Quality Control for RAW Data – Practical session

1. Open a terminal window
   The programs used in this tutorial are called from the command line. In order to do that, the first step is to open a Terminal window. To do this go to:

   Applications → Accessories → Terminal 

   A new window will open with a prompt ready for an input. Now change to the directory with the sequence data. Type on the terminal:

   cd $HOME/mda13/QC_Raw/

2. Open FastQC program
   To start the FastQC program, you have to type on the terminal window:

   fastqc &

   The FastQC application will start in a new window.
   You can minimize the terminal window, but **do not close it** while using the FastQC application. Otherwise, FastQC will be also closed.

3. Load a file into FastQC
   From the FastQC program, go to:

   File → Open

   And load from the folder **$home/mda13/QC_Raw** the file called

   mirna.fastq

4. Look at the different FastQC result sections and answer the following questions

Questions:

Sample **mirna.fastq**

1. Do a quality control for the sample **using FastQC.**
   What are the parameters you consider bad quality indicators?
   Write down your conclusions:

Per base sequence quality → Quality starts dropping at 23th base. The last 3 bases are predictably wrong. GC content → Not stable. Probably there is a bias due to library contamination or PCR artifact Overrepresented sequences → There are many PCR primers and adapters we should have removed

2. Trim your sample based on its quality with a **minimum quality threshold of 20.**
   You should use Fastx_toolkit, so type on the terminal:
   fastq_quality_trimmer -t 20 -i mirna.fastq -o mirna_t20.fastq

   And now you do the quality control for the new file *mirna_t20.fastq* from FastQC.
   What are the main changes?
   Write down your conclusions:

Per base sequence quality → Good. The mean ends in green. Actually, this is good.
GC content → No changes
Sequence length distribution → Reads from 0-39 length
Overrepresented → Still the same adapters and primers
110 reads have been deleted. → Lower quality than 20 along the sequence
* **IMPORTANT**: -t 20 just removes nucleotides with lower qualities from the end of the sequence !!

3. Trim the sample based on its quality with a **minimum quality threshold of 28.**
   You should use Fastx_toolkit, so type on the terminal:
   fastq_quality_trimmer -t 28 -i mirna.fastq -o mirna_t28.fastq

   And now you do the quality control for the new file *mirna_t28.fastq* from FastQC.
   Is there any quality improvement over the previous filter?
   Write down your conclusions:

Per base sequence quality → We have removed some variability.
GC content → The same. Minor changes.
Overrepresented → Still the same adapters and primers
Sequence length distribution → Increase in the number of short reads.
135 reads have been deleted. → Lower quality than 20 along the sequence

4. Trim the sample based on its quality with a **minimum quality threshold of 28**, removing the reads with a **length lower than 30.**
   You should use Fastx_toolkit, so type on the terminal:
   fastq_quality_trimmer -t 28 -l 30 -i mirna.fastq -o mirna_t28l30.fastq

   And now you do the quality control for the new file *mirna_t28l30.fastq* from FastQC.
   Is there any quality improvement over the previous step?
   How many reads have been removed?
   Write down your conclusions:

Per base sequence quality → Less variability in bases lower than 30.
GC content → Apparently an improvement, but it's just the same.
Overrepresented → Still the same adapters and primers
Sequence length distribution → Decrease in the number of short reads.
24865-21808 = We had 3057 reads shorter than 30 nucleotides.

5. Trim the sample based on its quality with a **minimum quality threshold of 28**, removing the reads with a **length lower than 35.**
   You should use Fastx_toolkit, so type on the terminal:
   fastq_quality_trimmer -t 28 -l 35 -i mirna.fastq -o mirna_t28l35.fastq

   And now you do the quality control for the new file *mirna_t28l35.fastq* from FastQC.
   Is there any quality improvement over the previous step?
   How many reads have been removed?
   Write down your conclusions:

Per base sequence quality → Less variability in bases lower than 35 GC content → the same. Overrepresented → Still the same adapters and primers Sequence length distribution → Decrease in the number of short reads. 21808-17378 = We had 4430 reads shorter between 30 and 35 nucleotides.

Sample **solid.fastq**

1. Do a quality control for the sample using **FastQC**.
   What are the parameters you consider bad quality indicators?
   Write down your conclusions:

Per base sequence quality → What happens in the 48th base?. So much variability in the rest
Per sequence quality scores → Peak indicating there are lots of reads with quality of 5
Per base sequence content → Nucleotides oscilates a lot along the bases
Per sequence GC content → Quite good
Per base N content → We know why the quality in 48th base is so bad. Plenty of N's.
No overrepresented sequences → Fine
Kmer content → We have to remove the poly-T and it seems we have kind of a poly-G. Those G's can be a bias or that a gene has a lot of expression and it's part of its sequence.

2. **Trim** your sample based on its quality with a **minimum quality** threshold of **20**.
   You should use Fastx_toolkit, so type on the terminal:
   fastq_quality_trimmer -t 20 -i solid.fastq -o solid_t20.fastq

   And now you do the quality control for the new file *solid_t20.fastq* from FastQC.
   What are the main changes?
   Do you consider the trimming to be effective?
   Write down your conclusions:

Nearly 10.000 reads have been deleted.
We have removed the peak of quality with 5.
There seems to be an improvement, but it haven't been effective.
GC content → Seems that it have been improved, although the distribution is weird.
We still have the problem with the 48th base.
We have removed the poly-T and poly-G sequences → FINE !!

3. **Trim** the sample based on its quality with a **minimum quality** threshold of **28.**
   You should use Fastx_toolkit, so type on the terminal:
   fastq_quality_trimmer -t 28 -i solid.fastq -o solid_t28.fastq

   And now you do the quality control for the new file *solid_t28.fastq* from FastQC.
   Is there any quality improvement over the previous filter?
   Write down your conclusions:

Nearly 4000 reads more have been deleted.
Seems to be better per base sequence quality.
Still the N in the 48th base.

4. Trim the sample based on its quality with a **minimum quality** threshold of **28**, removing the reads with a **length lower than 47.**
   You should use Fastx_toolkit, so type on the terminal:
   fastq_quality_trimmer -t 28 -l 47 -i solid.fastq -o solid_t28l47.fastq

   And now you do the quality control for the new file *solid_t28l47.fastq* from FastQC.
   Is there any quality improvement over the previous step?
   How many reads have been removed?
   Write down your conclusions:

125000-44119 = 80881 reads have been removed from the original file
Per seq. Quality score → Fine
GC content is more accurate. → More normal distribution
Still have the problem with N's

5. Remove the reads with **less than a 90%** with **quality above 20.**
   You should use Fastx_toolkit, so type on the terminal:
   fastq_quality_filter -q 20 -p 90 -i solid.fastq -o solid_q20p90.fastq

   And now you do the quality control for the new file *solid_q20p90.fastq* from FastQC.
   Has the filter been effective?
   How many reads have been removed?
   Write down your conclusions:

Nearly 100.000 reads have been removed !!
The quality now is much more better.
GC content nearly perfect.
Still the N problem which cannot be resolved.

Annex 1: Fastx_toolkit

fastq_quality_trimmer

usage: fastq_quality_trimmer [-h] [-v] [-t N] [-l N] [-z] [-i INFILE] [-o OUTFILE]
Part of FASTX Toolkit 0.0.13 by A. Gordon (gordon@cshl.edu)

```
  [-h]        = This helpful help screen.
  [-t N]      = Quality threshold - nucleotides with lower
                quality will be trimmed (from the end of the sequence).
  [-l N]      = Minimum length - sequences shorter than this (after trimming)
                will be discarded. Default = 0 = no minimum length.
  [-z]        = Compress output with GZIP.
  [-i INFILE]  = FASTQ input file. default is STDIN.
  [-o OUTFILE] = FASTQ output file. default is STDOUT.
  [-v]        = Verbose - report number of sequences.
                If [-o] is specified,  report will be printed to STDOUT.
                If [-o] is not specified (and output goes to STDOUT),
                report will be printed to STDERR.
```

fastq_quality_filter

usage: fastq_quality_filter [-h] [-v] [-q N] [-p N] [-z] [-i INFILE] [-o OUTFILE]
Part of FASTX Toolkit 0.0.13 by A. Gordon (gordon@cshl.edu)

```
  [-h]        = This helpful help screen.
  [-q N]      = Minimum quality score to keep.
  [-p N]      = Minimum percent of bases that must have [-q] quality.
  [-z]        = Compress output with GZIP.
  [-i INFILE]  = FASTA/Q input file. default is STDIN.
  [-o OUTFILE] = FASTA/Q output file. default is STDOUT.
  [-v]        = Verbose - report number of sequences.
                If [-o] is specified,  report will be printed to STDOUT.
                If [-o] is not specified (and output goes to STDOUT),
                report will be printed to STDERR.
```

Examples:

- Trimming of sequences with quality lower than 20:

```
fastq_quality_trimmer -t 20 -i <sample>.fastq -o <sample_out>.fastq
```

- Trimming of sequences with quality lower than 20 and minimum length of 30:

```
fastq_quality_trimmer -t 20 -l 30 -i <sample>.fastq -o <sample_out>.fastq
```

- Trimming of sequences with less than 90% of bases with quality above 20:

```
fastq_quality_filter -q 20 -p 90 -i <sample>.fastq -o <sample_out>.fastq
```