Institut Pasteur

Sequana: a set of flexible genomic pipelines for processing and reporting NGS analysis

Thomas Cokelaer - Dimitri Desvillechabrol

Institut Pasteur

Nov 9th 2017, Paris (séminaire bioinfo Toulouse)

Motivation

Jan 2015: provide NGS pipelines to Biomics sequencing platform https://research.pasteur.fr/en/team/biomics/ (Institut Pasteur)

- Genomics: QC + variant calling + de-novo
- Transcriptomics: RNA-seq + ChIP-seq
- Metagenomics
- Illumina but also Pacbio long reads technologies

How ?

a glue language, a scientific language

a pipeline framework mixing Python and Makefile

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

Dedicated standalone such as genome coverage characterisation or a graphical user interface for Snakemake pipelines (Sequanix).

Snakemake as a workflow manager

= Makefile +

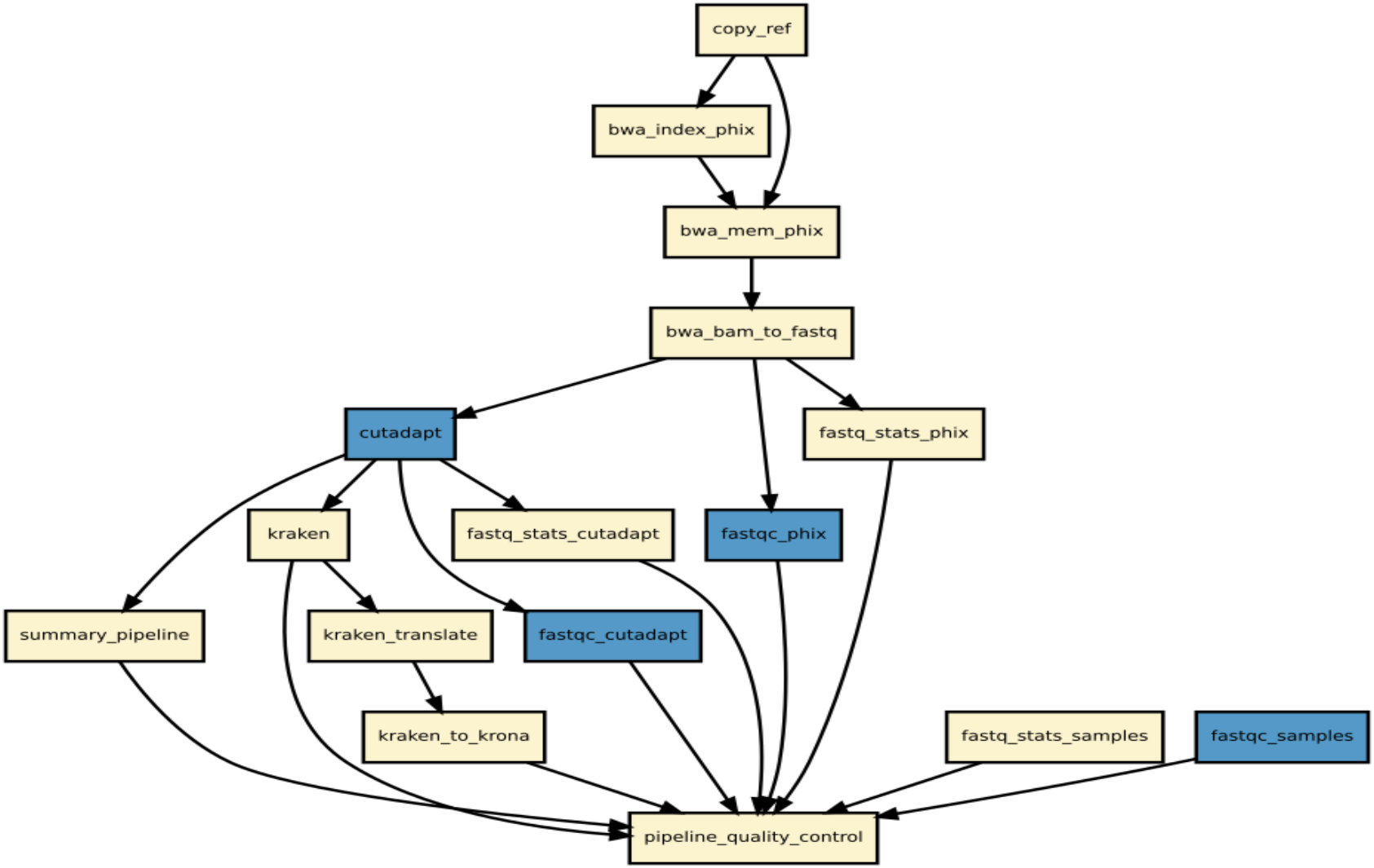
See dedicated slides for a Snakemake overview and tutorial on github.com/sequana/sequana_presentations

configuration file example in YAML format

```
##############################################################
# Input parameters for the fractal analysis
#
# :Parameters:
#
# - size: output image size formatted as NxM where N and M
#         are integers
# - depth: a integer (e.g. 200)
# - zoom: a positive value e.g. 0.5
# - N: number of random sets
gc:
    - window: 100
    - directory: /home/user/fastq_files
```

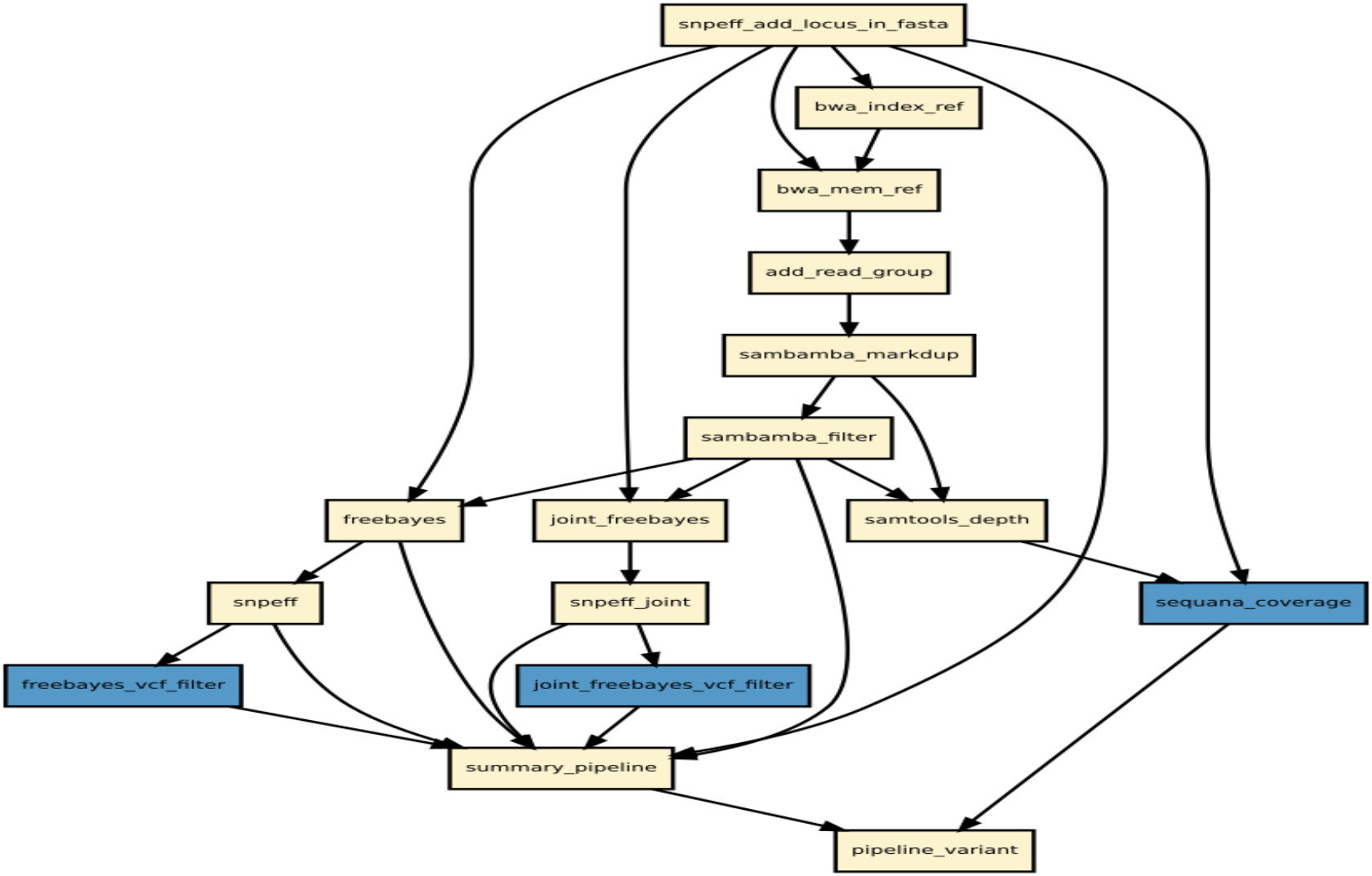
Sequana pipelines (an overview)

Pipeline example: quality control pipeline



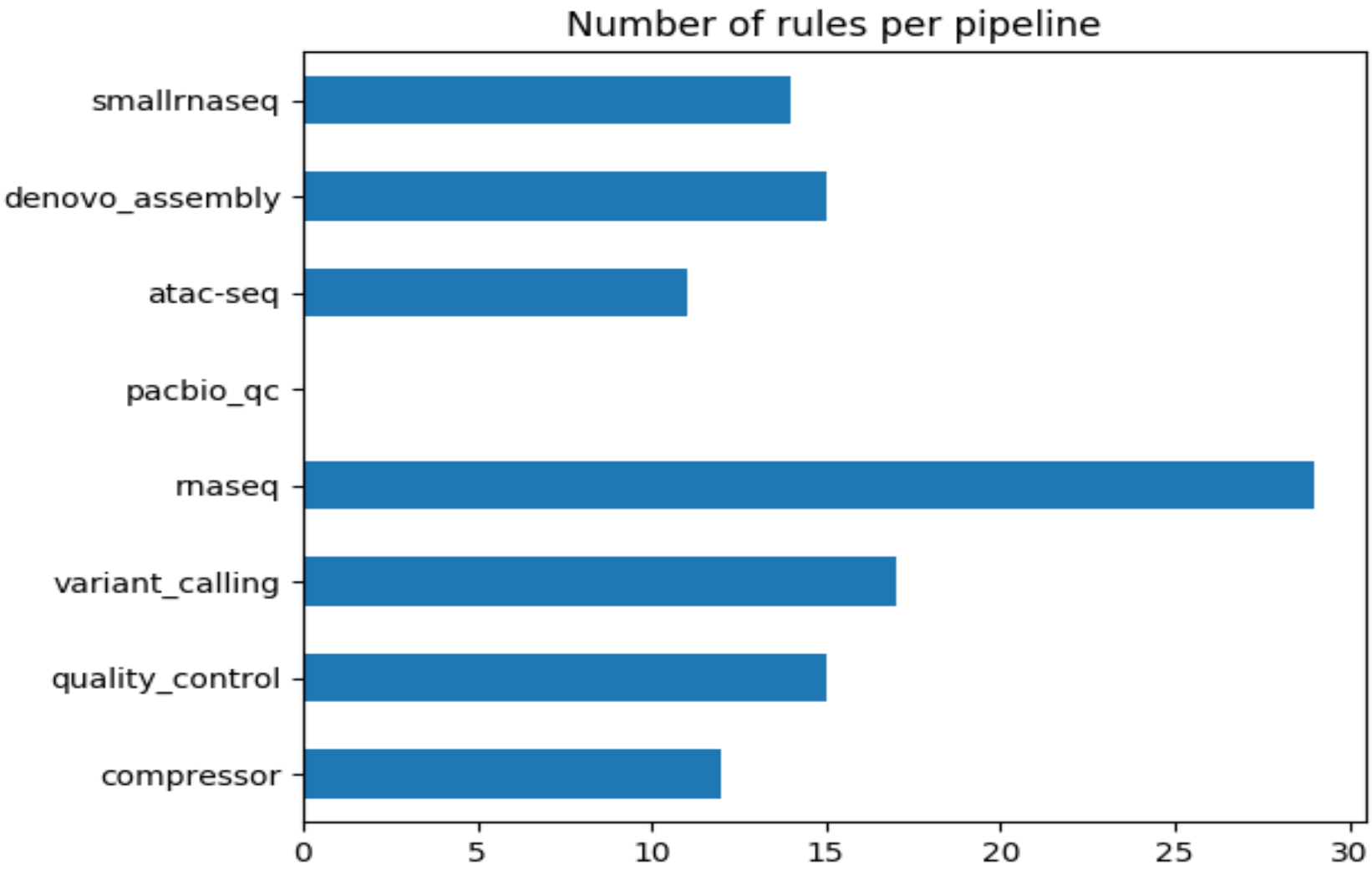
Quality control pipeline (sequana v0.5.2)

Pipeline example: variant calling



Variant calling pipeline (v0.5.2)

Pipeline complexity



Modularization: Factorise and reuse rules

Local standard rules

```
include: "path_to_rule_file"
```

Sequana rules

```
from sequana import snaketools as sm
include: sm.modules['rulegraph']
```

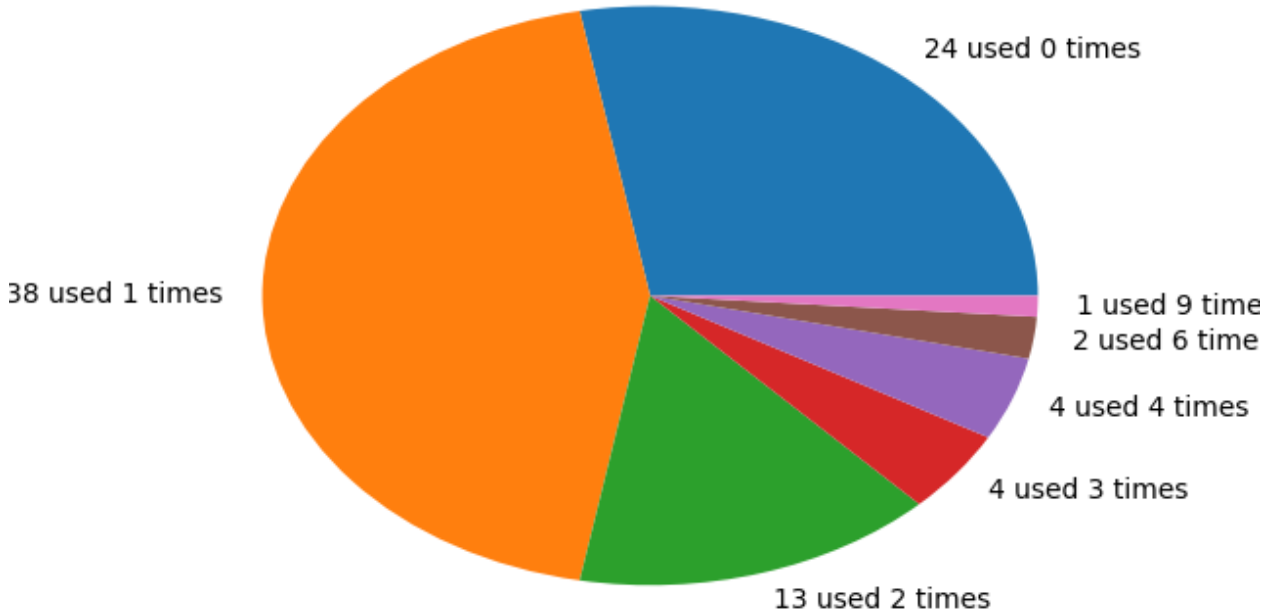
Dynamic rules:

```
sm.init("quality_control.rules", globals())
with open(sequana.modules["fastqc_dynamic"], "r")
    exec(dynrule.read())

manager = sm.PipelineManager("quality_control",
                             config)

include: fastqc_dynamic("example1", manager)
include: fastqc_dynamic("example2", manager)
```

Factorization



Once upon time there was a pipeline ... and a configuration file.

Once upon time there was a pipeline ... and a configuration file.

One need to edit the configuration file ... without typos

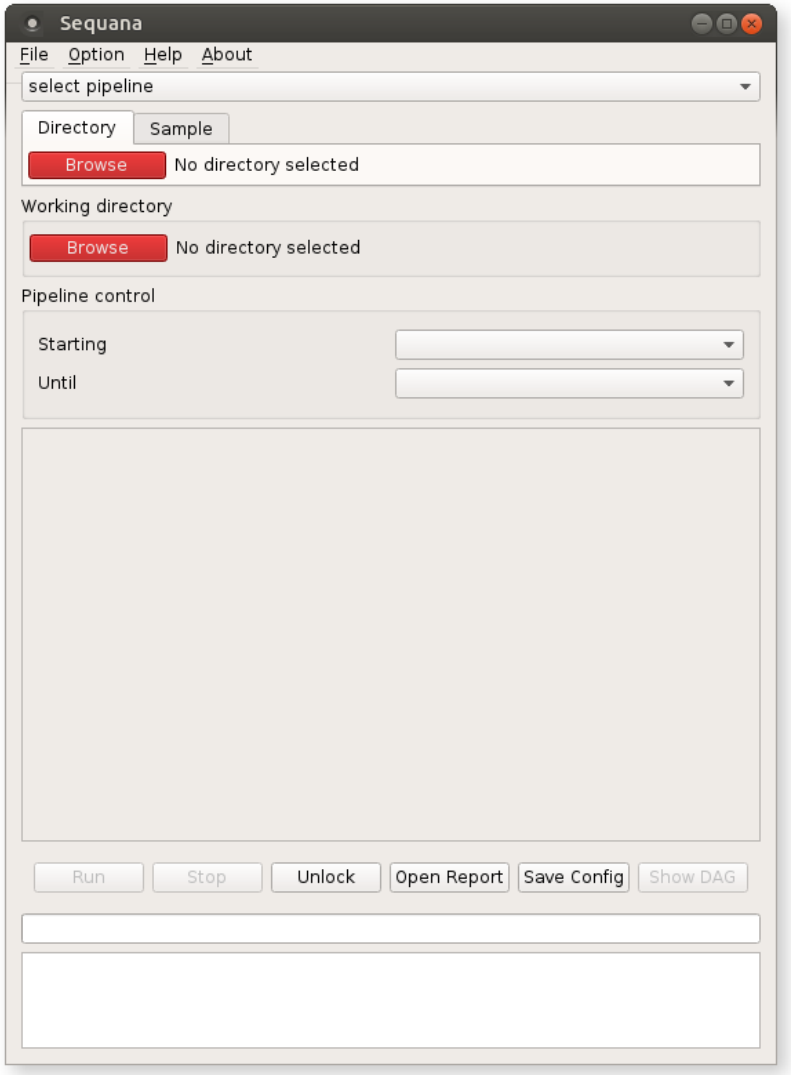
Once upon time there was a pipeline ... and a configuration file.

One need to edit the configuration file ... without typos

One need to launch the Snakemake command ... without typos

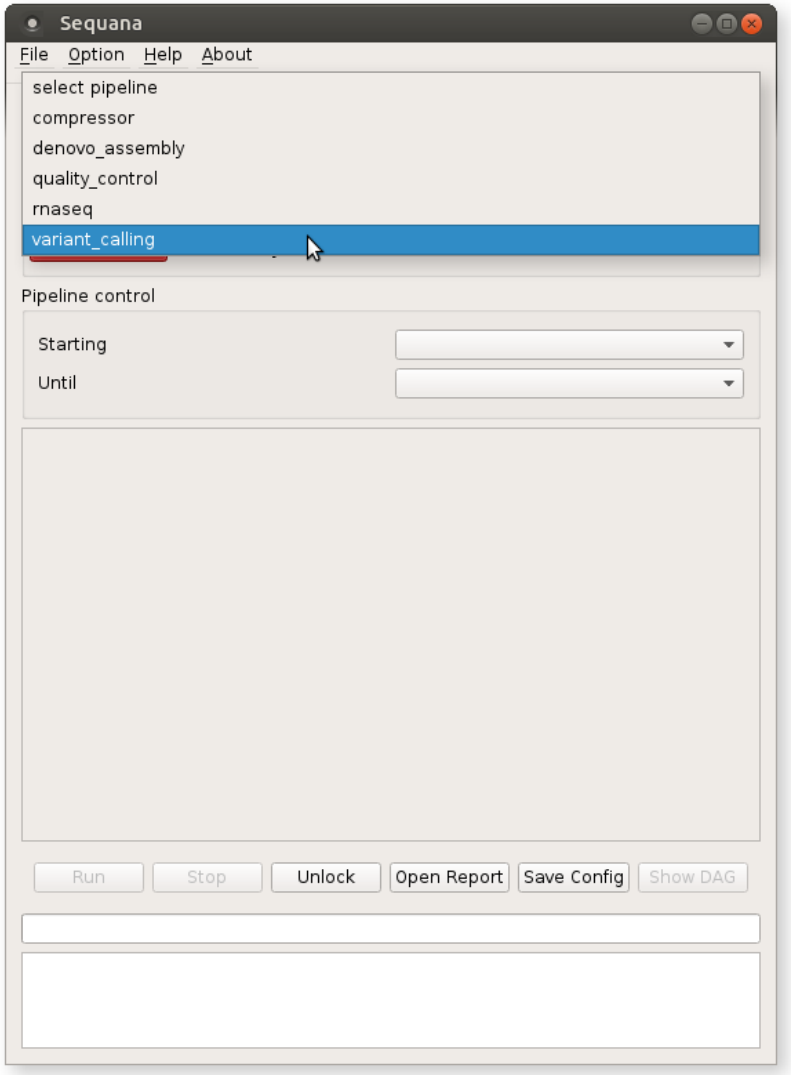
Sequanix

GUI to simplify the usage of snakemake



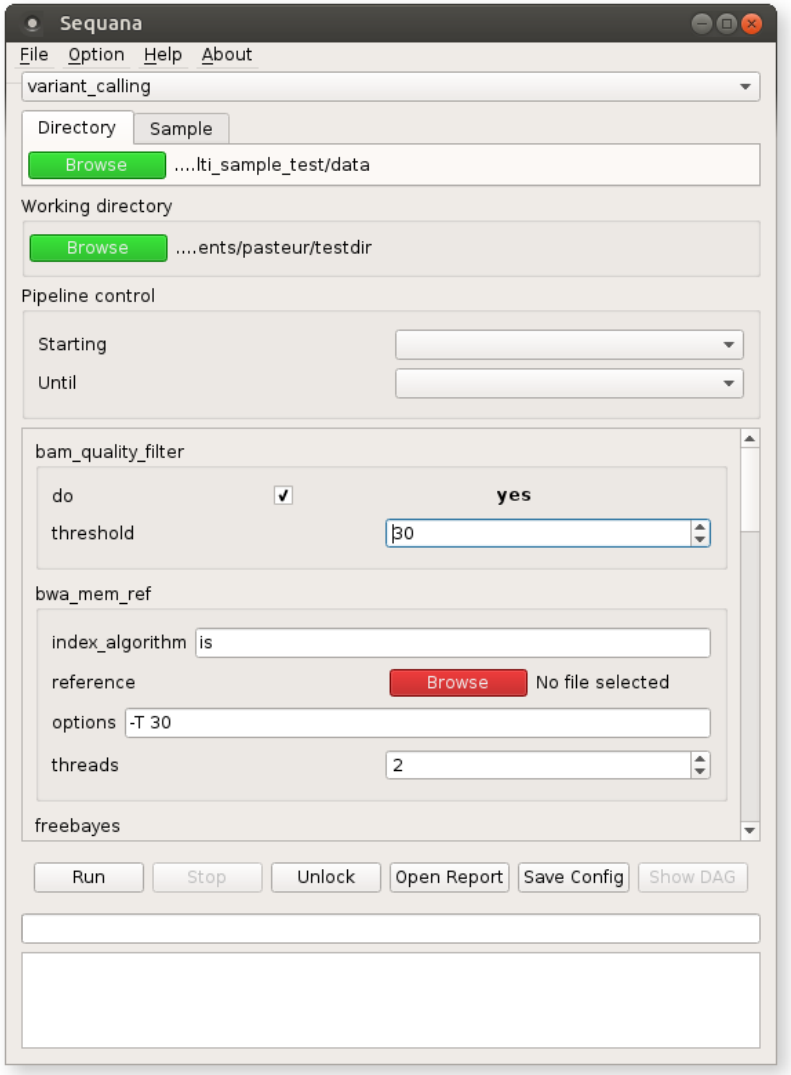
- Interface developed with PyQT5 and python
- Wrap our snakemake pipelines to ease the usage
- Usable on our cluster, which allows X11

GUI to simplify the usage of snakemake



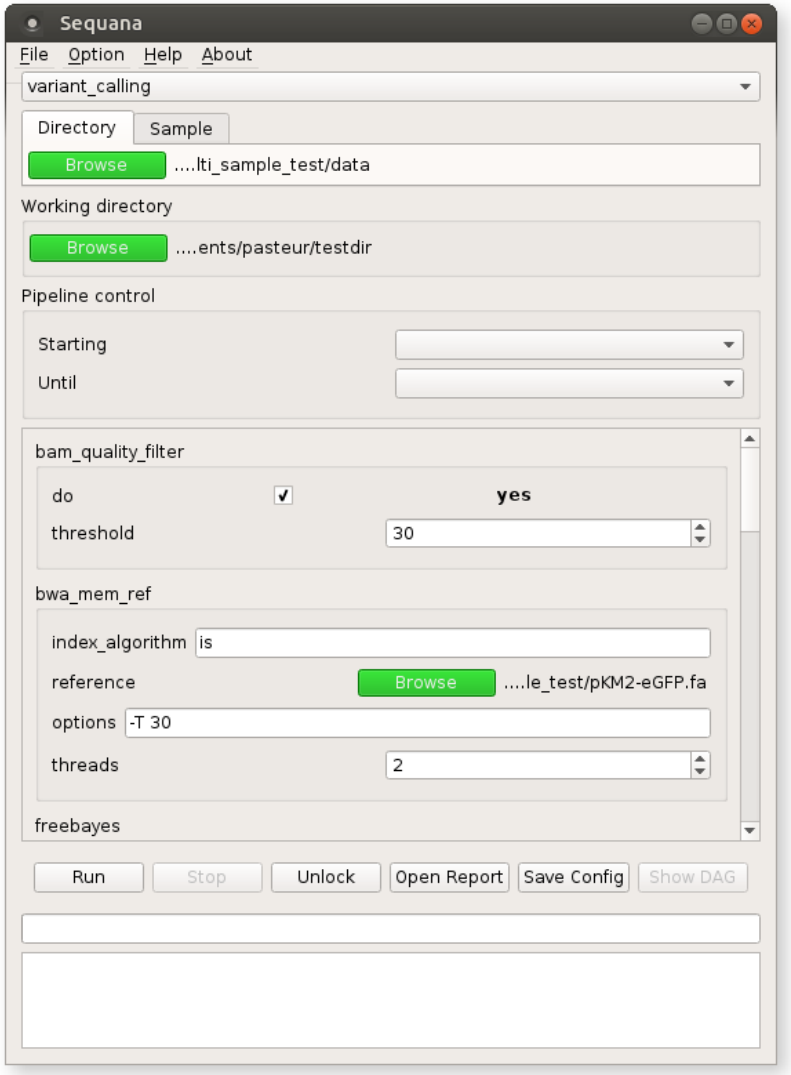
1. Choose a pipeline

GUI to simplify the usage of snakemake



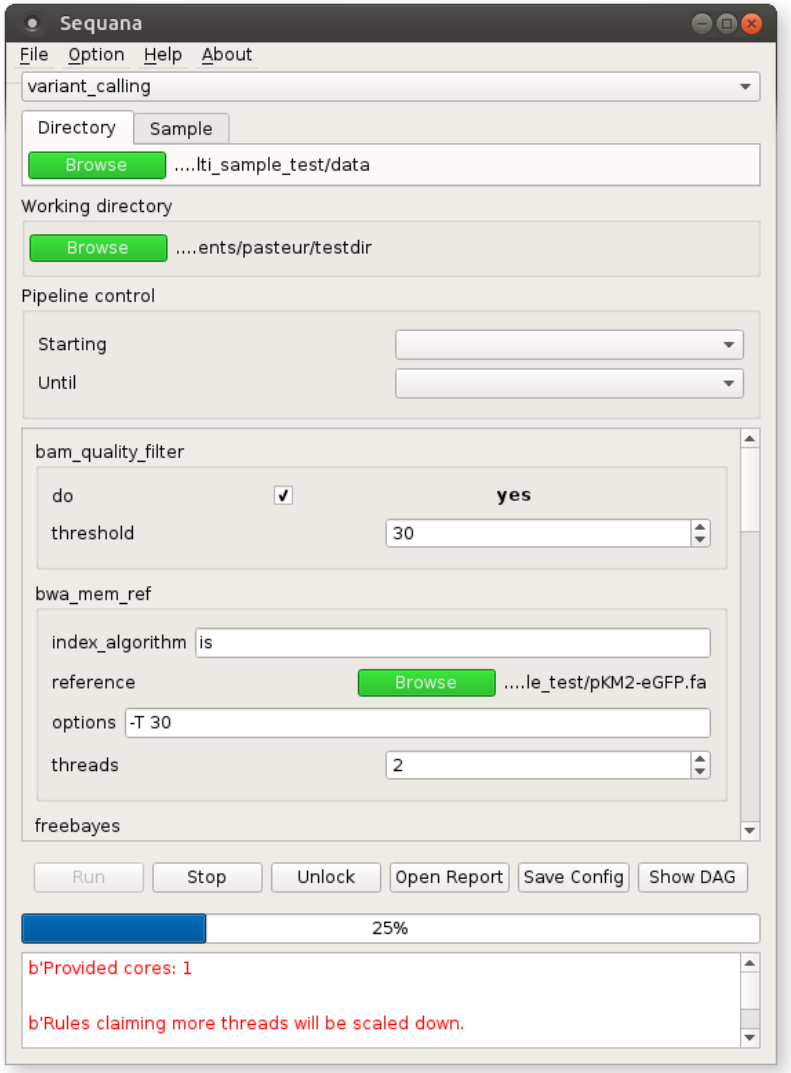
1. Choose a pipeline
2. Set input and output

GUI to simplify the usage of snakemake



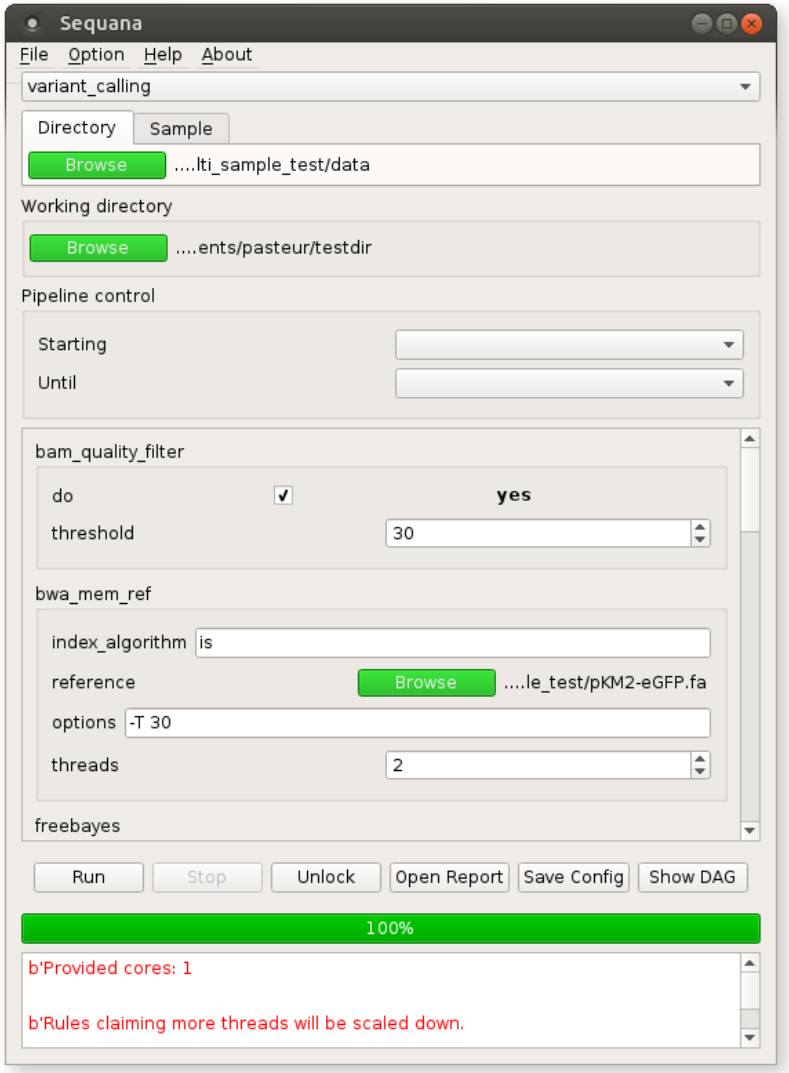
1. Choose a pipeline
2. Set input and output
3. Fill the config formular

GUI to simplify the usage of snakemake



1. Choose a pipeline
2. Set input and output
3. Fill the config formular
4. Run the pipeline

GUI to simplify the usage of snakemake



1. Choose a pipeline
2. Set input and output
3. Fill the config formular
4. Run the pipeline
5. Finished !

Reference

Sequanix: A Dynamic Graphical Interface for Snakemake Workflows

Dimitri Desvillechabrol, Rachel Legendre, Claire Rioualen, Christiane Bouchier, Jacques van Helden, Sean Kennedy, Thomas Cokelaer

https://www.biorxiv.org/content/early/2017/07/12/162701

Sequana coverage

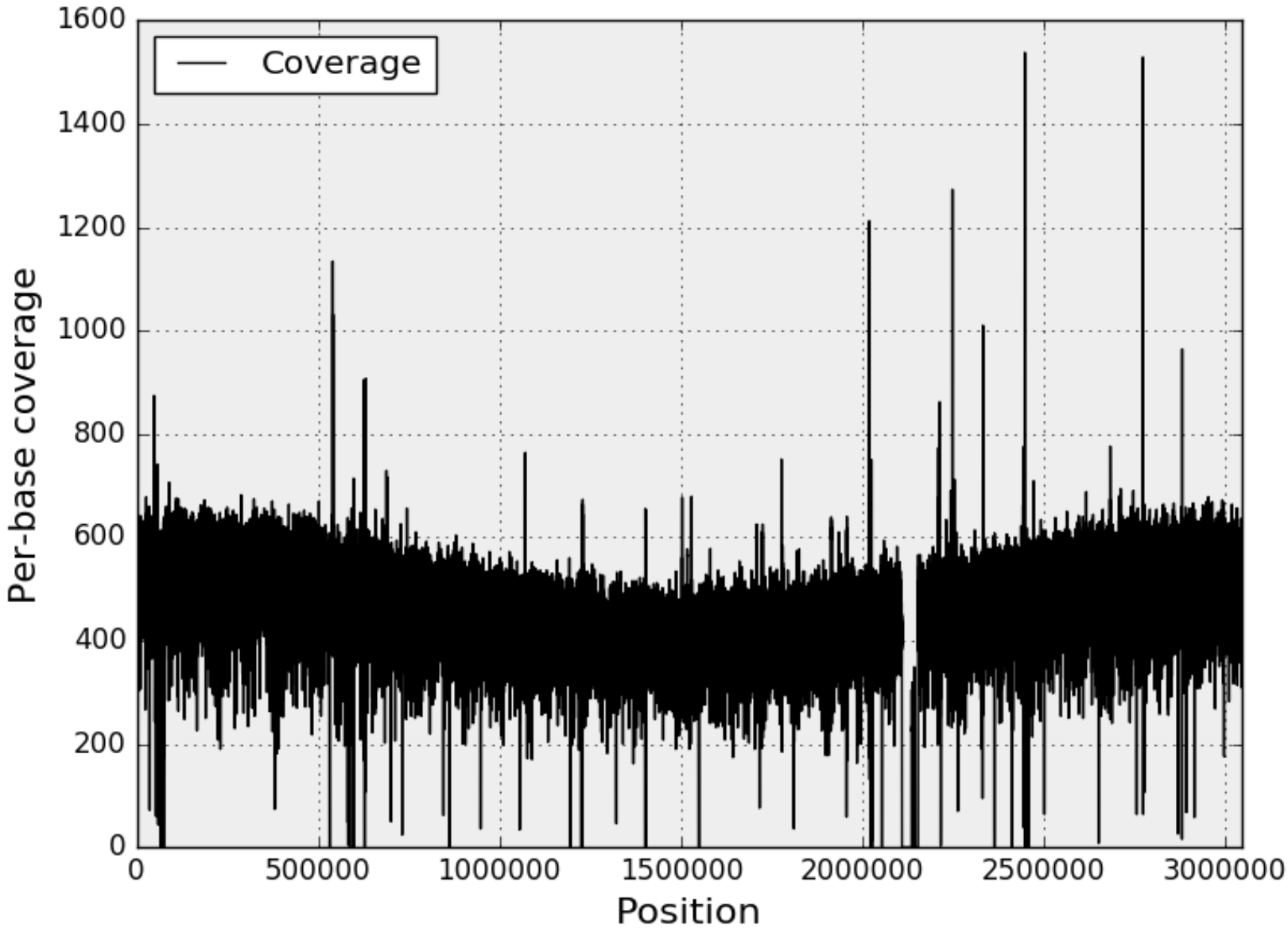
Genome coverage

Definition: The number of reads mapped to a specific position, b, within the reference genome.

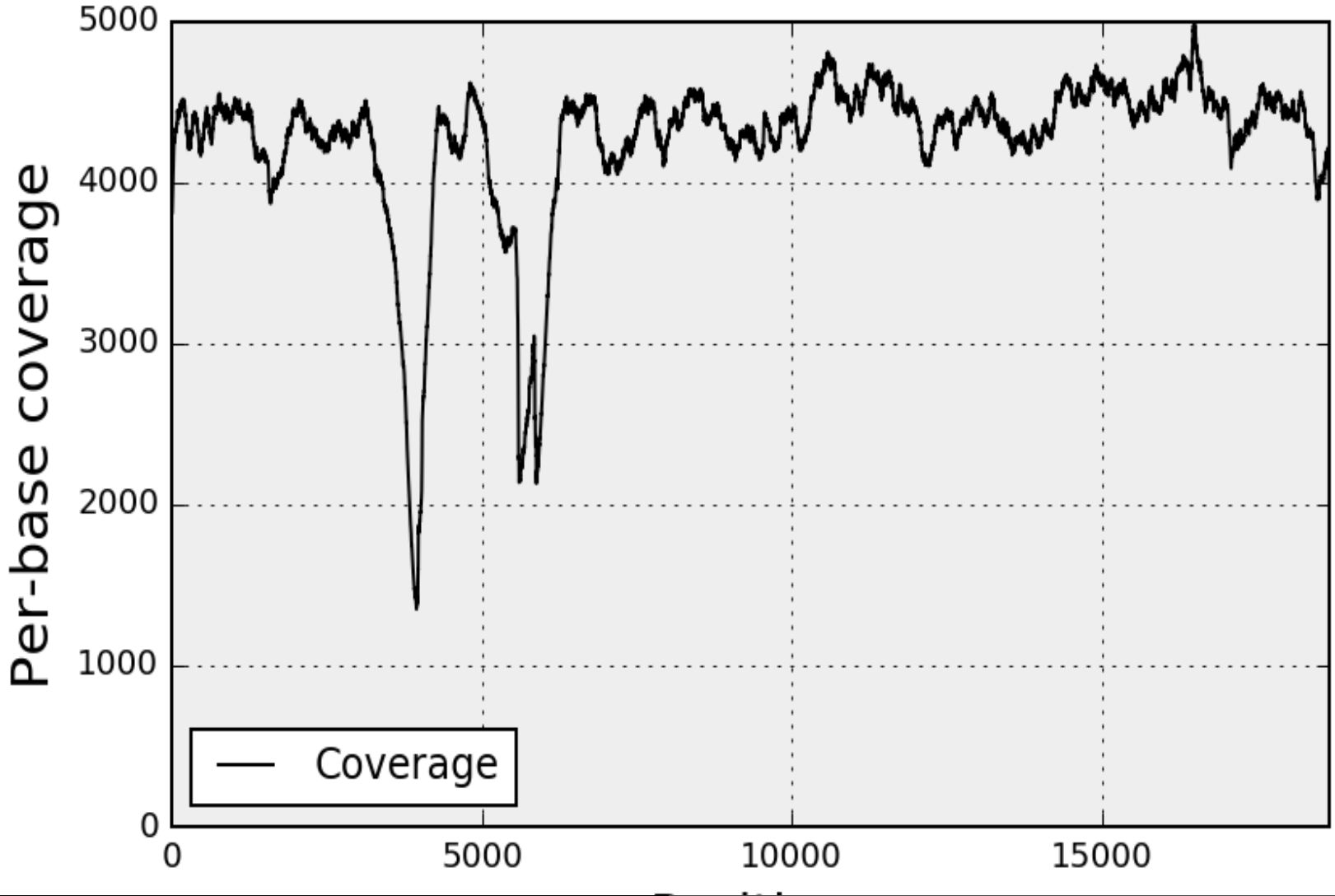
Notation: $C(b)$ also denoted C_b

Theoretical distribution: Poisson distribution but in practice over dispersed. The poisson parameter is distributed according to a Gamma hence leading to a negative binomial (See e.g., Linder et al 2013).

Bacteria case (low/high μ components and del. region)



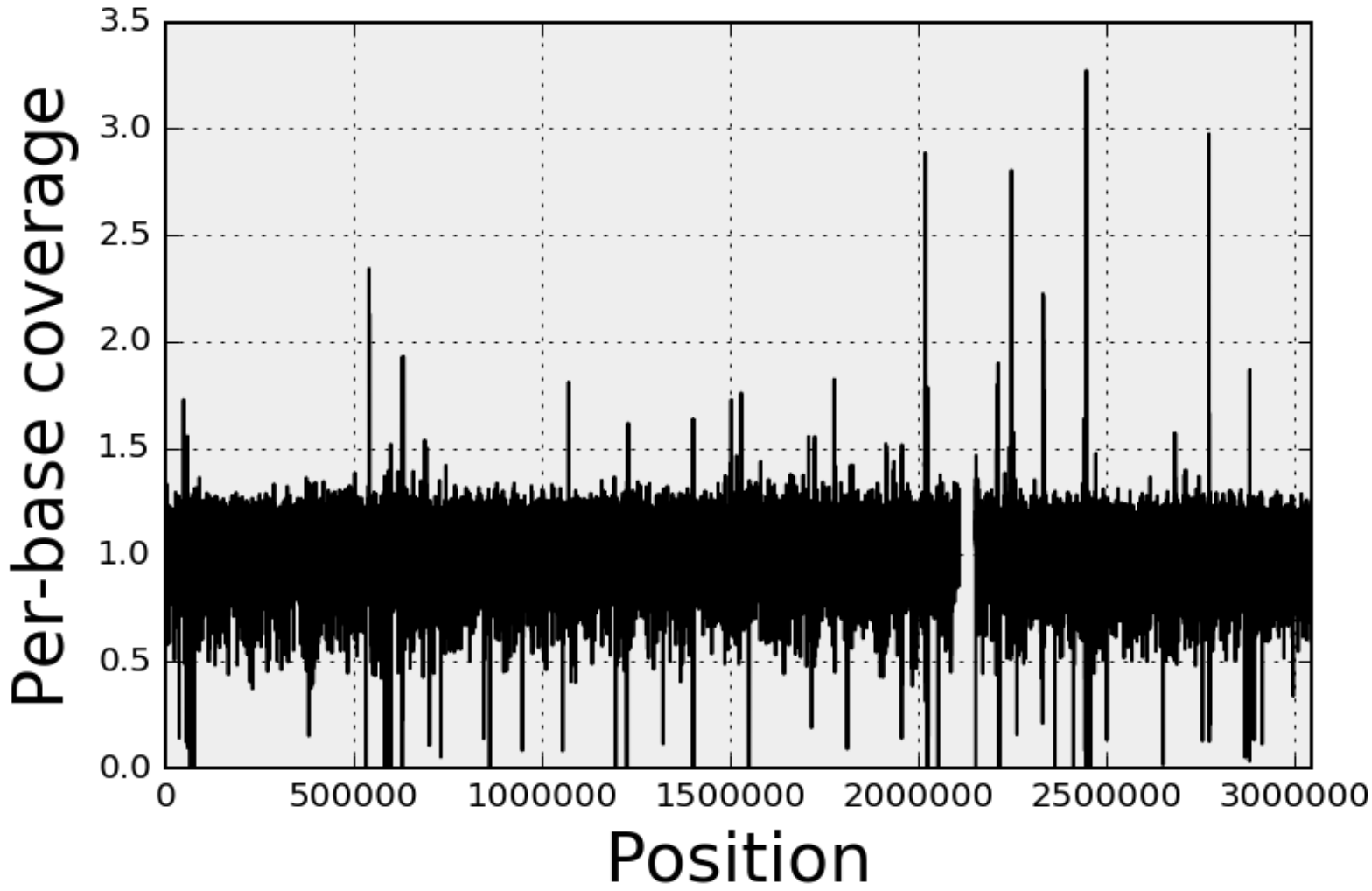
Virus case


Per-base coverage
5000
4000
3000
2000
1000
0
0
5000
10000
15000
Coverage

Question: how to automatically detect and characterise under and over covered genomic regions

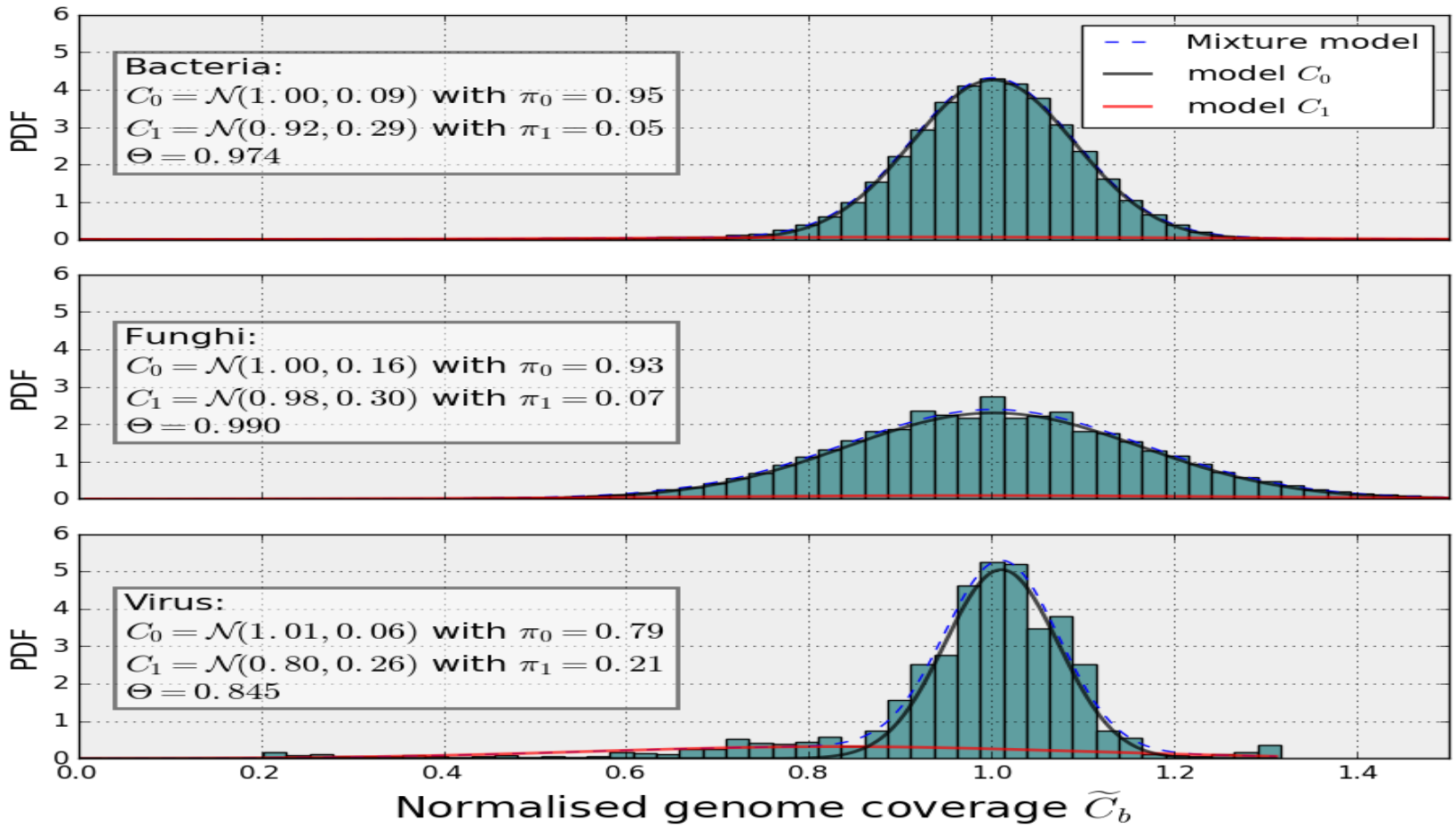
1. The algorithm
   1. Detrending (running median)
   2. Mixture model estimation (Gaussian approximation)
   3. Set a statistics (z-score)
   4. Clustering (double threshold)

1. Detrending (Normalised coverage)

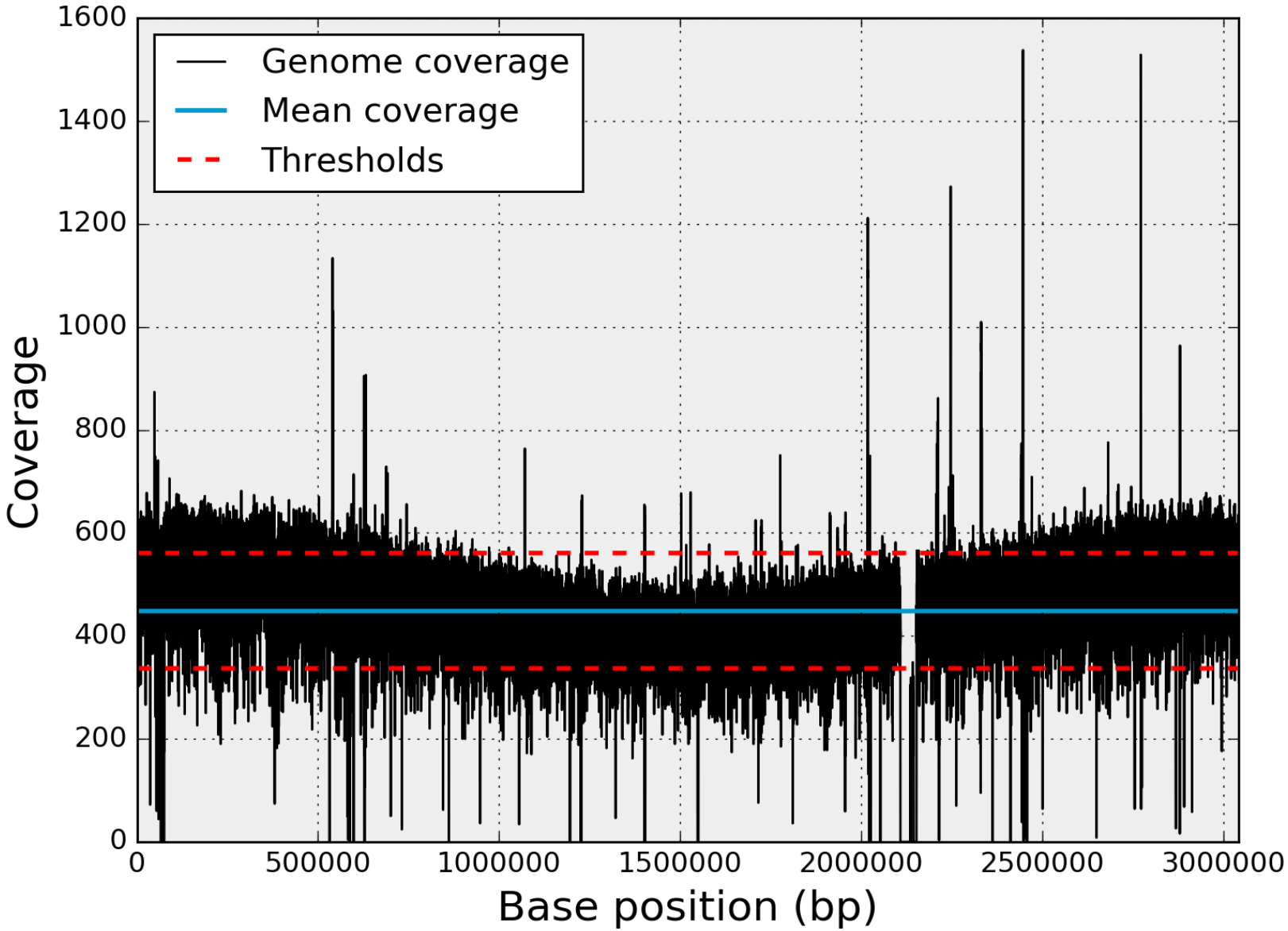


2. Building a statistics

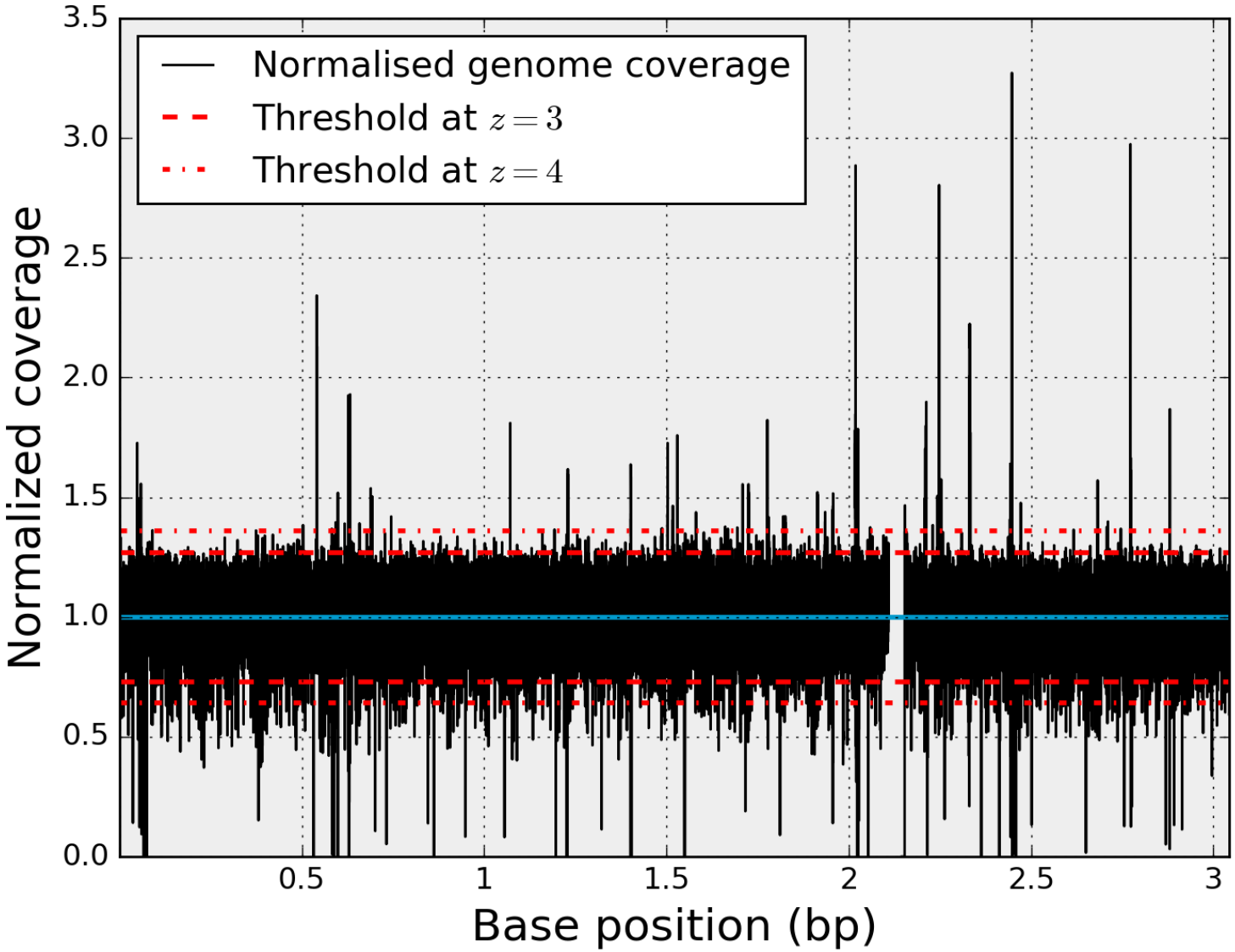
Hypothesis 2: The normalised genome coverage follows a Gaussian distribution in particular the central distribution



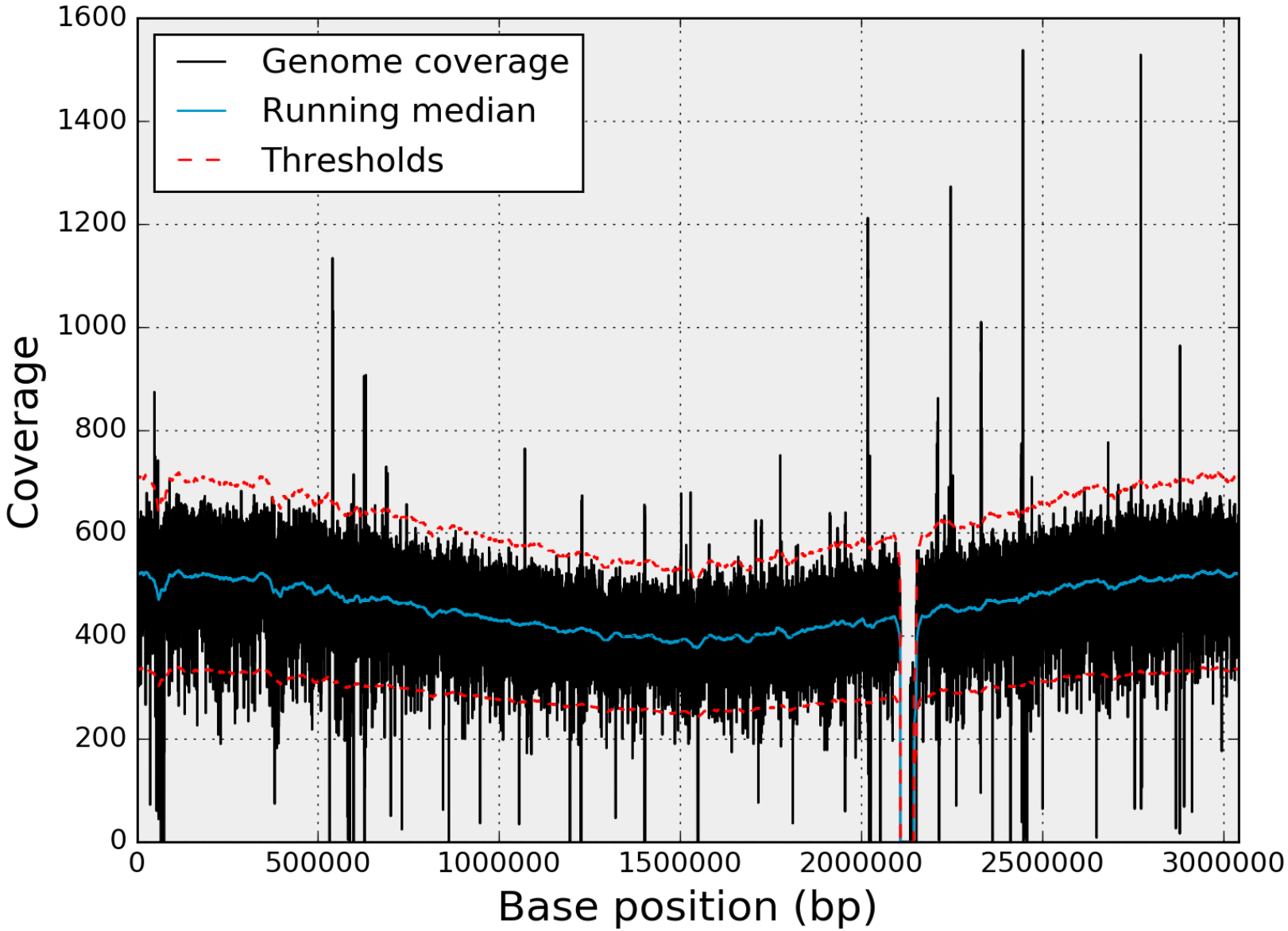
C. From a constant to adaptative *z*-score



C. From a constant to adaptative z-score



C. From a constant to adaptative *z*-score



Detection and characterization of low and high genome coverage regions using an efficient running median and a double threshold approach.

Dimitri Desvillechabrol, Christiane Bouchier, Sean Kennedy, Thomas Cokelaer

bioRxiv 092478; doi: http://dx.doi.org/10.1101/092478

Sequana: Continuous integration

Versioning, Test and Documentation

https://github.com/sequana/sequana

Continuous Integration on Travis with 185 tests with 85% coverage

Uses Sphinx (RST syntax) to document the source code and provides user guide.

Updated after each commits on sequana.readthedocs.io

Summary

Summary

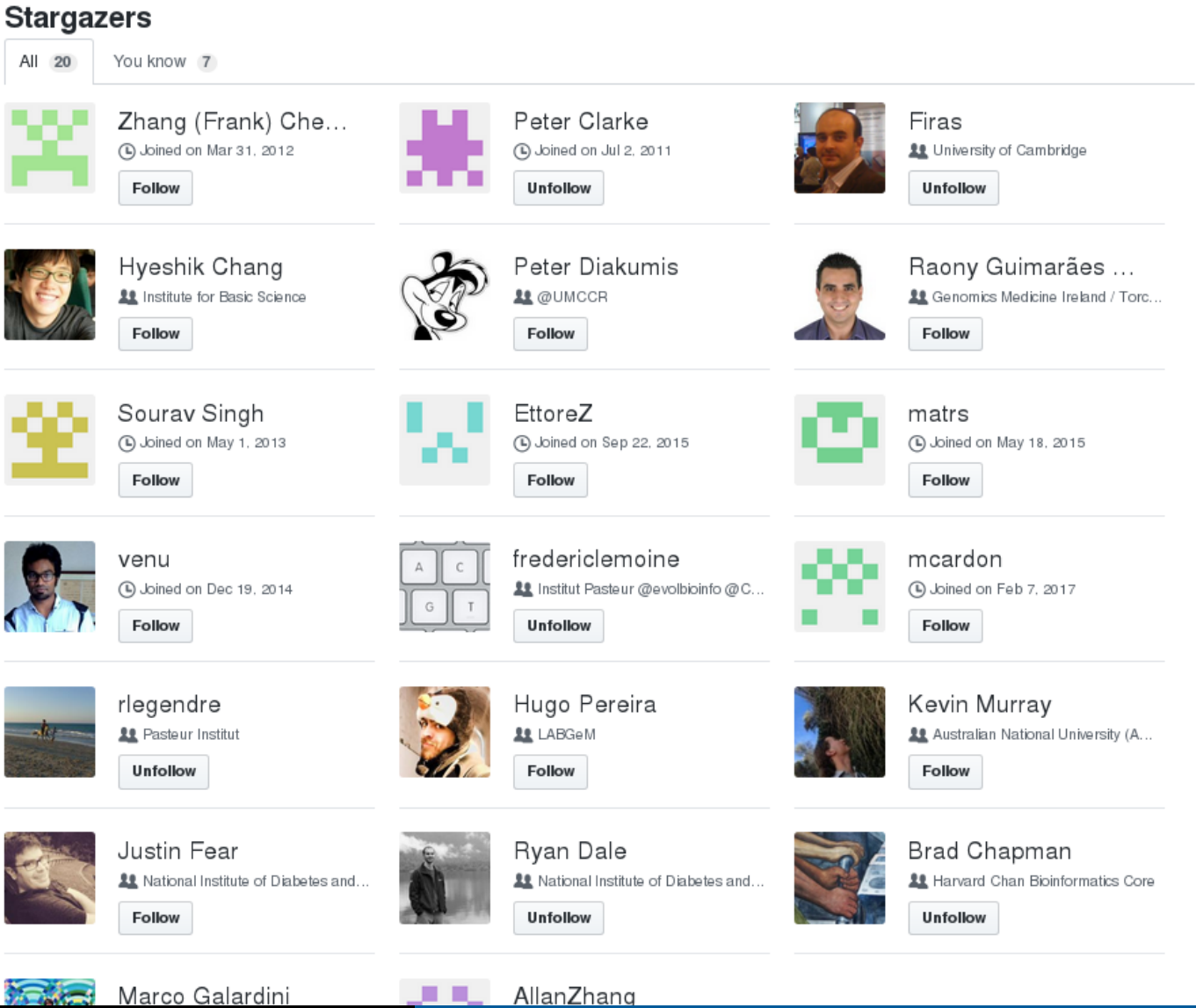
Sequana is a versatile tool that provides

1. A Python library dedicated to NGS analysis (e.g., tools to visualise standard NGS formats).
2. A set of snakemake workflows and rules dedicated to NGS
3. A GUI to execute them easily with Sequanix
4. HTML reports
5. Standalone applications:
   - **sequana_coverage** ease the extraction of genomic regions of interest and genome coverage information
   - **sequana_taxonomy** get a quick overview of read contents
   - ...

Please visit sequana.readthedocs.io for more info or check out github.com/sequana/sequana for the code.

You like it ? Please, add a star on our github

https://github.com/sequana/sequana/stargazers



You like it ?

Join us ! add rules and pipelines !

Acknowledgements

- Dimitri Desvillechabrol (variant calling, denovo, sequana, sequanix)
- Rachel Legendre (Transcriptomics)
- Mélissa Cardon (pacbio)
- Biomics users (Institut Pasteur)

Thank you

Slides available on http://github.com/sequana/sequana_presentations/