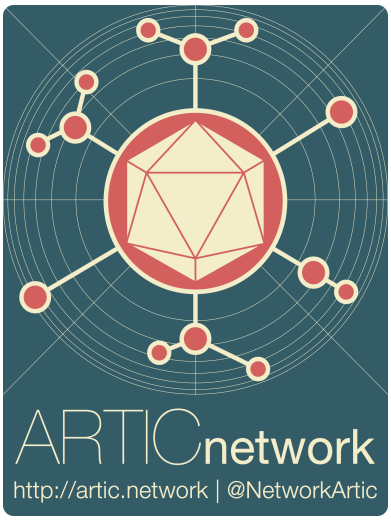

Ebola virus bioinformatics protocol

Nanopore | bioinformatics

Document: ARTIC-EBOV-bioinformaticsSOP-v1.0.1
Creation Date: 2018-05-26
Author: Nick Loman

Overview: A complete bioinformatics protocol to take the output from the sequencing protocol (/ebov/ebov-seq-sop.html) to consensus genome sequences. Includes basecalling, de-multiplexing, mapping, polishing and consensus generation.

This document is part of the Ebola virus Nanopore sequencing protocol package:

http://artic.network/ebov/

Related documents:

Ebola virus Nanopore sequencing protocol:

http://artic.network/ebov/ebov-seq-sop.html (/ebov/ebov-seq-sop.html)

Setting up the laptop computing environment using Conda:

http://artic.network/ebov/ebov-it-setup.html

Phylogenetic analysis and visualization:

http://artic.network/ebov/ebov-phylogenetics-sop.html

Funded by the Wellcome Trust
Collaborators Award 206298/Z/17/Z --- ARTIC network (artic.network)

Preparation

Set up the computing environment as described here in this document: ebov-it-setup (ebov-it-setup.html). This should be done and tested prior to sequencing, particularly if this will be done in an environment without internet access or where this is slow or unreliable. Once this is done, the bioinformatics can be performed largely off-line. If you are already using lab-on-SSD, you can skip this step.

Make a new directory for analysis

Give your analysis directory a meaningful name, e.g.. analysis/run_name

```
mkdir analysis
cd analysis

mkdir run_name
cd run_name
```

Activate the ARTIC environment:

All steps in this tutorial should be performed in the artic-ebov conda environment:

```
source activate artic-ebov
```

RAMPART

To run RAMPART on a current run:

```
artic rampart
```

Select your run and protocol, enter the names of your barcodes, then open http://localhost:3000 in your browser.

Basecalling with Guppy

If you did basecalling with MinKNOW, skip this step.

Run the Guppy basecaller on the new MinION run folder:

For fast mode basecalling:

```
guppy_basecaller -c dna_r9.4.1_450bps_fast.cfg -i /path/to/reads -s run_name -
x auto -r
```

For high-accuracy mode basecalling:

```
guppy_basecaller -c dna_r9.4.1_450bps_hac.cfg -i /path/to/reads -s run_name -x auto -r
```

You need to substitute `/path/to/reads` to the folder where the FAST5 files from your run are. Common locations are:

- Mac: `/Library/MinKNOW/data/run_name`
- Linux: `/var/lib/MinKNOW/data/run_name`
- Windows `c:/data/reads`

This will create a folder called `run_name` with the base-called reads in it.

Consensus sequence generation

We first collect all the FASTQ files (typically stored in files each containing 4000 reads) into a single file.

```
artic gather --min-length 400 --max-length 700 --prefix run_name
```

The command will show you the runs in /var/lib/MinKNOW/data and ask you to select one. If you know the path to the reads use:

```
artic gather --min-length 400 --max-length 700 --prefix run_name --directory /path/to/reads
```

Here `/path_to_reads` should be the folder in which MinKNOW put the base-called reads (i.e., `run_name` from the command above).

We use a length filter here of between 400 and 700 to remove obviously chimeric reads.

You may need to change these numbers if you are using different length primer schemes. Try the minimum lengths of the amplicons as the minimum, and the maximum length of the amplicons plus 200 as the maximum.

I.e. if your amplicons are 300 base pairs, use –min-length 300 –max-length 500

You will now have a file called: `run_name_pass.fastq` and a file called `run_name_sequencing_summary.txt` , as well as individual files for each barcode (if previously demultiplexed).

Demultiplex with Porechop with stringent settings

This stage is obligatory, even if you have already demultiplexed with Guppy, due to significant barcoding misassignments that can confound results:

```
artic demultiplex --threads 4 run_name_pass.fastq
```

Now you will have new files called:

```
run_name_pass_NB01.fastq
run_name_pass_NB02.fastq
run_name_pass_NB03.fastq
```

Create the nanopolish index (once per sequencing run, not per sample)

```
nanopolish index -s run_name_sequencing_summary.txt -d /path/to/reads run_name
_pass.fastq
```

Again, alter `/path/to/reads` to point to the original location of the FAST5 files.

Run the MinION pipeline

For each barcode you wish to process (e.g. run this command 12 times for 12 barcodes), replacing the file name and sample name as appropriate:

E.g. for NB01

```
artic minion --normalise 200 --threads 4 --scheme-directory ~/artic/artic-ebo
v/primer-schemes --read-file run_name_pass_NB01.fastq --nanopolish-read-file r
un_name_pass.fastq IturoEbola/V1 samplename
```

Replace `samplename` as appropriate.

E.g. for NB02

```
artic minion --normalise 200 --threads 4 --scheme-directory ~/artic/artic-ebo
v/primer-schemes --read-file run_name_pass_NB02.fastq --nanopolish-read-file r
un_name_pass.fastq IturoEbola/V1 samplename
```

Output files

- `samplename.primertrimmed.bam` - BAM file for visualisation after primer-binding site trimming
- `samplename.vcf` - detected variants in VCF format
- `samplename.variants.tab` - detected variants
- `samplename.consensus.fasta` - consensus sequence

To put all the consensus sequences in one filei called my_consensus_genome, run

```
cat *.consensus.fasta > my_consensus_genomes.fasta
```

To visualise genomes in Tablet

Open a new Terminal window:

```
conda activate tablet
tablet
```

Go to “Open Assembly”

Load the BAM (binary alignment file) as the first file.

Load the refernece file (in artic/artic-ebov/primer_schemes/IturiEbola/V1/IturiEbola.reference.fasta) as the second file.

Select Variants mode in Color Schemes for ease of viewing variants.