

Sequanix: A Dynamic Graphical Interface for Snakemake Workflows

Dimitri Desvillechabrol¹, Rachel Legendre^{2,1}, Melissa Cardon¹, Sean Kennedy¹, Thomas Cokelaer^{2,1}
1 : Institut Pasteur - Pôle Biomix, CITECH - Paris, France
2 : Institut Pasteur - Hub Bioinformatique et Biostatistique - C3BI, USR 3756 IP CNRS - Paris, France

Motivation

Within the biomix pole (Institut Pasteur, CITECH), large genomic data sets are produced covering several transcriptomics, genomics, metagenomics. Since analysis pipelines evolve quickly, we needed a flexible framework and decided to use the Snakemake framework. We incorporated several Snakemake pipelines within a common project called Sequana (<http://github/sequana/sequana>). The pipelines are flexible for developers, but in order to expose them to a wider audience, we designed a simple yet highly useful graphical interface -- Sequanix -- also aiming at democratizing the use of Snakemake pipelines. Although the primary goal of Sequanix was to facilitate the execution of NGS Snakemake pipelines available in the Sequana project (<http://sequana.readthedocs.io>), it can also load any Snakemake pipelines.

Building pipelines based on Snakemake

Rules

```
rule vcf_filter:
    input:
        vcf = __vcf_filter__input
    output:
        vcf = __vcf_filter__output
    params:
        filter_dict = config["vcf_filter"]
    run:
        from sequana import vcf_filter

        vcf_record = vcf_filter.VCF(input["vcf"])
        vcf_record.filter_vcf(params["filter_dict"], output["vcf"])
```

Snakefile

```
...
include: sm.modules["snpeff"]
__vcf_filter__input = __snpeff__output
__vcf_filter__output = manager.getname(
    "vcf_filter", ".filter.vcf")
include: sm.modules["vcf_filter"]
...
```

Shell

```
$ sequana --pipeline variant_calling \
--input-directory samples \
--output-directory analysis \
--reference sequence.fasta
$ cd analysis
$ snakemake -s variant_calling.rules
```

Köster, Johannes and Rahmann, Sven. "Snakemake - A scalable bioinformatics workflow engine". Bioinformatics 2012

YAML file and tooltips

YAML configuration file

```
#####
# BWA used to remove a contaminant
# :Parameters:
#
# - do: if unchecked this rule is ignored
# - reference_file: the name of the reference
#   file to be found in the analysis directory.
#   If set to phiX174, it is downloaded
#   automatically from Sequana, otherwise you
#   will need to copy it yourself in the
#   working directory.
# - index_algorithm: the BWA index algorithm
# - options: any options recognised by BWA tool
# - threads: number of threads to be used
bwa_mem_phix:
  do: yes
  reference_file: 'phiX174.fa'
  index_algorithm: 'is'
  options: '-T 30'
  threads: 4
```

Tooltip

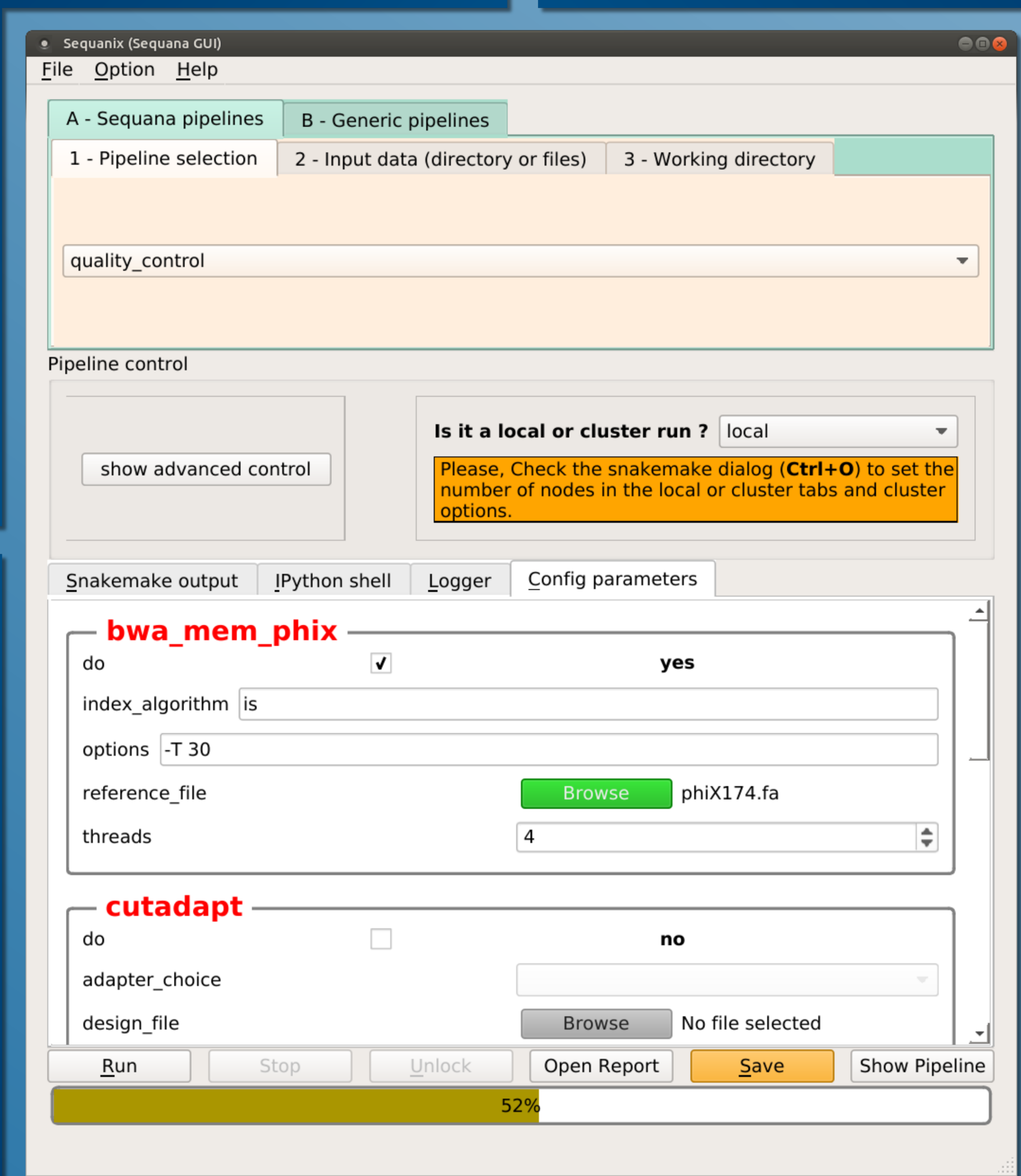
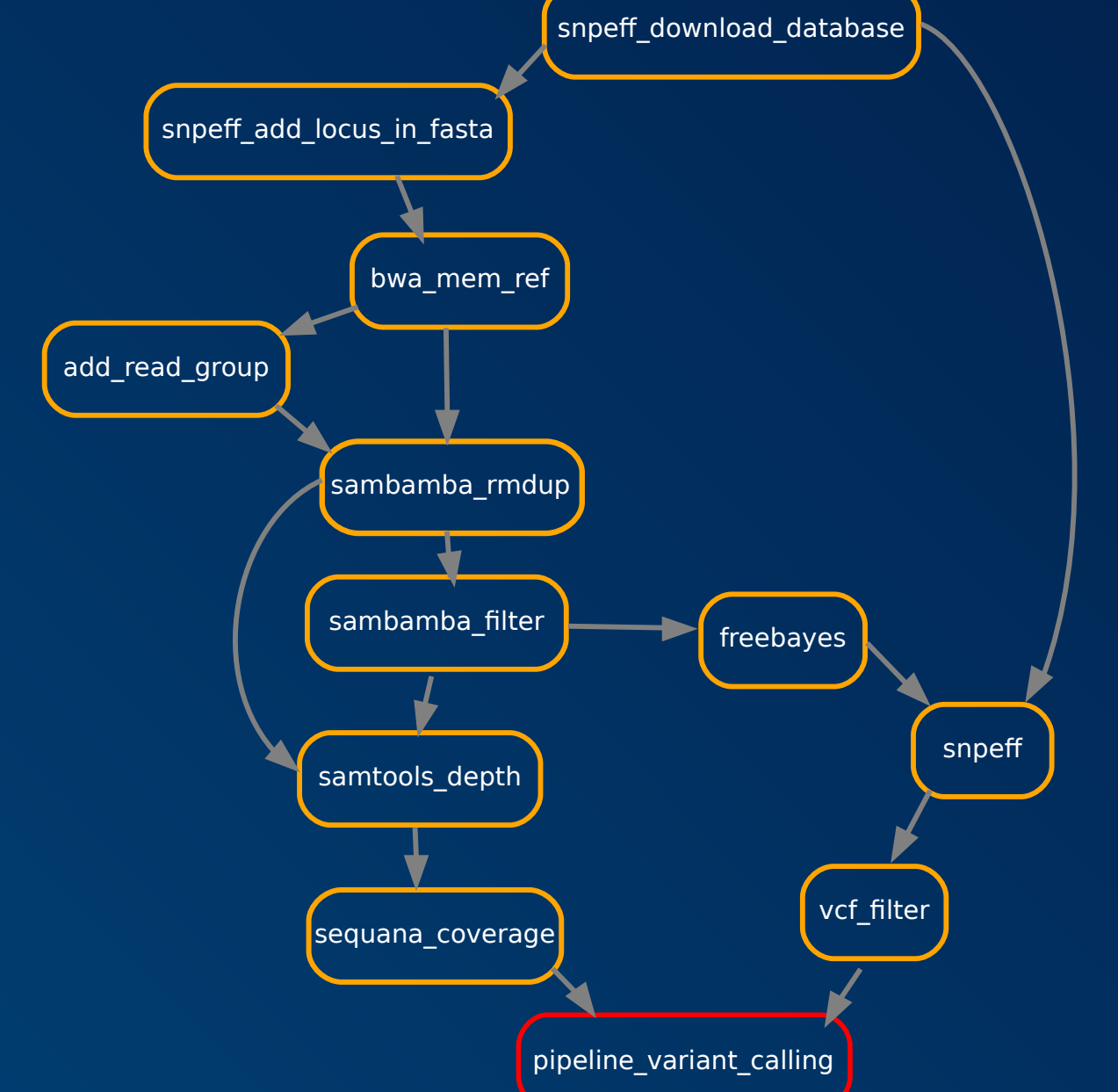
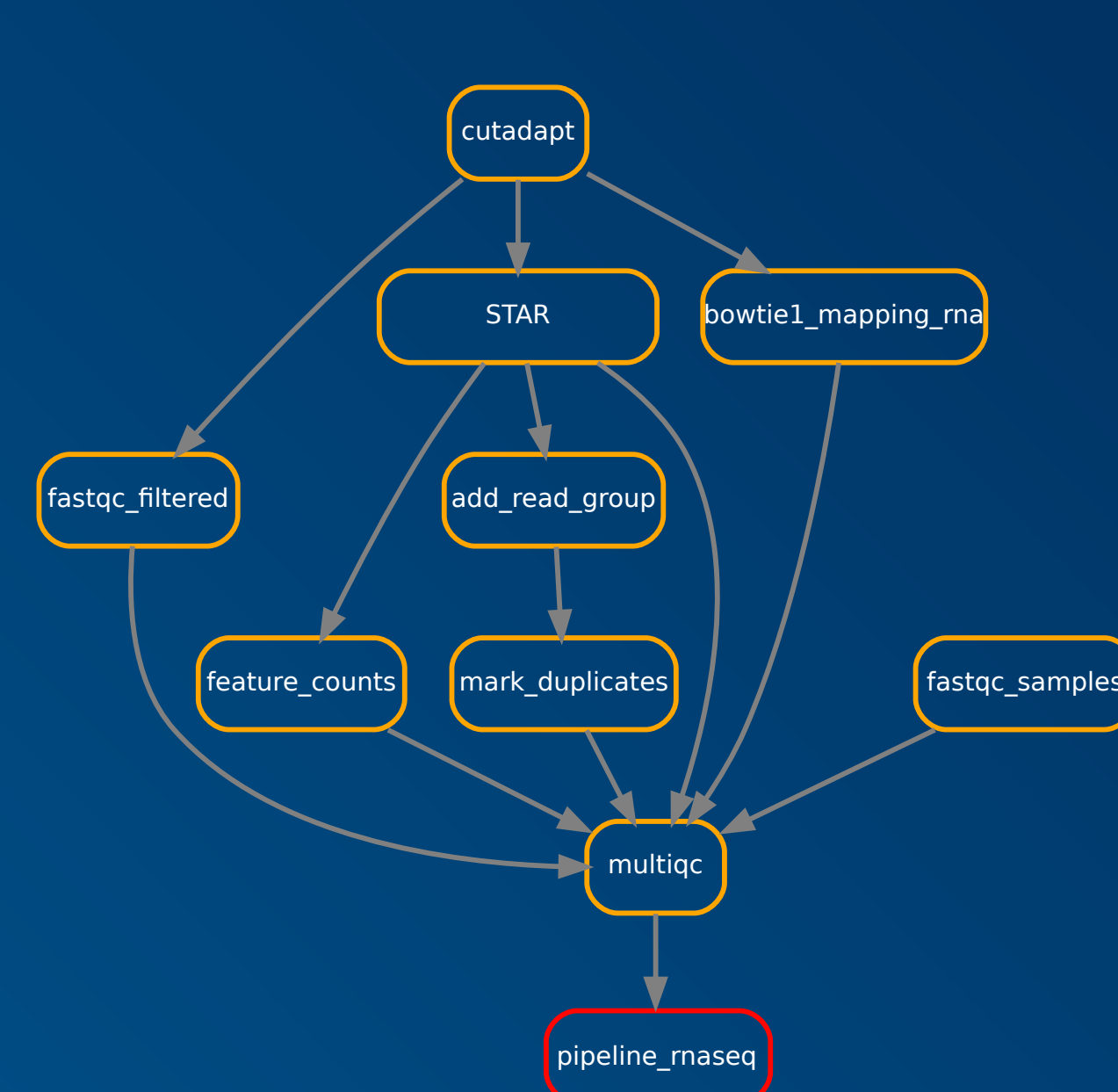
BWA used to remove a contaminant

Parameters:

- do: if unchecked, this rule is ignored
- reference_file: the name of the reference file to be found in the analysis directory. If set to phiX174.fa, it is downloaded automatically from Sequana, otherwise you will need to copy it yourself in the working directory.
- index_algorithm: the BWA index algorithm
- options: any options recognised by BWA tool
- threads: number of threads to be used

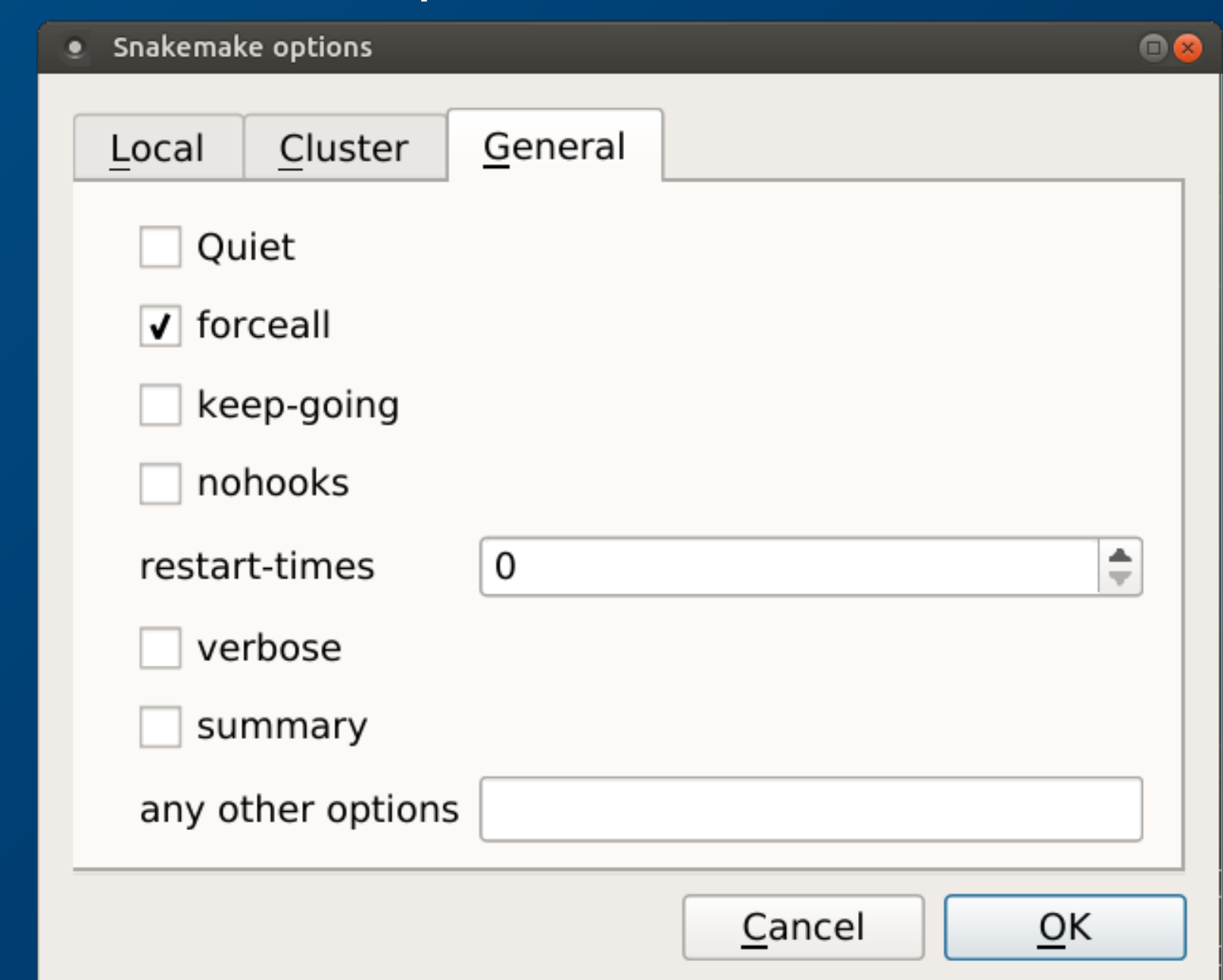
Within Sequanix, we designed a non-intrusive documentation framework for Snakemake configuration file (YAML file). Since YAML format allows comments, we propose to add a Python-like docstring before a section (top left figure) to describe the parameters of the section. This docstring is then interpreted and a tooltip is shown when users hover the pointer over a section.

Examples of Sequana pipelines

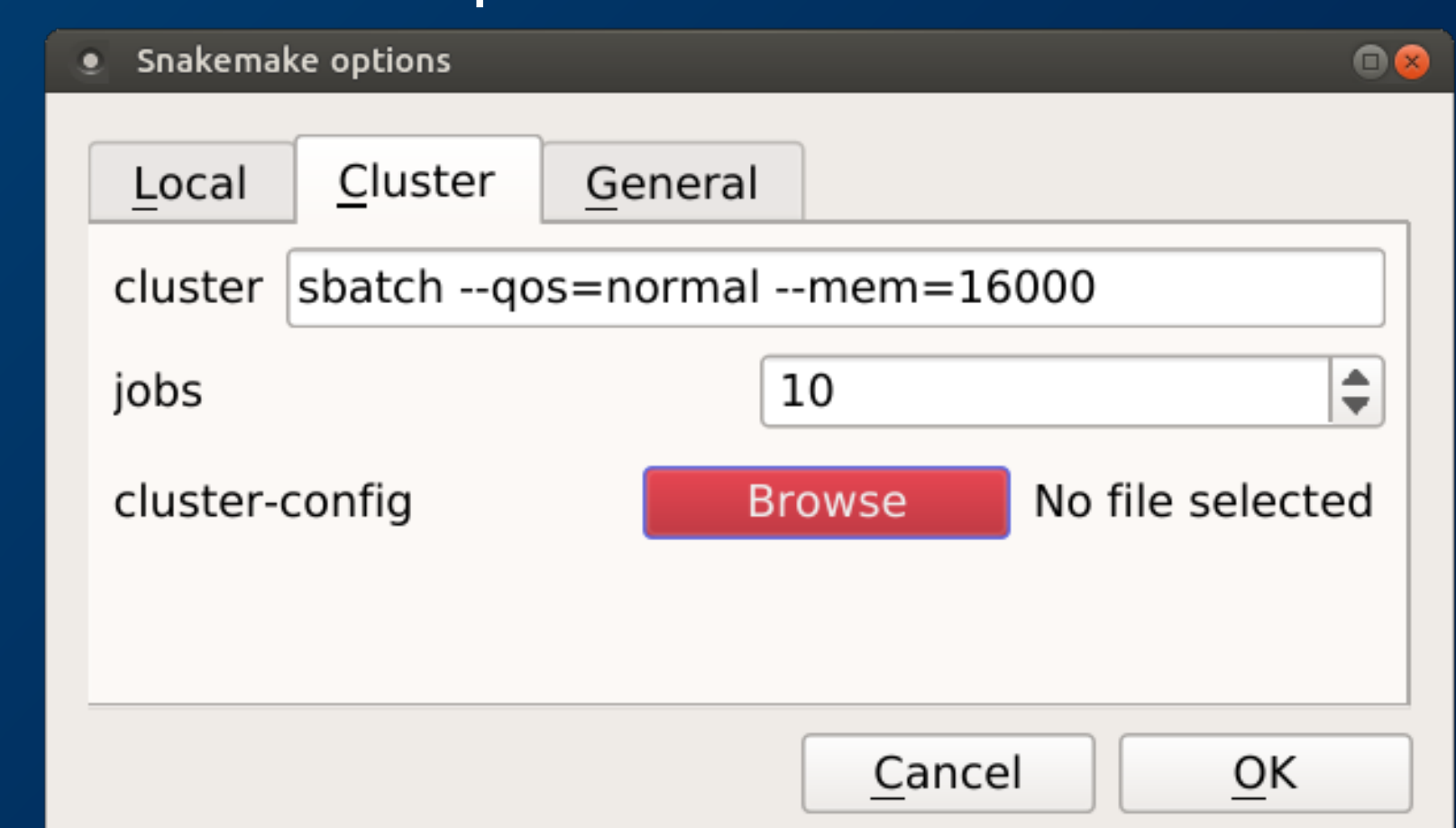


Snakemake options

General options



Cluster options



Conclusion and future directions

Following the development of several NGS pipelines based on the snakemake framework (<http://sequana.readthedocs.io>), we wanted to expose our pipelines through a graphical interface. This was achieved with Sequanix.

Although Sequanix was primarily developed to expose Sequana pipelines to its end-users, it should also benefit a wider community since any Snakemake pipelines can be loaded and executed within Sequanix.

Note that Sequanix is now available on bioconda (<http://bioconda.github.io>, package named sequana).

Using Sequanix, we do not lose the flexibility of Snakemake: the most important parameters are available as a dialog; other parameters can be added manually.

Moreover, Sequanix can be run locally or on a cluster (that allows display).

Github: <https://github.com/sequana/sequana>

Doc: <http://sequana.readthedocs.io>

Contacts: thomas.cokelaer@pasteur.fr

dimitri.desvillechabrol@pasteur.fr

