Quality control and artefact removal

Dóra Bihary

15 September, 2017

Contents

---

1 Introduction

The data-set for this practical is a publicly available dataset downloaded from the NCBI GEO repository with the accession: GSE15780. It looks at the genome wide binding of tp53 and tp73 (TAp73beta isoform) transcription factors in the human osteosarcoma cell lines Saos-2.

We downloaded the dataset (fastq files) from the Sequence Read Archive using the SRA-toolkit. There are multiple ways of doing this.

1. Browse the **SRA dababase** and download the data.

   https://www.ncbi.nlm.nih.gov/sra

2. Use **SRA toolkit**. You need to install and configure this on your computer first. Detailed instructions are here:

   https://ncbi.github.io/sra-tools/install_config.html

3. Use the Bioconduuctor package **SRAdb** to download the SRR files.

   https://bioconductor.org/packages/release/bioc/html/SRAdb.html

Genomic alignments can be time consuming and not realistic to do in the short time we have. Therefore, we downloaded and preprocessed a single chromosome from the above dataset to save time. This preprocessing step included aligning to a *GRCh38 genome* with a *sponge database* (which removes artefacts and non-chromosomal sequences) and then regenerating the chr3 fastq files.

In this tutorial we will first generate a quality report about a single sample (tp73_rep2) from this dataset using **FastQC**, based on this report we will decide what type of trimming is needed in order to improve the quality of our reads before alignment. We will use **Cutadapt** to trim the reads, and finally we will generate a new quality report from the trimmed reads and compare it with the original one.

2 Checking read quality using FastQC

Let's open a terminal window and change the directory to ~**/Course_Materials/Introduction/SS_DB/RawData/ChIPseq/**:

```
cd ~/Course_Materials/Introduction/SS_DB/RawData/ChIPseq/
```

We will use FastQC to check the quality of our sequence reads. Here we will use the command line version of FastQC. To check what kind of options this tool has, type:

```
fastqc --help
```

This command will display in your terminal window all the parameters you can use when running FastQC. Now we will run a fairly simple command:

```
fastqc -o "~/Course_Materials/Introduction/SS_DB/QC/" --noextract -f fastq tp53_r2.fastq.gz
```

The options we were using:

- -o *FOLDER_NAME*: you can define this way in which folder you want FastQC to output its results,
- –noextract: will tell FastQC not to uncompress the output file after creating it,
- -f fastq: here we define that our input file is a fastq file (valid formats are bam,sam,bam_mapped,sam_mapped and fastq).

Open the generated .html report (that you find in Course_Material/Practical1/ folder) and go through each section carefully.

3 Quality based trimming using Cutadapt

Once you had a closer look at the quality report you can realize that the data quality is not toot bad, however we still might be able to improve the quality with a quality based trimming since the quality drops towards the end of the reads:

We will use Cutadapt for trimming, so let's have a look at its help page:

```
cutadapt --help
```

As you can see Cutadapt has many options for:

- trimming based on quality threshold,
- trimming some bases from the 5' or 3' ends of reads,
- removing adapter contaminations.

In our case all we want to do is to remove low quality bases from our reads. We can use the following command to do this:

```
cutadapt -m 10 -q 20 -o tp53_r2.fastq_trimmed.fastq.gz tp53_r2.fastq.gz
```

Let's go through the parameters we are using in the command above:

- -m 10: will discard all reads that would be shorter than a read length of 10 after the trimming,
- -q 20: will trim low-quality bases from the 3' end of the reads; if two comma-separated cutoffs are given, the 5' end is trimmed with the first cutoff, the 3' end with the second,
- -o *FILE_NAME*: the output file name.

4 Checking the quality of trimmed reads using FastQC

Once the trimming has finished we will want to check the quality of our trimmed reads as well to make sure, we are happy with its results: the trimming improved the quality and it didn't introduce new artefacts. So let's run FastQC again on our trimmed .fastq file with the following command:

```
fastqc -o "~/Course_Materials/Introduction/SS_DB/QC/" --noextract -f fastq tp53_r2.fastq_trimmed.fastq.gz
```

Now open the generated report. As you can see after trimming:

- the quality doesn't drop that much at the end of the reads anymore,

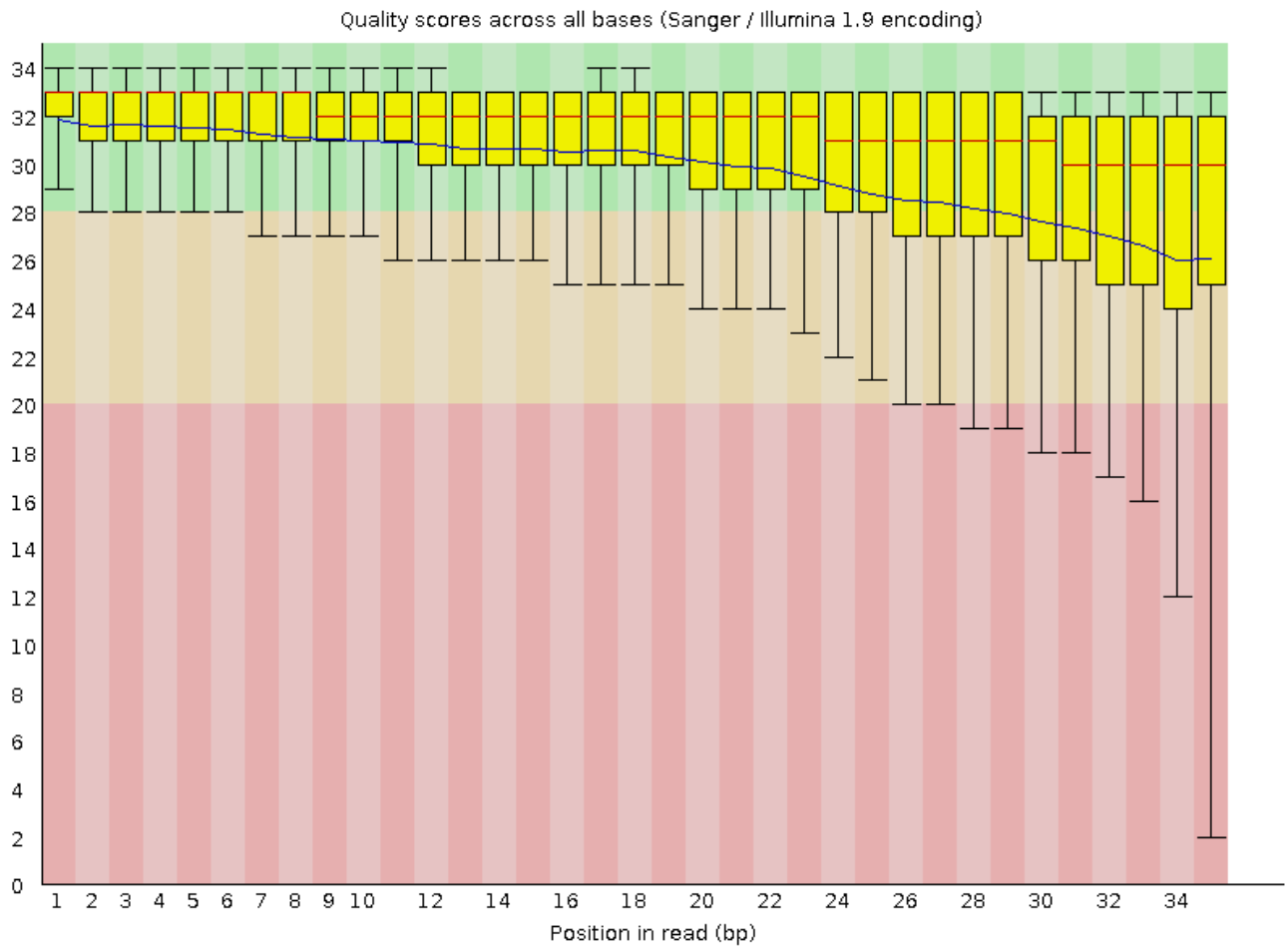

Figure 1: FastQC report of raw reads

- we see that a new warning appeared under "Sequence Length Distribution" which was expected since our read length is not constant anymore,
- if you check the "Basic Statistics" section you will see that we didn't actually loose much data, the amount of sequences decreased, but only slightly.