QAA Assignment

Mahmoud

2024-09-10

Part 1: Read quality score distributions

All tools used in this assignment are downloaded in QAA conda environment. The working directory is `/projects/bgmp/malm/bioinfo/Bi623/FastQC`. Input files consist of two samples of paired-end RNA-Seq data.

```
/projects/bgmp/shared/2017_sequencing/demultiplexed/31_4F_fox_S22_L008_R1_001.fastq.gz
/projects/bgmp/shared/2017_sequencing/demultiplexed/31_4F_fox_S22_L008_R2_001.fastq.gz

/projects/bgmp/shared/2017_sequencing/demultiplexed/10_2G_both_S8_L008_R1_001.fastq.gz
/projects/bgmp/shared/2017_sequencing/demultiplexed/10_2G_both_S8_L008_R2_001.fastq.gz
```

Note:

I will be referring to the first data set (fox_S22) as sample 1, and the second dataset (10_2G) as sample 2. All plots generated by FASTQC will have a caption starting with the word "FASTQC", otherwise the plots are generated by the author

Sample 1 Quality score distribution and per base N content

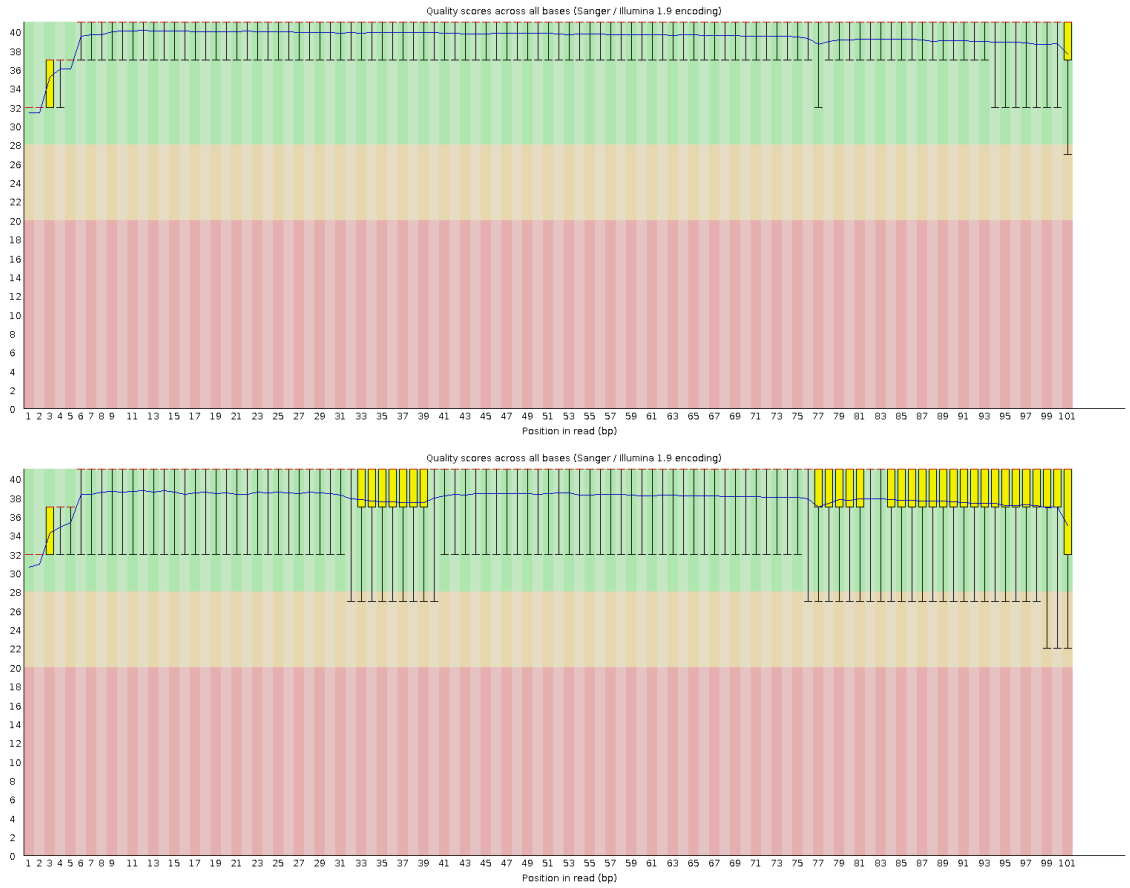

Figure 1: FASTQC Per Base Quality Score Distribution for Forward Read (top) and Reverse Read (bottom).

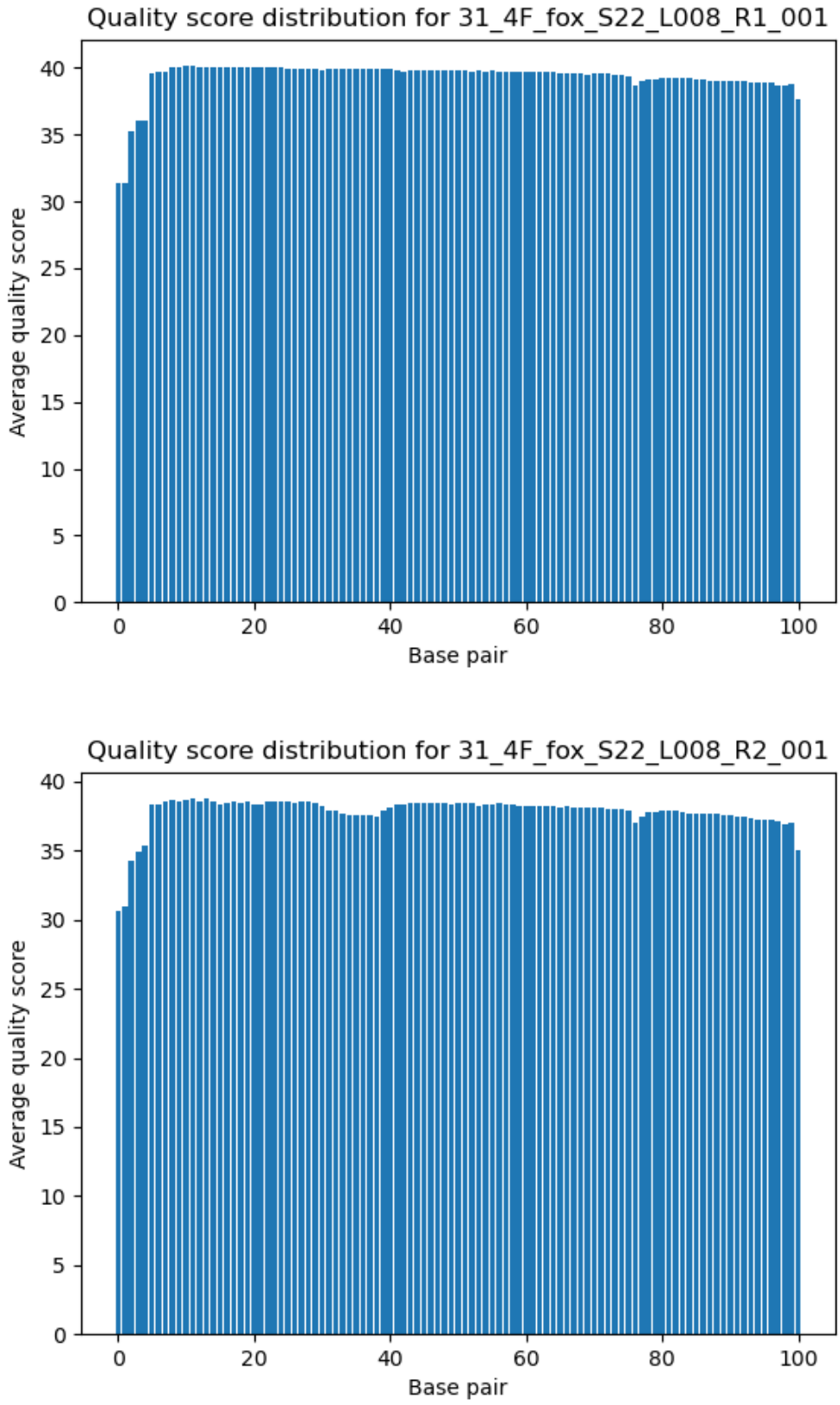

Figure 2: Per Base Quality Score Distribution for Forward Read (top) and Reverse Read (buttom).

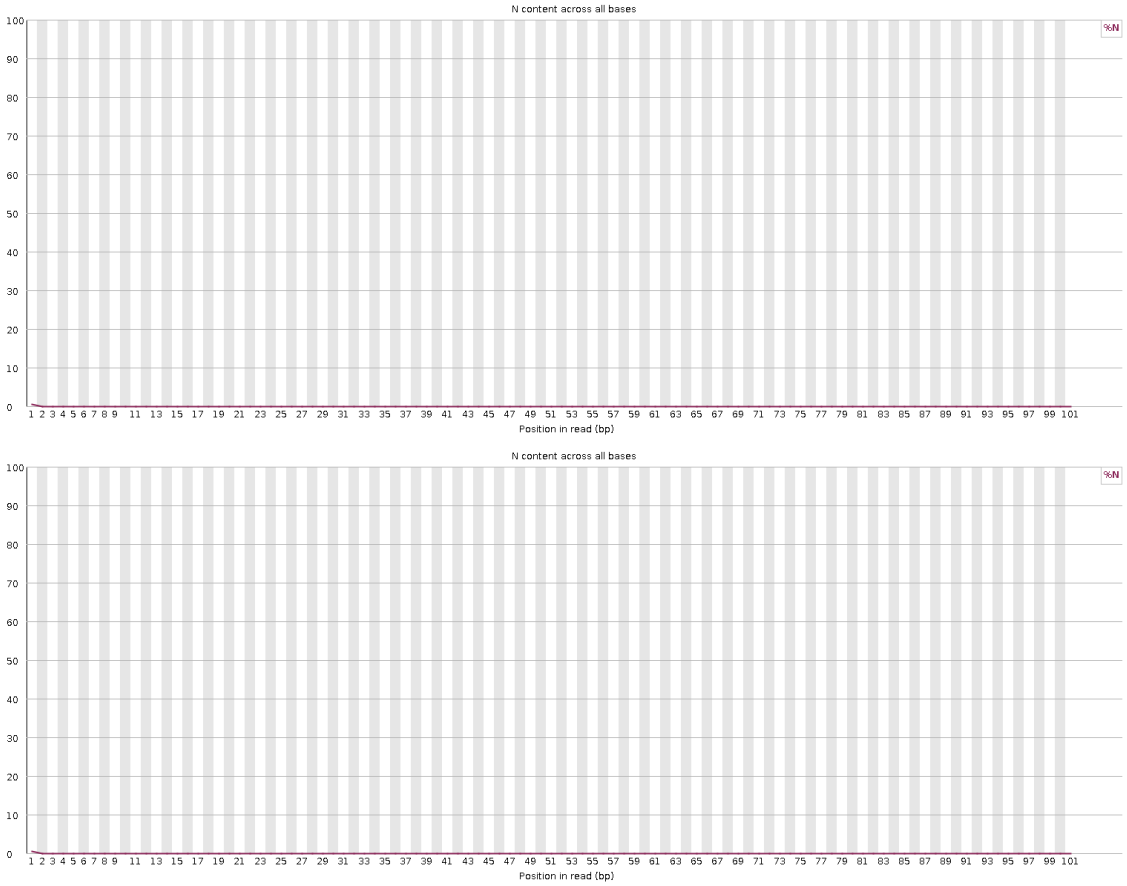

Figure 3: FASTQC Per Base N content Distribution For Forward Read (top) and Reverse Read (buttom).

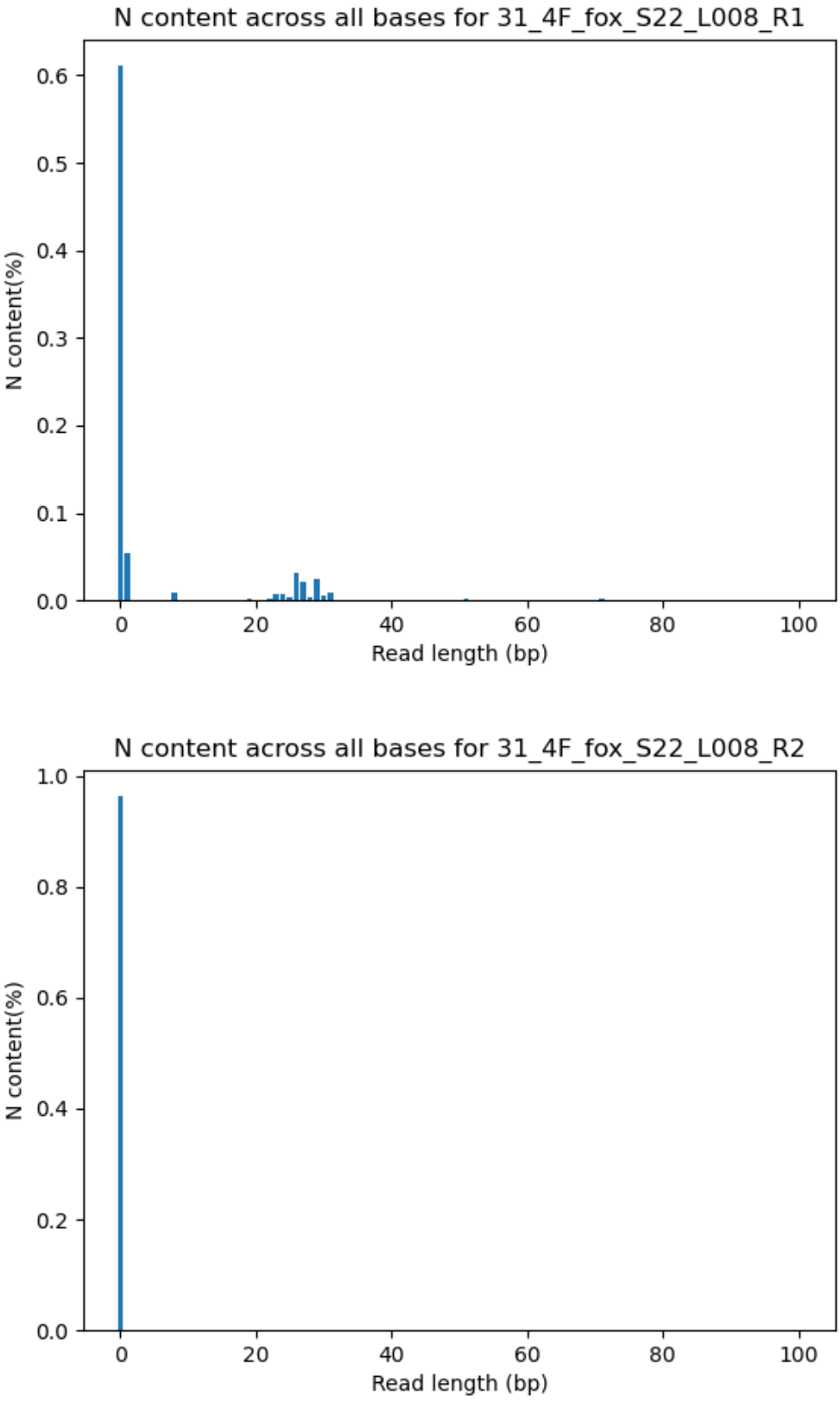

Figure 4: Per Base N content Distribution For Forward Read (top) and Reverse Read (buttom).

Sample 2 Quality score distribution and per base N content

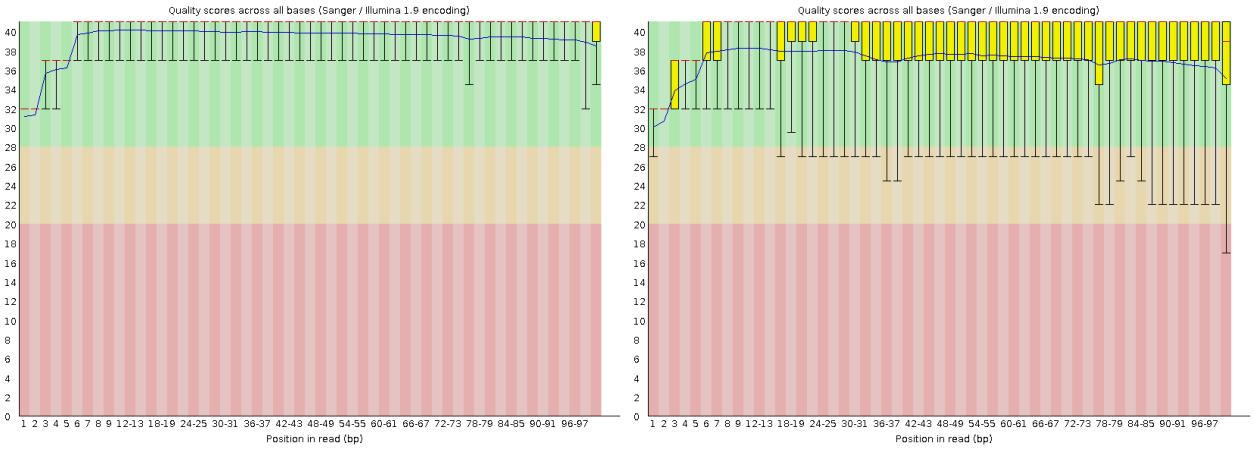

Figure 5: FASTQC per base quality score distribution for forward read (left) and reverse read(right).

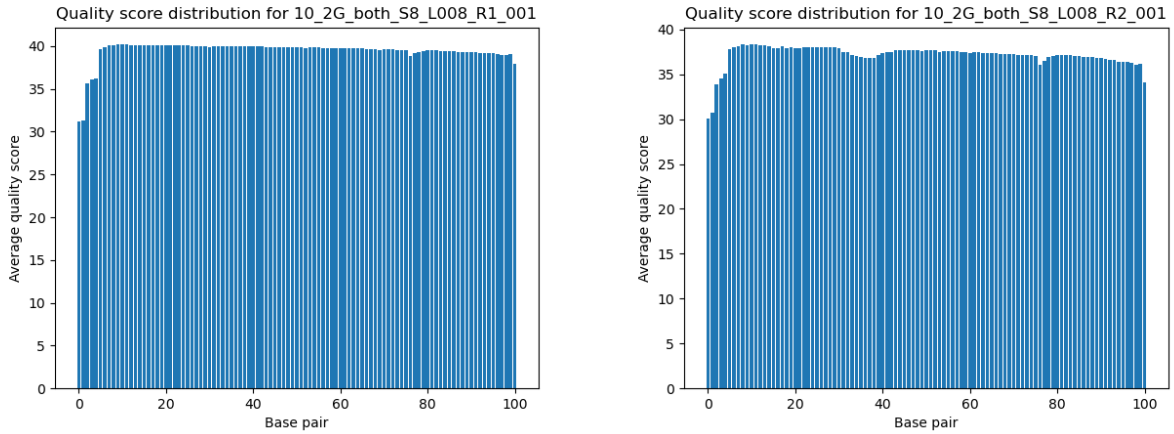

Figure 6: Per base quality score distribution for forward fead (left) and feverse fead (right)

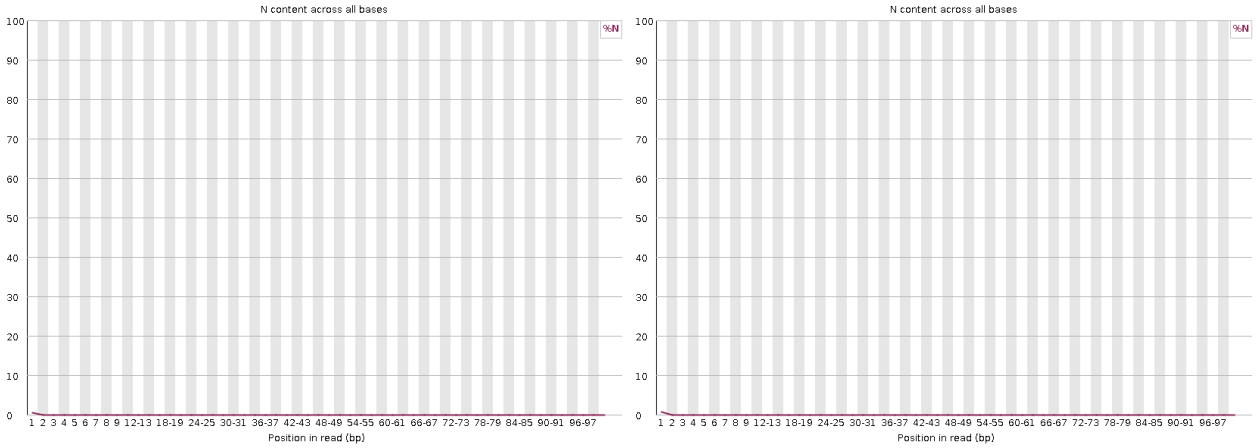

Figure 7: Per base N content distribution for forward read (left) and reverse read (right).

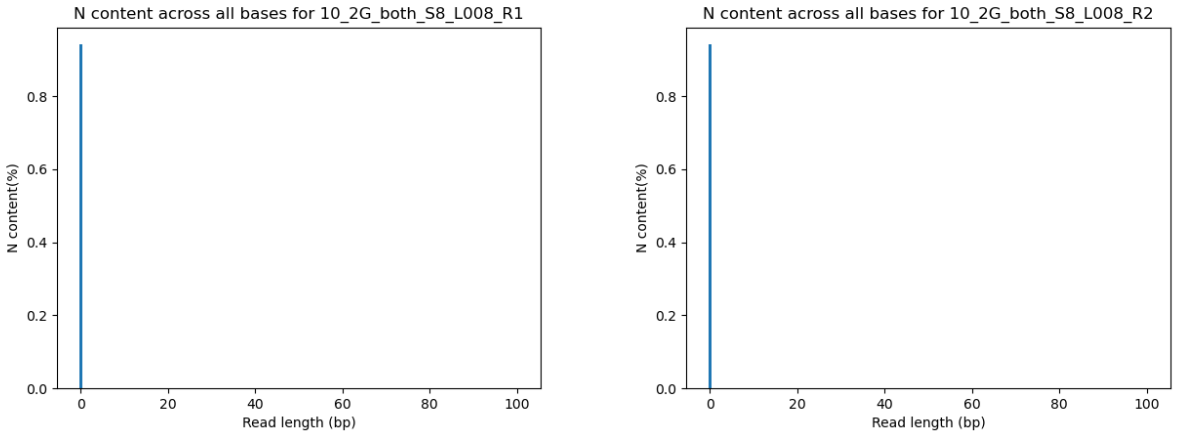

Figure 8: Per base N content distribution for forward read (left) and reverse read (right).

Data Interpretation

The plots produced by FASTQC presented above showed a high quality data averaging a q score of approximately 36, for both samples. This data was consistent with the per-base N content plots produced by FASTQC as well. Looking at Figures 3 and 7, we can notice the extremely low %N in both reads, supporting the validity of the data. However, sample 1 contained higher adapter content compared to sample 2, as seen the FASTQC adapter content plot, and the over represented sequence section.

In comparing plots produced by FASTQC software with plots produced by me, all plots seemed to be consistent with each other. By comparing Fig.1 with Fig.2 and Fig.5 with Fig.6, we see a consistency in the quality score distribution across both reads. It is hard hard to spot the difference between the plots due to the little variation in the data, but if we look at the reverse read of Fig.1 and it's replicate in Fig.2, we can the see the same dip in q score between base pairs 29 and 40.

The same trend was seen with the N content plots. The y-axis scales between the FASTQC plots and my plots are different, but they were generally consistent with each other, where the highest N percentage appeared at the lowest read length. The forward read in Fig.4 showed a small percentage of N content ($<0.05\%$) between read length of 20 and 30 that cannot be seen on the FASTQC plot since it is generated at a much lower resolution. This is the only observation that indicates a difference between the plots, but i think it is insignificant in terms of downstream analysis. The FASTQC HTML report was generated in under a minute for sample 1 and around 10 minutes for sample 2, using 97% and 99% of CPU respectively. The quality score distribution plots generated by me took around 1 minute for sample 1 and 17 minutes for sample 2.

Part 2: Adaptor Trimming

Cutadapt

Cutadapt produces a report for each run indicating the proportion of reads trimmed due to the presence of adapter sequences. In my cutadapt report for sample 1, 12% of reads were trimmed in the R1 read while 12.7% were trimmed for the R2 read. This seems to be a high percentage of adapters to be present, but by looking at the FASTQC plot for the adapter content, I can confirm the presence high percentage of adapters in both R1 and R2, confirming the cutadapt output.

For sample 2, the percentage of trimmed adapters was much lower, where 2.6% of reads were trimmed in R1 and 3.4% for R2. The proportion of reads trimmed was bigger for the R2 reads in both samples, possibly due to the fact that read 2 usually has lower quality as the sequencing machine uses up its reagents.

explain the high adapter content in sample 1 as shown in FASTQC report

Trimmomatic

Trimmomatic was run on both RNA-seq samples. For each paired-end sample, it outputs four files, two paired and two unpaired. Here I show the plots of the paired reads (R1 and R2) for each sample. The plots show the difference in read lengths between R1 and R2 of each sample. As shown in Fig.9 below, the R2 read is trimmed more extensively compared to R1 in both samples, since shorter reads can be spotted in the R2 read. This means that R2 has a lower quality sequence, which could be due to reagents being used up by the time R2 read is being sequenced. This can also be explained by the fact that the clusters size decreases during bridge amplification at the paired-end turnaround stage that occurs before read 2 is sequenced, resulting in lower quality reads in R2 compared to R1.

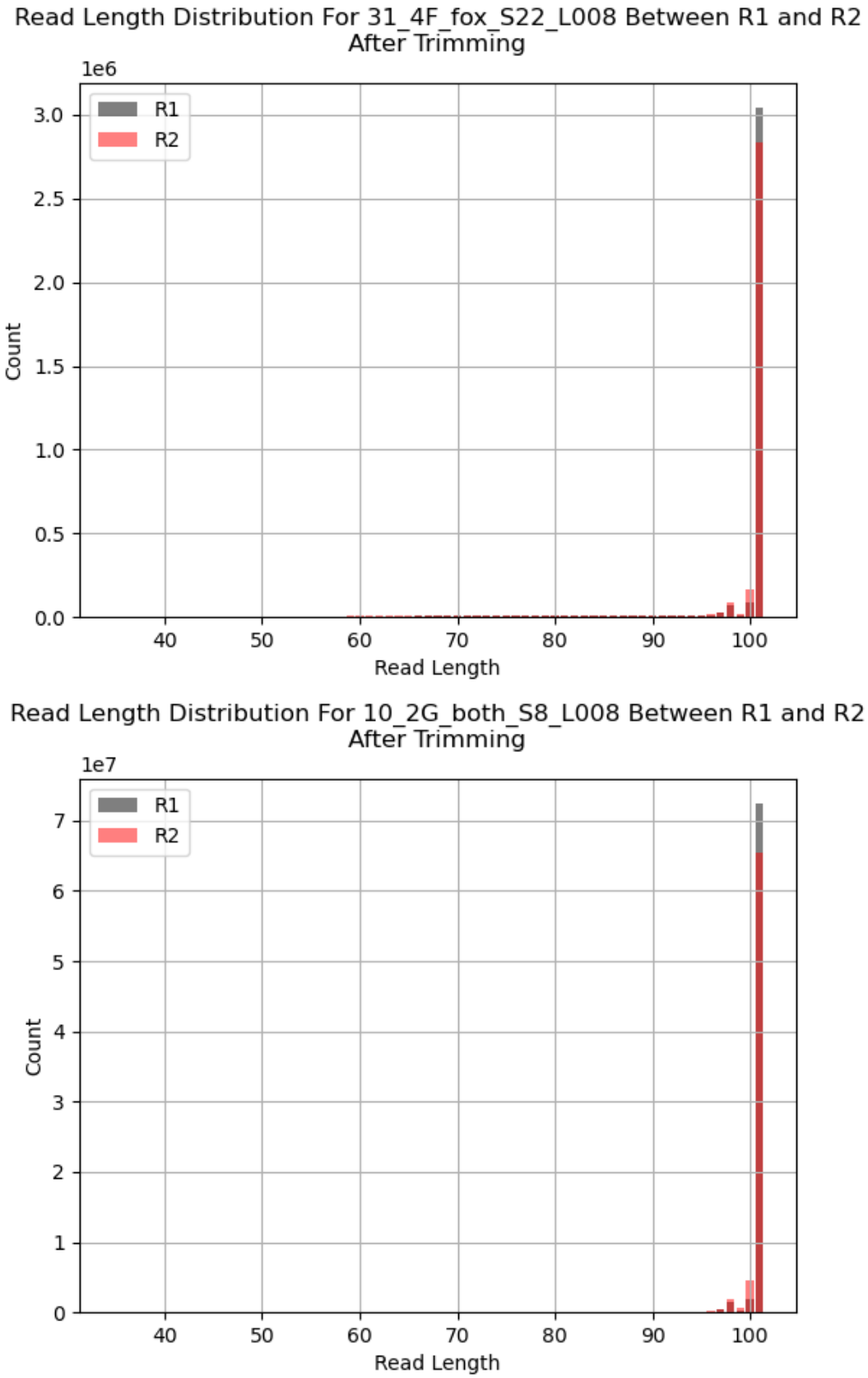

Figure 9: Read length distribution (R1 and R2) for sample 1 (top) and sample 2 (buttom) after trimming low quality reads using Trimmomatic.

Part 3: Alignment and Starnd-Specificity

STAR

I downloaded mouse genome FASTA file (Mus_musculus.GRCm39) from Ensemble release 112 as well as the GTF file. Then a databse was created and the reads were aligned to this genome. This produced `.sam` file for each sample. Then, I calculated the number of mapped and unmapped reads in each `.sam` file:

- **Sample 1** The number of mapped reads is **6969863** The number of unmapped reads is **225953**
- **Sample 2** The number of mapped reads is **6969863** The number of unmapped reads is **225953**

htseq

htseq-count tool was used to count the number of reads that mapped to a feature. It was run twice using two different flags. The first was `--stranded=yes` and the second is `--stranded=reverse`. Each run produced gene count file allowing us to calculate the total number of reads and compare the two flags output.

Read Count Summary The following bash command was used to count the total number of reads:

```
cat <filename> | cut -f 2 | awk '{sum+=$1} END {print sum}'
```

Total Number of Reads:

Sample	Total Reads
Sample 1	3,597,908
Sample 2	77,520,903

The following command was used to calculate the number of reads that mapped to a feature:

```
grep -v "^__" <filename> | cut -f 2 | awk '{sum+=$1} END {print sum}'
```

Reads Mapped to Features:

Sample	Stranded Option	Reads Mapped to a Feature	Percentage
Sample 1	stranded=yes	180,913	5.02%
Sample 1	stranded=reverse	2,957,009	82.19%
Sample 2	stranded=yes	2,957,148	3.81%
Sample 2	stranded=reverse	67,359,730	86.90%

Considering the percentages above, I propose that these data are strand specific, because 82.19% of reads mapped to a feature with the stranded=reverse flag, while only 5.02% percent were present with stranded=yes flag. The same trend was observed in both samples. This suggests that the RNA-Seq library prep was done using a protocol where reads are aligned to the reverse strand instead of the forward strand such as the dUTP protocol where reads are generated from the reverse strand. If counts using both flags were similar, then the library is unstranded. However, this does not seem to be the case here as the difference in counts is significant between the two cases.