QC

Nanopore does it's own QC for the reads. Ignore the failed reads. fast5 format has additional information about the sequenced reads

The "sequencing summary" file has information about each sequence and also has mean qscore: quality score

```
#Concatenate all your "pass" reads into one giant fastq file
cat *.fastq > ../../../../../trial_BR_assembly_VK/200214_BR_Refill_pass.fastq
```

Porechop 0.2.41 with default settings used for adapter trimming. can be used on barcoded reads for demultiplexing (separating based on barcodes) and adapter trimming. Porechop is no longer supported

```
#for porechop help
python3 /gpfs/ebg_data/programs/nanopore/Porechop/porechop-runner.py -h
```

```
#to run porechop
nohup python3 /gpfs/ebg_data/programs/nanopore/Porechop/porechop-runner.py -i
200214_BR_pass_catall.fastq -o 200214_BR_pass_catall.trimmed.fastq -t 8&
```

ASSEMBLY

Nanopore assembly. You have options of a "merged/hybrid" assembly using illumina short reads + long reads or just nanopore assembly with a illumina polish

Flye

```
#--meta: for metagenomes with uneven coverage
#--genome-size: estimated genome size for organism or for metagenome (mg size)
#-i: polishing iterations
#-t: threads

nohup /gpfs/ebg_data/programs/nanopore/Flye/bin/flye --nano-raw
200214_BR_pass_catall.trimmed.fastq --meta --genome-size 450m --out-dir
assembly_fly -i 0 -t 12&
```

What are metrics for assessing the quality of your assembly?
N50? circular contigs?, longest and shortest contigs
View your assembly graph (.GFA file) in Bandage to see problematic regions (bubbling in the graph)

POLISHING

Polishing step is to resolve sequencing error on draft assembled sequences using both long-read and short read (Illumina) sequence data. Sequencing errors present themselves as sections of sequence that do not have "consensus" I.e. the reads don't agree with each other at certain positions.

Racon

racon
"Racon can be used as a polishing tool after the assembly with either Illumina data or data produced by third generation of sequencing. The type of data inputed is automatically detected."

This Racon consensus step uses adapter-trimmed long-read fastq sequences "200214_BR_pass_catall.trimmed.fastq"

Iterative step of polishing four times. Index assembly --> mapping reads to assembly --> Polishing with racon

```
#ensure your fastq headers do not have any spaces
#replace spaces with an underscore:
sed 's/\s/_/g' 200214_BR_pass_catall.trimmed.fastq >
200214_BR_pass_catall.trimmed.nospace.fastq
```

OPTION: put the below into a script which will run one command after another

```
#indexing the fasta sequences
/gpfs/ebg_data/programs/bwa/bwa index ../assembly.fasta

#mapping step 1
/gpfs/ebg_data/programs/bwa/bwa mem -t 20 -x ont2d ../assembly.fasta
../../200214_BR_pass_catall.trimmed.nospace.fastq -o assembly.mapping.sam
```

Note: You can expect contigs to get dropped by Racon. "by default Racon does not output contigs which were not polished, meaning not a single window of 500bp had two reads mapped to it. Hence contigs which either of low quality, or contained in other contigs, or there are no reads supporting them, are dropped. You can get them with option -u." issue

```
#polishing step 1
/gpfs/ebg_data/programs/nanopore/racon/build/bin/racon -t 20
../../200214_BR_pass_catall.trimmed.nospace.fastq assembly.mapping.sam
../assembly.fasta > racon1.fasta
```

This command removes the extra the stuff at the start of the fasta file and also the headers after the first whitespace. This is required because BWA sometimes fails when there's whitespaces in the headers and sometimes doesn't (don't know why). The error you will get if that happens is "cannot parse the .amb file"

">contig_1 LN:i:60228 RC:i:454 XC:f:0.909091" will change to ">contig_1"

```
sed -n '/^>/,$p' racon1.fasta | sed 's/\s.*$//g' > racon1.mod.fasta
```

```
#index2
/gpfs/ebg_data/programs/bwa/bwa index racon1.mod.fasta
#map2
```

```
/gpfs/ebg_data/programs/bwa/bwa mem -t 20 -x ont2d racon1.mod.fasta
../../200214_BR_pass_catall.trimmed.nospace.fastq -o racon1.mapping.sam
#polish2
/gpfs/ebg_data/programs/nanopore/racon/build/bin/racon -t 8
../../200214_BR_pass_catall.trimmed.nospace.fastq racon1.mapping.sam
racon1.mod.fasta > racon2.fasta

sed -n '/^>/,$p' racon2.fasta | sed 's/\s.*$//g' > racon2.mod.fasta

#index3
/gpfs/ebg_data/programs/bwa/bwa index racon2.mod.fasta
#map3
/gpfs/ebg_data/programs/bwa/bwa mem -t 20 -x ont2d racon2.mod.fasta
../../200214_BR_pass_catall.trimmed.nospace.fastq -o racon2.mapping.sam
#polish3
/gpfs/ebg_data/programs/nanopore/racon/build/bin/racon -t 20
../../200214_BR_pass_catall.trimmed.nospace.fastq racon2.mapping.sam
racon2.mod.fasta > racon3.fasta

sed -n '/^>/,$p' racon3.fasta | sed 's/\s.*$//g' > racon3.mod.fasta

#index4
/gpfs/ebg_data/programs/bwa/bwa index racon3.mod.fasta
#map4
/gpfs/ebg_data/programs/bwa/bwa mem -t 20 -x ont2d racon3.mod.fasta
../../200214_BR_pass_catall.trimmed.nospace.fastq -o racon3.mapping.sam
#polish4
/gpfs/ebg_data/programs/nanopore/racon/build/bin/racon -t 20
../../200214_BR_pass_catall.trimmed.nospace.fastq racon3.mapping.sam
racon3.mod.fasta > racon4.fasta

sed -n '/^>/,$p' racon4.fasta | sed 's/\s.*$//g' > racon4.mod.fasta
```

Medaka

medaka

```
mkdir medaka
cd medaka
```

medaka_consensus will only work if medaka, minimap2, samtools and tabix are in your path.

```
#to check if something is in your path
echo $PATH| grep medaka
```

To add to path: open your .bashrc, add text inside square brackets to the file [export PATH="path/to/medaka/bin:$PATH"]. Do this for each of missing tools.

Note: Medaka_consensus will fragment contigs where there is no overlap in the reads (gaps in coverage) as this could be a region of misassembly. To keep contigs unfragmented use the -v option. This will also output a .vcf file which tells you where the fragmentation would have been. Issue

```
#-i input basecall (NANOPORE fastq sequences)
#-d draft sequences (racon #4)
#-t threads
#-m model - depends on the the type of basecaller used. "r941_min_high_g303" is
for the MinION. refer to the github page for others
/gpfs/ebg_data/programs/nanopore/medaka/bin/medaka_consensus -i
../../../200214_BR_pass_catall.trimmed.nospace.fastq -d ../racon4.mod.fasta -o .
-t 14 -m r941_min_high_g303
```

PILON

Only if you have also done short-read sequencing for the same sample

To be continued