Report QAA

Kyra Lindley

2023-09-12

Part 1

Objective: In this section, I employed FASTQC via the command line to conduct quality control assessments on raw sequencing data obtained from high-throughput sequencing pipelines. Subsequently, I conducted additional quality control checks, including an analysis of quality score distribution plots. To conclude, I will provide comments on the overall data quality and offer recommendations regarding the suitability of these data for further analysis.

Initial Quality Control

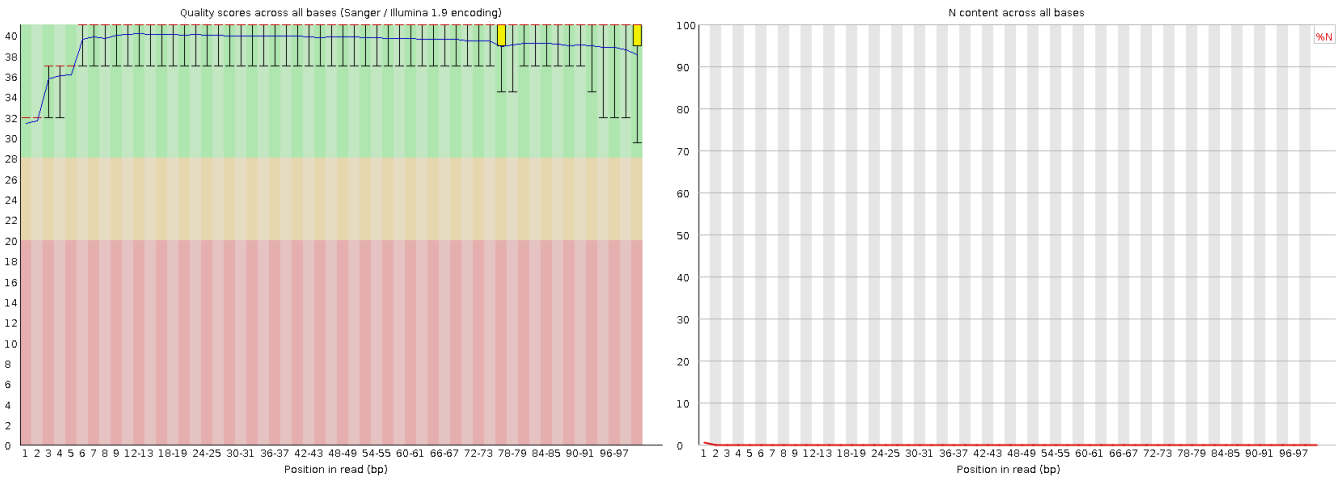

Fig 1. Read 1 21_3G: Employed FASTQC to produce per-base quality score and per-base N content plots. (a) Observed lower q-scores around the beginning of the reads (b) The low N content aligns consistently with the per-base quality score readings, while a minor peak is evident at the start of the reads, correlating with lower quality scores in that region.

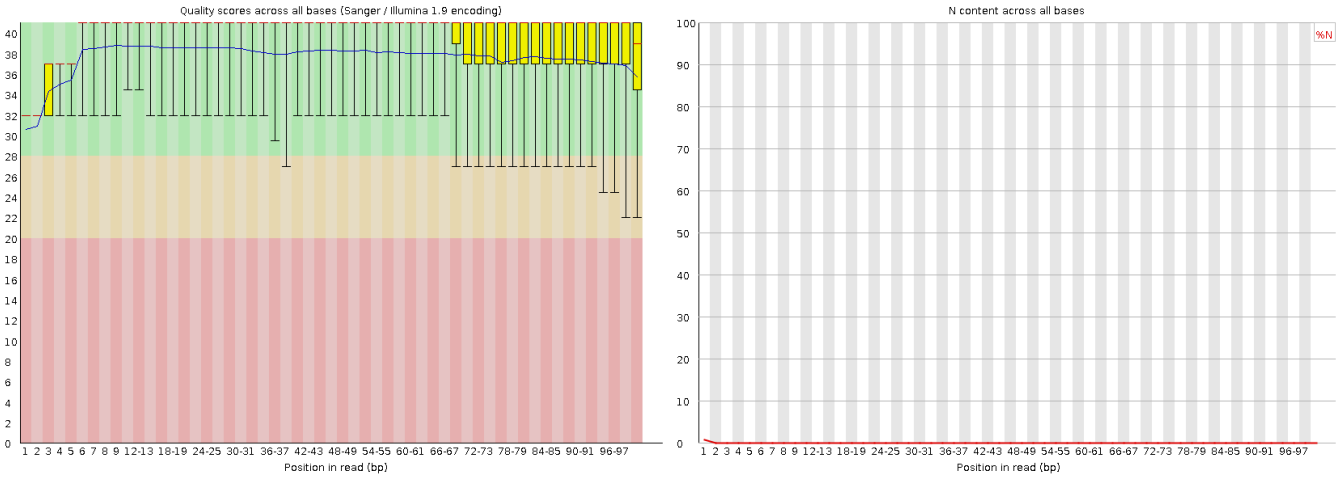

Fig 2. Read 2 21_3G: Employed FASTQC to produce per-base quality score and per-base N content plots. (a)Noticed lower quality scores in Read 2 compared to Read 1, which is expected due to Read 2's longer exposure to the sequencer, potentially resulting in more sequencing errors. While the error bars are slightly larger for Read 2, it's worth noting that the overall quality scores remain relatively high. Additionally, a similar trend of lower quality scores at the beginning of Read 2 was observed, mirroring the pattern observed in Read 1.(b)Minimal presence of per-base N content is noted, and a subtle peak towards the initial segment of the reads aligns consistently with lower quality scores in that region.

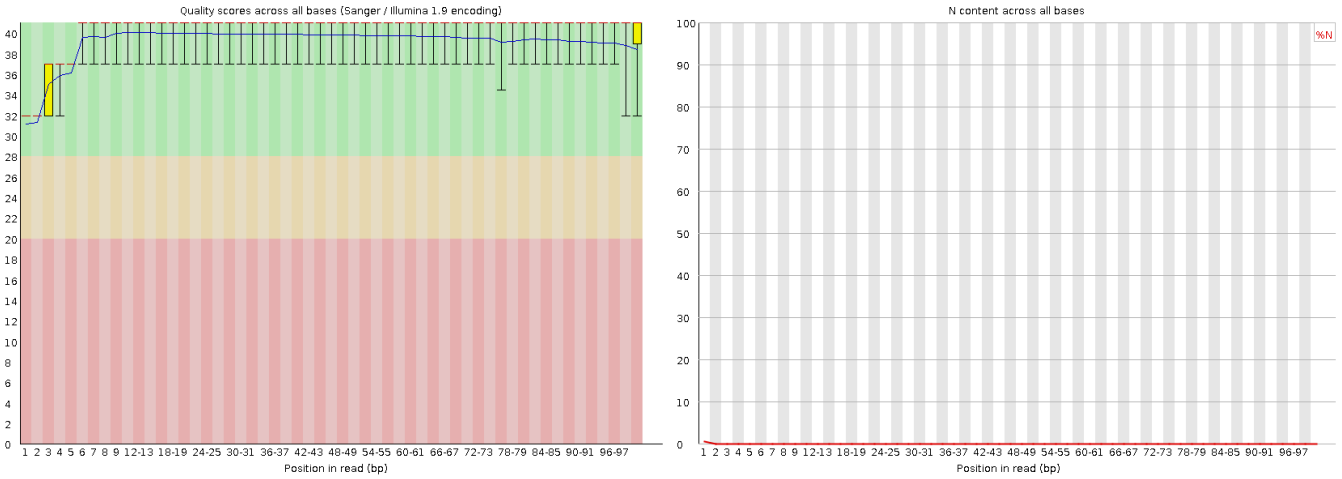

Fig 3. Read 1 34_4H: Employed FASTQC to produce per-base quality score and per-base N content plots.(a)Observed overall high quality scores, at the beginning of the reads the quality is slightly lower.(b)The low N content aligns consistently with the per-base quality score readings. Furthermore, a minor peak at the beginning of the reads corresponds to the lower quality scores observed in this region.

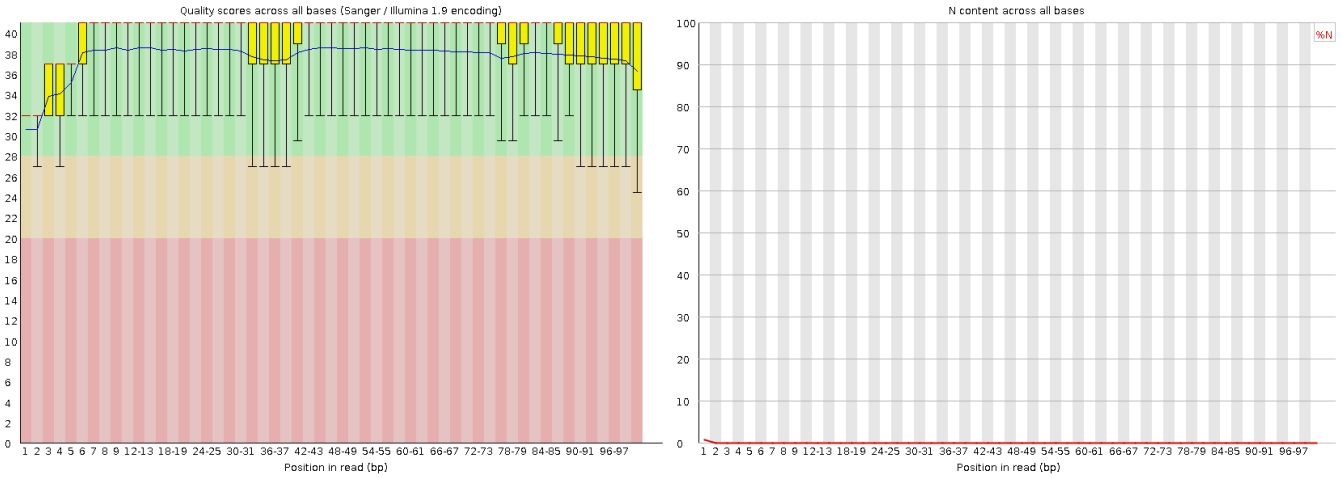

Fig 4. Read 2 34_4H: Employed FASTQC to produce per-base quality score and per-base N content plots.(a)Significantly reduced quality scores in comparison to Read 1, coupled with larger error bars, can be attributed to the prolonged sequencing time of Read 2.(b)Minimal N content is observed, accompanied by a subtle peak at the beginning of the reads, which corresponds to lower quality scores in this region

Further Quality Score For the Fig 5-6: Employed phred_encoding.py to plot the quality score distribution per base pair.

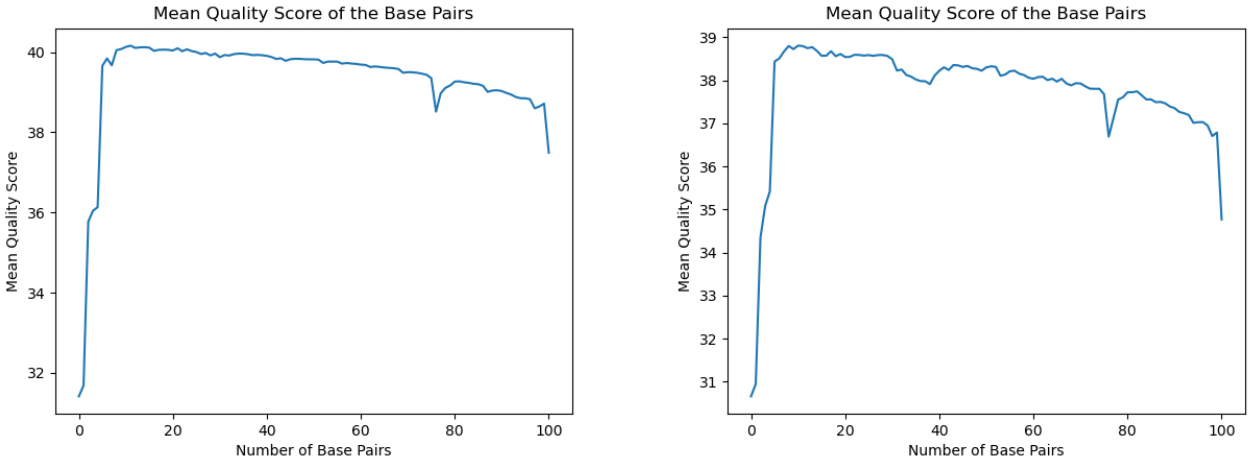

Fig 5. 21_3G (a) per-base quality score distribution for Read 1. (b) per-base quality score distribution for Read 2.

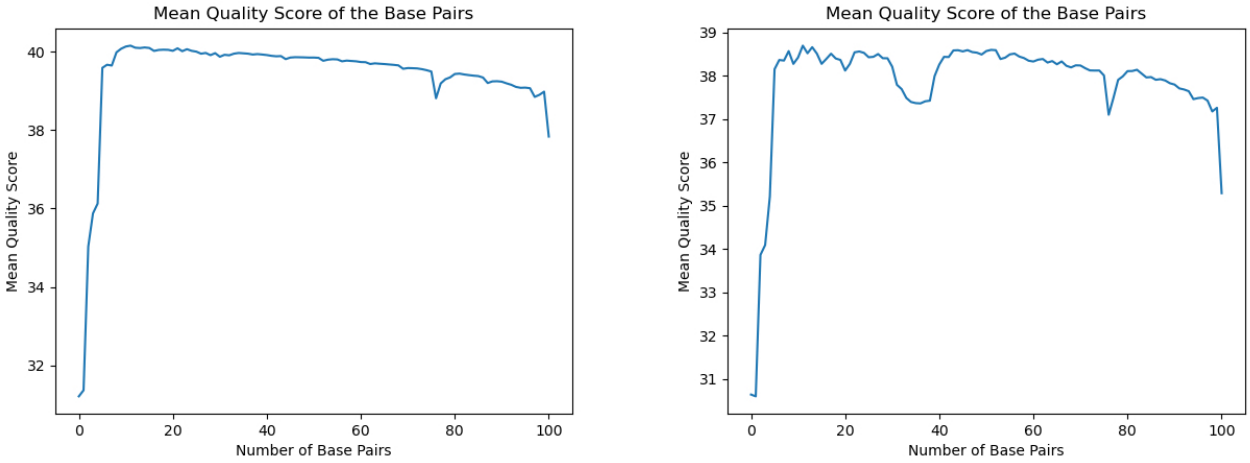

Fig 6. 34_4H (a) per-base quality score distribution for Read 1. (b) per-base quality score distribution for Read 2.

Comparing Output from FASTQC & length_distribution.py

The phred_encoding.py script generated a graph that displays more pronounced dips in quality score. Unlike FASTQC, which produces a plot spanning quality scores from 0 to 40, graphs from phred_encoding.py are specifically zoomed in on the distribution within the range of 40 to 32 quality scores. As a result, the dip in quality scores is not as readily apparent in their graph as it is in those from phred_encoding.py.

Overall data quality recommendation

The slightly lower quality scores surrounding the beginning of reads can be explained by the following: Lower quality scores are often observed at the beginning of RNA-seq reads, and this is an expected outcome given the nature of RNA sequencing. The initial step of the library preparation process involves reverse transcription, a stage known to introduce errors and occasional template switching. Additionally, during RNA fragmentation in the library preparation, there can be a bias towards the 3' end of the RNA molecules, resulting in shorter fragments at the 5' end. These shorter fragments are more susceptible to errors, ultimately contributing to the lower quality scores observed in this region.

With that in mind, it is advisable to continue the investigation into the data. All quality score graphs depict scores exceeding 30, which signifies a base call accuracy of 99.99%. The limited per-base N distribution instills confidence in the viability of proceeding with the data analysis.

Part 2

Objective: In this section, the objective was to utilize Cutadapt for the removal of adapter sequences from the provided files and report the proportion of trimmed reads. Following that, Trimmomatic was employed to perform quality trimming on the reads. Subsequently, the read length distributions for Read 1 and Read 2 were generated. Finally, an analysis was conducted to determine whether the adapter-trimming rates are expected to be consistent or different between the two reads.

Cutadapt Proportions

Samples	Read1	Read2
21_3G	6.6%	7.4%
34_4H	9.1%	9.8%

Read Length Distribution Using the command line to find the length distribution of the reads. The following command outputted a file with the length and occurrences of each length: ''zcat

trimmed_saved_data.fastq.gz | grep "^@" -A1 | grep -v "^@" | grep -v "^@" | grep -v "^–" | awk'{print length($0)}' | sort | uniq -c | sort -n > outfile name'''

Then the python script, length_distribution.py to parse through the output file from the bash command. It saved the lengths as a key, and the value was its occurrences. This dictionary was used to plot the read length distribution for both Read 1 and Read 2.

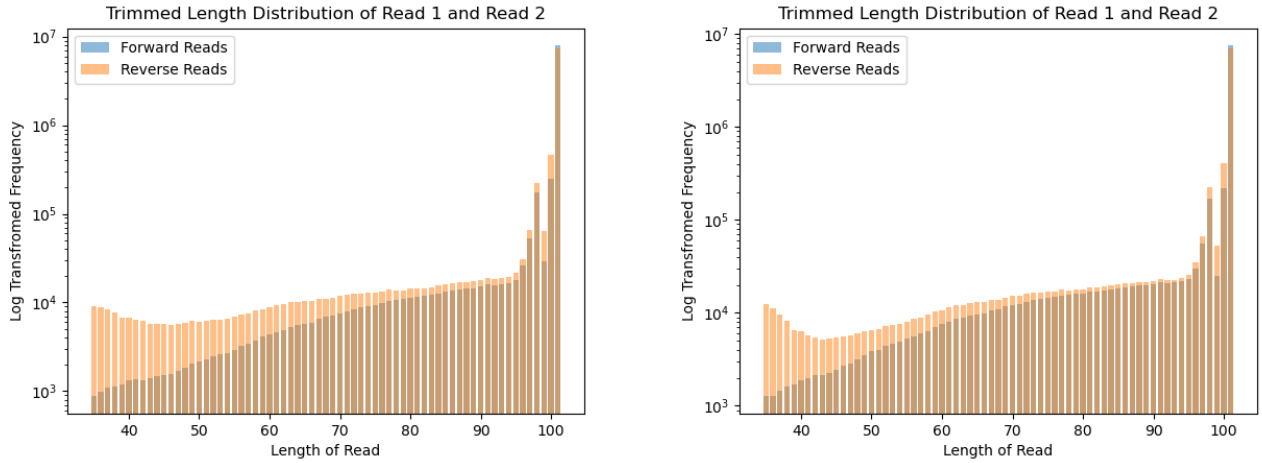

Fig 7 Read length distribution for Read 1 and Read 2 per sample. (a) Trimmed length distribution for 21_3G (b) Trimmed length distribution for 34_4H

I anticipate comparable adapter trimming rates for both Read 1 and Read 2. This expectation is based on the high overall quality scores for both reads, with Read 2 showing slightly lower scores leading to a minimal difference of less than 1% in the proportion of trimmed reads. Additionally, since the adapter content represents less than 10% of the reads, there is no strong indication that one read would require significantly more trimming due to concentrated adapters. However, Figure 7 reveals that the reverse strand was trimmed more because it has a larger distribution of shorter reads.

Part 3

Objective: In this final section, the aim was to construct a mouse genome database by employing STAR and Ensemble release 110, thus creating an alignment database. Subsequently, this database was utilized to align the reads using a splice-aware aligner, resulting in the generation of two SAM files. The task also involved reporting the counts of mapped and unmapped reads, employing htseq-count for read-to-feature mapping. Lastly, the analysis aimed to determine the strand specificity of the RNA-seq data.

samples	mapped	unmapped	total	Percentage of Read Mapped
21_3G	17061162	645462	17706624	96.35469
34_4H	16822690	483592	17306282	97.20569

samples	fw	rv
21_3G	3.78%	81.11%
34_4H	5.50%	83.37%

In HTSeq, the '-stranded' argument offers three options. When used without an argument, it informs us whether a read has aligned to any feature, regardless of whether it aligns to the same or opposite strand. If we specify 'stranded=yes' as the argument, it implies that for a paired-end read to be counted, the

first read must align to the same strand as the feature, while the second read must align to the opposite strand. Conversely, when we use 'stranded=reverse' as the argument, the rules are inverted compared to 'stranded=yes.'

An analysis of the files using 'stranded=yes' revealed that only 3.78% (21_3G), 5.5% (34_4H) of the reads mapped accordingly, while 'stranded=reverse' resulted in 81.11% (21_3G), 83.37% (34_4H) of reads mapping in a stranded manner. These results make me conclude that the RNA-seq is indeed strand specific.