QAA Write-up

Layaa Sivakumar

2023-09-13

Contents

Part 1 – Read quality score distributions

FASTQC Plots

4_2C_mbnl_S4

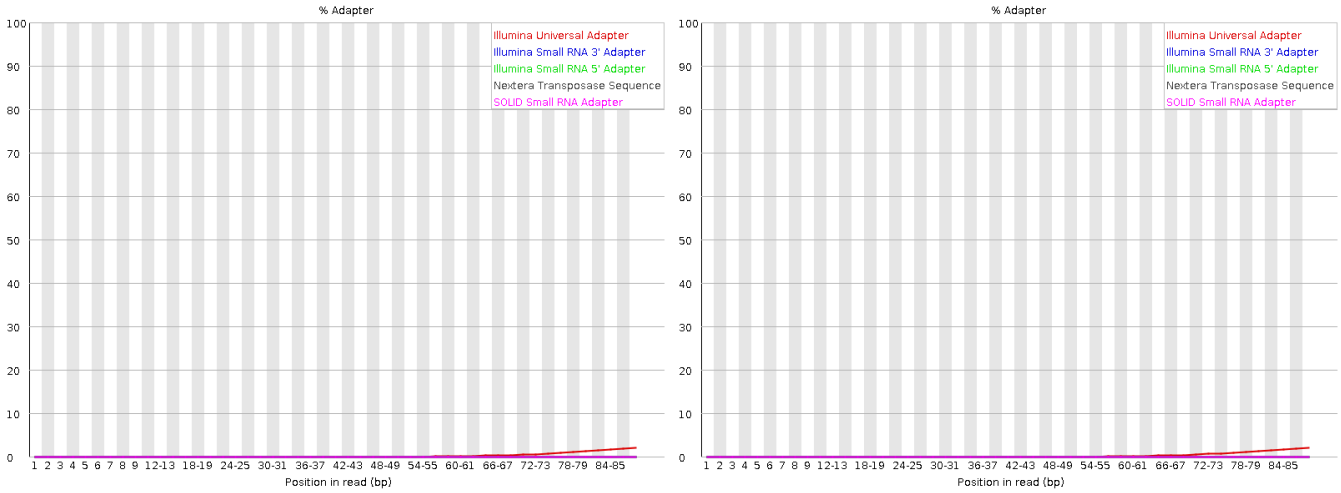

Figure 1: Percent adapter content for read1 (left) and read2 (right).

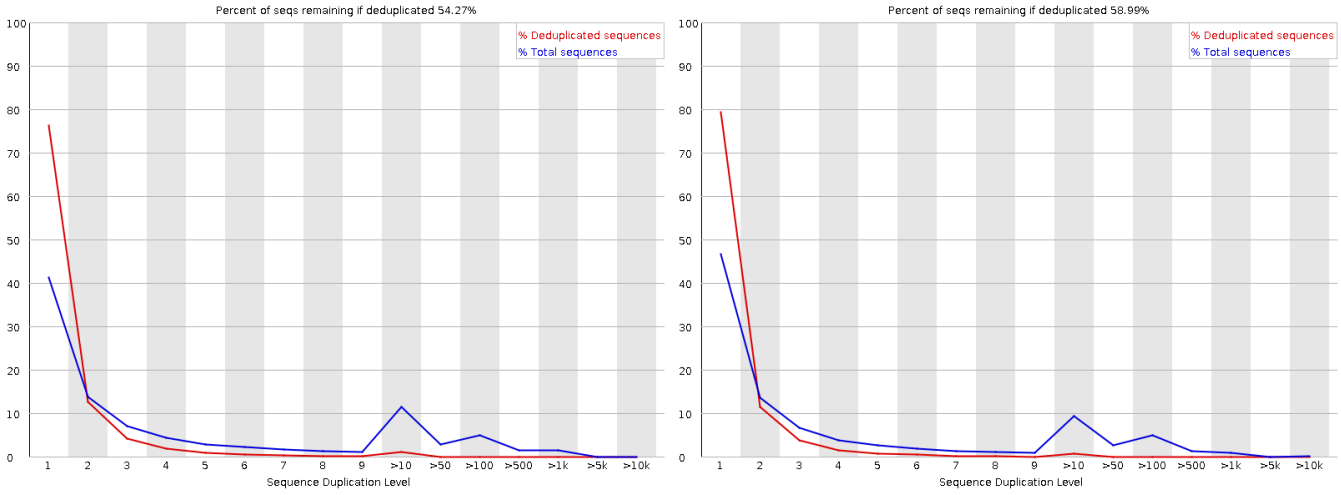

Figure 2: Duplication levels displayed by the reads for read1 (left) and read2 (right). The blue line represents duplication in the raw data while the red line represents duplication in the data when they have been deduplicated.

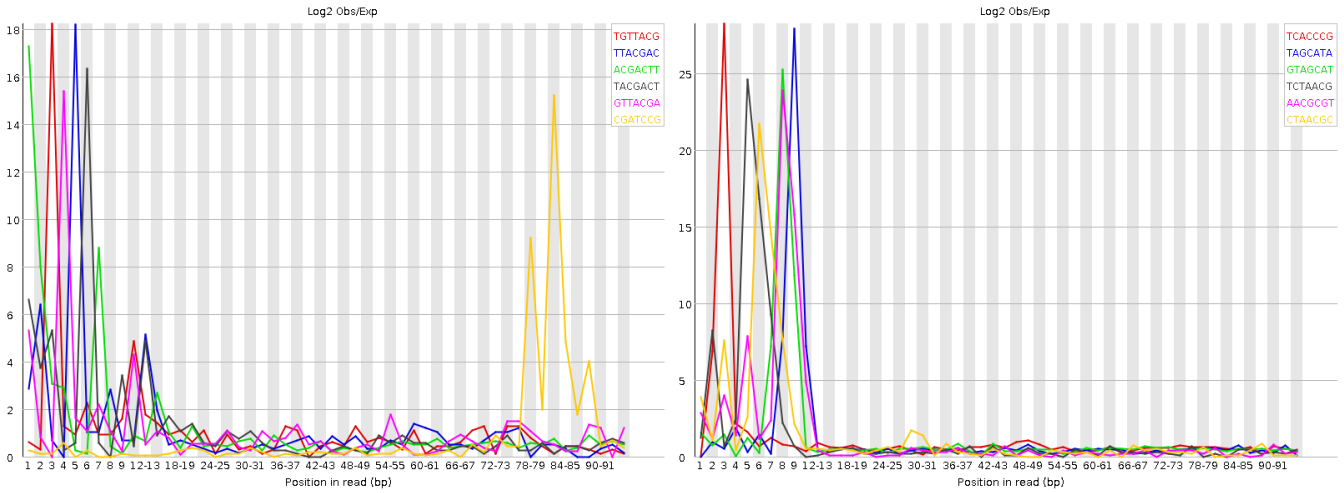

Figure 3: Kmer enrichment at each position along a read for Read1 (left) and Read2 (right).

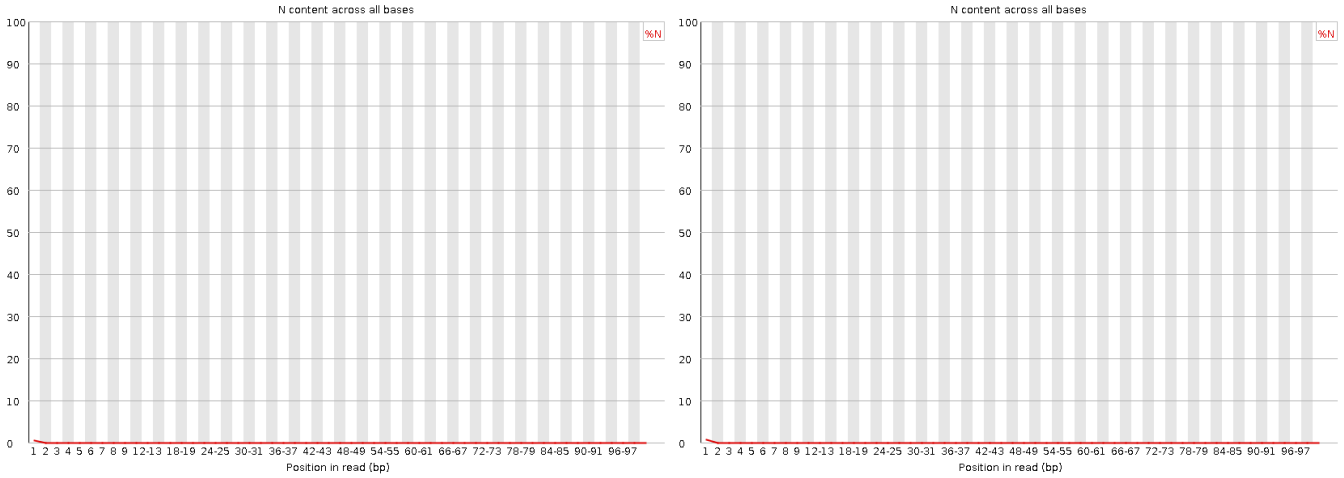

Figure 4: The number of 'N' base calls at every position in the read for Read1 (left) and Read2 (right).

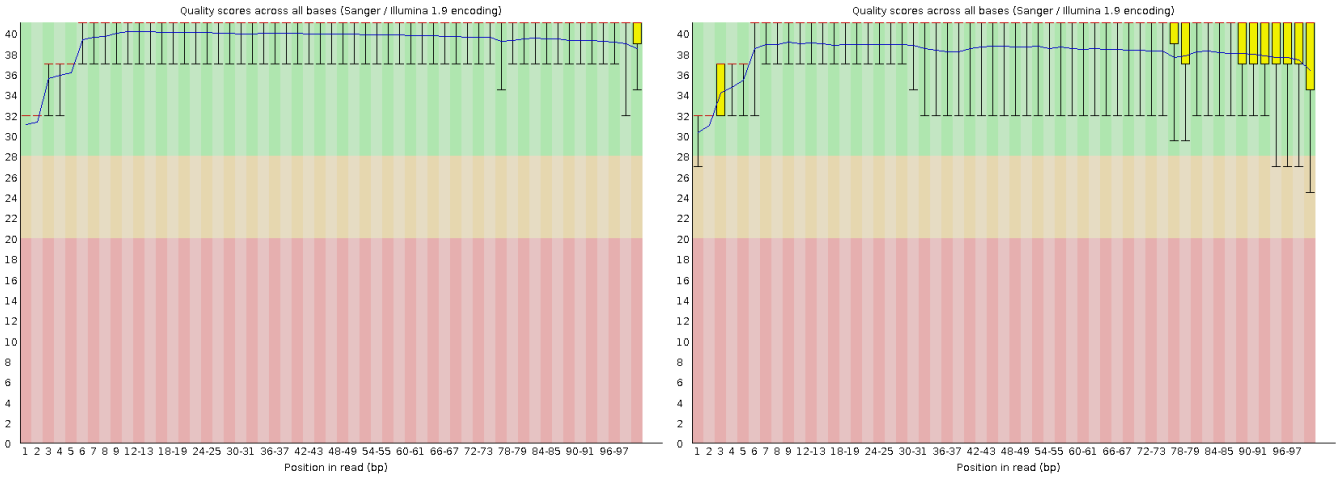

Figure 5: Mean quality score per base position across the sequence for Read1 (left) and Read2 (right).

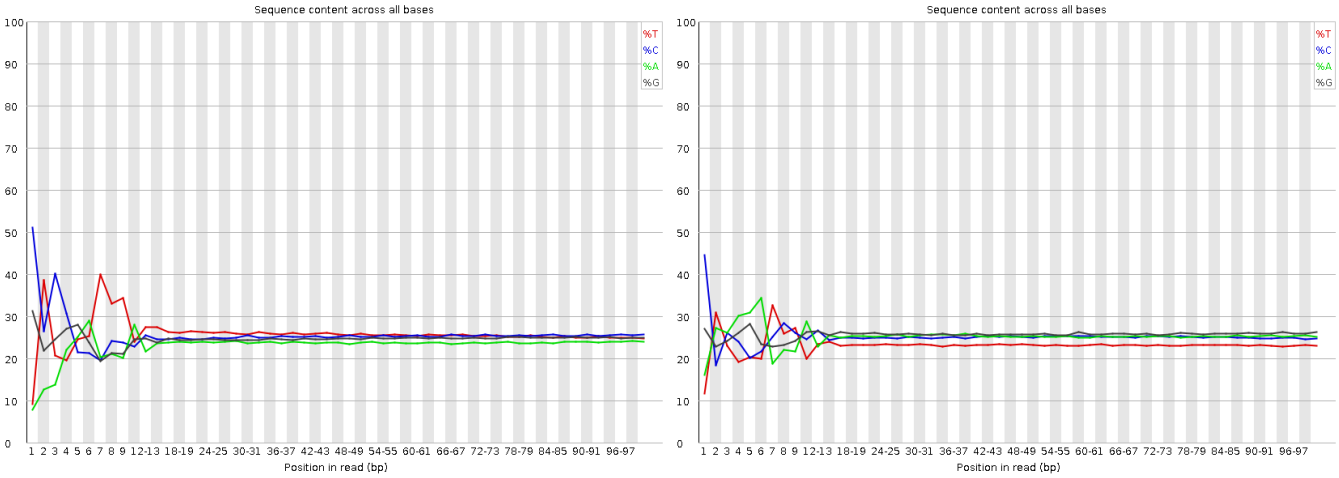

Figure 6: Percent of each base present at each position in a sequence for Read1 (left) and Read2 (right).

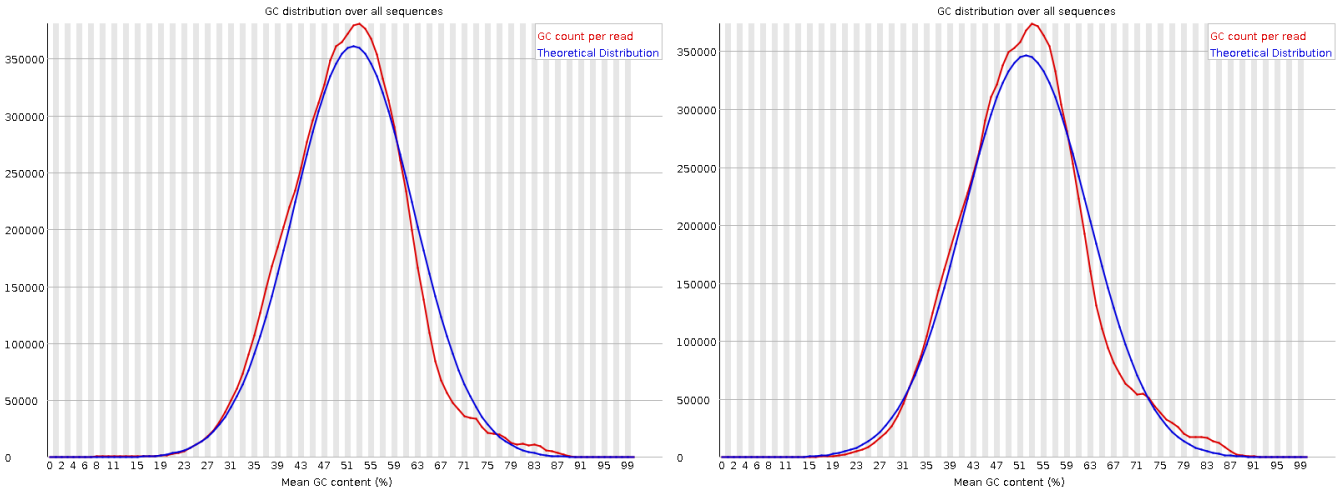

Figure 7: Distribution of percent GC content across all sequences in Read1 (left) and Read2 (right).

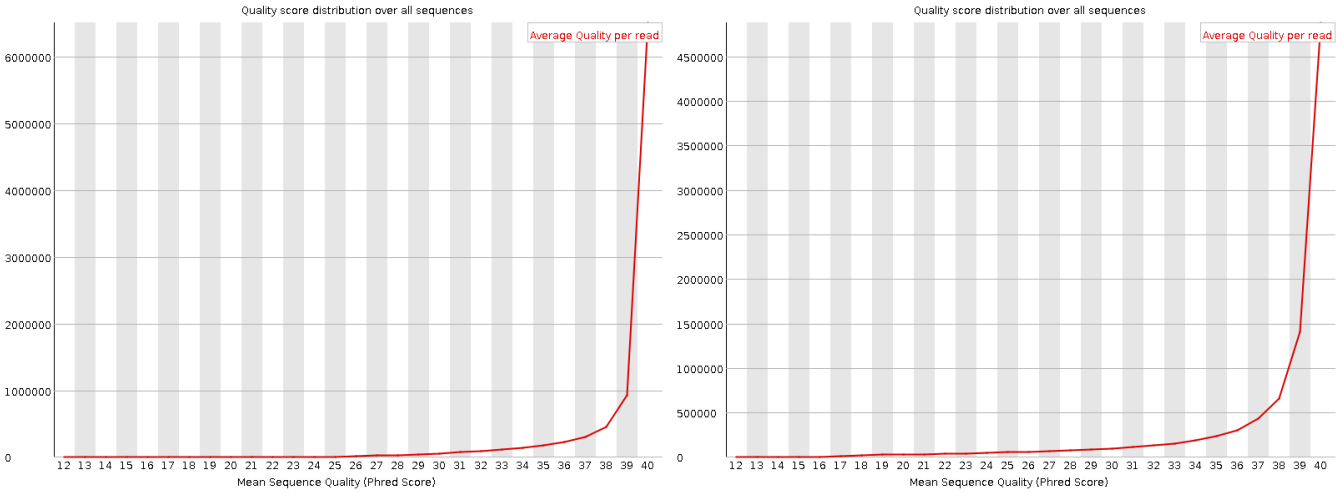

Figure 8: Distribution of mean quality score across all sequences for Read1 (left) and Read2 (right).

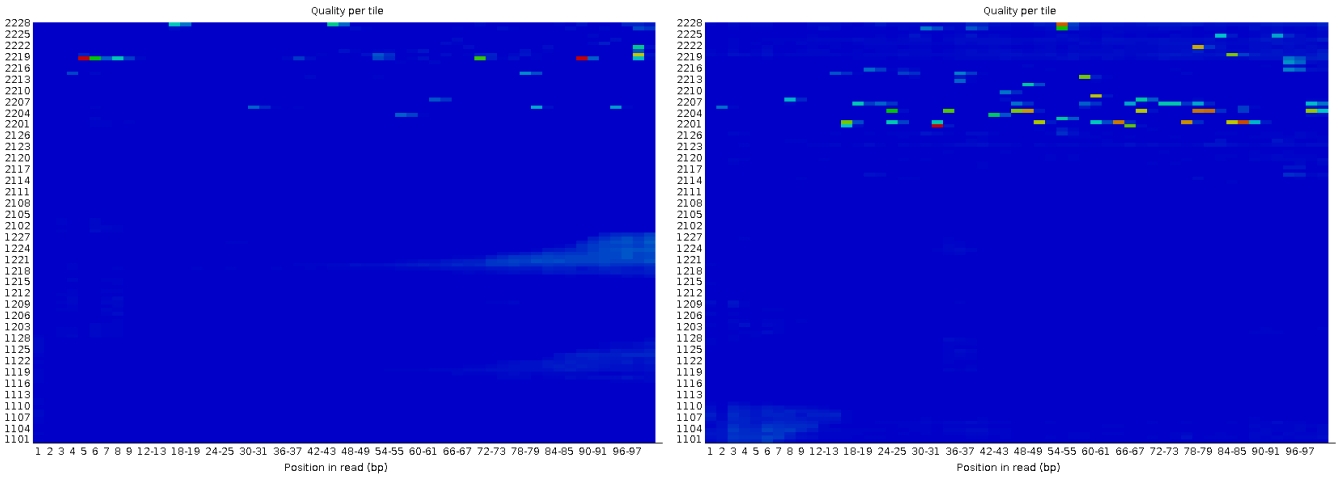

Figure 9: Deviation from the average quality for each tile. The colours are on a cold to hot scale, with cold colours being positions where the quality was at or above the average for that base in the run, and hotter colours indicate that a tile had worse quality than other tiles for that base. Read1 (left), Read2 (right).

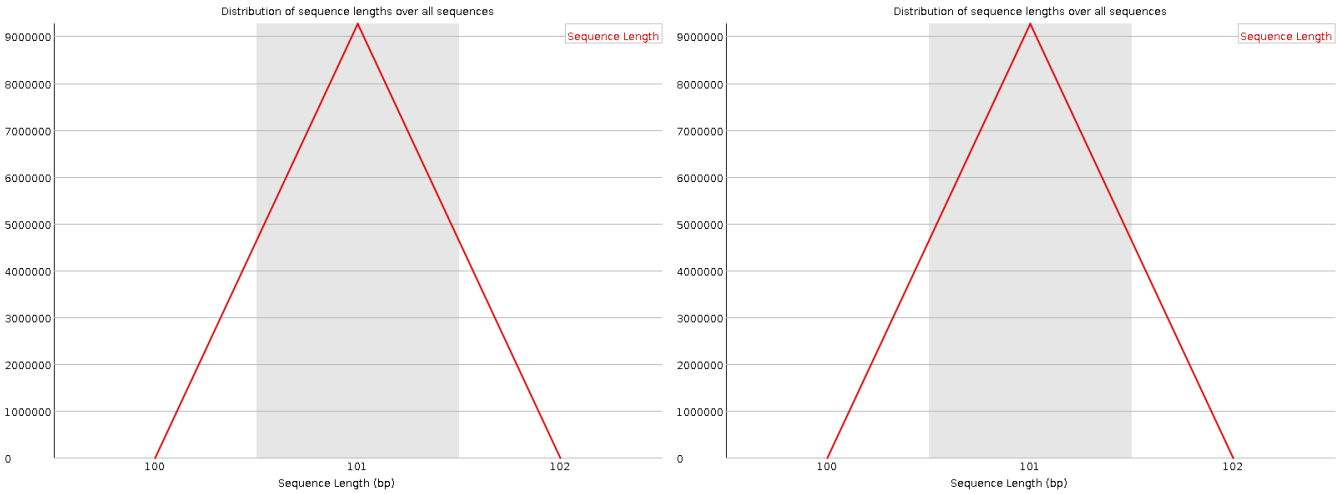

Figure 10: Distribution of sequence length across all sequences in Read1 (left) and Read2 (right).

21_3G_both_S15

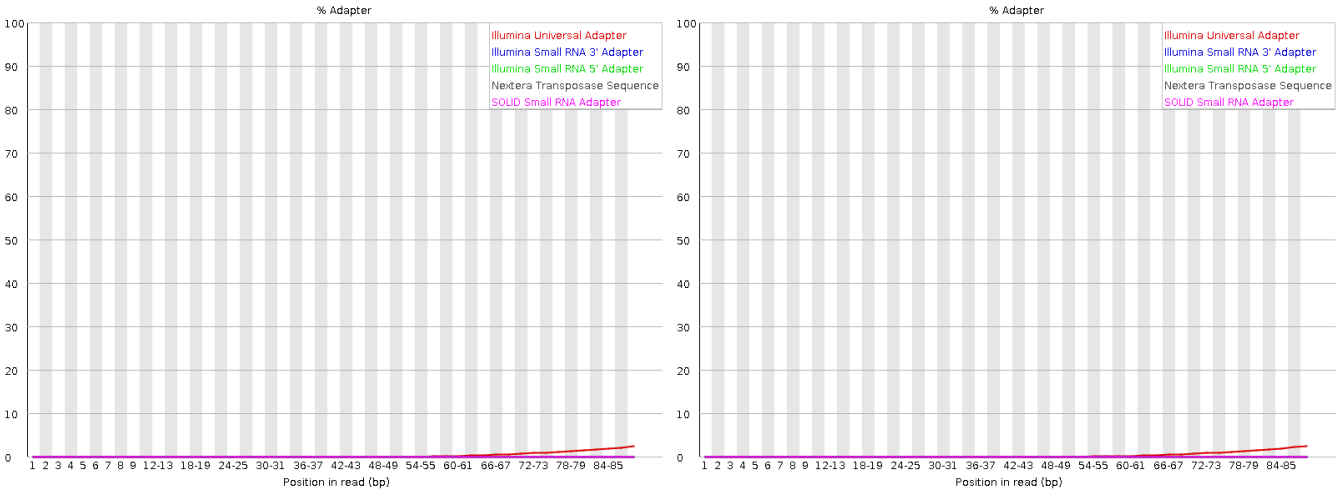

Figure 11: Percent adapter content for read1 (left) and read2 (right)

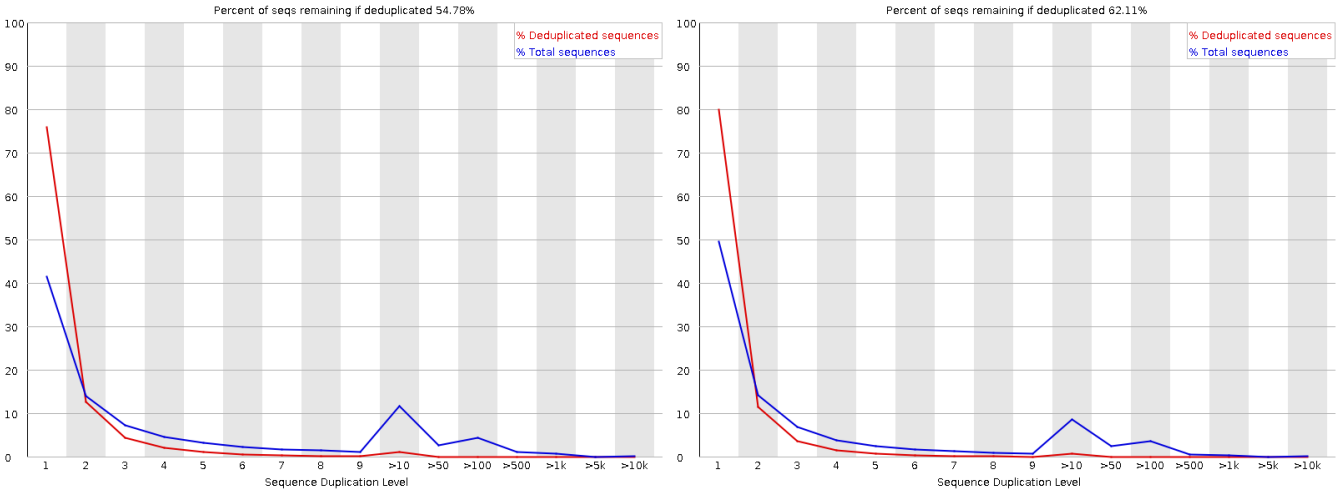

Figure 12: Duplication levels displayed by the reads for read1 (left) and read2 (right). The blue line represents duplication in the raw data while the red line represents duplication in the data when they have been deduplicated.

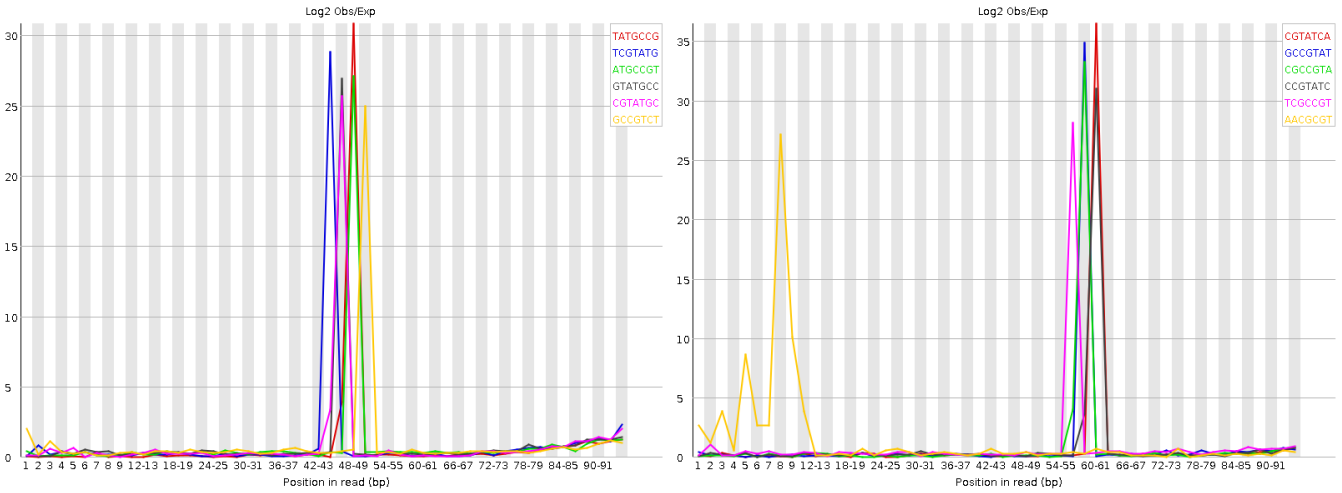

Figure 13: Kmer enrichment at each position along a read for Read1 (left) and Read2 (right).

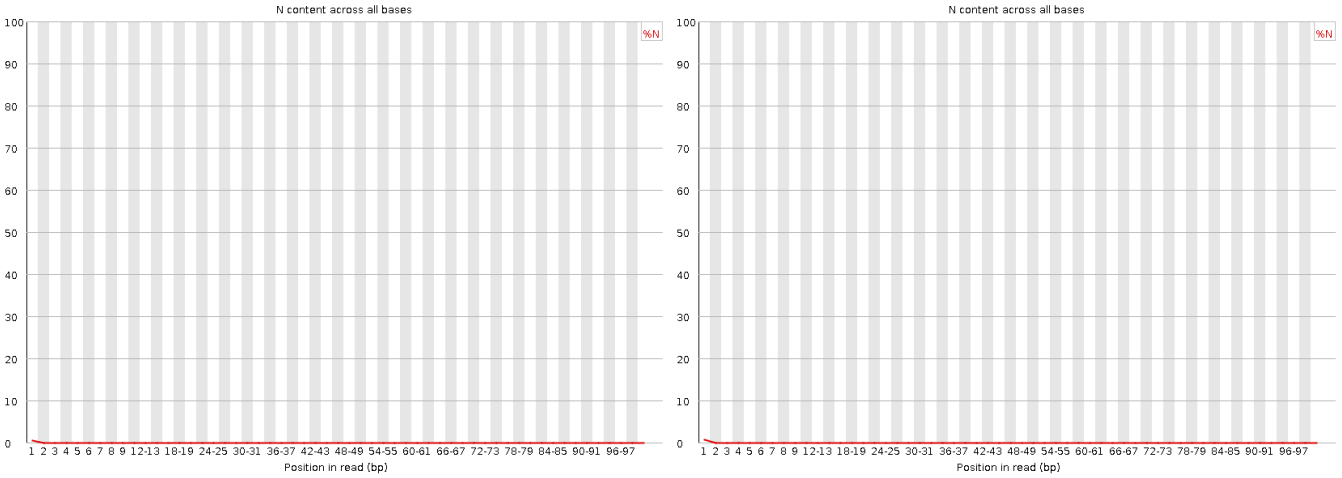

Figure 14: The number of 'N' base calls at every position in the read for Read1 (left) and Read2 (right).

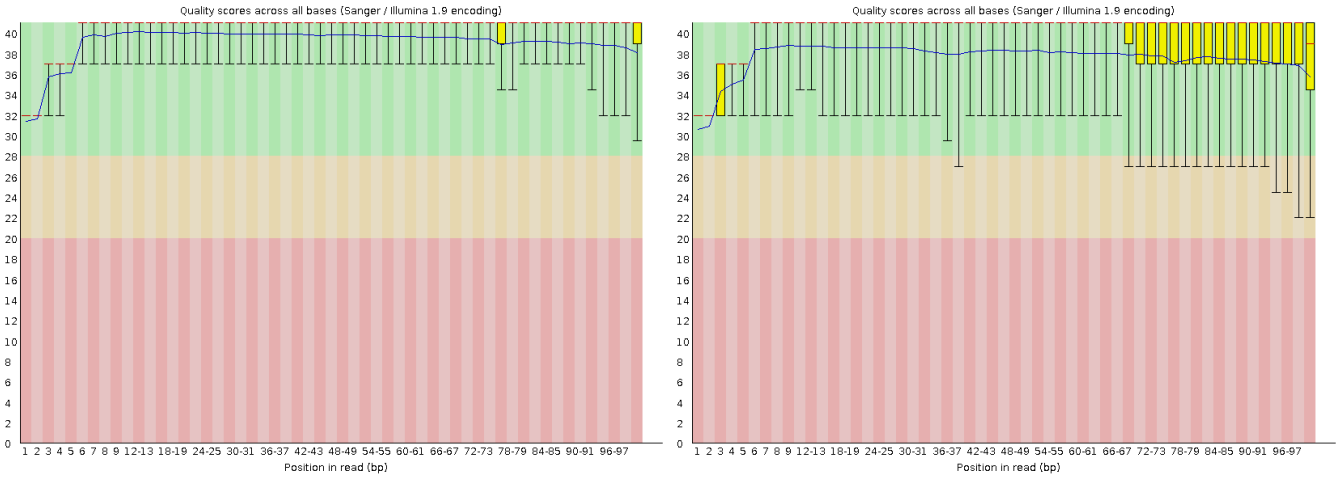

Figure 15: Mean quality score per base position across the sequence for Read1 (left) and Read2 (right).

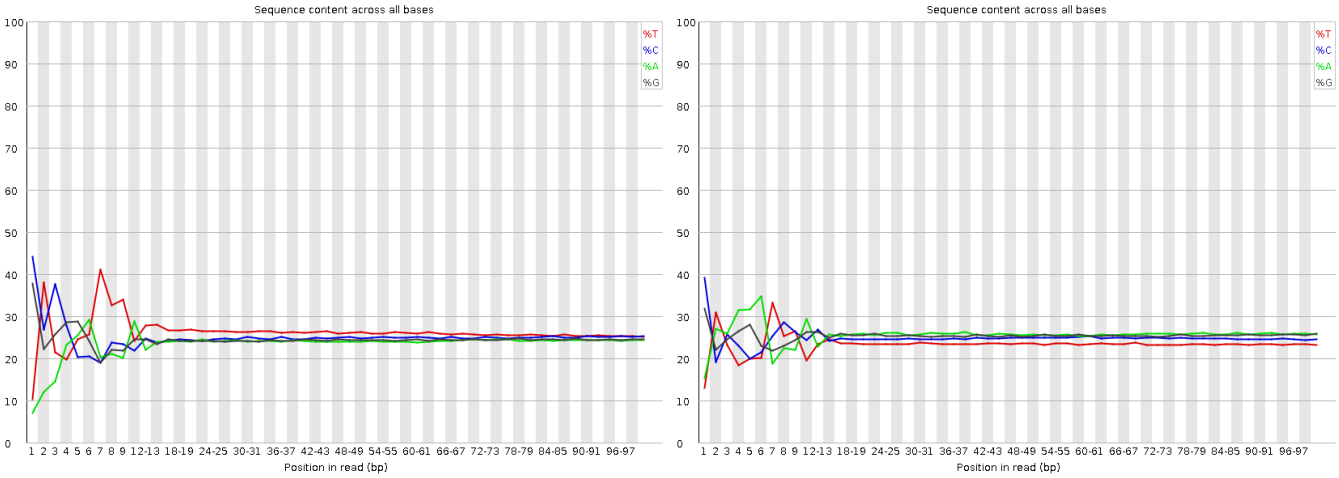

Figure 16: Percent of each base present at each position in a sequence for Read1 (left) and Read2 (right).

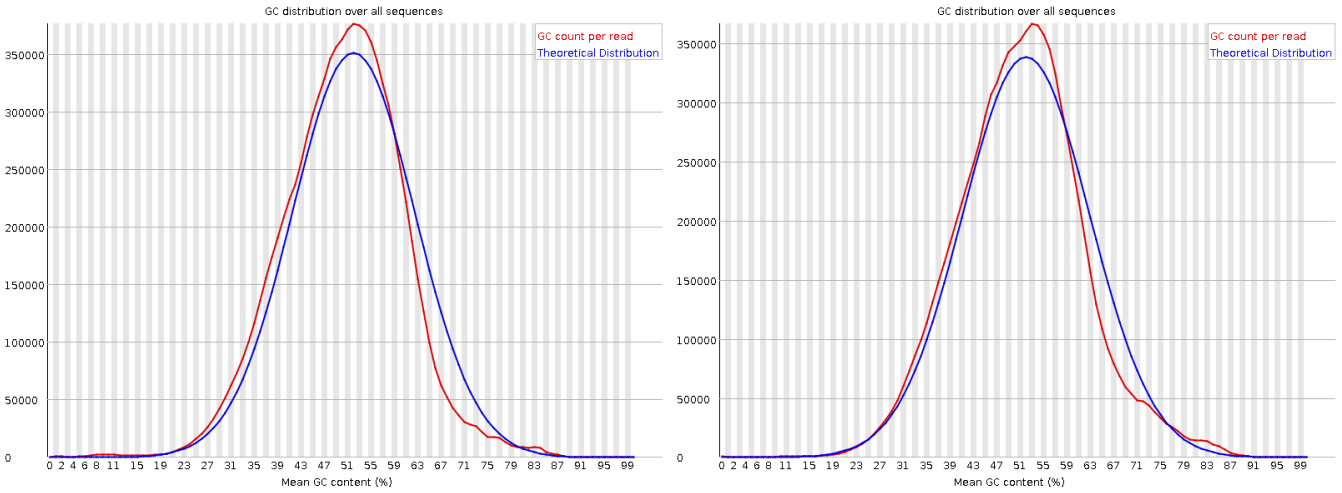

Figure 17: Distribution of percent GC content across all sequences in Read1 (left) and Read2 (right).

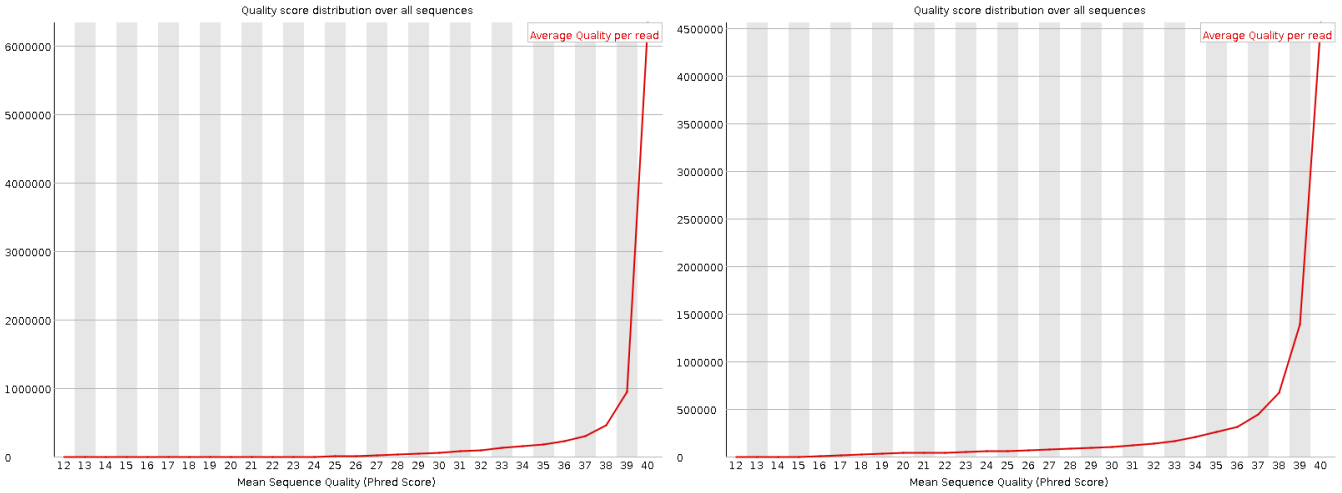

Figure 18: Distribution of mean quality score across all sequences for Read1 (left) and Read2 (right).

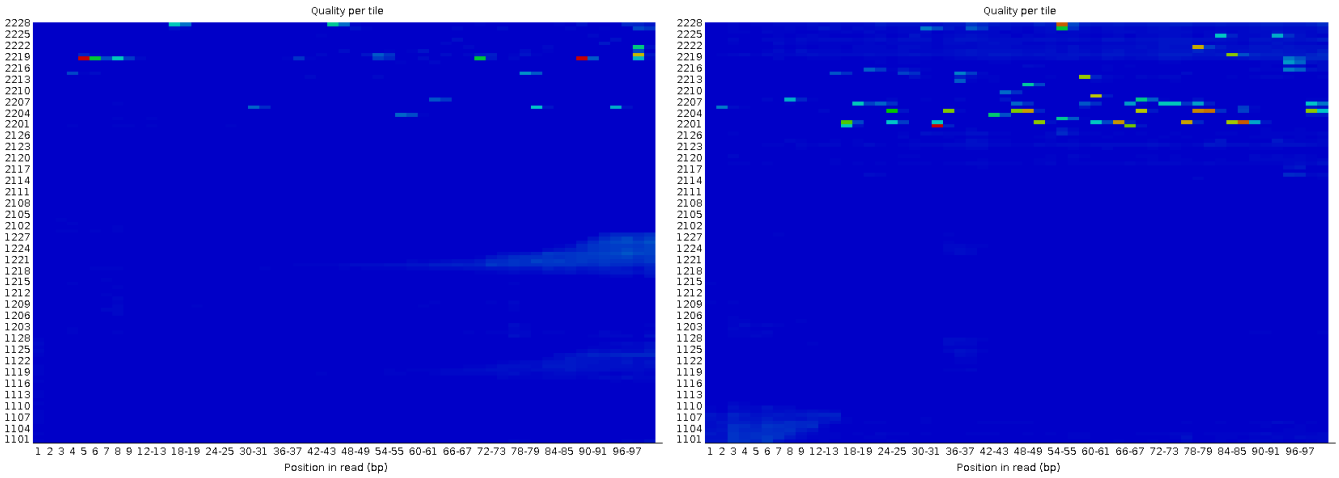

Figure 19: Deviation from the average quality for each tile. The colours are on a cold to hot scale, with cold colours being positions where the quality was at or above the average for that base in the run, and hotter colours indicate that a tile had worse quality than other tiles for that base. Read1 (left), Read2 (right).

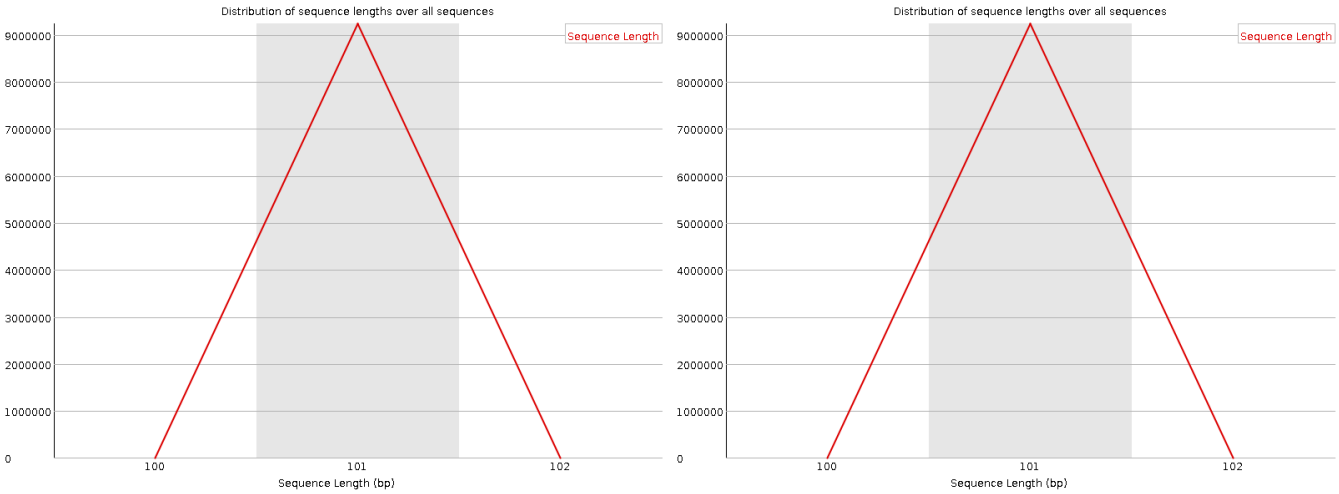

Figure 20: Distribution of sequence length across all sequences in Read1 (left) and Read2 (right).

Comparison of average per base quality score distribution plots by FASTQC vs Demultiplex program

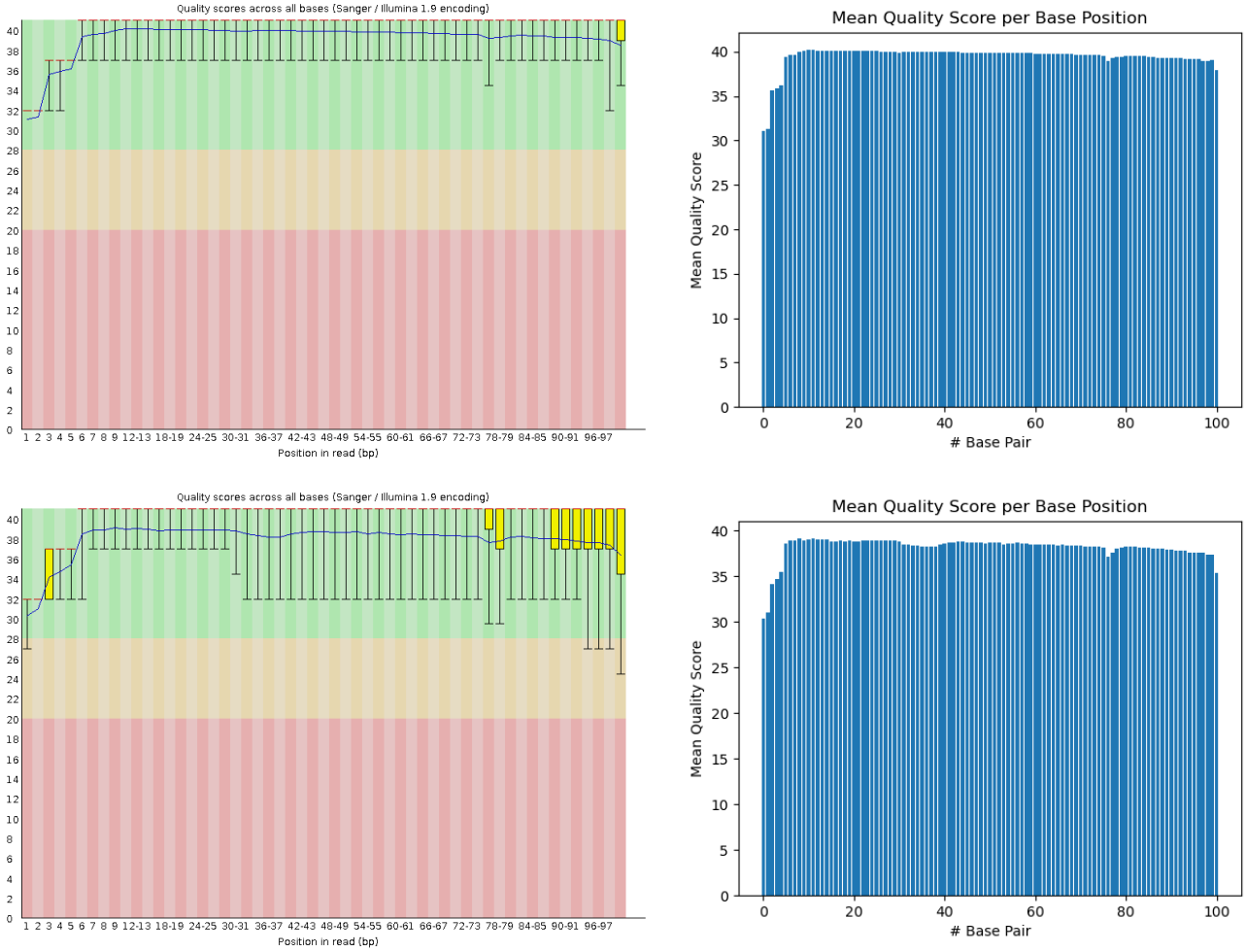

Figure 21: Distribution of mean quality score per base position across all sequences in Read1 (top) and Read2 (bottom) as computed by FASTQC (left) and my demultiplex program (right) in 4 2C mbnl S4 samples.

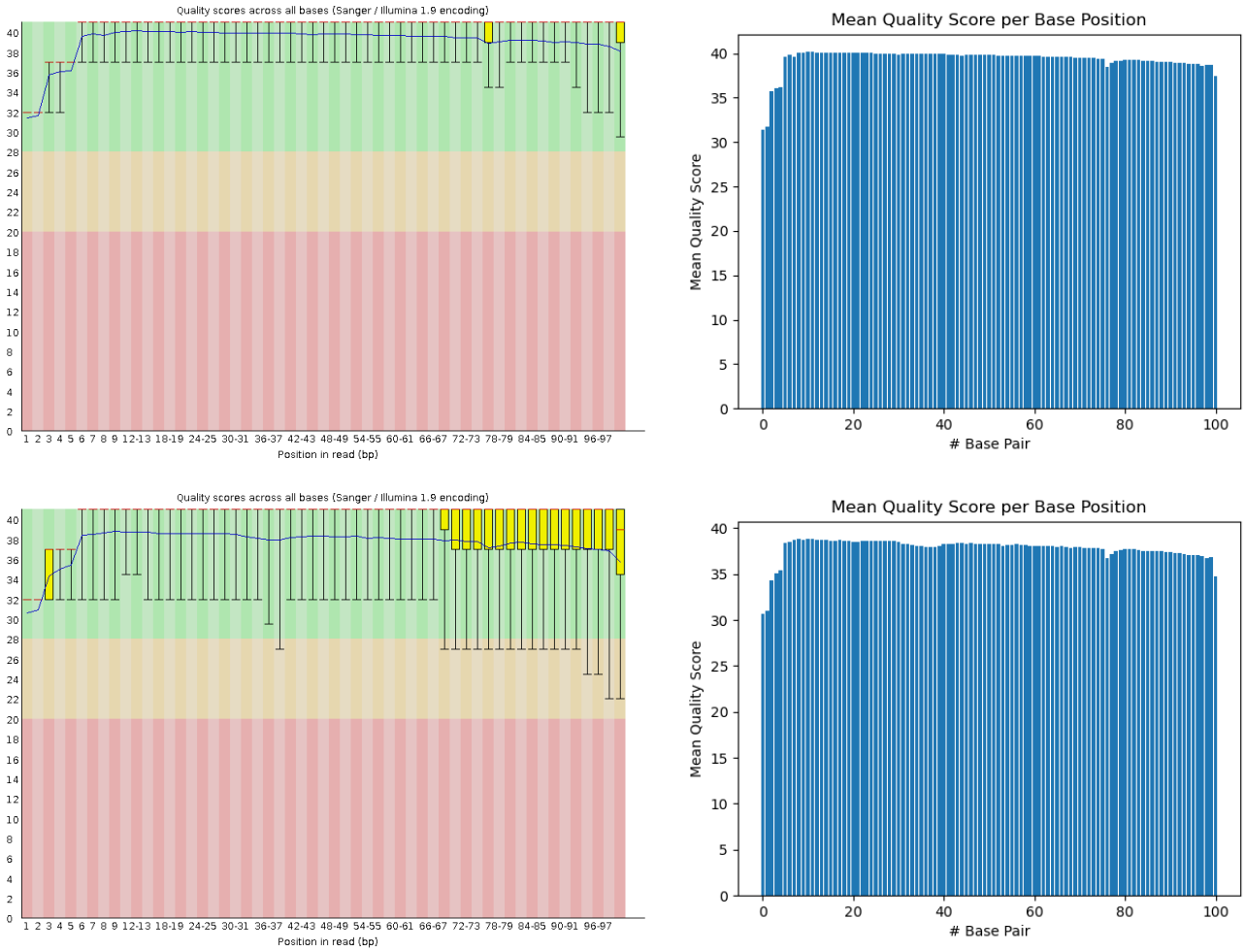

Figure 22: Distribution of mean quality score per base position across all sequences in Read1 (top) and Read2 (bottom) as computed by FASTQC (left) and my demultiplex program (right) in 21 3G both S15 samples.

The distribution plots produced by my demultiplex program looks identical to the plots produced by FASTQC for for all 4 files. However, the run time for FASTQC is slightly longer than the demultiplex program. I would imagine this is because the FASTQC program was running many different analyses on the fastq files and producing multiple plots, whereas the demux code was only creating the per-base distribution plot. However, considering the number of analyses FASTQC was running to the number of analyses the demultiplexing code was running, FASTQC appears to be more efficient.

Analysis of data quality

The median per base quality scores are very high, and the IQR for R1 files are very small, indicating that there is very little variance in these values at each position. In the R2 files, the median is a bit lower and there is more variance since the IQR looks larger. However, this is expected in R2 reads, and the scores are still reasonably high enough to warrant confidence. Per base N content is low across all files, which indicates there were very few positions that got multiple conflicting base calls.

The great majority of sequences have significantly high mean quality score in the R1 files. In the R2 files, the more sequences had slightly lower mean quality scores (phred score ~20-30 range). Still, most are of high

mean quality. Again, this is expected of R2 reads because of the nature of the Illumina sequencing protocol: There are multiple sequencing cycles between which the plates are washed with chemicals, and "read2" is the last sequence to be read by the sequencer

Adapter content is very low across all files. It does increase a bit at the end of the sequence, but they contain less than 5% adapter sequence, so I would conclude that this can be managed with adapter trimming.

Overall, I believe these libraries are of high enough quality to be used for downstream analyses.

Part 2 – Adaptor trimming comparison

Cutadapt analysis

4_2C_mbnl_S4

Total read pairs processed: 9,265,284

File	Number of reads with adapter
Read1	570,274 (6.2%)
Read2	637,307 (6.9%)

The proportion of R1 that was trimmed was 6.2%, and the proportion of R2 that was trimmed was 6.9%.

21_3G_both_S15

Total read pairs processed: 9,237,299

File	Number of reads with adapter
Read1	613,874 (6.6%)
Read2	679,275 (7.4%)

The proportion of R1 that was trimmed was 6.6%, and the proportion of R2 that was trimmed was 7.4%.

Adapter analysis

Regardless of whether it is read1 or read2, the sequences are given in 5'-3' direction in the fastq files as per convention. The adapters are also given in 5'-3' direction. Therefore, the adapters (if any are present in the read) should be found at the 3' end, in the same sequence as provided (no reverse complement necessary).

To check for presence and directionality of the proposed adapter in the fastq files, I used this code:

```
zcat <zipped_fastq_file> | grep -c <adapter seq>
```

I repeated this for pairs of (R1-adapter1, R2-adapter2, R2,-adapter1, R1-adapter2). As expected, the appropriate read-adapter pairs (i.e. R1-adapter1 or R2-adapter2) had the adapters in them. However, in the mismatched pairs, the adapter was not found. However, it is also important to note that the numbers produced by this command are not accurate to the number of reads that will be trimmed by cutadapt because some may contain partial adapter sequences.

Trimmed read length distribution

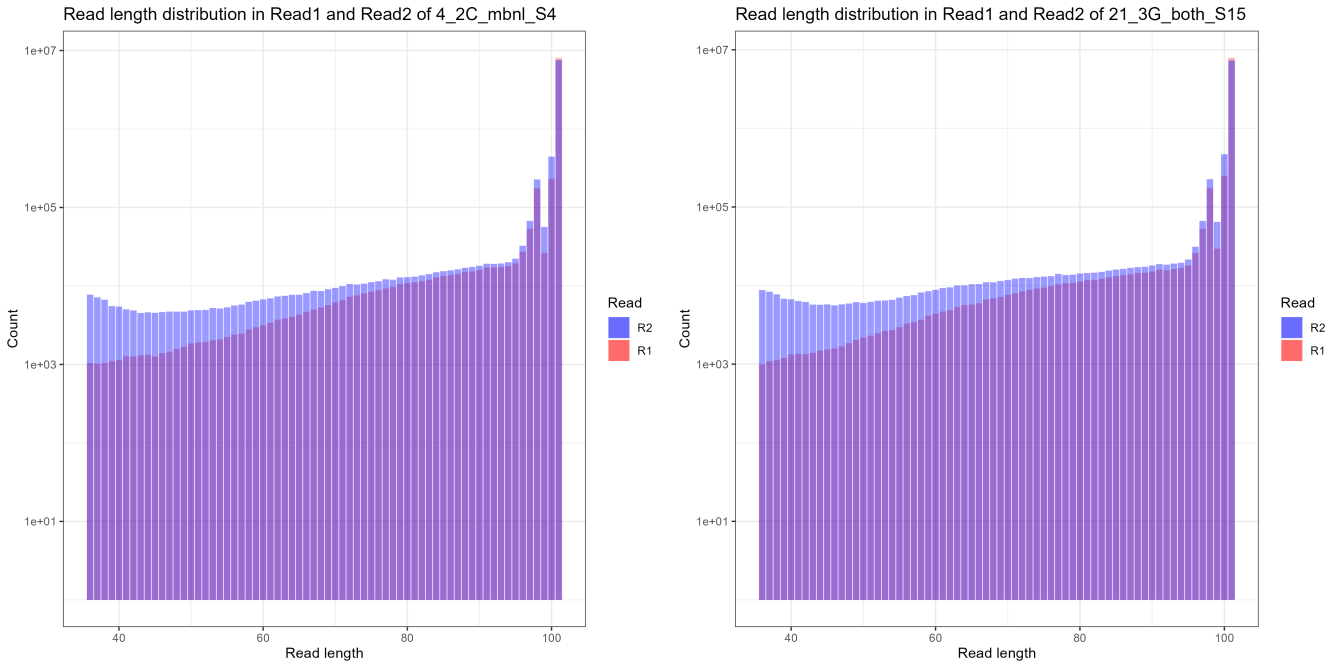

Figure 23: Comparison of the distribution of read lengths in Read1 and Read2 in 4 2C mbnl S4 (left) and 21 3G both S15 (right) samples.

In both samples, read2 has more reads that are shorter in length. Moreover, read2 has slightly fewer reads that are 101nt long (maximum length). The general shape of the distributions look very similar in both samples.

The function of trimmomatic is to read one sequence at a time in a sliding window of selected size, and trim the read when the quality of the window falls below a set quality score cutoff. Since read2 sequences are generally expected to be of lower quality than read1, as confirmed by FASTQC, I had expected for the read2 sequences to be shorter than read1 sequences after trimming.

Part 3 – Alignment and strand-specificity

Number of mapped/unmapped reads

File name	Mapped	Unmapped
aligned_21_Aligned.out.sam	17061162	645462
aligned_4_Aligned.out.sam	17172681	788079

Strandedness of library

```
grep "^ENSMUSG" <genecounts_file> | awk '{sum+=$2} END {print sum}
```

File name	Total gene count
genecounts_21_stranded.tsv	334465
genecounts_21_revstranded.tsv	7180828
genecounts_4_stranded.tsv	356074
genecounts_4_revstranded.tsv	7236906

The genecounts when running the data through htseq-count with stranded=yes were lower than those for the data run with stranded=reverse. This difference in alignment between the 2 strands indicates that the library was stranded, because sequences will only have been sequenced in one direction, and therefore will only align to one strand of the reference genome. If the library was unstranded, then there should be equally as many sequences of a specific gene aligning to one strand as the other. Therefore, I would expect approximately 50-50 distribution between the two htseq-count runs (stranded=yes and stranded=reverse).