

A Snakemake workflow for WGS data analyses of *Chlamydia trachomatis* genome.

Whole genome sequencing data analyses of *Chlamydia trachomatis* genome from clinical samples was performed using Snakemake - a Python-based pipeline management tool for creation of reproducible workflows.

General information

The pipeline requires the reference genome of human host and reference genome of *Chlamydia trachomatis*. The reference genome for *Chlamydia* that is used in this pipeline is the pan genome that was generated by seq-seq-pan (Jandrasits et al., 2018) by combining *Chlamydial* reference genomes of serovars A, B, D, E, F, G, J and K. The index files for *C. trachomatis* reference genome is created by the following command

```
bowtie2-build ../mac/samples/reference/ct_pangenome_full.fasta  
../mac/samples/reference/ct_fullpangenome/ct_pan
```

```
samtools faidx ../mac/samples/reference/ct_pangenome_full.fasta
```

```
picard CreateSequenceDictionary R=../mac/samples/reference/ct_pangenome_full.fasta  
O=../mac/samples/reference/ct_pangenome_full.dict
```

Bowtie2 is used to map the *Chlamydia* sequences from the isolates and the mapped sequences are pre-processed, preparing them for variant calling with GATK. The output files from all the samples are merged and then SNPs and INDELs are called in separate files. Both, SNPs and INDELs are filtered and then a table with all the required information from the SNPs and INDELs is generated. The quality checks are performed at various stages of the analysis and towards the end a final report is generated that lists the statistics for each performed analysis step.

Platform specific installation notes

The implementation was tested on a Linux and a MACOS system, as long as the majority of tools are available for the two platforms. Moreover RAM-intensive jobs will most likely be executed on a cluster, or take too long on dual core processor. Mostly, the environment built using Miniconda is analogous between the two platforms, however some packages require platform-specific implementation of dependencies (e.g. libgcc and libcxx). Therefore, I have included different environment YAML-files for the two implementations. The normalization step using bbnorm takes too long to perform on 2.9 GHz Intel Core i5, so the normalization step output files are provided for the test run with the samples (CTD_1 and CTD_2).

Configuring the workflow

The configuration file config.yaml offers some flexibility to the user – some (or all) of the parameters can be passed onto the executables using the specified fields. The user can keep or remove the duplicates from the sequencing data. The read groups can be named to suit the user requirement. MULTIQC parameter field refers to a separate configuration file stored in the rules folder, which provides an order of the reports that are generated during the execution of the workflow. Furthermore, an on/off switch is provided for the execution of BBNORM. If the coverage is homogeneous then normalisation step is not advisable. But since in these

samples, it was not homogeneous (fastqc report) so by default, it can be changed. To introduce this option in the workflow, an if-statement has been added in the mapping step with BOWTIE2 and when generating the MULTIQC report.

Step by step instructions:

1). First, you have to install the Miniconda Python3 distribution from this link (<https://conda.io/en/latest/miniconda.html>). Make sure you answer yes to the question whether conda shall be put into your PATH.

2). Obtain the copy this workflow
git clone https://github.com/casrssk/WGS-analysis-in-C_trachomatis.git

Make it your current working directory
cd WGS-analysis-in-C_trachomatis.git

3). The folders common to linux and mac platform are outside the mac and linux folders. The user can move them into the samples folder (mac or linux) depending on the implementation platform. Make samples as your current directory for running the rest of the pipeline. The annotation file is too large to be put on github and so I am providing a link from onedrive from where the folder can be downloaded from this link <https://1drv.ms/u/s!AoZAmrQJEC56gQKjbO-kElU8uZcZ?e=JvYqHv>

The sample data and the normalized data can also be downloaded from this link

4). You need to set up conda channels and then create a virtual environment and install snakemake and python into it using the following command

```
cd <path to the folder containing environment.yml>
conda install --channel "AgBiome" package
conda config --append channels AgBiome
conda env create -c conda-forge -c bioconda -c defaults -c R -f environment.yml
```

5). Execute the workflow by first testing it using dry-run option
snakemake -n

If it runs without errors
snakemake --cores <specify the number of cores>

6). After successful completion, you can investigate the results by
snakemake --report report.html

