Artic pipeline

Sandeep Kasaragod

2024-02-22

1 Bioinformatic pipeline

During this practical session will be running the Artic-nf workflow which requires nextflow and artic packages to be installed on the conda environment. This workflow processes the raw data (fastq) to produce the consensus sequences.

Important Note: Due to the size of the data files used for the workshop we are unable to share the full data in github (there are size limitations on github), therefore we have provided a subset of data to use as practice (beyond the workshop) that is part of the workshop's downloadable folder. If you still have access to the SSD then the larger data will be available for this tutorial.

1.1 Structure of Artic-nf directory

Artic-nf workflow contains 8 major components - Evnironment setup - Main workflow - Meta_data directory - Modules - Config file - Raw files - Scripts - Results

Environment setup: Contains necessary modules/tools required to support the workflow. It is recommended use the environment.yml file to setup the workflow. Alternatively, manual_package_install.txt can be used to install the package manually when the environment.yml is unable to run sucessfully.

Main workflow: This enables the execution of the entire workflow. Generally all the sub-tasks are stitched together and executed by the "main.nf" module.

Meta_data directory: This directory contains the sample_sheet, providing information about the barcode and its corresponding sample names.

Modules: All the sub workflow's are stored here, enabling in easy maintanance, replacement and troubleshooting.

Raw files: It is recommended to store the raw fast5/pod5 files inside raw_file directory to keep the well organized project directory. Althought program can accept the raw_files from any directory.

Scripts: Any programs/scripts used in the workflow, other than *.nf are saved in this directory.

Results: All the workflow outputs are stored here in a sub-folders. This directory is generated during the workflow run. It can be "results" or "analysis" or in other names (as per the user defination).

Tree structure below may give you a better idea of how the files are organized in Artic-nf.

1.2 Mandatory parameters

The workflow requires some of the mandate parameters to run the workflow, this includes

- output_dir: To store all the analysis results
- meta_file: Contains barcode and their sample details along with primer schema and version
- basecaller: Enables user to select basecaller software [Guppy or Dorado]
- fast5_or_pod5_dir: Directory for raw files
- primer_schema: Directory of primer_schema
- kit_name: Barcode kit name for demux
- dorado_dir: Location of the Dorado software (not required if Guppy is used)
- guppy_dir: Location of the Guppy software (not required if Dorado is used)

1.3 Tasks run by the Artic-nf workflow

Artic-nf workflow runs multiple tasks simultaneously, Here are the list of tasks run by the Artic-nf. - Basecalling: Basecalling is performed by the user specified basecalling tool - Barcodering: Performs demultiplexing for the basecalled fastq files - Plex/demux: Aggregate pre-demultiplexed reads - Medaka: This steps performs multiple steps related to medaka and data filtering Some of the major steps are mentioned below. - Alignment - Variant calling - Variant filter - Concat: Concatinating all the sequences to a single fasta file - Mafft: Performing mafft alignment on the concatinated sequence - Summary: Produces the summary of the aligned reads

The medaka step executes multiple commands, if you are curious to know about the commands you can have a look at

```
less results/medaka/<sample_name>.minion.log.txt
```

1.4 Notes on these instructions

This is not simply a copy and paste exercise! The commands you are instructed to run require some editing, e.g. to tell the pipeline where your data is. Parts of the commands with *<some text>* are highlighing sections that need your input- i.e. you need to edit the code.

1.4.1 Using pipeline without RAGE-on-ssd

Note that this tutorial has been written according to the use of the RAGE-on-SSD environment. If you want to run this without the SSD there are some extra steps.

You need to install the tools Dorado and weeSam following instructions here: Dorado weeSam

Create a directory called "tools" in the tutorial directory (i.e. inside RAGE-workshop-2024/4_command_line_and_phylogene tutorial-bioinformatic_pipeline) of the material downloaded after the workshop.Place the Dorado and weeSam downloads in this folder.

Continue with the instructions below (skip 1.6.2) but replace command.sh with command_github.sh in section 1.8.

If you continue to struggle and need extra help, please reach out via the RAGE mailing list: rage-project@lists.cent.gla.ac.uk

1.5 Download Artic-nf and activate the conda environment

First we need to ensure we have access to all the tools needed to run the pipeline commands. We have a custom conda environment specifically for this: Artic-nf

```
git clone https://github.com/RAGE-toolkit/Artic-nf.git
cd Artic-nf
conda env create --file environment.yml
conda activate artic_nf
```

1.6 Data organization

It important that raw data is left untouched - we don't want to risk modifying these files. We can use it for input but not direct manipulation of the data. It is best to create a well defined space for any processed data. Now let us have a look at number of raw data available and get some more details of it.

1.6.1 Task 1

1.6.2

list the number of barcodes available in fastq_pass directory:

```
ls -lh raw_files/fastq/
```

The '-l' in the command to list all the files in a given directory and 'h' for providing human redable information.

1.6.2 Task 2

Similarly, du command can be used provide the disk space occupied by certain directory or files

```
du -h scripts
```

The "-h" parameter provides the human readable information, such as showing each file size in MB/GB instead of bytes. Alternatively you can try without "h" option to see how the du command prints.

```
du scripts
```

1.7 Running the workflow

Artic-nf can be run in two ways. One with passing all the mandate parameters to the terminal.

```
nextflow main.nf \
 --meta_file "meta_data/sample_sheet.csv" \
 --fast5_dir "projects/fast5/" \
 --guppy_dir "projects/ont-guppy-cpu/bin/" \
 --primer_schema "projects/Artic-nf/meta_data/primer-schemes/" \
 --guppy_barcode_kits "EXP-NBD104" \
 --output_dir "results"
```

Another method with editing all the parameters in the **nextflow.config** file.

```
nextflow main.nf
```

1.8 Running workflow for the current session

```
open->terminal (Ctrl+Alt+T)
cd workshop_dir/backup/Artic-nf
conda activate artic_nf_backup
bash command.sh
```

Tasks

1.9.1 Task 3

Go through the meta data file and understand the format of the sample sheet

```
less meta_data/sample_sheet.csv
```

1.9.2 Task 4

View the workflow config file using “less” command

```
less nextflow.conf
```

1.9.3 Task 5

Go through each folder of the result section and list down each folders/directory created by the Artic_nf workflow. Example command given below.

```
ls -lh results/dorado_basecaller
```

1.9.4 Task 6

View the summary stat result

```
less results/summary_stats2/summary_stats.txt
```