

NextFlow pipeline for SARS-CoV-2 Illumina data

Table of contents

Quickstart	2
Pipeline overview	3
Updates	4

Quickstart

Installation and usage

1. Install NextFlow

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

2. Build two containers:

```
docker build --target production -f Dockerfile-main -t
nf_illumina_sars-3.0-main .
docker build --target prodcutioin -f Dockerfile-manta -t
nf_illumina_sars-3.0-manta .
```

3. Install pangolin-data package locally in data/pangolin directory:

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

Substitute <VERSION> with desired tag name from git repository. List of tags is available here (<https://github.com/cov-lineages/pangolin-data/tags>).

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

Updates

Two components of pipeline require access to their databases which are updated roughly 2 once two weeks.

tl;dr

Some updates require additional software which is installed in `updater` container. Build updater 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:

```
nextclade/
└─ sars-cov-2.zip

pangolin/
├─ bin
│   └─ pangolin_data
├─ pangolin_data
│   └─ data
│       ├── alias_key.json
│       ├── lineages.hash.csv
│       └─ lineageTree.pb
├─ __init__.py
├─ __pycache__
│   └─ __init__.cpython-311.pyc
└─ pangolin_data-1.25.1.dist-info
    ├── direct_url.json
    ├── entry_points.txt
    └─ INSTALLER
```

```
|— LICENSE
|— METADATA
|— RECORD
|— REQUESTED
|— top_level.txt
|— WHEEL
```

Nextclade

Nextclade (<https://docs.nextstrain.org/projects/nextclade/en/stable/user/datasets.html>) 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 `nextclade` tool.

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

Pangolin

Pangolin (<https://github.com/cov-lineages/pangolin>) (Phylogenetic Assignment of Named Global Outbreak LINEages) 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 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 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
```

i 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 automatically 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 user have to do two things:

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
```