Covid sequencing and clade assignment workflow at NCCT

Author: A. Angelov, J. Geißert, NCCT
Last edit: 2021-06-21

Note:
This workflow is executed on the workstation server (10.24.11.83), use your login credentials to log in via RDP-Client. No installations are required, everything runs in docker.
The workflow is adjusted for data generated by the ARTIC for Illumina sequencing pipeline.
Map workstation project folder under windows - \\10.24.11.83\NCCTprojects

- **Copy Illumina run folder to workstation before running**

runs are located under:
`/home/sysgen/illuminadata/nextseq500` (or respective run folders of e.g. MiSeq)

General locations of raw Illumina data - /home/sysgen/novaseqdata (IMGAG machines) and /home/sysgen/illuminadata (own machines)

currently they are copied to a temporary run folder in the ncct-projects directory:
`/home/sysgen/Desktop/ncct-projects/20-bcls`

example for copy command:
```
cp -rf 210618_NB501351_0143_AHWM5LAFX2
/home/sysgen/Desktop/ncct-projects/20-bcls
```

1. Processing of raw sequencing data (Illumina bcl folder)

- you can find the run info for each Covid run together with a mapping file of samples and index distribution under: https://airtable.com/shrsepm2uhXt0l3DK

- Create a project folder for the current run e.g. “2106-cov-im-exp25” within the “ncct-projects” folder

For fastq files generation and demultiplexing, the nxf-bcl pipeline is used.

Prerequisites:
- docker, nextflow
- in case docker is not available, the conda environment can be recreated using this env file:

https://github.com/angelovangel/nxf-bcl/blob/master/environment.yml

Input:

- A **sample sheet**, which can be prepared with the Illumina samplesheet generator app (does not work in the Klinikum network, from within the Klinikum use this link). Herefore the mapping file deposited in Airtable is very helpful.

  → Paste sample-well data

  → Select index kit and machine: select IDT for Illumina DNA/RNA UD Indexes Tagmentation ver2 for illumina covidseq!

  → Add run details

- Download the sample sheet into the project folder

Output:

- results-bcl folder with fastq files (results-bcl/fastq/NA)
- multiqc report (results/bcl)

For example, if the samplesheet is in the current project working directory, the nxf-bcl can be run with the following command:

```
nextflow run ncct-mibi/nxf-bcl --runfolder /path/to/runfolder --samplesheet 20210118_samplesheet.csv
```

Check the output per sample and correct index mapping in the multiqc-report.

2. COVID-19 genome mapping and SNP calling

Here two pipelines can be used:
A. nf-core/viral recon pipeline
B. RKI ncov_minipipe

Prepare reference sequences (fasta, gff, bed) and put them in a “refseq” folder. The refseq folder from the last Covid run can be copied into the new/current Covid run folder, as refseq does not change)
Currently we are always running both pipelines.

A - nf-core/viralrecon
settings are updated to meet RKI QC

Prerequisites:

- docker, nextflow, internet connection to download the nf-core pipeline
- reference files (fasta, gff, bed) in the **refseq folder**, can be obtained from previous runs
- fastq sequence files, generated in the previous step

- **viralrecon-samplesheet** used as input, see example. The viralrecon samplesheet can be generated e.g. with Excel and the “Verketten” function. To make it easier you can copy the Excel from a previous run into the project folder and enter the new sample information. Then export to .csv and open with Notepad ++. Here the EOL Conversion has to be set from Windows (CR LF) to Unix (LF) under “Edit” and semicolons have to be replaced by commas. Safe the samplesheet in the Covid project folder.

Command:

```
nextflow run nf-core/viralrecon -r 1.1.0\
--input
/home/sysgen/Desktop/ncct-projects/2009-cov-exp4/viralrecon_covid_exp4_sample
sheet.csv \
--fasta
/home/sysgen/Desktop/ncct-projects/2009-cov-exp4/refseq/MN908947.3.fasta \
--gff /home/sysgen/Desktop/ncct-projects/2009-cov-exp4/refseq/MN908947.3.gff
\
--amplicon_bed
/home/sysgen/Desktop/ncct-projects/2009-cov-exp4/refseq/nCoV-2019-artic-v3.be
d \
--protocol amplicon \
--min_coverage 20 \
--skip_assembly \
--outdir /home/sysgen/Desktop/ncct-projects/2009-cov-exp4/results-viralrecon
\
-profile docker

# note on --max_allele_freq parameter
# in the varscan2 process, the filtering is done as:
bcftools filter \\
-i 'FORMAT/AD / (FORMAT/AD + FORMAT/RD) >= $params.max_allele_freq'

# where AD = unfiltered allele depth, i.e. the number of reads that support
each of the reported alleles
# and RD = Depth of reference-supporting bases
```

We use varscan2 as variant caller, ivar and bcftools variants are also available in the variants folder.
Note: after running the 2 steps above, it is good to delete the temporary folder:

```
rm -rf work

# sometimes, prefix with sudo
```

comment:
Sometimes one or more of the three callers (ivar, bcftools, varscan2) fail. Then you can re-run the pipeline with excluding the failing caller(s)
If e.g. bcftools fails run the command above and include:

```
--callers ivar, varscan2
```

- under `results-viralrecon/variants/summary_variants_metrics_mqc.tsv` there are general run statistics which are important to monitor. Import them into an Excel and save it into the Covid project folder.

B - RKI's ncov_minipipe

Prerequisites:
- conda
- the RKI CoVpipe
- reference genome fasta file (in folder refseq)

Before executing, cd into the project folder and **switch to bash**, after executing run **conda deactivate**.
The kraken2 database is specifically built for this pipeline and contains only the human and COVID-19 genomes. Check the db/kraken/ncov_kraken_db folder for the correct name of the database (parameter is the folder containing the database files).
If you want variant annotation then make sure that the ref genome is supported by SNPeff (supported genomes are listed in the ncov_minipipe appendix).

```
conda activate covpipe_env

ncov_minipipe \
--reference refseq/reference.fasta \
--input results-bcl/fastq/NA \
--no-var-annotation \
--cpus 96 \
-k ~/db/kraken/ncov_kraken_db \
-o results-minipipe

# after executing:
conda deactivate
```

3. Clade assignment

This step is performed using the pangolin command line tool and with the Nextclade online tool.
In both cases, a multifasta file is needed containing all the sequences to be analysed.
https://cov-lineages.org/

A Pangolin:

Input:
A multifasta file with all genomes to be analysed (masked fasta, positions not covered are N)
For example, to generate a multifasta from files which are in a variants/bcftools/consensus directory:

```
# check that wildcard ok
ll variants/bcftools/consensus/*.consensus.masked.fa
```

```
# cat into one file
cat variants/bcftools/consensus/*.consensus.masked.fa > project-masked.fa
```

Output:
By default, the output file of the pangolin command line tool is called **lineage_report.csv.** Do not rename the file as it will be needed as input for the Covid report generator (description under point 5. You can import the data to Excel and name it e.g. pangolin-lineages.xlsx.

Update pangolin before each run

pangolin is installed with conda on the workstation, and the git repo is located in **/home/sysgen/git/pangolin**, so to update you first have to cd in this directory
pangolin is worked on intensively, so it is good to update before running (especially as new clades arise):

```
cd ~/git/pangolin
conda activate pangolin

# pulls the latest changes from github
git pull

# re-installs pangolin
python setup.py install

# updates the conda environment
conda env update -f environment.yml

# updates if there is a new data release
pip install git+https://github.com/cov-lineages/pangoLEARN.git --upgrade
```

Run pangolin

```
# activate the pangolin conda env
conda activate pangolin

# run pangolin
pangolin query-multi.fasta

# deactivate
conda deactivate
```

B [Nextclade online tool:]

Here you can upload the multifasta file generated above and get the Nextclade assignments. Download the clade assignments as **nextclade.tsv** and then import it to Excel and name it e.g. nextclade-assignments.xlsx.

4. Check if reconstructed genomes meet RKI criteria with president

```
conda activate president
```

```
president -r refseq/NC_045512.2.fasta -q multifasta.fa -p president-viralrecon


conda deactivate
```

Usually we check with president the viralrecon-multifasta as well as the multifasta generated from the RKI-minipipe output and compare.

5. Generate NCCT report

The outputs of nf-core/viralrecon and pangolin/nextclade assignments can be used as input to generate a R Markdown report in R Studio. The template for this report is located in `/home/sysgen/Desktop/ncct-projects/21-covid-report-generator`.

Prerequisites:

- local R and RStudio installation
- installed `renv` library - execute `install.packages("renv")` in the R console
- the contents of the `covid-report-generator` folder - a backup copy is available under `/home/sysgen/git/private`
- (optional) output folders of either `nf-core/viralrecon` or RKI's `ncov_minipipe`
- (optional) output of pangolin and/or nextclade
- (optional) output of RKI's `president` script

Important: new users have to execute `renv::restore()` in the R console.

Generate report by

- start Rstudio,
- open covid-report.Rmd (`/home/sysgen/Desktop/ncct-projects/21-covid-report-generator`.)
- click on "Knit with Parameters".
- fill-in the fields and press "Knit". The path to the analysis results folder (last point to be filled) has to be given in relation to the 21-covid-report-generator folder

  e.g. `../2106cov-im-exp25/results-viralrecon`

A file named covid-report.html will be generated in the same folder. Copy and rename into the current Covid project folder as needed.

6. RKI upload

General and most recently updated information about the upload of sequencing data to the RKI can be found here:
https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/DESH/DESH.html

- The output from president is a (masked) FASTA file which contains all sequences which meet the RKI criteria “valid.fasta”
- check which of the valid files can be uploaded in our “COVIDSEQ-lab-allsamples” Excel-sheet.
- delete fasta files from the multifasta which should not be uploaded
- prepare a samplesheet csv file as described here Anleitung Bereitstellung Sequenzierdaten (Version 2021-02-20 - Check for updates regularly!) Make sure that the OWN_FASTA_ID is identical with the headers in the multifasta file. The DATE_DRAW can be found in the “COVIDSEQ-lab-allsamples” Excel-sheet.

SENDING_LAB	DATE_DRAW	SEQ_TYPE	SEQ_REASON	SAMPLE_TYPE	PUBLICATION_STATUS	OWN_FASTA_ID
10276	20210301	ILLUMINA	N	s002	P	2103-cov-IM-r1-378
10276	20210302	ILLUMINA	N	s002	P	2103-cov-IM-r1-379
10276	20210302	ILLUMINA	N	s002	P	2103-cov-IM-r1-381
10276	20210303	ILLUMINA	N	s002	P	2103-cov-IM-r1-382
10276	20210302	ILLUMINA	N	s002	P	2103-cov-IM-r1-385

When the multifasta and the csv file are ready login under: https://desh.bdr.de:

login: norbert.brenner@med.uni-tuebingen.de
Pw.: desH#2021

- After successful upload a csv file will be generated where one demis transaction id per individual sample is listed “YYYYMMDDNachweisBereitstellungSequenzierdaten.csv”. Safe this file, it is essential for the identification of uploaded samples.

7. Share results with the diagnostics

- Safe NachweisBereitstellungSequenzierdaten, the multifasta and the samplesheet under Q/MK/Seq_SARS_CoV/RKI_uploads/folder of individual run
- Generate a run folder under Q/MK/Seq_SARS_CoV/Results_Pangolin_Nextclade and safe there for all samples from this run (independent from meeting RKI criteria):
    - ncct customer mapping file
    - nextclade assignments
    - pangolin lineages
    - covid report.html
    - masked and unmasked multifasta of all samples
    - summary_variants_metrics_mqc.xlsx

8. Tipps: Working with fasta files

- Make a multifasta file from individual fasta files

```
cat path/to/files/*.fa > multifasta.fa
```

- Select a subset of a multifasta file, prepare a text file with the headers needed, one line per entry.

```
seqkit grep > -n -f headers.txt multifasta.fa > selected.fa
```

- Get N-content (in %) from a multifasta file

```
seqkit fx2tab -B N -n -l multifasta.fa

# to directly get N counts in tab-separated form
seqkit fx2tab -B N -n -l multifasta.fa | awk '{printf ("%s \t %.0f\n", $1,
$3/100*$2)}'
```