**Running NextFlow Pipelines: ONT-Seq-Analysis & Mpox\_AmpSeq**

This guide provides instructions for running the CDCgov/ONT-Seq-analysis and CDCgov/Mpox\_AmpSeq NextFlow pipelines on a computing cluster with pre-configured NextFlow modules.

ONT-Seq-analysis is for 10kb and 15kb amplicon data.

Mpox\_AmpSeq is for F13L amplicon data.

**For information about the pipelines, please see:**

[CDCgov/ONT-Seq-analysis: ONT-Seq-analysis is a specialized pipeline designed for analyzing Oxford Nanopore Technologies (ONT) sequencing data from Mpox isolates. It performs reference-based assemblies, identifies variants, designates clades, and generates a detailed quality control (QC) report.](https://github.com/CDCgov/ONT-Seq-analysis)

[CDCgov/Mpox\_AmpSeq: Mpox\_AmpSeq is a custom nextflow-style pipeline for mpox F13L amplicon sequencing. It generates reference-based consensus sequences, variant reports, NextClade outputs including clade designation, and multiple quality control metrics.](https://github.com/CDCgov/Mpox_AmpSeq)

**1. Pipeline Setup and Configuration**

1. **Clone Pipelines:** In your chosen location, clone the pipelines from GitHub:

Bash

git clone https://github.com/CDCgov/ONT-Seq-analysis.git

git clone https://github.com/CDCgov/Mpox\_AmpSeq.git

1. **Ensure Executability:** Navigate into each cloned pipeline directory and ensure that the pipeline scripts and any additional Python scripts are executable. This is crucial for NextFlow to run them.
   * **For the main pipeline directory:**

Bash

cd /full/path/to/ONT-Seq-analysis/ # or /full/path/to/Mpox\_AmpSeq/

chmod +x \*.nf # Makes NextFlow scripts executable

chmod +x ./bin/check\_samplesheet.pys

* + **For additional scripts (e.g., in assets/ or bin/ directories):**

Bash

cd /full/path/to/ONT-Seq-analysis/assets/ # or /full/path/to/Mpox\_AmpSeq/assets/

chmod +x \*.sh # Makes shell scripts executable

chmod +x \*.py # Makes Python scripts executable

* + **Important Note:** If you plan on using any additional post-processing scripts not explicitly mentioned, you will need to make them executable in the same way.

**2. Creating a samplesheet.csv**

Your sequencing data will be in one of two formats:

**Option A: Data Already in FastQ Format**

If your data is already in FastQ format (e.g., .fastq or .fastq.gz files), you can use the provided scripts to create your samplesheet.csv.

1. **Use the provided script:**
   * **For ONT-Seq-analysis:**

Bash

/full/path/to/ONT-Seq-analysis/assets/ont\_fastq\_concat\_and\_samplesheet\_create.sh

* + **For Mpox\_AmpSeq:**

Bash

/full/path/to/Mpox\_AmpSeq/assets/ont\_fastq\_concat\_and\_samplesheet\_create.sh

1. **Follow the script's prompts:**
   * Press enter and carefully read the on-screen message regarding file naming standards.
   * If your FastQ data contains characters other than those specified in the message, enter y. You will then need to **manually update the naming of your FastQ data** to meet the described standards.
   * If your FastQ data meets the specified standards (no other characters), enter n.
   * When prompted, **input the full file path** to your FastQ data directory.
     + **Example:** /scicomp/instruments-pure/23-4-631\_Nanopore-MinION-NP234947/2024/PRB-2024-MPXV-10Kb-15kb-Amplicons-JiushengDeng/PRB-MPXV-10K-15K-barcodes-cladeIIb-TPOXX-4tissues-3-4-2025/4tissues/20250304\_1249\_MC-113388\_AXD563\_57180a83/fastq\_pass/
   * Upon completion, a samplesheet.csv will be generated in your current working directory. You can now proceed to run your NextFlow pipeline.

**Option B: Data in Raw Fast5 Format**

If your data is in raw Fast5 format (e.g., .fast5 files), you'll need to convert it to FastQ using pod5 and dorado before creating your samplesheet.

1. **Convert Fast5 to POD5:**

Bash

pod5 convert fast5 /full/path/to/your/raw/fast5/data/ -o /full/path/to/merged.out.pod5

* + **Example Fast5 path:** /scicomp/instruments-pure/23-4-631\_Nanopore-MinION-NP234947/2023/PRB02232023-EC-RABV-MPOX/seq/20230223\_1655\_MN18626\_ANV188\_26d73d90/fast5/

1. **Basecall and Demultiplex with Dorado:** This step can be time-consuming, so run it through the shared bash script

Bash

qsub /full/path/to/daisy\_dorado\_pod5\_new\_minqscore\_automodel.sh -f /full/path/to/merged.out.pod5

* + **Note:** The output will be in a directory named dorado (or similar, specified by the daisy\_dorado\_pod5\_new\_minqscore\_automodel.sh script), containing subfolders per barcode with compressed FastQ reads.

1. **Create a Samplesheet from Dorado Output:**
   * **For ONT-Seq-analysis:**

Bash

/full/path/to/ONT-Seq-analysis/assets/create\_samplesheet\_only.sh

* + **For Mpox\_AmpSeq:**

Bash

/full/path/to/Mpox\_AmpSeq/samplesheet.sh

* + When prompted, **enter the full path** to the data output directory containing your FastQ files (e.g., /full/path/to/dorado/).
  + Upon completion, a samplesheet.csv will be generated in your current working directory. You can now proceed to run your NextFlow pipeline.

**3. Running Your NextFlow Pipeline**

**Before running any NextFlow pipeline, you *must* load the NextFlow module on your computing cluster.**

Bash

module load nextflow

**Critical Note on File Paths:** The biggest issues arise from incorrect file paths. To avoid errors, always use **full, absolute file paths** for all inputs (samplesheet, reference files, output directories), not paths relative to your current working directory.

**A. For ONT-Seq-analysis (10kb and 15kb amplicon data)**

* **Reference Information:** All necessary reference files are available in: /full/path/to/ONT-Seq-analysis/assets/references/NC063383/
* **General Usage Example:**

Bash

nextflow run ~/path/to/ONT-Seq-analysis \

-profile <docker/singularity/…/institute> \

--input ~/path/tosampleshee.csv \

--outdir <OUTDIR> \

-resume <#if applicable> \

--fasta <fasta\_path\*> \

--bed\_file <bed\_path\*> \

--fai\_file <fai\_path\*> \

--gff\_file <gff\_path\*> \

--mmi\_file <mmi\_path\*> \

--nextclade\_dataset\_name ‘nextstrain/mpox/all-clades’

* + **Explanation of parameters:**
    - --input: Full path to your generated samplesheet.csv.
    - --outdir: Full path to the directory where all pipeline results will be stored.
    - -resume: use to pick up wherever the pipeline left off without having to rerun all analyses
    - --fasta to --mmi\_file: Reference information for various pipeline steps.
    - -profile: Specifies the execution profile. Use docker if Docker is available and configured, or singularity if using Singularity. For CDC cluster use singularity.
* **Enabling Ivar\_Variants Module:** The Ivar\_Variants module is turned off by default. To include it in your run, add --run\_ivar\_variants true to the nextflow run command:

Bash

nextflow run ~/path/to/ONT-Seq-analysis \

-profile <docker/singularity/…/institute> \

--input ~/path/tosampleshee.csv \

--outdir <OUTDIR> \

-resume <#if applicable> \

--fasta <fasta\_path\*> \

--bed\_file <bed\_path\*> \

--fai\_file <fai\_path\*> \

--gff\_file <gff\_path\*> \

--mmi\_file <mmi\_path\*> \

--nextclade\_dataset\_name ‘nextstrain/mpox/all-clades’

--run\_ivar\_variants true \

* **Filtering NextClade Output:** To remove unnecessary columns from NextClade output, run the filtering script **inside the NextClade output directory**:

Bash

cd /full/path/to/your/output\_directory/nextclade/

/full/path/to/ONT-Seq-analysis/assets/nextclade\_tsv\_column\_filter.sh

The resulting TSV file will contain: index, seqName, clade, lineage, outbreak, qc.overallScore, qc.overallStatus, totalSubstitutions, totalDeletions, totalInsertions, totalFrameShifts, totalMissing, totalNonACGTNs, failedCdses, warnings, errors. Feel free to modify the script as needed.

* **Extracting Mutations with Python Script:** To extract mutations from each NextClade output file and highlight key mutations (Clade I or Clade II lineages), run the Python script **inside the NextClade output directory**:

Bash

cd /full/path/to/your/output\_directory/nextclade/

/full/path/to/ONT-Seq-analysis/assets/pythonX\_nextclade\_parser.py # Replace pythonX with actual python version like python3

This will generate a directory /nextclade/cleaned/ containing a TSV file for each specimen with mutation details.

**B. For Mpox\_AmpSeq (F13L amplicon data)**

* **Reference Information:** All necessary reference files are available in: /full/path/to/Mpox\_AmpSeq/assets/genome/
* **General Usage Example:**

Bash

nextflow run /path/to/Mpox\_AmpSeq \

-profile <docker/singularity/…/institute> \

--input sampleshee.csv \

--outdir <OUTDIR> \

-resume <#if applicable> \

--fasta <fasta\_path\*> \

--bed\_file <bed\_path\*> \

--fai\_file <fai\_path\*> \

--gff\_file <gff\_path\*> \

--mmi\_file <mmi\_path\*> \

--nextclade\_dataset\_name ‘nextstrain/mpox/all-clades’

* + **Explanation of parameters:**
    - --input: Full path to your generated samplesheet.csv.
    - --outdir: Full path to the directory where all pipeline results will be stored.
    - -resume: use to pick up wherever the pipeline left off without having to rerun all analyses
    - --fasta to --mmi\_file: Reference information for various pipeline steps.
    - -profile: Specifies the execution profile. Use docker if Docker is available and configured, or singularity if using Singularity. For CDC cluster use singularity.

**4. Post-processing Scripts**

**FOR ONT\_SeqAnalysis**

Two scripts are available in ONT\_SeqAnalysis/assests directory to streamline the nextclade output.

1. /assets/nextclade\_tsv\_column\_filter.sh
   * **Description:** This script generates a TSV file will contain the following colums: 'index', 'seqName', 'clade', 'lineage', 'outbreak', 'qc.overallScore', 'qc.overallStatus', 'totalSubstitutions', 'totalDeletions', 'totalInsertions', 'totalFrameShifts', 'totalMissing', 'totalNonACGTNs', 'failedCdses', 'warnings', 'errors'. Feel free to modify as needed.
   * **Usage:**

Bash

cd /full/path/to/your/nextflow\_output\_directory/

/full/path/to/ONT-Seq-analysis/assets/ nextclade\_tsv\_column\_filter.sh

1. /assets/pythonX\_nextclade\_parser.py
   * **A python script to extract mutations from each nextclade output file and for each specimen, returning a table highlighting key mutations that could indicate a designation of Clade I or Clade II, including the different lineages. Run the python script (assets/pythonX\_nextclade\_parser.py) inside the Nextclade output directory. It will generate a directory named parser that will contain a tsv file for each specimen with the following format:**

| **seq\_name** | **Mutations\_found\_on\_sequence** | **Found\_in\_Clade\_I\_or\_Clade\_II** | **Clade\_designation** |
| --- | --- | --- | --- |
| sample1 | G01234A | yes | Clade Ia |
| sample1 | G56789A | yes | Clade IIb |

* **Usage:**

Bash

cd /full/path/to/your/nextflow\_output\_directory/

/full/path/to/ONT-Seq-analysis/assets/pythonX\_nextclade\_parser.py #specify python2 or python3 based on your system, CDC use python3

**FOR Mpox\_AmpSeq**

Two additional scripts are available in the Mpox\_AmpSeq/assets/ directory to help summarize and analyze your results. **These should be run from within your NextFlow output directory**.

1. /assets/table\_summary.sh: MPOX\_AMPSEQ ONLY
   * **Description:** This script generates a comprehensive summary table of sequencing metrics and NextClade mutation information. It integrates outputs from SAMTOOLS and NextClade to provide read statistics (total reads, mapped reads, average coverage, average read length) and NextClade-specific details (nucleotide/amino acid substitutions, deletions, clade assignment, overall coverage). The final table is output in the ./summary\_stats/ directory within your main output folder.
   * **Usage:**

Bash

cd /full/path/to/your/nextflow\_output\_directory/

/full/path/to/Mpox\_AmpSeq/assets/table\_summary.sh

(Note: The script path should point to the correct pipeline based on which output you're summarizing.)

1. /assets/match\_mutations.sh: MPOX\_AMPSEQ ONLY
   * **Description:** This script performs a targeted analysis by cross-referencing amino acid substitutions identified by NextClade against a custom mutation database. It requires your mutation database to be provided as a tab-separated file named mutation\_database.tsv within the respective pipeline's /assets/ directory (e.g., /full/path/to/Mpox\_AmpSeq/assets/mutation\_database.tsv). For each sample, if a NextClade amino acid substitution matches an entry in your database, the script will report the sample name, the NextClade amino acid substitution, and all corresponding details from your database (Mutation, AminoAcid, TotalCount, APOBEC3\_Context). The report is saved to ./summary\_stats/matched\_mutations\_report.tsv.
   * **Usage:**

Bash

cd /full/path/to/your/nextflow\_output\_directory/

/full/path/to/Mpox\_AmpSeq/assets/match\_mutations.sh

Bash

These two scripts will generate new files in nextflow\_output\_directory/summary\_statistics/