NextFlow pipeline for SARS-CoV-2 Illumina data

Table of contents

Quickstart

Installation and usage

1. Install NextFlow

```
curl -s https://get.nextflow.io | bash
mv nextflow ~/bin
```

2. Build three containers:

```
docker build --target production -f Dockerfile-main -t
nf_illumina_sars-3.0-main .
docker build --target prodcution -f Dockerfile-manta -t
nf_illumina_sars-3.0-manta .
docker build --target updater -f Dockerfile-main -t nf_illumina_sars-
3.0-updater:latest .
```

3. Download latest version of external databases:

In project root dir run:

```
./update_external_databases.sh
```

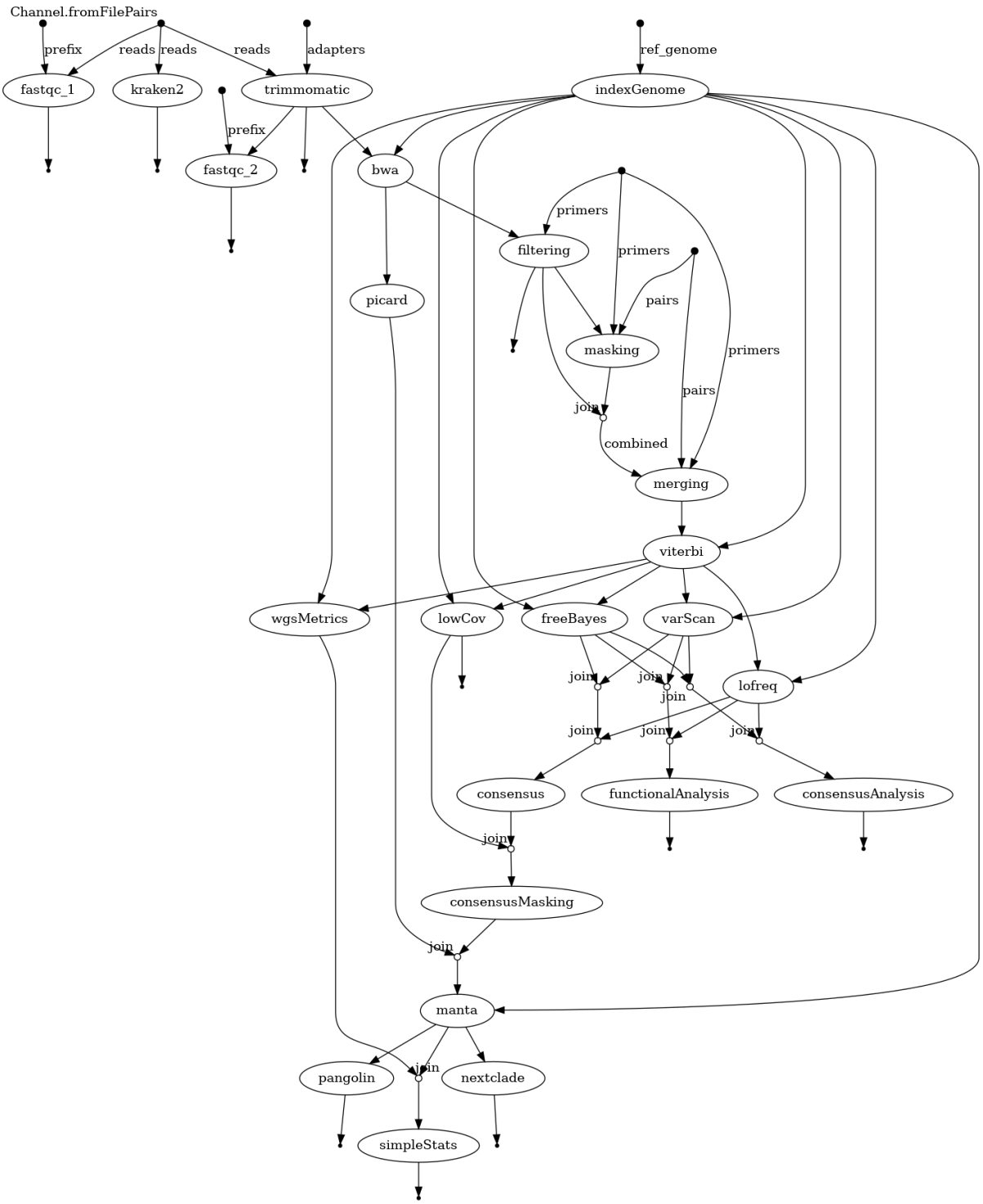
This should fill directories in `data/pangolin` and `data/nextclade`. For more details read the chapter External databases updates.

4. Copy `run_nf_pipeline.sh.template` to `run_nf_pipeline.sh` and fill in the paths to the reads and output directory.

5. Run the pipeline:

```
./run_nf_pipeline.sh
```

Pipeline overview



overview of the pipeline

External databases updates

Some components of the pipeline require access to their two databases, which are updated roughly once every two weeks. Different software pieces need to be updated in different ways. To make this process as smooth and painless as possible, we prepared a dedicated Docker container exactly for this task, along with a bash script for running it with appropriate volume mounts. The script should be placed in either the cron or systemd timer and run on a weekly basis.

Updates procedure

Build the dedicated container:

```
docker build --target updater -f Dockerfile-main -t nf_illumina_sars-3.0-updater:latest .
```

Run the updater script. The working dir must be in project root directory.

```
update_external_databases.sh
```

Total size of download is ~2-3 MiB. If everything work fine in directories `data\pangolin` and `data\nextclade` you should see downloaded content like below:

```
data/nextclade/
└── sars-cov-2.zip

data/pangolin/
├── bin
├── pangolin_data
└── pangolin_data-1.25.1.dist-info
```

Updates internals

The following section contain information what and how is updated. Unless you need to debug or refactor the code, and you followed guides in chapter "Updates procedure" in "External databases updates" you can safely skip it.

List of components that require updates

- Nextclade
- Pangolin

Nextclade

Nextclade (https://docs.nextstrain.org/projects/nextclade/) is software for assigning evolutionary lineage to SARS-Cov2. To make it work properly, it requires a database which is updated roughly once every two weeks.

The recommended way of downloading dataset is using `nxtclade` tool.

```
nextclade dataset get --name sars-cov-2 --output-zip sars-cov-2.zip
```

Detailed manual is available https://docs.nextstrain.org/projects/nextclade/en/stable/user/datasets.html. Nextclade is downloading `index.json` from site: https://data.clades.nextstrain.org/v3/index.json, and based on that files it decide what to download and from where. Probably the same data are available directly on GitHub: https://github.com/nextstrain/nextclade_data/tree/master/data/nextstrain/sars-cov-2/wuhan-hu-1/orfs.

The command above will download a `sars-cov-2.zip` file in desired destination (default: `data/nextclade`). That directory have to be mounted inside `main` container. It is done by Nextflow in the `modules/nextclade.nf` module.

```
process nextclade {
    (...)
    containerOptions "--volume
${params.nextclade_db_absolute_path_on_host}:/home/SARS-
CoV2/nextclade_db"
    (...)
```

Pangolin

Pangolin (https://github.com/cov-lineages/pangolin) (**P**hylogenetic **A**ssignment of **N**amed **G**lobal **O**utbreak **LIN**eages) is alternative to Nextclade software for assigning evolutionary lineage to SARS-Cov2.

To make it work properly, it requires a database that is stored in the Git repository pangolin-data (https://github.com/cov-lineages/pangolin-data).

Pangolin-data is actually a regular python package. Normal update procedure is via command: `pangolin --update-data`. It also can be installed by `pip` command. Keeping it inside main container is slightly tricky. We don't want to rebuild entire container just to update the database. We also don't want to keep the database inside the container, because it would force us to run the update before every pipeline run, which is stupid. The best solution is to mount the database from the host.

To achieve this goal we install the package externally to the container in designated path using host native `pip`.

```
pip install \
    --target data/pangolin \
    --upgrade \
    git+https://github.com/cov-lineages/pangolin-data.git@v1.25.1
```

> Make sure you entered proper version in the end of git url. The version number is also git tag. List of available tags with their release dates is here (https://github.com/cov-lineages/pangolin-data/tags).

Then that dir is mounted as docker volume inside the container (which is done automagically in the Nextflow module file):

```
process variantIdentification {
    containerOptions "--volume
${params.pangolin_db_absolute_path_on_host}:/home/SARS-CoV2/pangolin"
    (...)
```

During container build the `$PYTHONPATH` environment variable is set to indicate proper dir.

```
(...)
ENV PYTHONPATH="/home/SARS-CoV2/pangolin"
(...)
```

So the manual download consist of two steps:

1. Install the desired version of `pangolin-data` package in `data/pangolin` directory.

2. Provide absolute path to that dir during starting pipeline

```
--pangolin_db_absolute_path_on_host /absolute/path/to/data/pangolin
```