Computational practical 2: Accessing Data and Quality Control

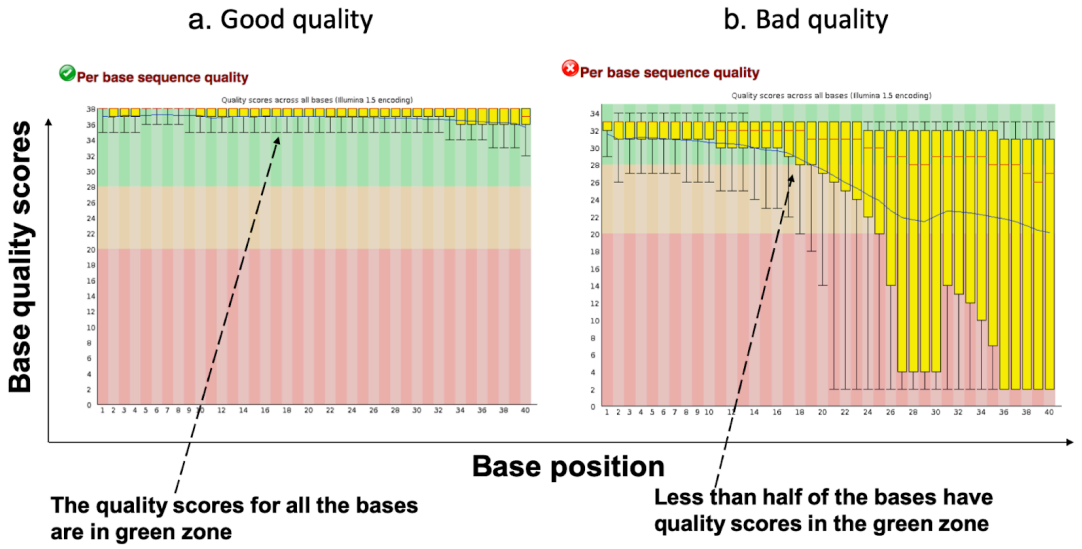
Module Developers: Dr. Stanford Kwenda, Mr. Collins Kigen and Mr Mishalan Moodley

Table Of Contents

Introduction

A typical whole genome sequencing process involves genomic DNA isolation, library preparation and sequencing. Errors occurring at any of these steps can negatively impact the quality of the sequence information and hence affect downstream analysis. For example, the samples could be mixed during sample or library preparation, or errors can be encountered during the sequencing itself. If these errors are not removed from the raw reads, they might be incorporated into your analysis output and would be harder to resolve later on. Therefore, it is important to perform quality checks on the raw sequence reads before starting your analysis.

There are several bioinformatic tools available for evaluating read data quality, here we will discuss one of the widely used tools, called FastQC which is made available from Babraham Institute. The tool can be run by both command-line and also has a graphical user interface enabling its use without knowledge of the command-line. The tool provides you with a report on the quality of sequence reads using a traffic light system, red, amber and green. There are a number of parameters which we will learn in this module that help us in assessing the sequence data quality. Below are the two screenshots generated by the FastQC tool (a) good sequence data and (b) bad sequence data.



Example: Assessing quality of sequence reads

Downloading the sequence reads

Let's assess the sequence data quality of an isolate of *Salmonella typhi* (ERR2093245).

First, open the terminal and change your working directory to **cp2**.

cd /home/manager/course/cp2

Now download the sequence reads (fastq files) for analysis by entering the following commands one after the other.

wget ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR209/005/ERR2093245/ERR2093245_1.fastq.gz

wget ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR209/005/ERR2093245/ERR2093245_2.fastq.gz

Running the FastQC tool with command line

Run the following command to run fastqc tool on both the reads files:

fastqc ERR2093245_1.fastq.gz ERR2093245_2.fastq.gz

Note: The program will exit with error if the downloaded ".gz" files are truncated/not downloaded completely. If you face such an error, please download the file again using the wget commands and rerun fastqc.

Upon successful completion fastqc will create an analysis report in "html" format one for each read file. We can see the report by opening the html file in the web-browser.

Let's take a look at the important graphs generated below that represent overall quality of the sequence reads.

1. Basic statistics

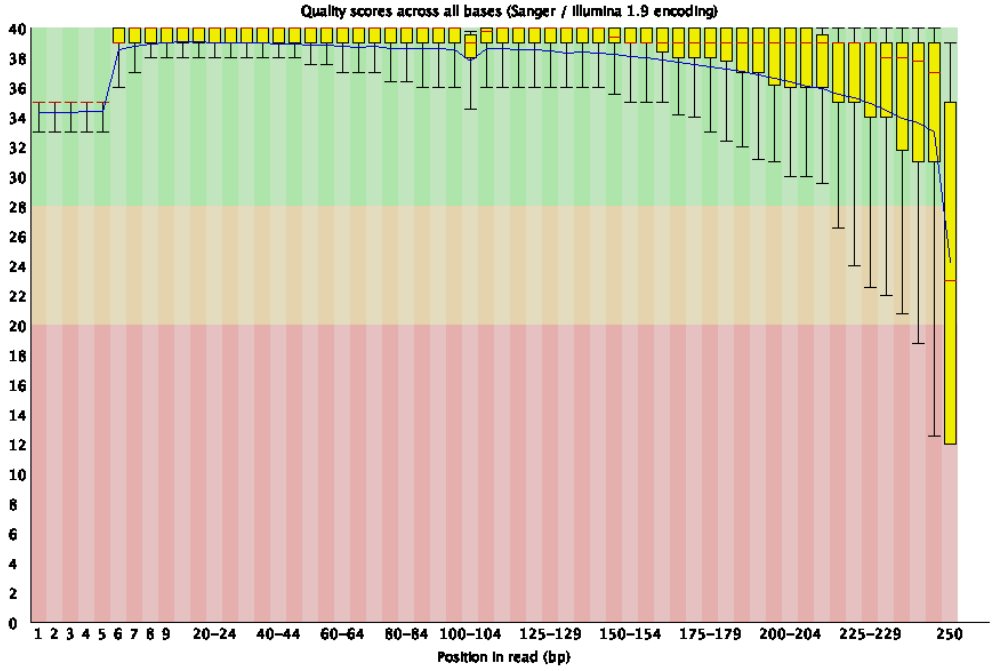
Basic Statistics

Measure	Value
Filename	ERR2093245_1.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	1339517
Sequences flagged as poor quality	0
Sequence length	250
%GC	51

This is a table containing basic information gleaned from the sequence reads such as total number of reads, length (range) of sequence reads and GC%. From this alone we can infer average coverage (total number of reads and length of reads and compare the GC content with the species that we expect the isolate to belong to.
For example, in the example here, we have a *Salmonella typhi* isolate that has GC percent of around 50% which matches with the GC% in the reads.

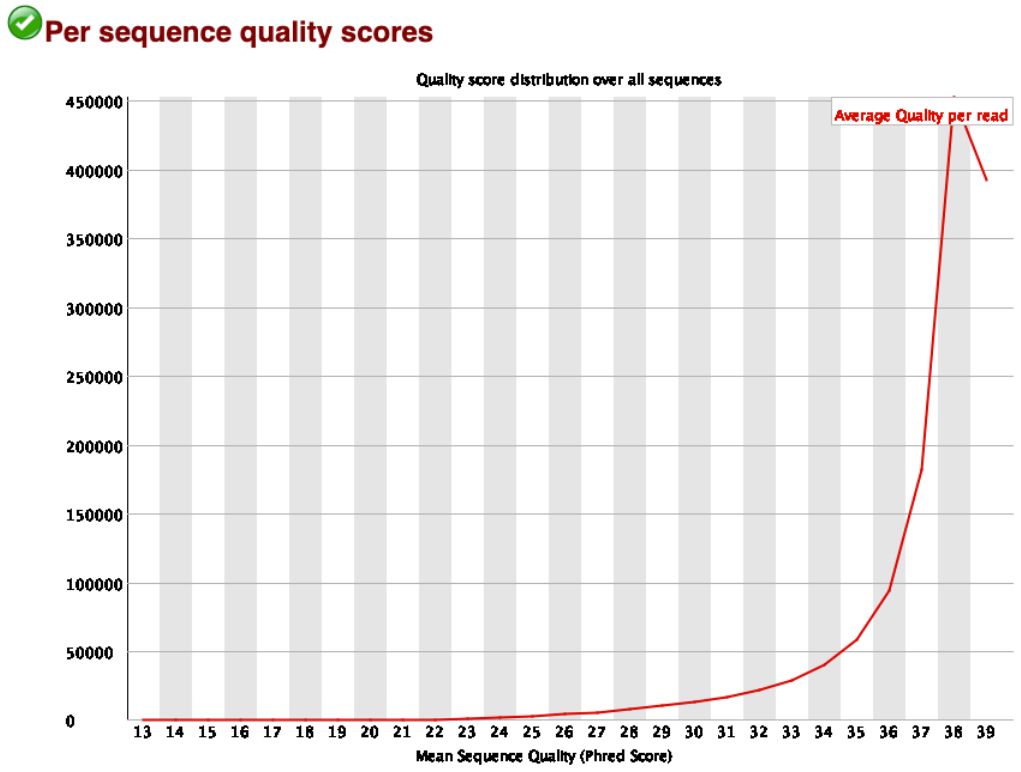
2. Per base sequence quality

Per base sequence quality



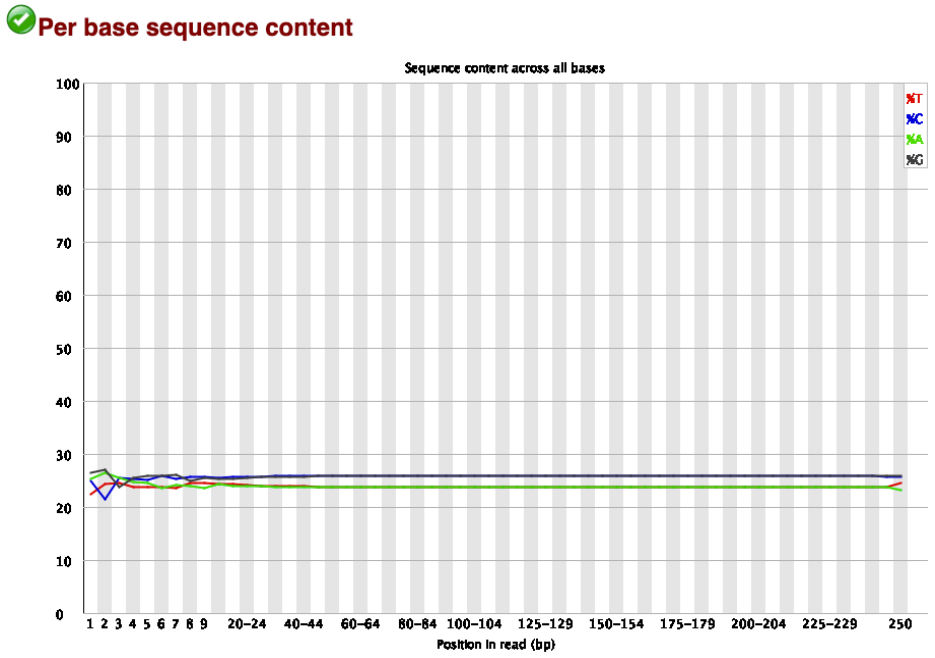
The y-axis on the graph shows the quality scores and the x-axis represents the base positions in the reads. The blue line shows the mean quality of bases for each position in the reads. The space coloured in the green regions shows high quality, the amber coloured region below reflects acceptable quality and the regions in the red shows low quality. Therefore, if you observe the blue line in the red region for your sequence reads, it means lots of errors in the sequence reads and might need trimming before any downstream analysis.

3. Per sequence quality scores



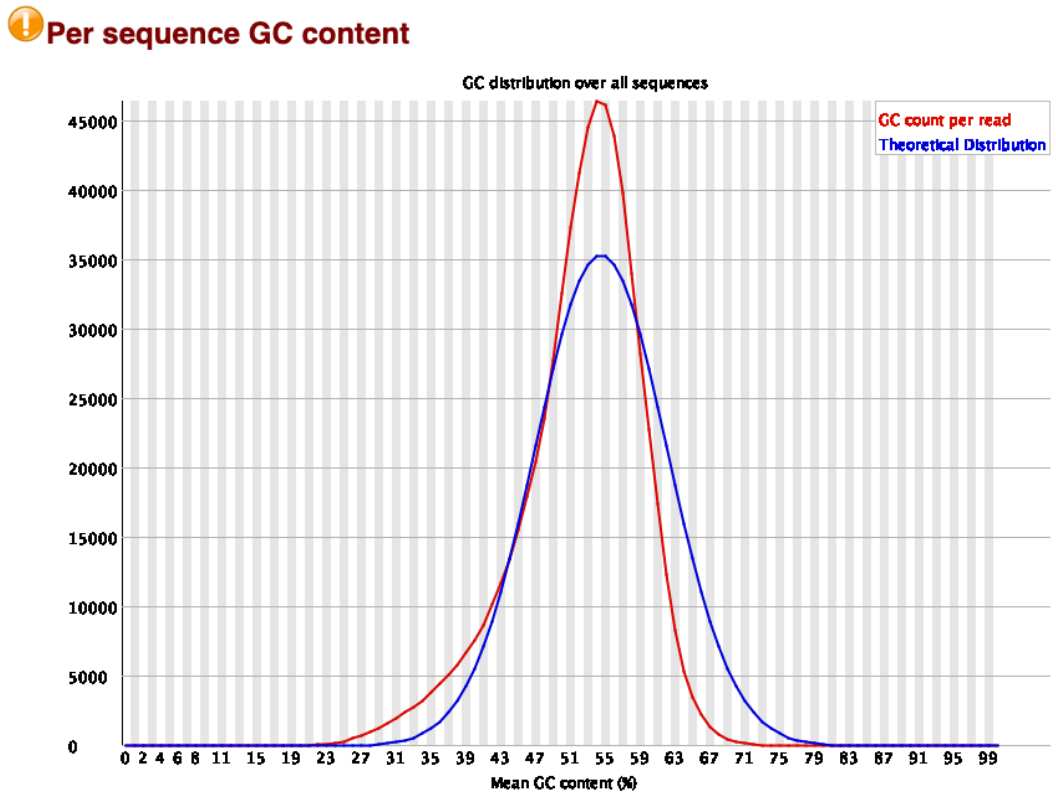
The per sequence quality score report shows how many reads have overall low quality values. If a significant proportion of the sequences in a run have overall low quality then this could indicate some kind of systematic problem - possible with just part of the run (for example one end of a flowcell).

4. Per base sequence content



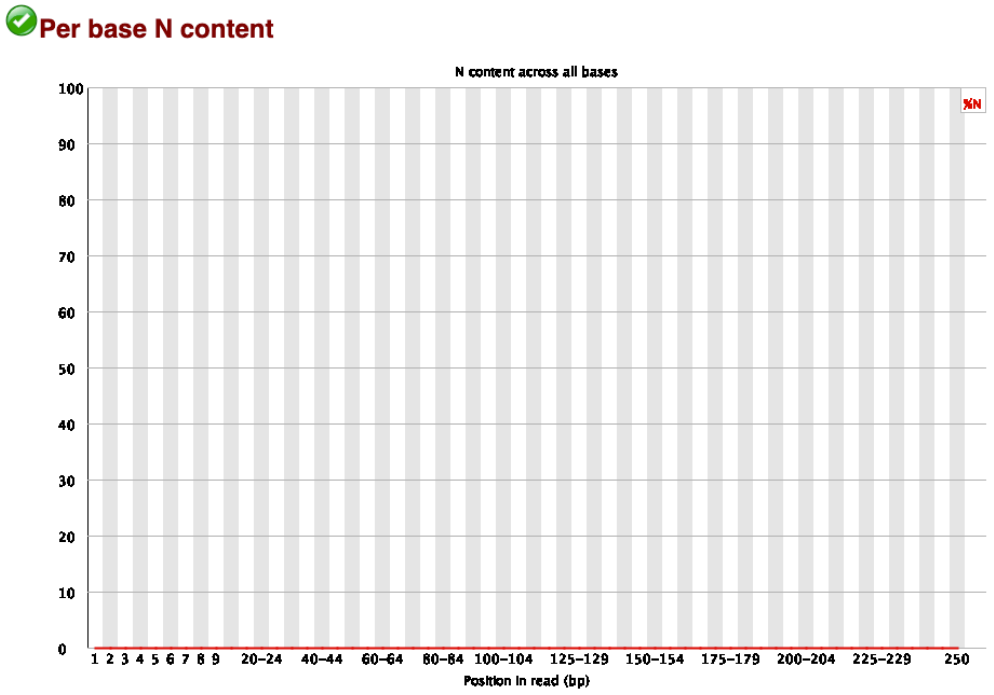
The graph above shows the overall GC content at each of the read positions with X-axis being the base position in the reads and Y-axis showing the percentage of A,T,G and C. For a good library the lines should run parallel reflecting no difference in base calling for each of the 4 bases. The relative amount of each base should reflect the overall composition of the genome.

5. Per sequence GC content



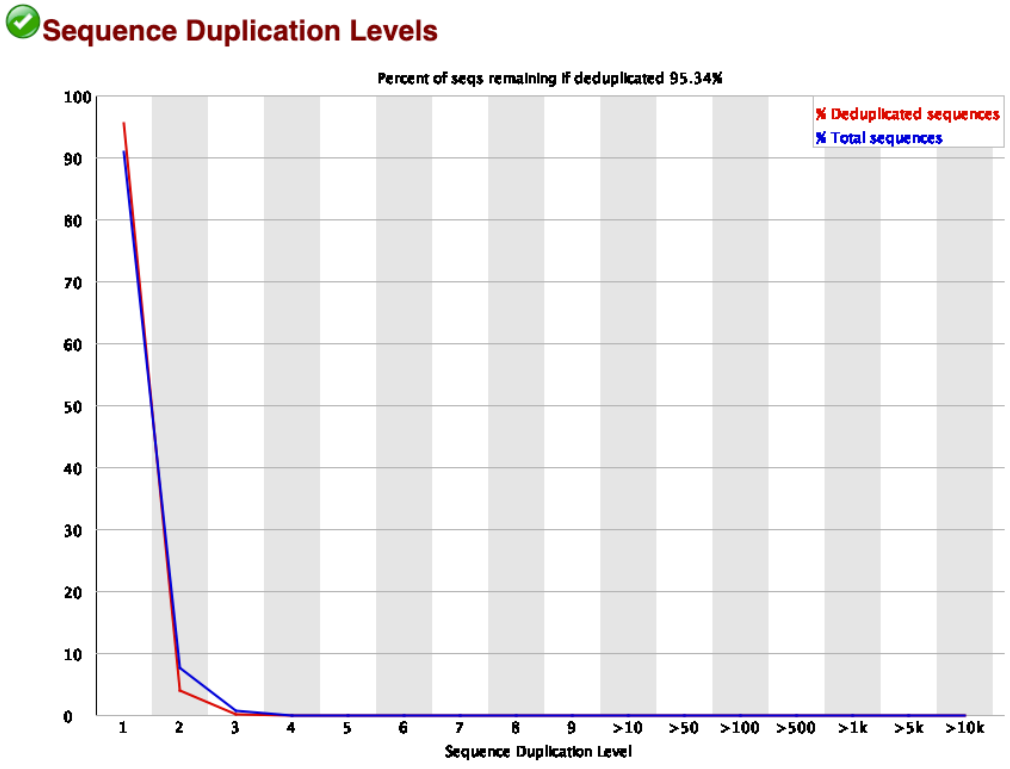
In a normal random library you would have a roughly normal distribution of GC content (a single peak) where the peak corresponds to the overall GC content of the underlying genome. An unusually shaped distribution could indicate a contaminated library or some other kind of bias in library prep.

6. Per base N content



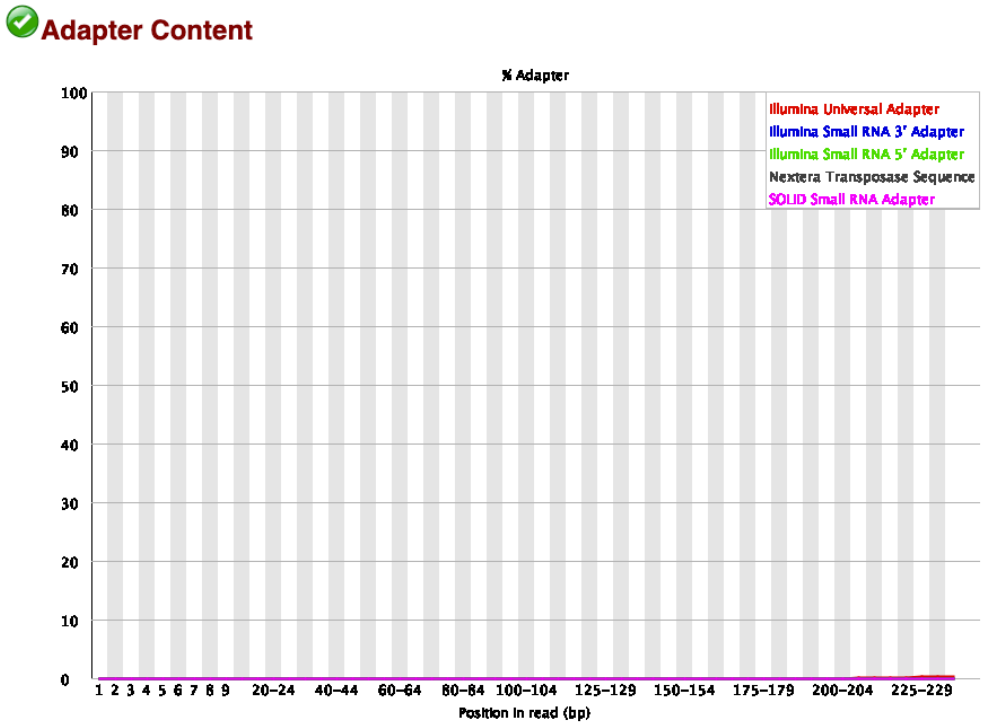
If a sequencer is unable to make a base call with sufficient confidence then it will normally substitute an N rather than a conventional base call. This module plots out the percentage of base calls at each position for which an N was called. It's not unusual to have a very low proportion of Ns especially nearer the end of reads. However, if proportion is higher that could cause problems in downstream analysis.

7. Sequence duplication levels



In a diverse library most sequences will occur only once in the final set. A low level of duplicating may indicate a very high level of coverage of the target sequence, but a high level of duplication is more likely to indicate some kind of enrichment bias (e.g., PCR over amplification). This module counts the degree of duplication for every sequence in a library and creates a plot showing the relative number of sequences with different degrees of duplications.

8. Adapter content



It is important to ensure that the sequence reads are not contaminated with adapter sequences used in library preparation. This module reports the abundance of various adapters used in sequencing. The plot above shows a cumulative percentage count of the proportion of the reads at each position that matches adapter sequences. Once a sequence has been seen in a read, it is counted as being present right through to the end of the read so the percentages you see will only increase as the read length goes on.

From all the above data metrics it appears that the sequence reads of the isolate ERR2093245 are of good quality and can be used to run the downstream analysis.

Quiz

Now in a similar manner, run fastqc tool on the sequence reads of another isolate DRR107117. The files are located within the folder cp2. Analyse the graphs generated by the tool and answer the following questions:

1. Do we need to trim the reads, if yes how many bases need to be trimmed from the end?
2. What is the GC content of the isolate?

3. Which of the two files have the lowest quality and state the reason for your conclusion?

Filtering and trimming of sequence reads

Once we have assessed the quality of the sequence reads, sometimes we spot bases with lower quality particularly in read2 files. These low quality bases need to be trimmed and here we are going to use another tool “Trimmomatic” for this purpose. The tool can only be used with command line and therefore is a quite popular choice for automated analysis pipelines.
Here, we will run trimmomatic on the isolate DRR107117 that we used in the quiz above. Run the following command (in a single line) to initiate the tool:

```
trimmomatic PE DRR107117_1.fastq.gz DRR107117_2.fastq.gz
DRR107117_1.trimmed.fastq.gz DRR107117_1un.trimmed.fastq.gz
DRR107117_2.trimmed.fastq.gz DRR107117_2un.trimmed.fastq.gz
ILLUMINACLIP:/home/manager/miniconda/share/trimmomatic-0.39-2/adapters/Next
eraPE-PE.fa:2:30:10 SLIDINGWINDOW:4:20 MINLEN:50
```

In the command above we have used a number of options which are explained below:
PE refers to paired-end sequencing (two read files); **DRR107117_1.fastq.gz** and **DRR107117_2.fastq.gz** refer to the two reads files; **DRR107117_1.trimmed.fastq.gz** and **DRR107117_1un.trimmed.fastq.gz** refer to the file names to store the paired and unpaired reads after trimming; in the same way next two files are for the read2 as for read; **ILLUMINACLIP:NexteraPE-PE.fa:2:30:10** specifies the parameters for identifying and trimming adapters used during sequencing; **SLIDINGWINDOW:4:20** specifies the parameters
Once successfully completed you will see a summary of the process on the terminal, similar to the image below:

```
Multiple cores found: Using 4 threads
Using PrefixPair: 'AGATGTGTATAAGAGACAG' and 'AGATGTGTATAAGAGACAG'
Using Long Clipping Sequence: 'GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAG'
Using Long Clipping Sequence: 'TCGTCGGCAGCGTCAGATGTGTATAAGAGACAG'
Using Long Clipping Sequence: 'CTGTCTCTTATACACATCTCCGAGCCCACGAGAC'
Using Long Clipping Sequence: 'CTGTCTCTTATACACATCTGACGCTGCCGACGA'
ILLUMINACLIP: Using 1 prefix pairs, 4 forward/reverse sequences, 0 forward only seque
nces, 0 reverse only sequences
Quality encoding detected as phred33
Input Read Pairs: 723761 Both Surviving: 622614 (86.02%) Forward Only Surviving: 7599
5 (10.50%) Reverse Only Surviving: 6515 (0.90%) Dropped: 18637 (2.58%)
TrimmomaticPE: Completed successfully
```

Here, we can see the proportion of reads remaining after trimming and filtering (86%) plus the proportion dropped as a result (2.58%). The filtered and trimmed reads (**DRR107117_1.trimmed.fastq.gz and DRR107117_2.trimmed.fastq.gz**) will be in the same folder and can be used for the downstream analysis steps.

Quiz

You can run fastqc on the filtered files (**DRR107117_1.trimmed.fastq.gz and DRR107117_2.trimmed.fastq.gz**) and check how have the metrics improved by comparing the reports before and after trimmomatic run.