Adapted from https://artic.network/ncov-2019/ncov2019-bioinformatics-sop.html

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 should be performed in the artic-rabv conda environment:

```
source activate artic-rabv
```

Basecalling with Guppy

If you did basecalling with MinKNOW, you can skip this step and go to Demultiplexing.
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
```

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.

Demultiplexing

For the current version of the ARTIC protocol it is essential to demultiplex using strict Parameters to ensure barcodes are present at each end of the fragment.

```
guppy_barcoder --require_barcodes_both_ends -i run_name -s
output_directory --arrangements_files "barcode_arrs_nb12.cfg
barcode_arrs_nb24.cfg"
```

The following steps must be done for every barcode

Read filtering

This step is performed for each barcode in the run.
We first collect all the FASTQ files (typically stored in files each containing 4000 reads) into a single file.
To collect and filter the reads for barcode03, we would run:

```
artic guppyplex --skip-quality-check --min-length 350 --directory
output_directory/barcode03 --prefix run_name
```

You will now have a files called: run_name_barcode03.fastq

Run the MinION pipeline: medaka

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 barcode03

```
artic minion --medaka --normalise 200 --threads 4 --scheme-directory
~/artic-rabv_ea/primer_schemes --read-file run_name_barcode02.fastq
rabv_ea/V1 samplename
```

Replace samplename as appropriate.

Output files

samplename.rg.primertrimmed.bam - BAM file for visualisation after primer-binding site trimming
samplename.trimmed.bam - BAM file with the primers left on (used in variant calling)
samplename.merged.vcf - all detected variants in VCF format
samplename.pass.vcf - detected variants in VCF format passing quality filter
samplename.fail.vcf - detected variants in VCF format failing quality filter
samplename.primers.vcf - detected variants falling in primer-binding regions
samplename.consensus.fasta - consensus sequence
To put all the consensus sequences in one file called my_consensus_genomes.fasta , run

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

To visualise genomes in Tablet

Open a new Terminal window:

```
tablet
```

Go to “Open Assembly”
Load the BAM (binary alignment file) as the first file.
Load the reference file (in artic/artic-rabv/primer_schemes/rabv_ea/V1/rabv_ea_reference.fasta) as the second file.
Select Variants mode in Color Schemes for ease of viewing variants.