Quality Control Laboratory

Modern sequencing technologies can generate a huge number of sequence reads in a single experiment. However, no sequencing technology is perfect, and each instrument will generate different types and amounts of error. Therefore, it is necessary to understand, identify and exclude error types that may impact the interpretation of downstream analysis.
The objective of this activity is to understand some relevant properties of raw sequence data. We will focus on properties such as length, quality scores and base and k-mer distribution in order to assess the quality of the data and discard low quality or uninformative reads.

Filtering of low quality data and trimming of erroneous sequences such as adaptors are important for all applications. For example, low quality data is more likely to contain errors. If this is used for a genome assembly it may lead to errors in the genome.

We will be using bioinformatics tools to review, analyse and filter our sequencing data based on quality. This part of the workshop will involve using basic UNIX commands, FastQC, cut-adapt and fastq-mcf (more programs are available - see end of document). We will be reviewing datasets from a number of different experiments including genome shotgun, RAD-Seq and amplicons. By the end of this activity, you should feel more comfortable with assessing the quality of raw sequencing data and with filtering the data to improve the quality.

All data for this workshop can be found in

```
~/workshop data/quality control/
```

FastQC

FastQC is a software programme that assesses the quality of sequencing data and produces a report summarising the outcomes in graphical format. For more information take a look at the FastQC help pages. FastQC can be run using a graphical user interface or using the command line. Today we will explore both options.

Exercise One:

1. Firstly, we will take a look at a subset of Illumina sequencing of a single isolate of a Bovine bacteria *Bartonella.* Using the command line take a look at the bartonella_illumina.fastq data.
2. How many reads are there? Remember that the fastq format has four lines per sequencing read.
3. Launch FastQC from the command line:
   ```
   $ fastqc
   ```
4. Load the *Bartonella* data using File and then Open.
5. Inspect the report produced by FastQC. Have a look at the numbers output on the “Basic Statistics” page. How many sequences do we have? What is the sequence length? And the GC content?

6. Examine the “Per base sequence quality” and “Per sequence quality scores” pages. Roughly, how many incorrect base calls are expected at most positions? Do you think this sequencing run gave good quality sequences?
7. Examine the “Per base sequence content” and “Per sequence GC content” pages. On the “Per base sequence content” page, FastQC points out a “potential problem” with an orange exclamation mark. Do you think we should worry about it in this particular case?
8. Examine the “Overrepresented sequences” page. Why does FastQC give a warning message?
9. Save a copy of the report.

Exercise Two

1. Now we will look at Illumina sequencing data from the Earth Microbiome Project. This is an environmental sample sequenced for 16S amplicons. This data is in the file EMP_Misc_16v4EMP_NoIndex_L005_R1_001-sample.fastq.gz.
2. Load the data into FASTQC (note that there is no need to unzip it first).
3. What can explain the pattern observed in the “Per base sequence content”? What can explain the particular pattern observed at the first 8 bases?
4. Based on the warning messages, if this was your data how would you approach the trimming and filtering? Why?

Exercise Three

1. Now we will look at two sets of RAD sequencing data. Firstly 12 RAD sequencing samples saved in the compressed folder CAN.tgz.
2. Uncompressed the file using tar:
   ```
   $ tar -xzvf CAN.tgz
   ```
3. Analyse each of the samples using FastQC with the command line. This shows how to run it for the first file:
   ```
   $ fastqc CAN/can152.fq
   ```
   HINT - why not think about using a wildcard symbol to analyse multiple files at once?
4. Using your web browser look at the graphical results in the html files. From the command line type for the first file:
   ```
   $ firefox CAN/can152_fastqc.html
   ```
   HINT - Think about using the wildcard again.
5. How can you explain the pattern observed in the first six bases?
6. Now let’s look at the second data set. This is a paired-end Illumina run. The forward read is in the file KE1_S49_R1_001.fastq.gz and the reverse read is in the KE1_S49_R2_001.fastq.
7. Use fastqc from the command line again to analyse these reads.
8. With Read1, using the “Sequence duplication level”, “Overrepresented sequences” and “Adapter Content” graphs, what can we infer from this data?
9. Now look at Read 2, what is different about the “Per base sequence content”. What can we say about this RAD data?
10. Also take a look at the “Per base N content” and “Overrepresented sequences”. Do you think this is good data?

Filtering and Trimming

Depending on the downstream application of the data it may be important to remove low quality sequences or adaptor sequences. Various software is available to do this. Today we will look at cutadapt and fastq-mcf. Both achieve the same outcome but work in slightly different ways with different parameter choices.

Exercise Four

1. Let's look at some Illumina MiSeq paired end sequencing data from an amplicon study - 1_TAAGGCGA-TAGATCGC_L001_R1_001.fastq and 1_TAAGGCGA-TAGATCGC_L001_R2_001.fastq
2. Load the data into FastQC and assess the quality.
3. Why is read 2 poorer quality than read 1?
4. Now filter the data. For this we will use fastq-mcf. Take a look at the manual:
   ```
   $ fastq-mcf -h
   ```
5. Run fastq-mcf to filter based on a q score of 35 and a minimum length of 80 bp. The adaptors can be found in the file adaptors.fasta. Make sure you specify an output file for both the forward and reverse reads with a different name than the raw data.
6. Run FastQC on the filtered reads - can you see any changes?
7. Think about cases where such strict quality control might not be necessary.

Exercise Five

1. We will use cutadapt to filter and trim the data from an Illumina sequencing run of some microRNA. This data can be found in the file SRR026762-sample.fastq.gz.
2. Load the data into FastQC and make a diagnosis of the data. What is the source of the overrepresented sequences?
3. Use cutadapt to remove low quality bases and trim the adaptor sequence (these are on the 3' end). The adaptor used to generate this library was SmallRNA3pAdapter_1.5 ATCTCGTATGCCGTCTTCTGCTTG. To look at the cutadapt manual use:
   ```
   $ cutadapt --help
   ```
   You won't need to unzip the file. Make sure that you specify an output file.
4. Look at the filtered file in FastQC and make a new diagnosis of the data. What do you observe?
5. Now let's try removing the remaining adaptor contamination from the SmallRNASequencingPrimer (CGACAGGTTCAGAGTTCTACAGTCCGACGATC). Use cutadapt again with the filtered reads specifying this adaptor.
6. Examine the resulting file in FastQC once more. Are you satisfied with the outcome?

Exercise Six

1. Let's take a look at some PacBio data. These long reads have a higher error rate than Illumina data and a different quality encoding.

2. This data can be found in the m130929_060824_42175_c100588662550000001823089804281482_s1_p0.1.subset.fq
3. Use the command line to look at the first five reads. What's the main difference you notice when looking at PacBio data compared to Illumina data?
4. Analyse the data using FastQC (either GUI or command line).
5. Look at the "Per base sequence quality" and the "Sequence Length Distribution". How do the lengths explain the pattern seen in the "Per base sequence quality" graph?
6. Try filtering the data (you can choose which software) with a q score cutoff of 30. What's happened and why? **Note: look at the help page for fastq-mcf to view the option for running without an adaptors file.**

Other software available:

Trimmomatic (http://www.usadellab.org/cms/?page=trimmomatic)
Trim Galore! (http://www.bioinformatics.babraham.ac.uk/projects/trim_galore/)
BBMap/BBTools (http://sourceforge.net/projects/bbmap/)
seqtk (https://github.com/lh3/seqtk)
PRINSEQ (http://prinseq.sourceforge.net/)