Molecular Microbiology 2024

Protocol 5

Assessing quality, trimming, and filtering raw Oxford Nanopore sequencing reads

This tutorial covers:

- Assessing Oxford Nanopore read abundance, length, and quality using NanoPlot
- Trimming adapter sequences from Oxford Nanopore reads using Porechop
- Filtering reads based on quality and length using Filtlong

It has been tested using NanoPlot version 1.42.0, Porechop version 0.2.4, and Filtlong version 0.2.1.

1. Log into your server using **ssh** or **MobaXterm** as described in protocol 1, for example in ssh:

   ```
   ssh molmicroX@dnaseq1a.bio.au.dk
   ```

2. Change into the genome directory where you have saved your raw reads (fastq file):

   ```
   cd genome
   ```

3. Run `NanoPlot` to assess read abundance, length, and quality. This will take about 10-20 minutes to run.

   ```
   screen -L \
   NanoPlot \
   --loglength \
   --threads 3 \
   --outdir pretrimming_nanoplot_output \
   --fastq input_file.fastq.gz
   ```

   - `screen -L` - This runs all that follows within `screen`, which means that if the ssh connection is disrupted while the command is running it will continue to run to completion.
   - `NanoPlot` - This is the name of the program to be run.
   - `--loglength` - This specifies that the plotted read length should be logarithmic.
   - `--threads 3` - This specifies the number of CPU threads to be used (in this case 3).
   - `--outdir pretrimming_nanoplot_output` - This specifies the name of the output directory, in this case `pretrimming_nanoplot_output` - This should be changed if `NanoPlot` is run more than once in the same directory, for example before and after trimming and filtering.
   - `--fastq input_file.fastq.gz` - This specifies the name of the input file. `input_file.fastq.gz` should be changed to the name of the file with your reads.

4. Using **CyberDuck**, copy the NanoPlot output directory to your own computer. Open the file called *NanoPlot-report.html* and look at the

assessment of your reads. Does the read length match your expectation based on TapeStation results? Look at the total number of bases sequenced - is there enough data to assemble your genome? Look through the plots and get a feeling for the length and quality distribution of your reads.

5. Use `Porechop` to remove adapter sequences from your reads.

```
screen -L \
porechop \
-i input_file.fastq.gz \
-o input_file_trimmed.fastq.gz \
--threads 3
```

   - `porechop` - This is the name of the program to be run.
   - `-i input_file.fastq.gz` - This specifies the name of the input file. `input_file.fastq.gz` should be changed to the name of the file with your reads.
   - `-o input_file_trimmed.fastq.gz` - This specifies the name of the output file with trimmed reads. `input_file.fastq.gz` should be changed to something that makes sense for you.
   - `--threads 3` - This specifies the number of CPU threads to be used (in this case 3).

6. After adapters are trimming from your reads, you will need to filter the reads for the longest, highest quality reads. For this we will use `Filtlong`, which subsamples your reads based on various weighted criteria. This step will reduce the memory and time required for the final genome assembly. Note that screen is run in a slightly different way for filtlong (entering screen then exiting again rather than running it all in a single command) compared to previous steps. This is necessary due to the way that filtlong produces its output (to STDOUT rather than a file directly).

```
screen #First type this to enter screen
filtlong \ #Then enter this line and the following two to run filtlong
--target_bases X \
input_file_trimmed.fastq.gz | gzip > input_file_trimmed_filtered.fastq.gz
exit #Finally exit screen once filtlong has finished running
```

   - `filtlong` - This is the name of the program to be run.
   - `--target_bases X` - `X` should be replaced with the number of bases we are aiming to extract from the file. `filtlong` will extract the longest, highest-quality reads based on a weight-based algorithm. This number must be written out in full (i.e. 1000000000 not 1.0 for 1 gigabase).
   - `input_file_trimmed.fastq.gz` - This should be replaced by the trimmed reads file (output from the previous step)
   - `| gzip > input_file_trimmed_filtered.fastq.gz` - This pipes the output of `filtlong` into `gzip` to compress the output fastq file, and then pipes this into a file called `input_file_trimmed_filtered.fastq.gz`. This filename should be replaced without a filename that makes sense for you.

7. Now rerun `NanoPlot` on the output from `FiltLong` - does the number,

length, and quality of reads match your expectations?