QAA_report

Luiza Ostrowski

2022-08-31

RNA-seq Quality Assessment Assignment - Bi 622

Files used:

- 27_4C_mbnl_S19_L008_R1_001.fastq.gz
- 27_4C_mbnl_S19_L008_R2_001.fastq.gz
- 28_4D_mbnl_S20_L008_R1_001.fastq.gz
- 28_4D_mbnl_S20_L008_R2_001.fastq.gz

The initial exploratory analysis showed the number of records in each file, as following:

File	Number of records
27_4C_mbnl_S19_L008_R1_001.fastq.gz	7226430
27_4C_mbnl_S19_L008_R2_001.fastq.gz	7226430
28_4D_mbnl_S20_L008_R1_001.fastq.gz	12428766
28_4D_mbnl_S20_L008_R2_001.fastq.gz	12428766

Table 1: Number of records in each fastq file.

This result is consistent with pair-ended data, where R1 and R2 have the same amount of records. This was also confirmed checking the headers for the first records.

Part 1 - Read quality score distributions

27_4C_mbnl_S19

After running FASTQC, I obtained the plots for the quality scores distributions for R1 and R2 (Fig. 1), as well as the per-base N content plots (Fig. 2).

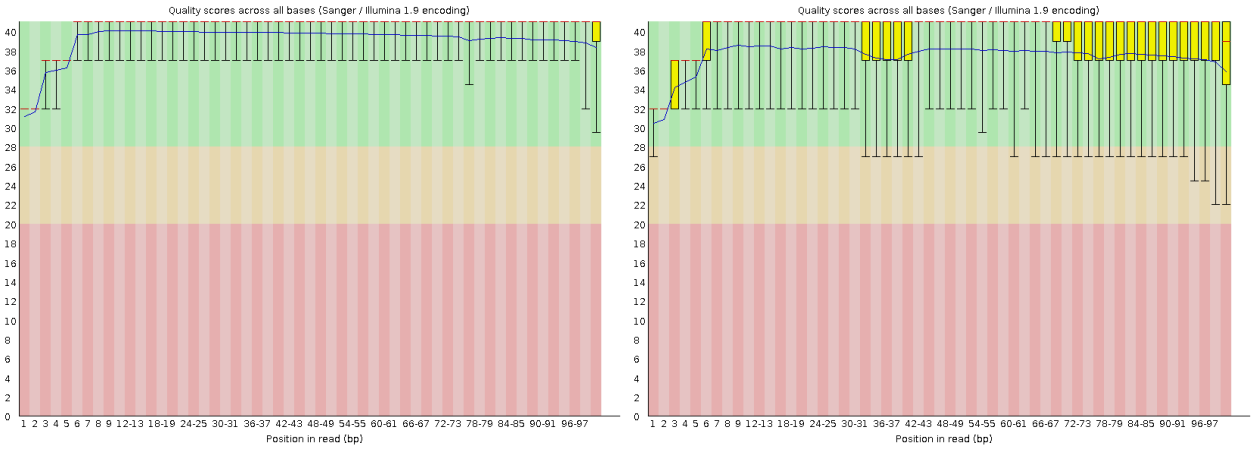

Fig. 1: Quality scores plots for R1 (left) and R2 (right) generated by FASTQC. We can observe that R2 has worse quality in comparison to R1, which can be caused by several reasons. For example, the R2 usually stays in the flowcell for more time before being sequenced, which can degrade RNA and decrease overall quality.

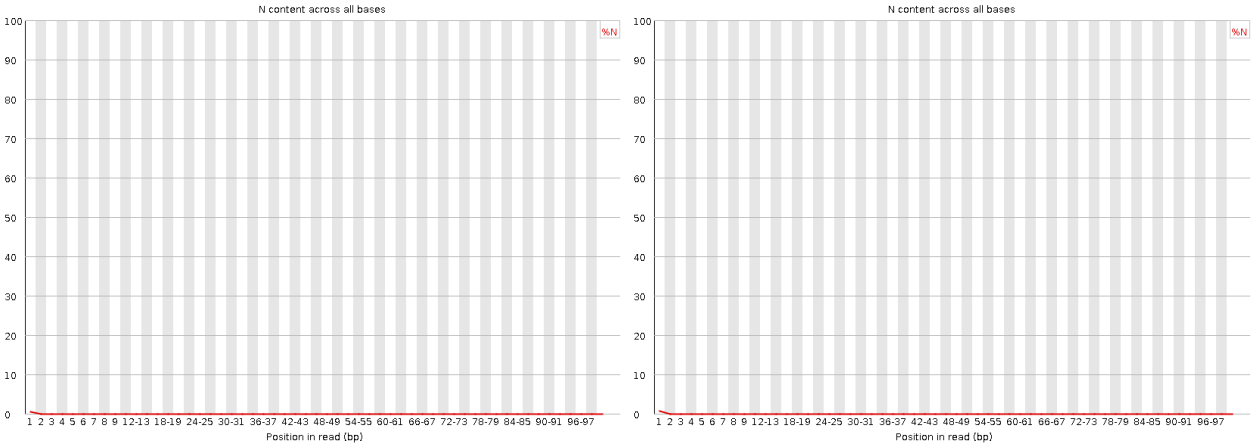

Fig. 2: N content plots for R1 (left) and R2 (right) generated by FASTQC. For both reads, N content was higher at the beginning of the reads, which is consistent with the qualities plots since the qualities scores at this position were also low.

The plots obtained after running the Demultiplex script were similar to the ones obtained with FASTQC. Both had a low quality score in the beginning of the read and R2 had a lower quality in comparison with R1 (Fig. 3).

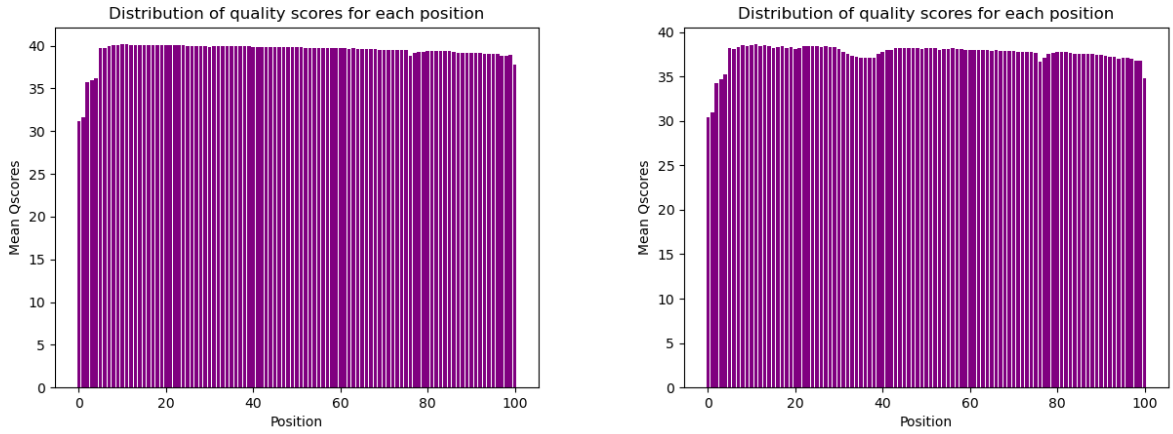

Fig. 3: Distribution of quality scores for each position of R1 (left) and R2 (right).

28_4D_mbnl_S20

After running FASTQC, I obtained plot for the quality scores distributions for R1 and R2 (Fig. 4), as well as the per-base N content plots (Fig. 5).

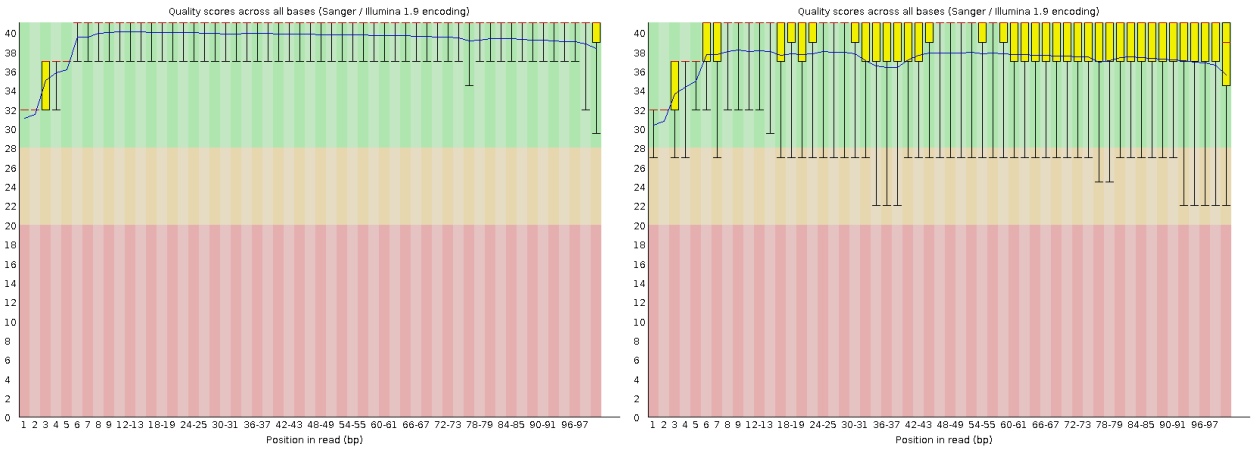

Fig. 4: Quality scores plots for R1 (left) and R2 (right) generated by FASTQC. We can also observe that R2 has worse quality in comparison to R1.

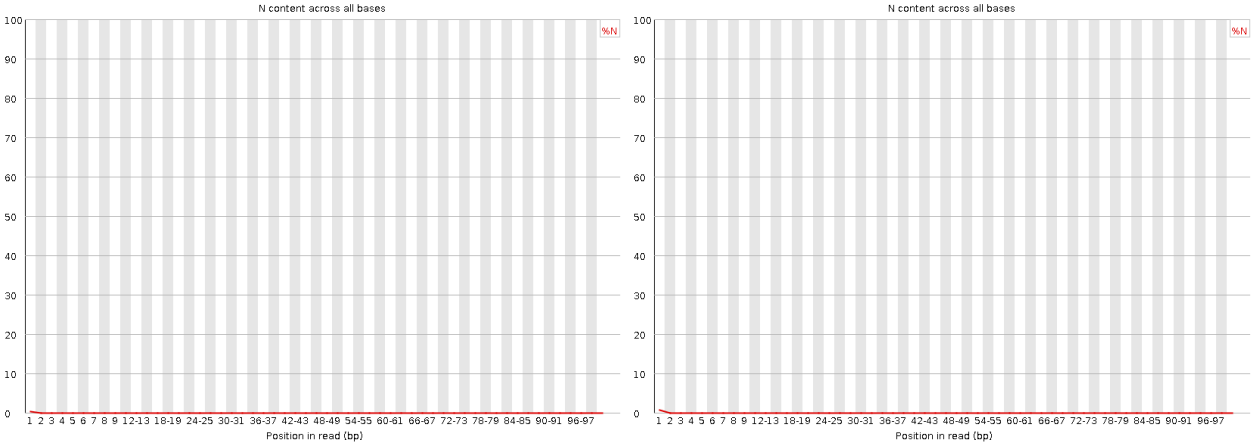

Fig. 5: N content plots for R1 (left) and R2 (right) generated by FASTQC. For both reads, N content was higher at the beginning of the reads, which is consistent with the qualities plots because the qualities scores at this position were also low.

The plots generated by Demultiplex script were also similar to the ones obtained with FASTQC. Both had a low quality score in the beginning of the read and R2 had a lower quality in comparison with R1 (Fig. 6).

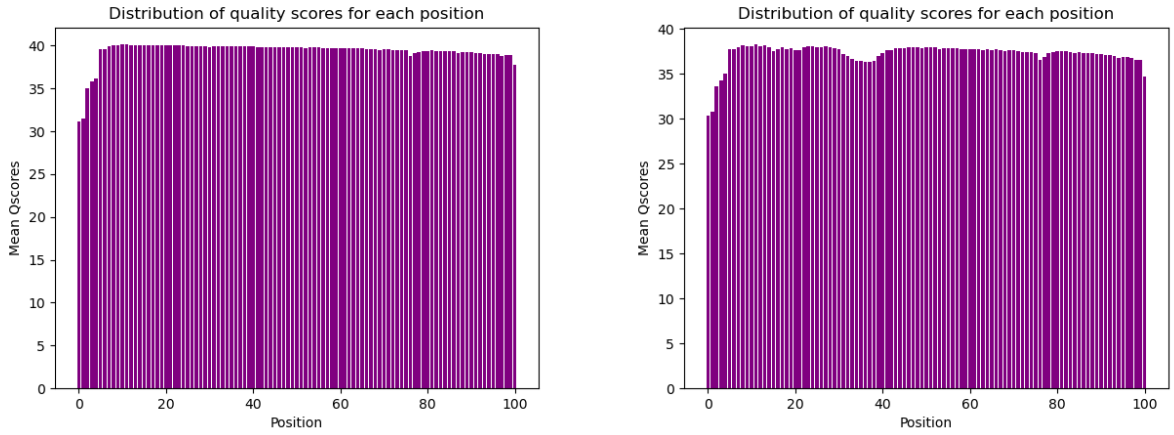

Fig. 6: Distribution of quality scores for each position of R1 (left) and R2 (right).

For all the files the running time for FASTQC was 296 seconds on average per file. For the demultiplex script it took 319 seconds per file on average. They were not very different, but if we consider bigger files, FASTQC would perform better. Also, FASTQC can handle multiples files at once, which also increases the productivity.

The overall quality of the libraries was good, especially for R1, with the majority of the reads with high quality scores.

Part 2 – Adapter trimming comparison

After running FASTQC on the files, the overrepresented sequences, suggested a TruSeq adapter. Looking at the Illumina website (https://support-docs.illumina.com/SHARE/AdapterSeq/illumina-adapter-sequences.pdf), the most common TruSeq adapters are:

- R1: AGATCGGAAGAGCACACGTCTGAACTCCAGTCA
- R2: AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT

Both of them were found in the reads, using Unix commands, as following:

command:

zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/27_4C_mbnl_S19_L008_R1_001.fastq.gz | grep -c “AGATCGGAAGAGCACACGTCTGAACTCCAGTCA”

output:

132794

command:

zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/27_4C_mbnl_S19_L008_R1_001.fastq.gz | grep -c “AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT”

output:

0

command:

zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/27_4C_mbnl_S19_L008_R1_001.fastq.gz | grep -c "AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT"

output:

0

command:

zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/27_4C_mbnl_S19_L008_R2_001.fastq.gz | grep -c "AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT"

output:

135250

As expected, the R1 adapter was only found in R1 files and the R2 adapter was only found in R2 files.

Cutadapt outputs

For both files, the proportion of reads trimmed was slightly higher for R2 in comparison to R1.

The proportion of reads trimmed for 27_4C_mbnl_S19:

- Total read pairs processed: 7,226,430
- Read 1 with adapter: 751,117 (10.4%)
- Read 2 with adapter: 803,568 (11.1%)
- Pairs written (passing filters): 7,226,430 (100.0%)

The proportion of reads trimmed for 28_4D_mbnl_S20:

- Total read pairs processed: 12,428,766
- Read 1 with adapter: 743,440 (6.0%)
- Read 2 with adapter: 841,389 (6.8%)
- Pairs written (passing filters): 12,428,766 (100.0%)

Read length distribution

For both files, we can observe that the length distribution varies between R1 and R2 (Fig. 7). The R2 apparently was more trimmed than R1, so R2 have a larger amount of shorter reads. This was expected since R2 had a worse overall quality, so both poor quality reads and poor quality bases were filtered after Trimmomatic.

Fig. 7: Read length distribution plot for 27_4C_mbnl_S19 (left) and 28_4D_mbnl_S20 (right). The overlapping data is showed in the darker color.

Part 3 – Alignment and strand-specificity

The number of mapped and unmapped reads for **27_4C_mbnl_S19** was:

Type	Counts
Mapped	13316430
Unmapped	432638

Table 2: Counts for mapped and unmapped reads.

The number of mapped and unmapped reads for **28_4D_mbnl_S20** was:

Type	Counts
Mapped	22653883
Unmapped	791959

Table 3: Counts for mapped and unmapped reads.

Strand-specificity

After counting reads that map to features with htseq-count, I analyzed the files to counts how many sequences aligned with genes. I did that as following:

command:

cat htseq-count_27_4C_reverse.txt | grep "^ENS" | cut -f 2 | awk '{s+=$1}END{print(s)}'

output:

5686831

command:

cat htseq-count_27_4C_stranded.txt | grep "^ENS" | cut -f 2 | awk '{s+=$1}END{print(s)}'

output:

281914

command:

cat htseq-count_28_4D_reverse.txt | grep "^ENS" | cut -f 2 | awk '{s+=$1}END{print(s)}'

output:

9700449

command:

cat htseq-count_28_4D_stranded.txt | grep "^ENS" | cut -f 2 | awk '{s+=$1}END{print(s)}'

output:

442000

The data is not strand specific, because the strand reverse option has a 20 times higher count in comparison with the stranded option. For both files, around 95% of genes aligned to a sequence in the reverse mode, meaning that the reads mapped to the same strand as the feature.