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

Dimitri Desvillechabrol[1], Christiane Bouchier[2], Thomas Cokelaer[3], Sean Kennedy[1]
1 : Institut Pasteur - Pôle Biomics, CITECH - Paris, France
2 : Institut Pasteur - Plate-forme Génomique - Pôle Biomics, CITECH - Paris, France
3 : Institut Pasteur – Hub Bioinformatique et Biostatistique – C3BI, USR 3756 IP CNRS - Paris, France

Motivation

Sequencing platforms that produce sequencing data must developed robust, fast and flexible pipelines to guarantee that good-quality data are delivered to research labs. The Biomics Pole (Institut Pasteur) produces 100 runs a year combining MiSeq and HiSeq 2500 systems. In order to analyse the platform sequencing data on a daily-basis, we developed Sequana (Sequence Analysis), a Python-based software dedicated to the development of NGS pipelines. Sequana provides (1) NGS pipelines for production (in the form of Snakefile workflows), (2) a Python library with convenient tools to bridge the gap between heteregenous tools and (3) a set of HTML/ Javacript reports. Here, we present some of the pipelines currently available from quality control to variant calling.

Building pipelines based on Snakemake

Rules

```
__vcf_filter__output = cfg.PROJECT + "/vcf_filter/%s_filter.vcf" % cfg.PROJECT

rule vcf_filter:
    input:
        vcf = __vcf_filter__input
    output:
        vcf = __vcf_filter__output
    run:
        from sequana import vcf_filter

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

Snakefile

```
include: sm.modules["freebayes"]
__vcf_filter__input = __freebayes__output

include: sm.modules["vcf_filter"]
__report_vcf__input = __vcf_filter__output

include: sm.modules["report_vcf"]
```

Shell

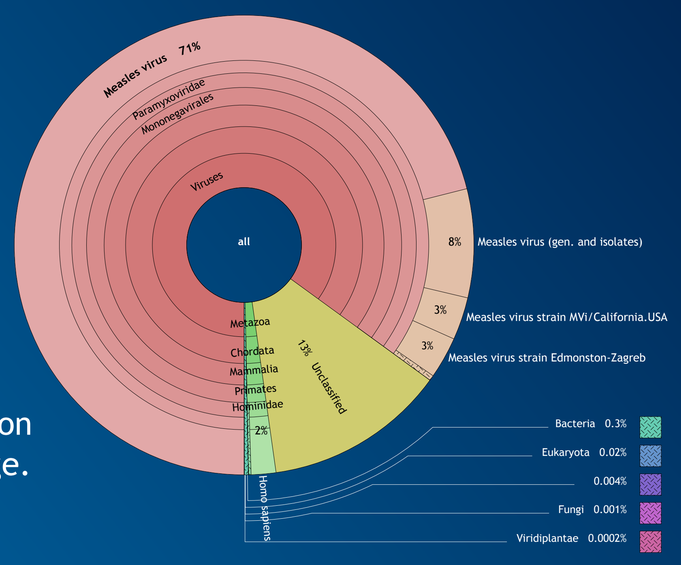
```
> snakemake --configfile config.yaml
```

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

Example 1: Fast contaminant detection

Within Sequana, we provide a tool to quickly assess the taxonomic content of FastQ reads to search for known contaminants. It is based on Kraken for the taxonomic identification and Krona for the visualisation. Sequana is used to bridge the gap between Kraken and Krona. Analysis performance is ~1,000,000 reads/minutes.

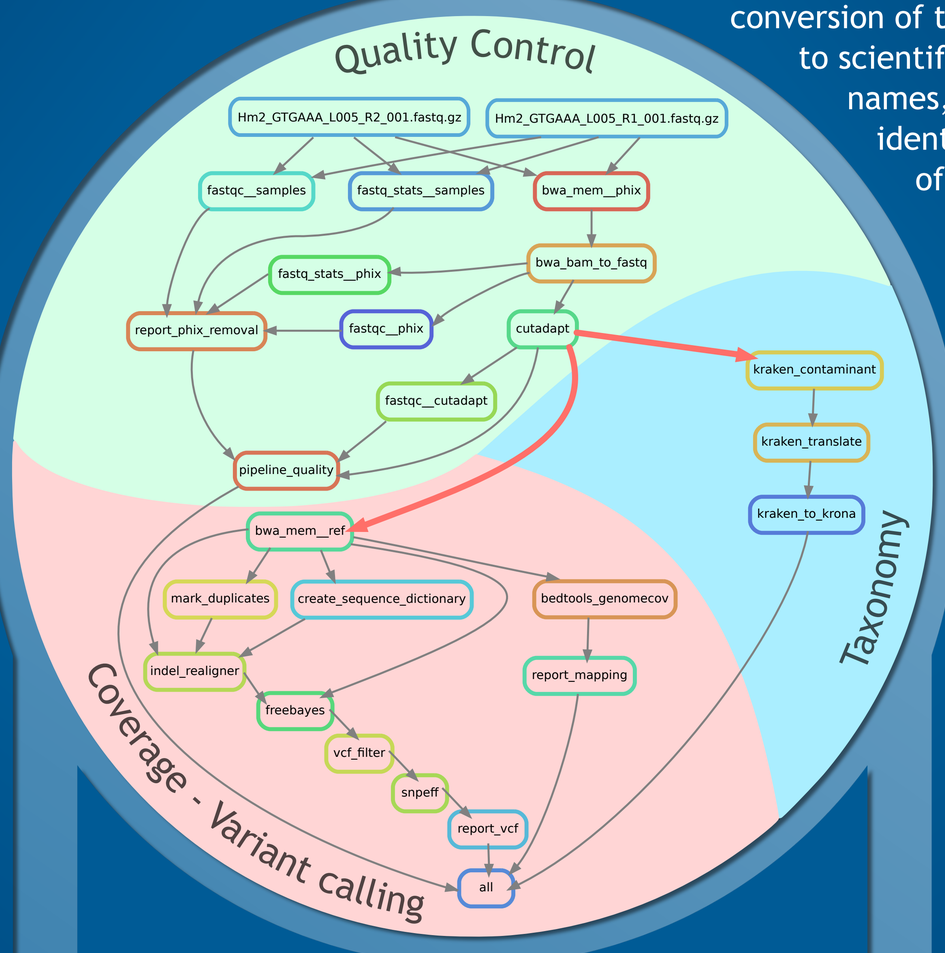
Sequana also provides tools to create custom Kraken databases from scratch: using Bioservices, the following tasks are perfomed seamlessly: download of ENA accession numbers, conversion of taxon to scientific names, identification of lineage.



Ondov BD, Bergman NH, and Phillippy AM. Interactive metagenomic visualization in a Web browser. BMC Bioinformatics. 2011 Sep 30; 12(1):385.

Cokelaer et al. BioServices: a common Python package to access biological Web Services programmatically Bioinformatics (2013) 29 (24): 3241-3242

Wood DE, Salzberg SL: Kraken: ultrafast metagenomic sequence classification using exact alignments. Genome Biology 2014, 15:R46.

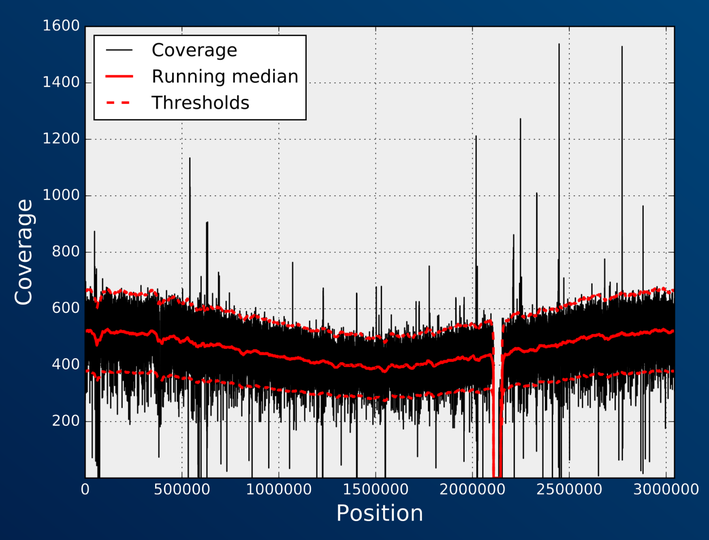


Example 2: Coverage characterization

Sequencing depth (coverage) can be used to assess the mapping quality and to identify genomic regions with unusual coverage.

In Sequana, we implemented an algorithm to detect regions with low or high coverage. In a nutshell, a running median estimates the trend in the coverage (see Figure). It is used to normalize the coverage, from which we estimate the profile (assuming a Gaussian distribution) to scale the original data onto z-scores.

The figure below shows the original coverage (black), the running median (central red) and z-scores mapped back to the original data corresponding to 2 thresholds(± 3σ).



Example 3: Variant calling

SNP and short INDEL detection is a standard method to compare a sequenced genome against a reference. Sequana provides a flexible pipeline to perform variant calling (middle figure).

Many tools exist to perform the variant calling analysis. We chose Freebayes. Indeed, it takes into account information held by reads and can be used without filtering (keeping all variants).

We developed tools to filter VCF files and to report results in a convenient HTML table. Thus, users can change their filters options without rerunning Freebayes to obtain their new results almost instantly. A wrapper of snpEff to annotate variants is also available in Sequana. It automatically handles the creation of custom database when the user provides an annotation file.

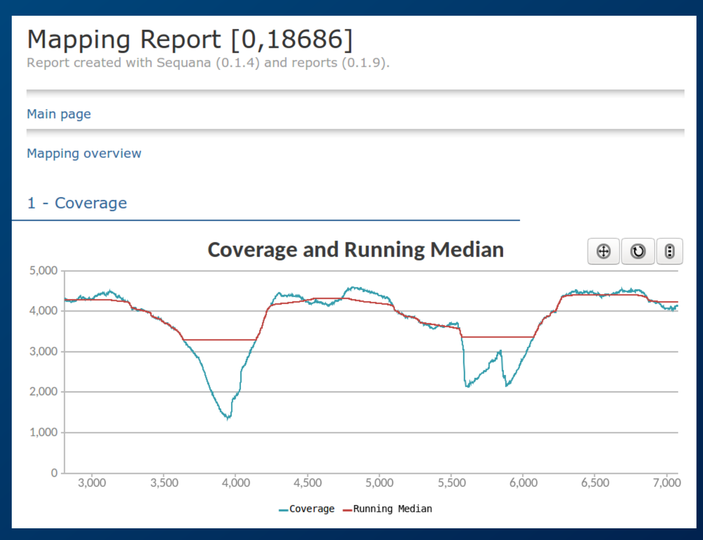
Garrison E, Marth G. Haplotype-based variant detection from short-read sequencing. arXiv preprint arXiv:1207.3907 [q-bio.GN] 2012

Reports

In addition to pipelines and original algorithms, Sequana provides a re-usable set of JINJA templates used to create HTML reports. These templates alleviate the needs to code directly in HTML, which may be cumbersome on long-term development. Furthermore, re-usable templates helps developers to **quickly design and develop** new types of reports.

Reports include the input/output, the config file, the pipeline used and all versions of the different tools that were used in the pipeline.Thus, analyses are **reproductible**.

Some reports include **interactive** Javascript like in the coverage report case shown here below.



Conclusion and future directions

We have designed a software, Sequana, that allows us to quickly design NGS pipelines, which are based on Snakemake framework. Currently, we have implemented pipelines to access quality control of reads, perform a fast taxonomic classification, detect variants, characterise coverage. Although we use established tools, we also developed original algorithms and tools. We provide HTML reports and ability to reproduce the results. For more information, thorough documentation is available on github.

Future extension to the Sequana pipelines is to consider long reads. Other pipelines may also be included to cover the needs of collaborators or sequencing platforms.

Github: https://github.com/sequana/sequana
Doc: http://sequana.readthedocs.io
Contacts: thomas.cokelaer@pasteur.fr
dimitri.desvillechabrol@pasteur.fr