User Manual for Nextflow Pipeline Tool

Introduction

Antimicrobial resistance (AMR) is one of the top global health threats, as stated by the World Health Organization (WHO). It leads to resistance against antibiotics, making infections harder to treat. *Pseudomonas aeruginosa* is a major concern in AMR due to its resistance, which makes treating infections such as bloodstream infections and respiratory system infections more challenging.

To address this issue, there is a need for a variant database that aids in clinical studies, and therapeutic research. This tool is designed to create a variant database using FASTA, FASTQ, and metadata CSV files. It generates VCF, CSV files as output .

Workflow

The tool follows these key processes:

- Indexing
- Trimming and quality check
- Alignment and sorting
- Duplicate marking
- Variant calling and annotation
- Variant filtering

Step-by-Step Guide

1. Install Docker.
2. Use Docker commands to build and run the image.
3. First, build the database for SnpEff.
4. Use `mkdir` to create a new folder, making it easier to run and access system data
5. Store all files in a single directory, based on the input parameters mentioned below.
6. Use the Nextflow command with the required parameters to process the data.

---

1. System Requirements:

Minimum Requirements:

- **CPU**: Dual-core processor
- **RAM**: 6 GB
- **Storage**: 5 GB free space
- **Python**: 3.8 or higher
- **Nextflow**: Latest version
- **Tools:** Latest version
- **Docker:** Latest version
- **OS**: Windows 10/11, Linux (Ubuntu 20.04+), macOS (Monterey or later)

Recommended Requirements:

- **CPU**: Quad-core processor
- **RAM**: 8GB or more
- **Storage**: 8 GB free space
- **Python**: 3.10 or higher
- **Nextflow**: Latest version

- **Tools:** Latest version
- **Docker**: Latest version
- **OS**: Windows 10/11, Linux (Ubuntu 22.04+), macOS (Ventura or later)

---

2. Installation:

A. Installing Nextflow Without Docker

Linux (Ubuntu/Debian-based)

- **Install dependencies:**
    - `sudo apt update && sudo apt install -y openjdk-21-jdk curl`
- **Install Nextflow:**
    - `curl -s https://get.nextflow.io | bash`
- **Move Nextflow to a directory in your system PATH:**
    - `sudo mv nextflow /usr/local/bin/`
- **Verify installation:**
    - `nextflow -v`

B. Installing Required Tools for the Pipeline Without Docker

1. Installing SnpEff

Build snpEff database

1. Download SnpEff

Go to the SnpEff download page.

Download the .zip file and extract it:

```
  wget https://snpeff.blob.core.windows.net/versions/snpEff_latest_core.zip
  unzip snpEff_latest_core.zip
  cd snpEff
```

Set Up Environment: Ensure SnpEff is executable:

```
chmod +x snpEff/snpEff.jar
snpeff  - version   //  This ensures that if BWA is already installed.
```

2. Obtain Genome and Annotation Files

- Download your bacterial genome in FASTA format (.fasta or .fna) from NCBI, Ensembl Bacteria, or another source.
- Obtain the corresponding GFF annotation file (.gff).

3. Use Prokka to Format Files for SnpEff

Format Files for SnpEff:

- Ensure the FASTA file contains a single genome sequence (or properly labeled contigs).
- Verify that the **GFF** file has accurate annotations, especially for CDS regions.

If annotations are missing or need to be generated, use **Prokka**:

Note :

- Replace my_bacteria with the actual bacterial name.
- The output files my_bacteria.fasta and my_bacteria.gff will be used in SnpEff.

```
prokka --outdir ./prokka_output --prefix my_bacteria my_bacteria.fasta
```

4. Configure the SnpEff Database

Locate the snpEff.config file in snpEff directory and open the SnpEff directory then snpEff.config file.

```
nano snpEff.config
```

Add a new entry at the end of the file **above the Non-standard Databases section:**

Note:

- Replace my_bacteria with a unique identifier for your genome.
- Replace MyBacteria with a descriptive name.

```
***
#---
# Non-standard Databases
#---
my_bacteria.genome : MyBacteria
***
```

5. Set Up the Genome Directory

Inside the data folder in the SnpEff directory, create a subdirectory for your genome:

```
mkdir snpEff/data/my_bacteria
```

6. Copy the Genome and Annotation Files

Copy the genome FASTA and annotation GFF files into the directory:

```
cp my_bacteria.fasta snpEff/data/my_bacteria/sequences.fna
cp my_bacteria.gff snpEff/data/my_bacteria/genes.gff
```

Use the **Prokka output files**

- my_bacteria.fasta → sequences.fna
- my_bacteria.gff → genes.gff

7. Build the SnpEff Database

Run the following command to build the database:

```
java -Xmx4g -jar snpEff/snpEff.jar build -gff3 -v my_bacteria
```

8. Handling Errors

If you encounter an error like this:

```
>>>WARNING_FILE_NOT_FOUND: Rare Amino Acid analysis: Cannot read
protein sequence file ' path/to/snpEff/data/au/protein.fa ', nothing done.
ERROR: CDS check file ' path/to/snpEff/data/au/cds.fa ' not found.
ERROR: Protein check file ' path/to/snpEff/data/au/protein.fa ' not found.
ERROR: Database check failed.
00:00:01 Logging>>>
```

Note : You need to generate **cds.fa** and **protein.fa** files.

9. Generate Missing Files

Extract Coding Sequences (CDS)

Use the GFF3 and FASTA files to generate cds.fa.

Note : Take gff file from prokka.

```
gffread genes.gff -g sequences.fna -x cds.fa
```

This extracts CDS sequences and saves them in cds.fa.

Translate CDS to Proteins

Use transeq from the EMBOSS suite to generate protein.fa:

```
transeq -sequence cds.fa -outseq protein.fa
```

This generates protein.fa.

Place the Files in the Correct Directory

Move the generated files to the SnpEff directory:

```
mv cds.fa protein.fa path/to/snpEff/data/au/
```

10. Rebuild the Database

```
java -Xmx4g -jar snpEff/snpEff.jar build -gff3 -v au
```

2. Installing BWA

```
sudo apt update && sudo apt install -y bwa
```

`Bwa - - help` // This ensures that if BWA is already installed.

3. Installing Samtools

```
sudo apt update && sudo apt install -y samtools
```

`Samtools - - help` // This ensures that if **Samtools** is already installed.

4. Installing VCFtools

```
sudo apt update && sudo apt install -y vcftools
```

`Vcftools - - help` // This ensures that if **VCFtools** is already installed.

5. Installing BCFtools

```
sudo apt update && sudo apt install -y bcftools
```

`Bcftools - - help` // This ensures that if **BCFtools** is already installed.

6. Installing Fastp

```
conda install -c bioconda fastp
```

`Fastp - - help`   //   This ensures that if **Fastp** is already installed.

7. Installing Sed, Awk, and Cut

```
sudo apt update && sudo apt install -y sed awk coreutils
```

`Coreutils - - help`   //   This ensures that if **coreutils** is already installed.

8. Installing Python3

```
sudo apt update && sudo apt install -y python3 python3-pip
```

`Python3 - - version`   //   This ensures that if **Python3** is already installed.

9. Installing SQLite

```
sudo apt update && sudo apt install -y sqlite3 || echo "SQLite3 is already installed or an error
occurred."
```

`Sqlite3  - - version`   //   This ensures that if **SQLite3** is already installed.

---

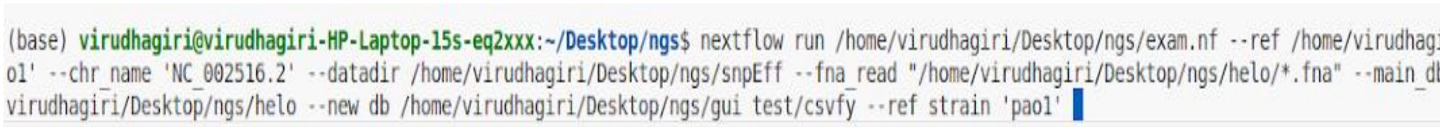
3. Installing and Running a Nextflow and Requirement Tools with Docker For Linux:

A. Running the Pipeline Manually

Without Docker in Linux:

- `nextflow run main.nf {parameters} .`

Example Nextflow command:



```
(base) virudhagiri@virudhagiri-HP-Laptop-15s-eq2xxx:~/Desktop/ngs$ nextflow run /home/virudhagiri/Desktop/ngs/exam.nf --ref /home/virudhag
o1' --chr_name 'NC_002516.2' --datadir /home/virudhagiri/Desktop/ngs/snpEff --fna_read "/home/virudhagiri/Desktop/ngs/helo/*.fna" --main_d
virudhagiri/Desktop/ngs/helo --new_db /home/virudhagiri/Desktop/ngs/gui_test/csvfy --ref_strain 'pao1'
```

Or

- Running the pipeline using GUI gives name,path,files and run nextflow.(if python 3 available).

With Docker in Linux:

- Update the apt package index:
  - sudo apt-get update
- Install dependencies required for Docker:
  - sudo apt-get install \
  - ca-certificates \
  - curl \
  - gnupg \
  - lsb-release \
  - sudo
- Add Docker's official GPG key:
  - curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
- Set up the stable repository for Docker:
  - echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
- Update the apt package index again:
  - sudo apt-get update
- Install Docker Engine:
  - sudo apt-get install docker-ce docker-ce-cli containerd.io
- Start and enable Docker:
  - sudo systemctl start docker
  - sudo systemctl enable docker
- Verify that Docker is installed and running:
  - sudo docker --version

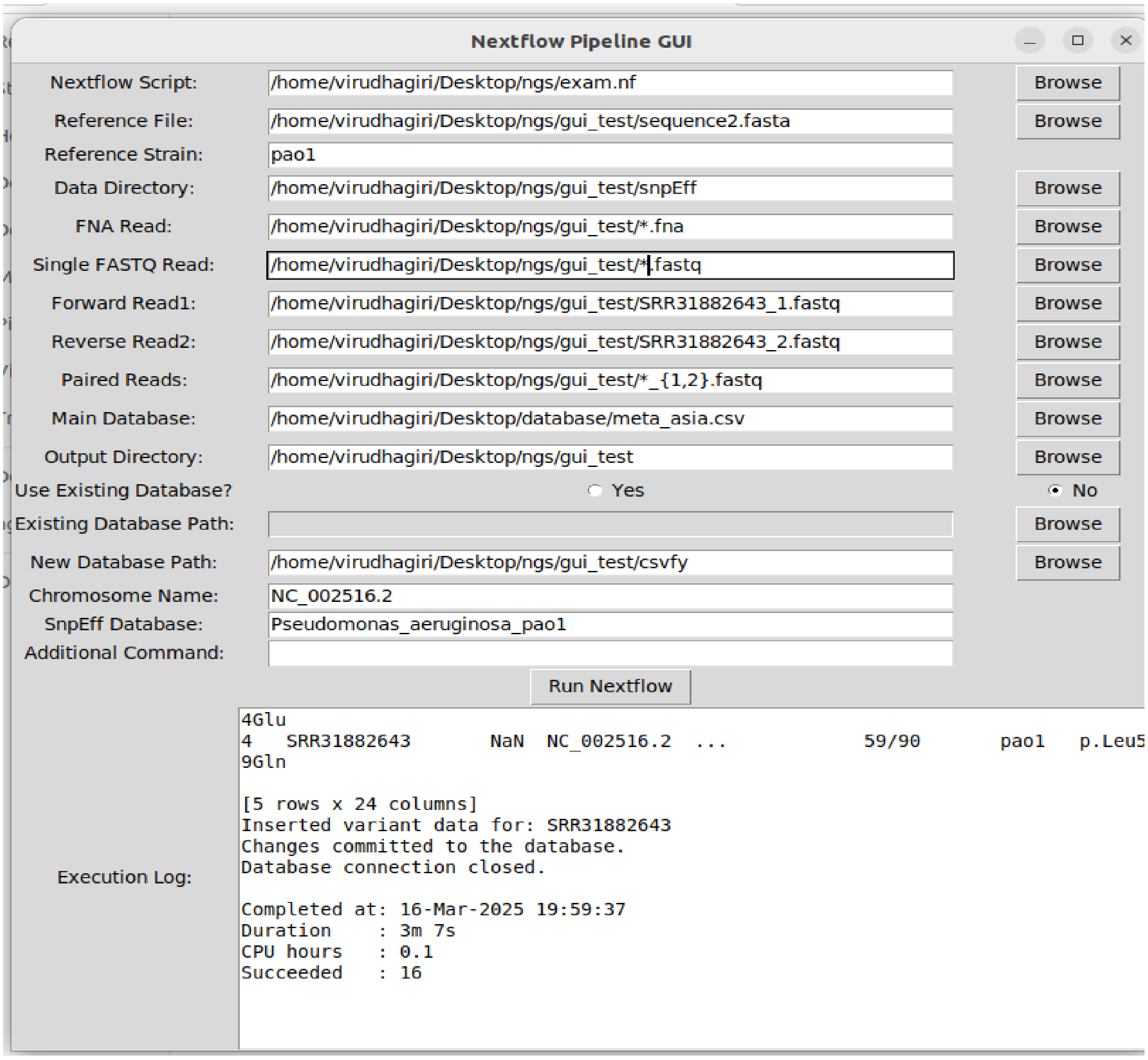
- Use mkdir to create a new folder, then download the Dockerfile and paste it into that folder. Open a terminal in the folder and build the Docker image:
    - docker build -t my_app **.** (Use **sudo** if need)
- Run Docker:
    - docker run -it -v /:/app/ my_app **.**
- Inside docker change directory using **cd ..** then enter the **"app"** directory.
- **@root: /app/** nextflow run main.nf {parameters} .

Or

- The Docker folder must contain the gui.py file, and it should be named exactly as gui.py.
- Navigate to the Docker folder - cd /path/to/docker
- Enable X11 access: xhost +
- Run the Docker container:
- sudo docker run -e DISPLAY=$DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix -v /home/admin:/home/admin -v /home/admin/Desktop/docker:/app my_app

  Note :
    - -e DISPLAY=$DISPLAY → Passes the display environment variable for GUI .
    - -v /tmp/.X11-unix:/tmp/.X11-unix → Shares the X11 socket for graphical output.
    - -v /home/admin:/home/admin → Mounts your home directory inside the container.
    - -v /home/admin/Desktop/docker:/app → Mounts your project directory inside the container.
    - my_app → The name of your Docker image.

Example GUI:



Nextflow Script:	/home/virudhagiri/Desktop/ngs/exam.nf
Reference File:	/home/virudhagiri/Desktop/ngs/gui_test/sequence2.fasta
Reference Strain:	pao1
Data Directory:	/home/virudhagiri/Desktop/ngs/gui_test/snpEff
FNA Read:	/home/virudhagiri/Desktop/ngs/gui_test/*.fna
Single FASTQ Read:	/home/virudhagiri/Desktop/ngs/gui_test/*.fastq
Forward Read1:	/home/virudhagiri/Desktop/ngs/gui_test/SRR31882643_1.fastq
Reverse Read2:	/home/virudhagiri/Desktop/ngs/gui_test/SRR31882643_2.fastq
Paired Reads:	/home/virudhagiri/Desktop/ngs/gui_test/*_{1,2}.fastq
Main Database:	/home/virudhagiri/Desktop/database/meta_asia.csv
Output Directory:	/home/virudhagiri/Desktop/ngs/gui_test
Use Existing Database?	Yes [✓ No]
Existing Database Path:	
New Database Path:	/home/virudhagiri/Desktop/ngs/gui_test/csvfy
Chromosome Name:	NC_002516.2
SnpEff Database:	Pseudomonas_aeruginosa_pao1
Additional Command:	

Execution Log:

```
4Glu
4   SRR31882643       NaN  NC_002516.2  ...          59/90       pao1    p.Leu5
9Gln

[5 rows x 24 columns]
Inserted variant data for: SRR31882643
Changes committed to the database.
Database connection closed.

Completed at: 16-Mar-2025 19:59:37
Duration    : 3m 7s
CPU hours   : 0.1
Succeeded   : 16
```

B. Input Parameters

: nextfow run path/to/main.nf

- -ref - Provide the reference genome file for a species (**.fasta** or **.fna**).(eg., path/to/sequence.fna or fasta)

- -datadir - Specify the path to the **SnpEff** directory. (eg., path/to/snpEff)

- -fna_read - Provide assembled genome files (**.fna**) or the path containing **.fna** files.

eg., Give "path/to/*.fna"

- -read - Provide raw sequencing reads (**.fastq**) or the path containing **.fastq** files.(Single end fastq)

eg., Give " path/to/*.fastq "

- -read1 - Provide the **forward raw read** file (**.fastq**) with _1 as a delimiter
(e.g., path/to/SRR12333_1.fastq).

- -read2 - Provide the **reverse raw read** file (**.fastq**) with _2 as a delimiter
(e.g., path/to/SRR12333_2.fastq).

- -reads - Provide the path containing multiple **.fastq** files.(Paired end fastq)

eg., Give "path/to/*_{1,2}.fastq "

- -main_db - Provide a **CSV metadata file** containing details of assembled genome **.fna** files or raw reads.(eg., path/to/meta.csv)

- - ref_strain - Provide the name of the reference strain as it appears in the FASTA file.(eg., 'pao1')

-output_dir - Specify the path where output CSV and VCF files will be stored. **This should be a new directory with nothing inside it.** (e.g., path/to/any directory)

- -chr_name - Provide the name of the chromosome as it appears in the FASTA file.(eg., 'NC_257237')

- -snpEff_db - Provide the name of the **SnpEff** database directory.(eg., 'psedo_au')

- -exist_db - Provide the path and filename of an existing database file (if available).
(eg., path/to/finaldata.db)

-new_db - If no existing database is available, provide this parameter along with the path where the new database file should be stored. **Importantly, the specified directory should not contain any existing database file.** (e.g., path/to/any directory)

C.Input Type

Ref: A reference FASTA or FNA file.

Fna_read:

- Can be a single file (e.g., path/to/287.1738.fna).
- Can be multiple files stored in a directory (path/to/*.fna).
- If using fna_read, the file extension must be .fna; otherwise, do not use this parameter.

Read:

- Can be a single file (e.g., path/to/SRR89484.fastq or path/to/ERR83938.fastq).
- Can be multiple files stored in a directory (path/to/*.fastq).
- If using read, the file extension must be .fastq, and the filename must start with SRR or ERR.
- If the filename does not start with SRR or ERR, rename it accordingly or do not use this parameter.

Read 1 and Read 2:

- Can be single files:
    - path/to/SRR89484_1.fastq and path/to/SRR89484_2.fastq.
    - path/to/ERR83938_1.fastq and path/to/ERR83938_2.fastq.
- If using read, the file extension must be .fastq, and the filename must start with SRR or ERR.
- _1 and _2 must be added before the file extension.
- If the filename does not follow this format, rename it accordingly or do not use this parameter.

Reads:

- Can be a single paired-end file (e.g., path/to/SRR89484_1 and _2.fastq or path/to/ERR83938_1 and _2.fastq .fastq).
- Can be multiple files stored in a directory (path/to/*_{1,2}.fastq).
- The files must not be clipped (e.g., each file should contain both forward and reverse reads).

- If using read, the file extension must be .fastq, and the filename must start with SRR or ERR.
- If the filename does not follow this format, rename it accordingly or do not use this parameter.

4. Troubleshooting

Common Issues

- Check whether the parameter is provided correctly, including case sensitivity as uppercase letters will also cause an error.
- **Nextflow command not found:** Ensure Nextflow is in your system PATH or docker environment PATH.
- **snpEff command error:** The snpEff configuration file is not found, or the data directory is missing.
    - If you encounter this issue, follow the snpEff manual (as mentioned in the Installation section) to properly recreate the snpEff database, ensure the correct path to the configuration file, and export the path using .bashrc. Additionally, set the necessary permissions using chmod 777 on the snpEff folder.
- **Java not found:** Install OpenJDK using sudo apt install openjdk-21-jdk.