REPORT:

Quality Control Summary of fastp Results

Objective

The goal of this analysis was to assess the sequencing quality of the 16S rRNA samples from the PRJEB61942 dataset using the fastp tool. The fastp_summary.R script was developed to automatically extract relevant quality metrics from the generated JSON and to produce both a summary and multiple visualizations.

Script overview

The fastp_summary.R script performs the following steps:

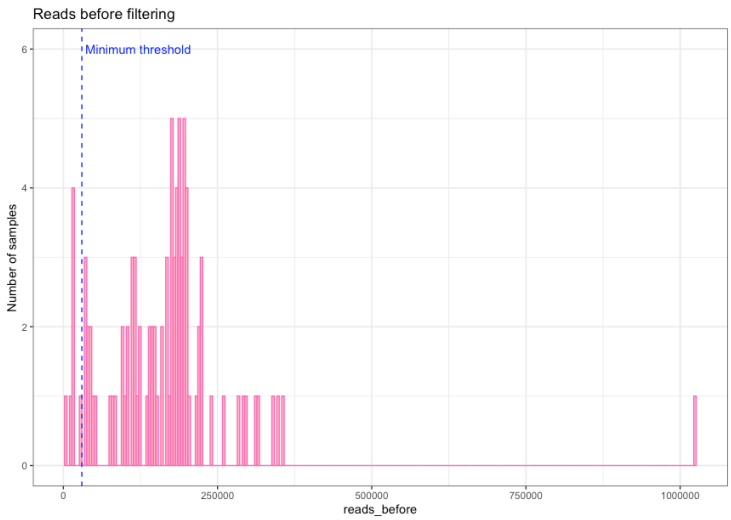
- Parsing all JSON files from the fastp_reports directory to extract:
  - the total number of reads before and after filtering
  - the quality scores: Q20 and Q30 rates
- Creating a summary table (fastp_summary_table.csv) including a quality status based on thresholds ("ok" or "low quality").
- Applying quality thresholds:
  - reads_before and reads_after >= 30,000
  - q20_rate >= 0,90
  - q30_rate >= 0,85
- samples that do not match all these criteria are marked "low quality" and the others as "ok"

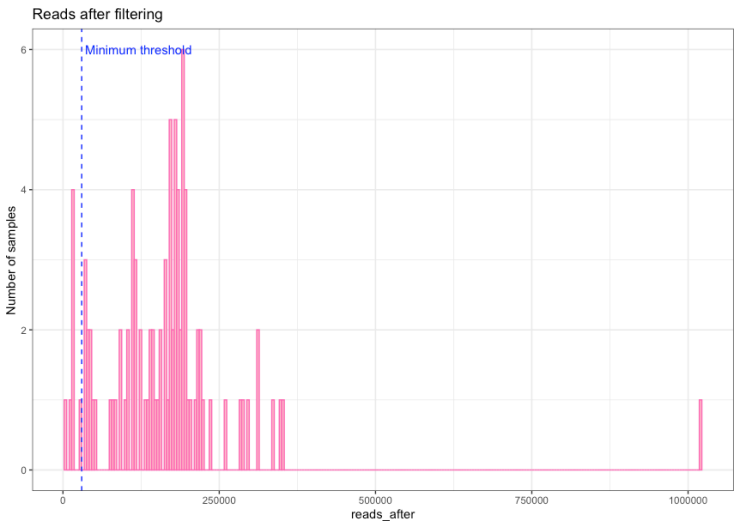
Visualizations

To better understand the quality of the samples processed with fastp, I generated several plots using R. These visualizations help confirm whether the trimming worked well and if the data is clean enough for the next steps, like using DADA2.
Here are the main plots generated to verify the quality of the data:

1. **histograms of read counts**

- reads before filtering
- reads after filtering

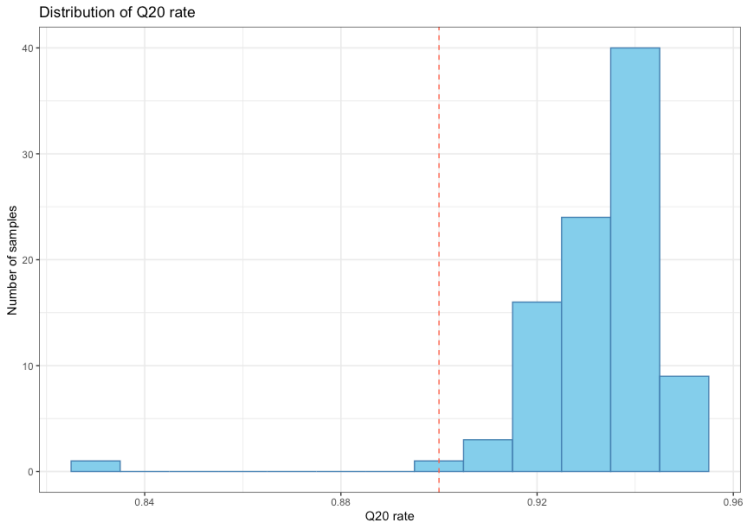


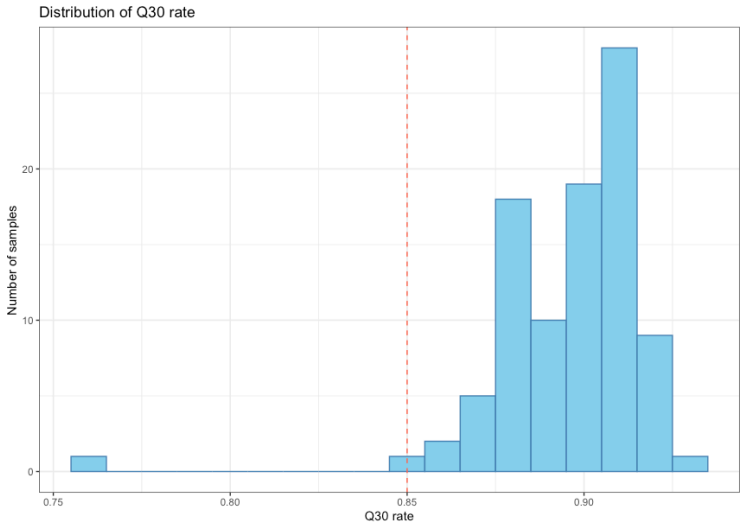


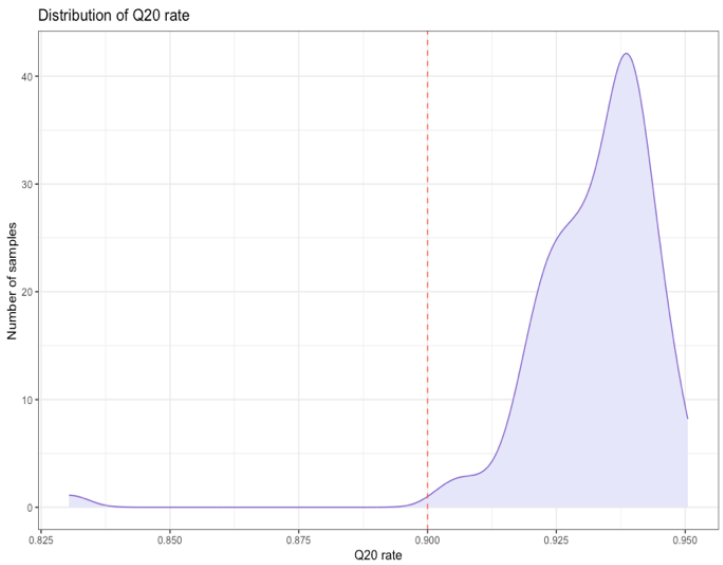
These histograms show the number of reads for each sample, before and after filtering with fastp. Each bar represents a group of samples with a similar number of reads.
For the two visualisations, we put a blue vertical dashed line at 30,000 reads to indicate the minimum threshold, which is the minimum number we want for good quality.
We can see that most of the samples are above this line, so they have more than 30,000 reads before and after filtering. So, the number of reads is good, and the samples are usable.

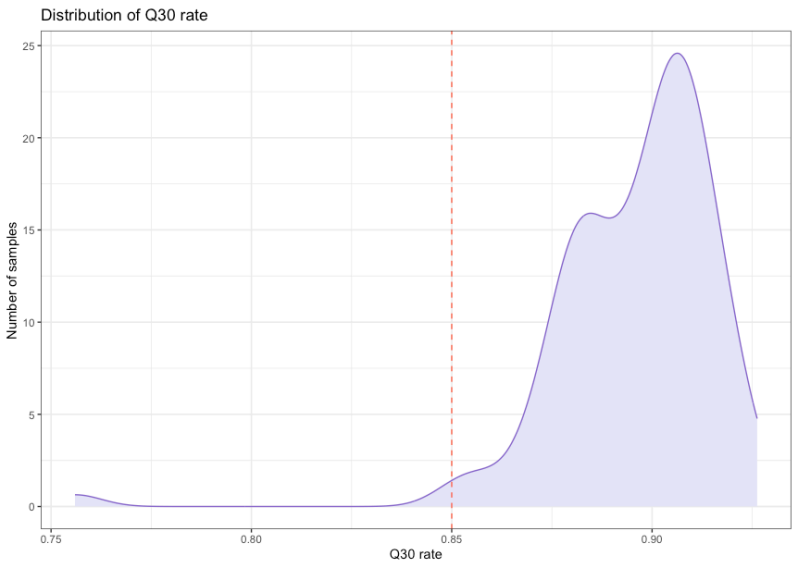
2. distribution of Q scores

- Q20 rate distribution
- Q30 rate distribution









We visualized the distribution of Q20 and Q30 scores across all samples using two types of plots: histograms and density plots. These quality scores indicate the proportion of bases in each sample that have a high sequencing quality.
Q20 means that the base call has 99% accuracy and Q30 means 99.9%.

In the histograms, each bar represents the number of samples that have a Q20 or Q30 rate in certain range. We added a dashed red line to indicate the minimum threshold we chose:

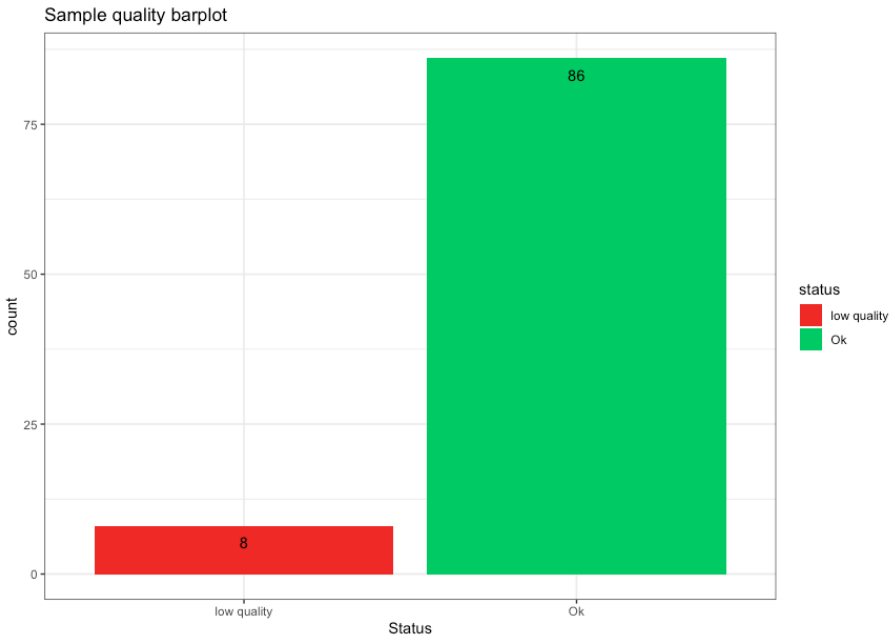
- 0.90 for Q20
- 0.85 for Q30

Most of the samples have values above the red line, which means their base quality is high enough to move forward in the analysis.

Then, we print the density plots that are similar, but they show a smooth curve instead of bars and it is better to visualize. The height of the curve tells how many samples are found at each quality level. These plots help to see the shape of the distribution more clearly.
Both Q20 and Q30 scores show a peak above the threshold which is a good sign.

So, these visualizations confirm that most of our data has high-quality reads, which is important before going further with DADA2.

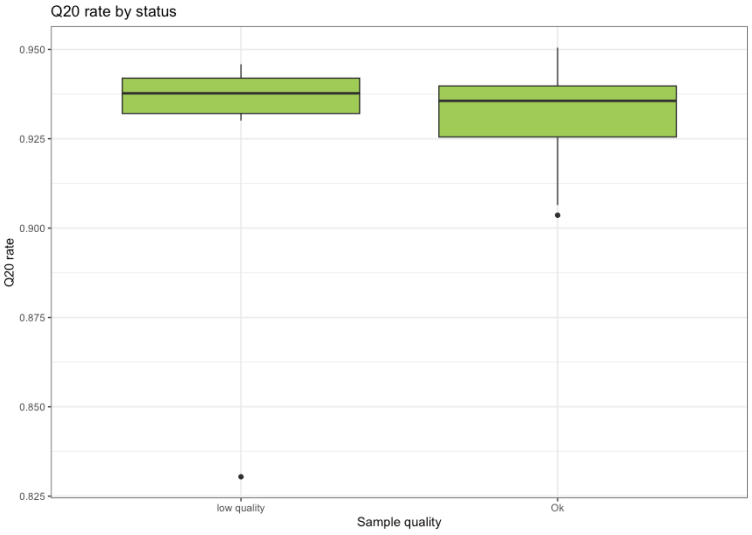
3. Barplot of sample status

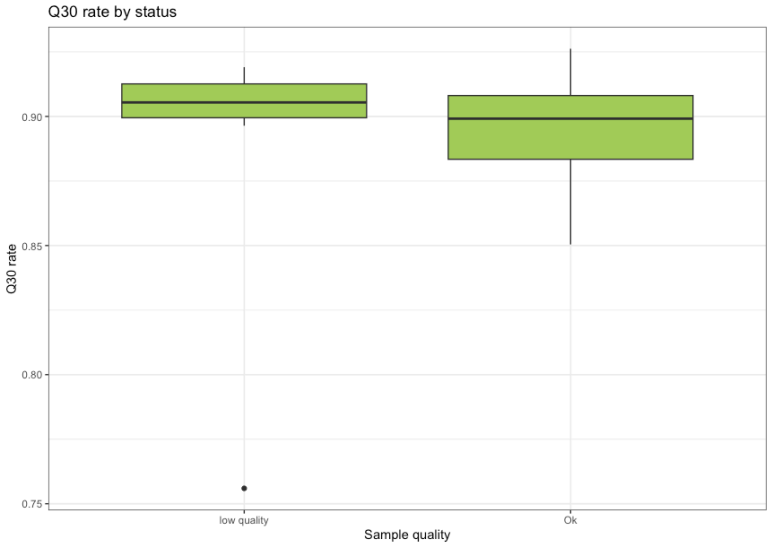


With this barplot, we want to show how many samples passed or failed our quality check.
The criteria of this low quality are based on the number of reads (it must be more than 30,000) and on the Q scores, as defined before.
Here, out of 94 samples, 8 were marked as low quality (red).
So, most of our dataset is usable.

4. Boxplots

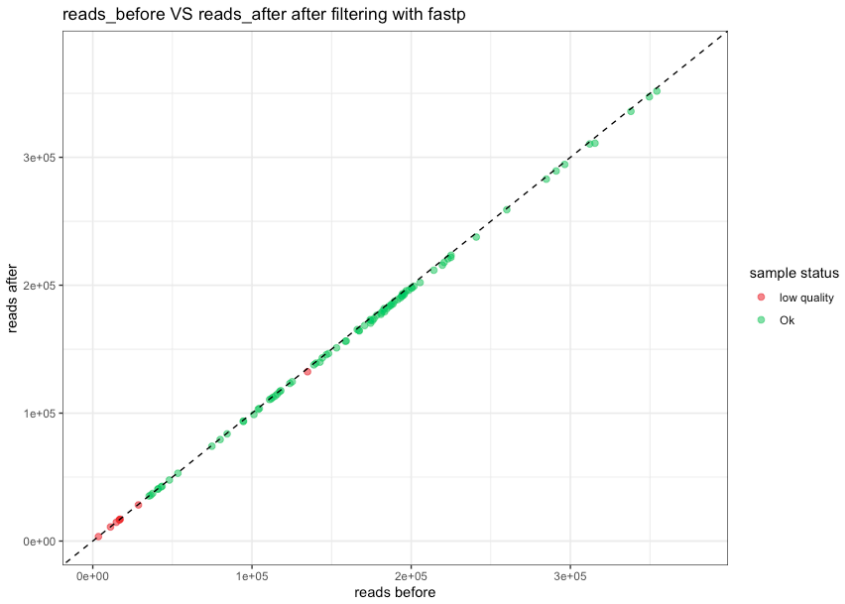
- Q20 rate by status
- Q30 rate by status





These boxplots show the distribution of Q20 and Q30 scores grouped by the quality status of the samples ("ok" vs "low quality").
For each group, the box shows where most of the data is concentrated (between the 1st and 3rd quartiles).
The line in the middle is the median and dots outside the box are outliers.
We can clearly see that low quality samples tend to have lower scores with some strong outliers.

5. Scatterplot

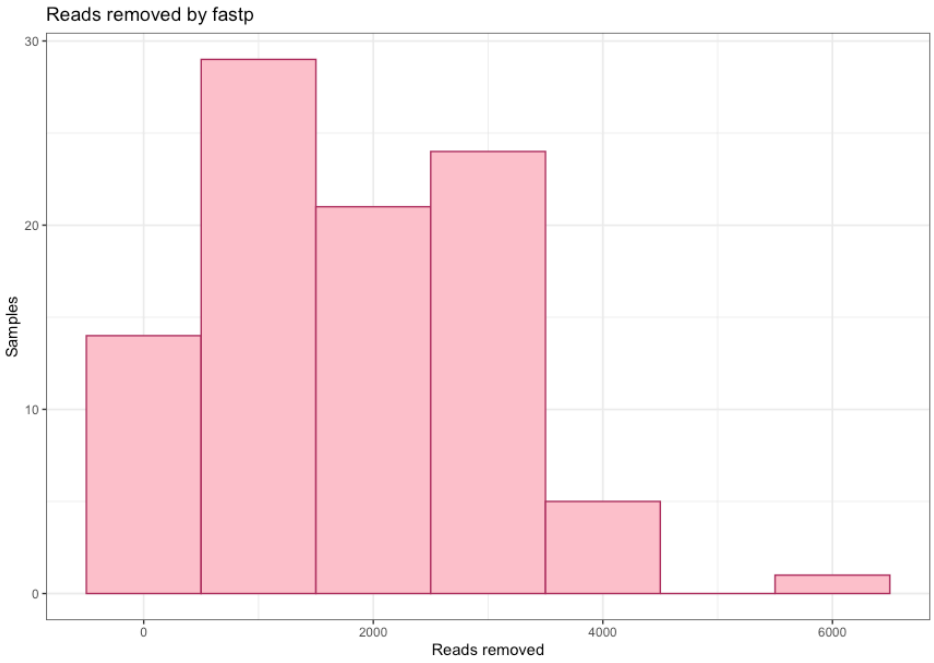


This plot compares the number of reads before and after fastp filtering. Here, each point represents one sample.

The dashed line represents the ideal situation: before=after because it means that no reads have been removed.

We can see that most points are a little bit below the line which is expected since fastp removes low quality reads.

6. Histogram of reads removed by fastp



The last visualization allows to see the number of reads that were removed during the filtering process.

Each bar represents the number of samples that had a similar number of reads removed.

Most samples had between 0 and 4000 reads removed.

A small number of samples ad a little bit more, up to about 6000 reads, but none lost more than that. This means that the filtering was not too aggressive and most of the original data was kept. The goal of fastp is to clean low quality and here it seems that only a small part of the data needed to be removed which is a good sign.

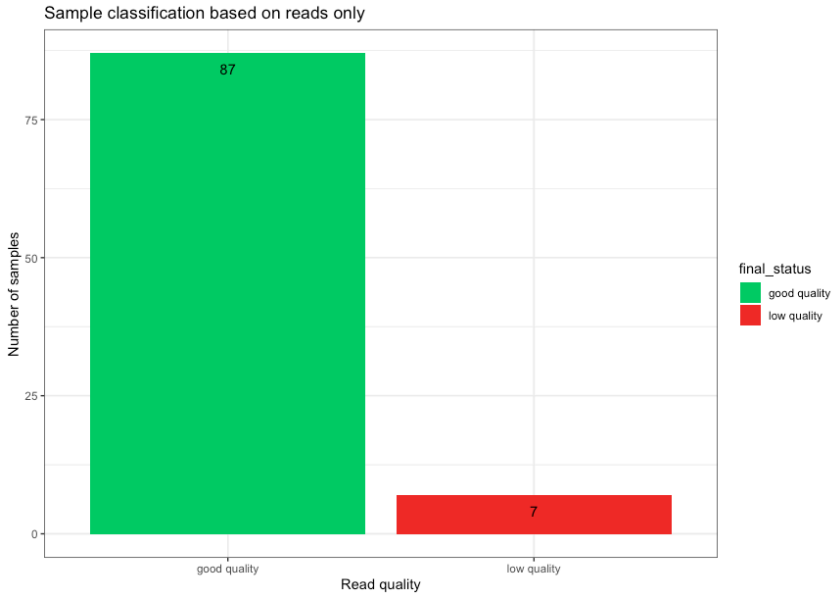
Final filtering criteria

To do the bioinformatics analysis, we need to remove the samples that are considered low quality.
Thanks to the previous visualizations, we noticed that most samples have good Q scores. The values of Q20 and Q30 are usually high and don't vary a lot between samples.
To double check whether it is necessary to exclude samples based on these Q scores, we printed the lowest values of Q20 and Q30 in our dataset.

```
> cat("Lowest Q20 scores: \n")
Lowest Q20 scores:
> print(lowest_q20[, c("sample_id", "q20_rate")])
    sample_id q20_rate
1 ERR13828018 0.830427
> cat("Lowest Q30 scores: \n")
Lowest Q30 scores:
> print(lowest_q30[, c("sample_id", "q30_rate")])
    sample_id q30_rate
1 ERR13828018  0.75597
```

We saw that even the lowest Q20 and Q30 rates stay close to the thresholds we had initially set. These values are still high enough to indicate good quality reads.
Because of this, we decided not to remove samples based on Q scores alone. Instead, we will only filter our samples with fewer than 30,000 reads after filtering. This threshold ensures that each sample has enough sequencing data for downstream analysis like DADA2, while keeping good quality data even if the Q score is a little bit below the cutoff.

To visualize this final filtering step, we created a barplot showing how many samples are considered "good quality" and how many are "low quality" based on the number of reads.



Out of 94 samples, 87 passed the filter and will be kept while 7 samples will be removed from the dataset.
To make it easier to know which are those 7 samples we must remove, we printed the list of them.

```
List of samples that are low quality based on the number of reads only:
     sample_id reads_after
20 ERR13827995       17072
24 ERR13827999       28240
34 ERR13828009        3490
39 ERR13828014       11038
49 ERR13828024       16674
52 ERR13828027       14666
53 ERR13828028       16590
```

So, we can remove them from the file “filereport_16S.tsv” with a bash command line: grep -v -E ‘ERR13827995|ERR13827999|ERR13828009|ERR13828014|ERR13828024|ERR13828027|ERR13828028' filereport_16S.tsv > filereport_16S_filtered.tsv