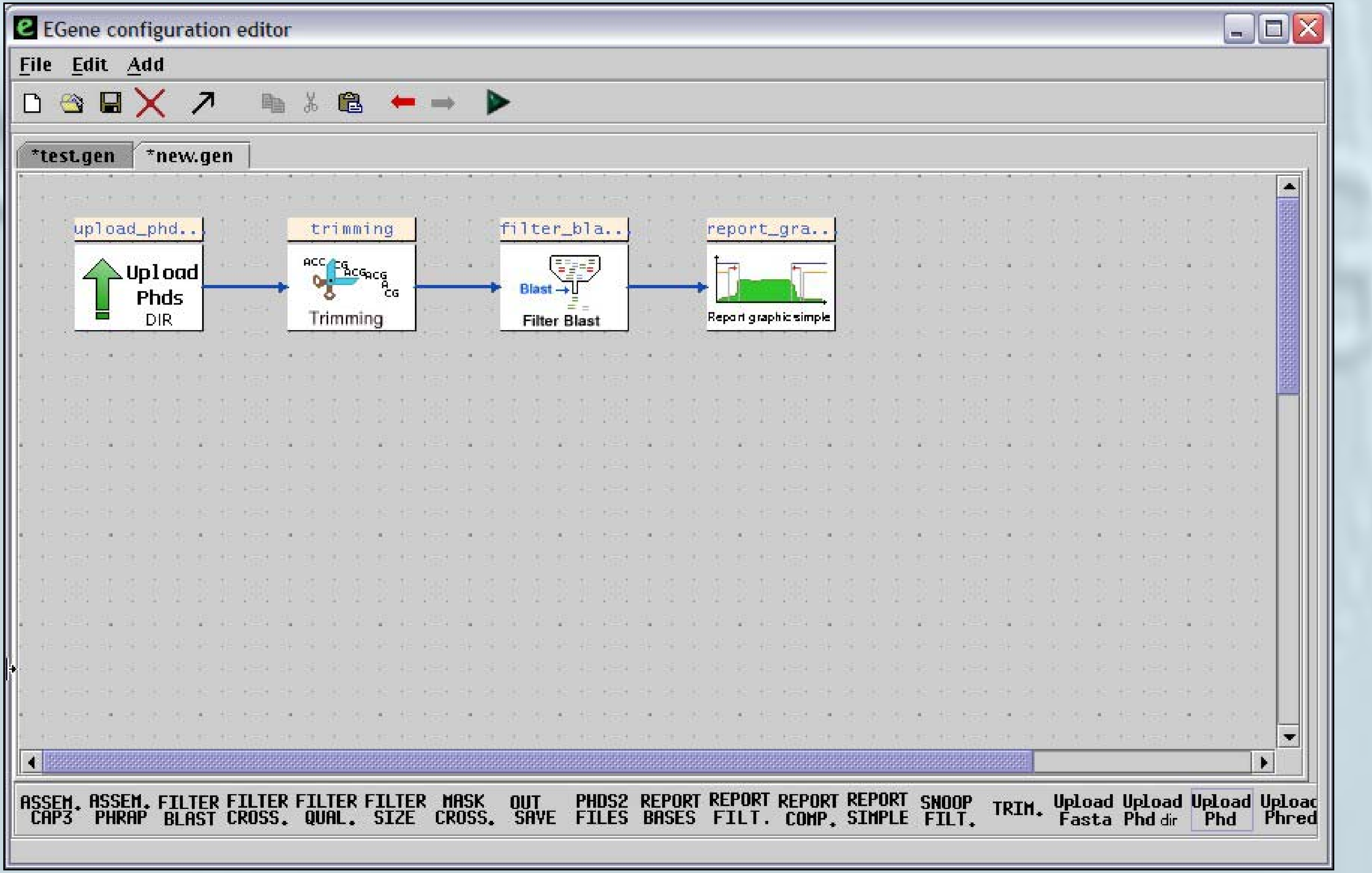


PHASE: upload_phd_dir

phd_directory: phd_dir

- After placing the component, use the arrow button to put it at the beginning of the pipeline.

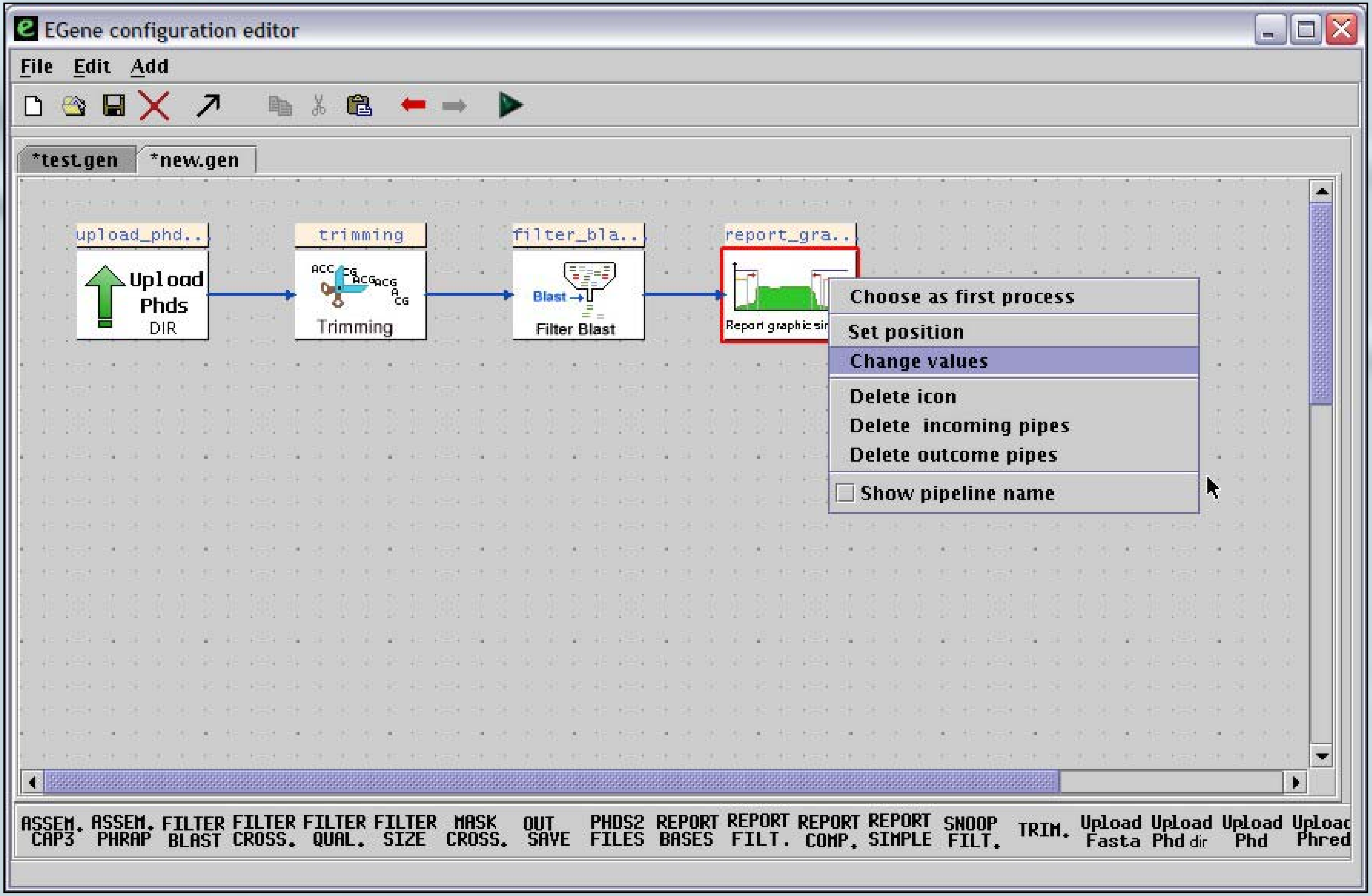
Pipeline with the added component



Editing the pipeline: changing the parameters of a component

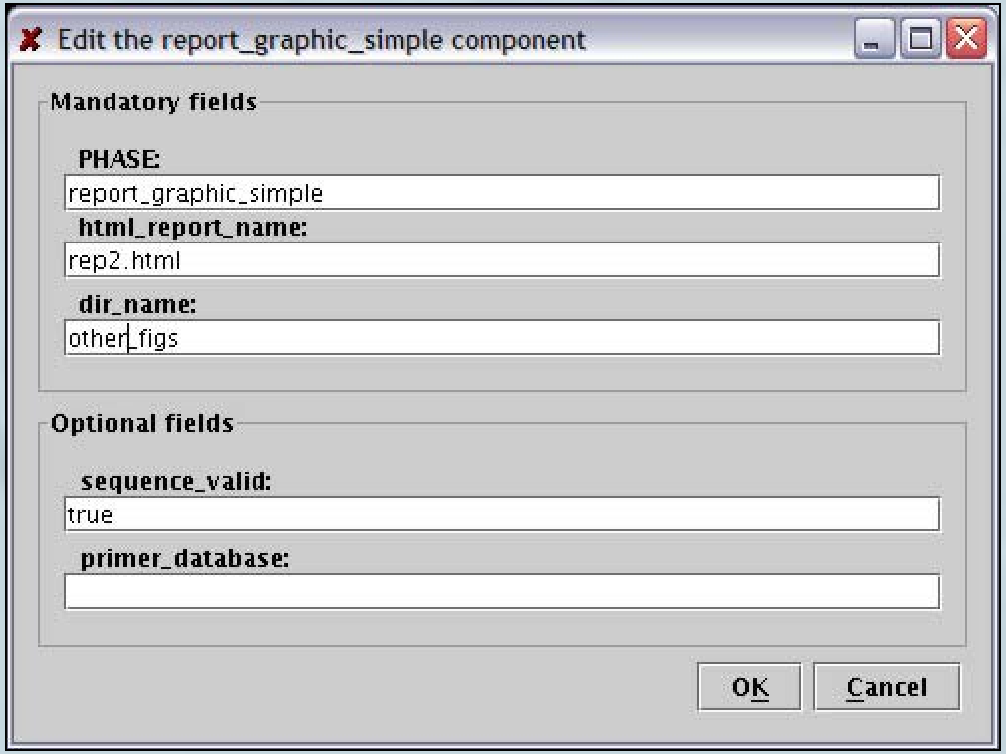
- To change the values of a component, place the mouse over it and use the right_button menu, selecting the “Change values” option.

Editing the pipeline: changing the parameters of a component



Editing the pipeline: changing the parameters of a component

- The configuration window will appear.
- Let's change the `html_report_name` to `rep2.html` and `dir_name` to `other_figs`.



Edit the report_graphic_simple component

Mandatory fields

PHASE: report_graphic_simple

html_report_name: rep2.html

dir_name: other_figs

Optional fields

sequence_valid: true

primer_database:

Running the new pipeline

- Choose the upload step as the first process.
- Now, use the “run pipeline” ▶ icon in the tool bar to run the new pipeline.
- Use the browser to compare the results with those of the previous files.
- You can also save the pipe in EGene format and use `bigou.pl` from the UNIX command line to run it again.

We support pipes! For peace.

THE END!