QAA Report

Sasha Golubeva

2023-09-12

Part 1 – Read quality score distributions

I was assigned one sample from condition "both", and one sample from condition "control" for this assignment. The id's of the demultiplexed samples on Talapas as follows:

3_2B_control_S3_L008_R1_001.fastq.gz

3_2B_control_S3_L008_R2_001.fastq.gz

32_4G_both_S23_L008_R1_001.fastq.gz

32_4G_both_S23_L008_R2_001.fastq.gz

I ran FastQC and my python scripts to asses quality of the cDNA library sequencing per forward and reverse read.

Read quality assessment for forward and reverse reads, "both" condition

Figure 1 shows per base distribution of quality scores produced by the FastQC software. Per base quality distribution plots for read 1 and read 2 show overall good quality for both reads. Read 1 quality is slightly better than the read 2 quality which is a known phenomenon for Illumina sequencing which could be attributed to the flowcell biochemistry when another round of bridge amplification and complimentary strand washing should happen before read 2 sequencing start. Additionally, the sequencing reagents has been exposed to room temperature for at least 8-10 hours at this point and might not work as good as fresh reagents. Nevertheless, the quality scores range from 31 to 38 for the read 1 and from 31 to 38 for the read 2 which makes error rate which makes the error rate = 1:1000. Illumina considers good quality scores to be equal 30 or above which makes quality of sequencing for both reads acceptable for the downstream analysis.

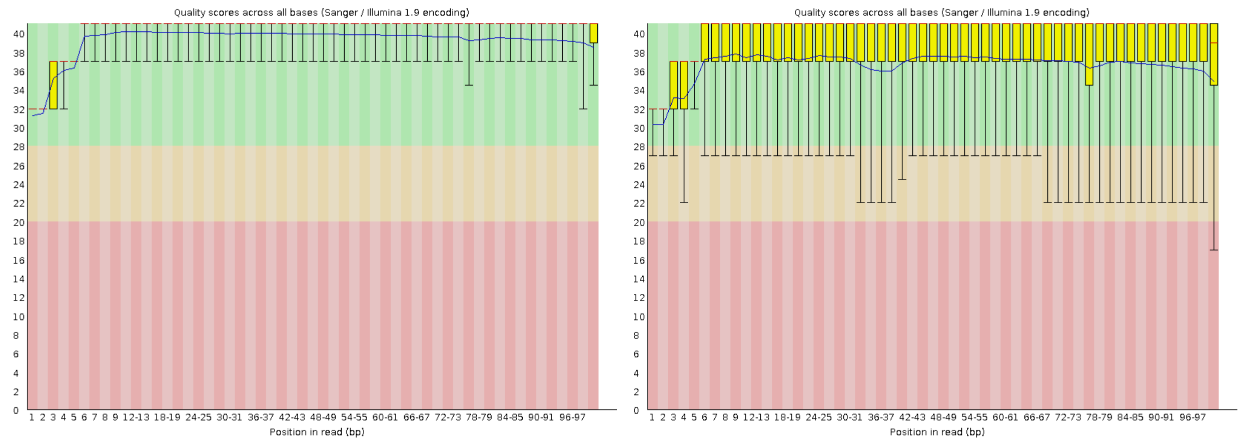

Figure 1: Quality score distributions by position on the read, both condition produced by FastQC. Read 1 (left graph) shows better overall quality per base position than Read 2 (right graph)

I compared the output of the FastQC with my script output. Figure 2 compares per base quality scores for read 1 and read 2. The results of my script are in line with the FastQC results for the quality scores distributions. Read 2 shows slightly worse overall quality, but both reads have acceptable quality by Illumina standards for the downstream analysis.

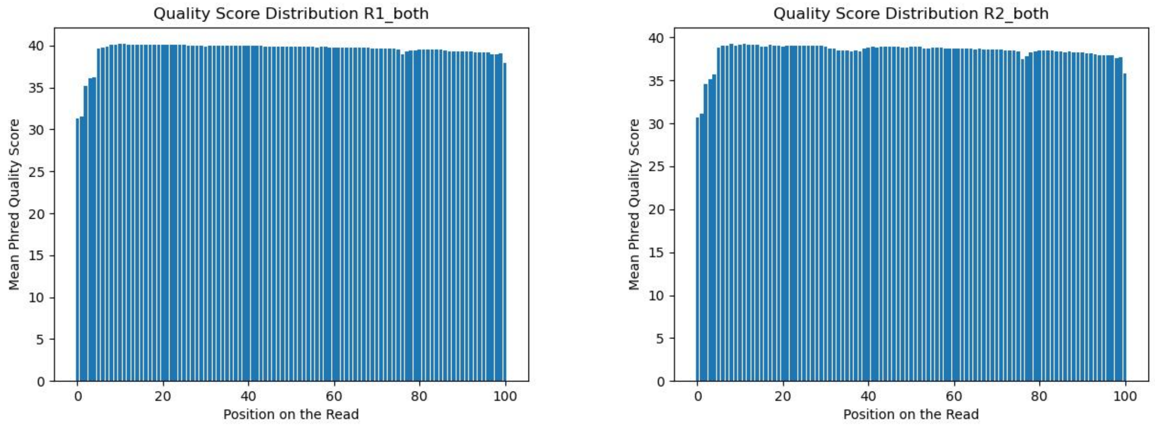

Figure 2: Quality score distributions by position on the read, both condition produced by my script. Read 1 shows better overall quality per base position than Read 2

Quality assessment per flowcell tile shows that most of the tiles had acceptable sequencing quality except a few tiles which had less than ideal sequencing quality. There are slightly more worse quality tiles for the read 2 which is inline with the previous findings of that read 2 had a slightly worse overall quality compared to the read 1.

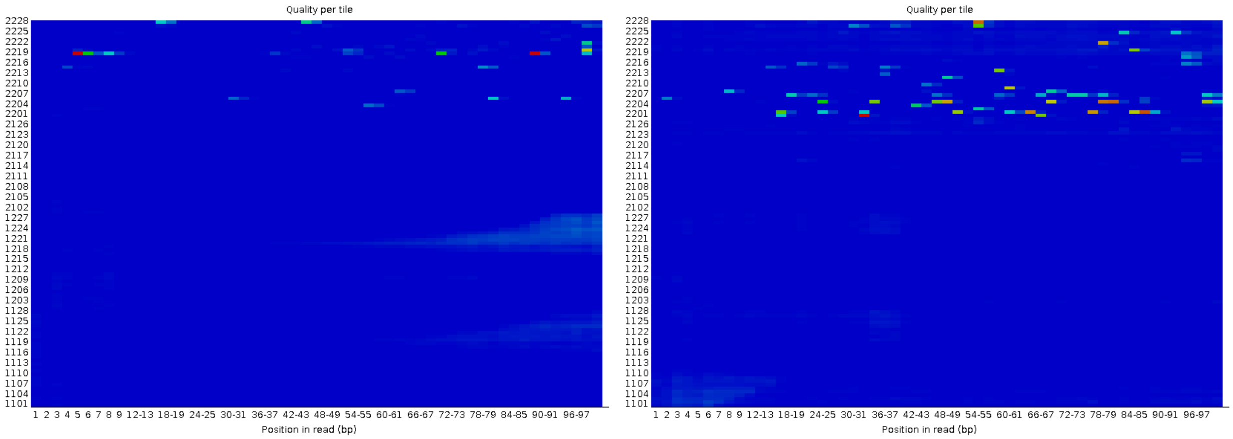

Figure 3: Quality assessment per tile on a flowcell for read 1 and read 2.

Looking at per read quality score distributions, majority of reads for read 1 and read 2 have quality scores higher than 30. Which makes most of the reads acceptable for the downstream analysis by Illumina standards.

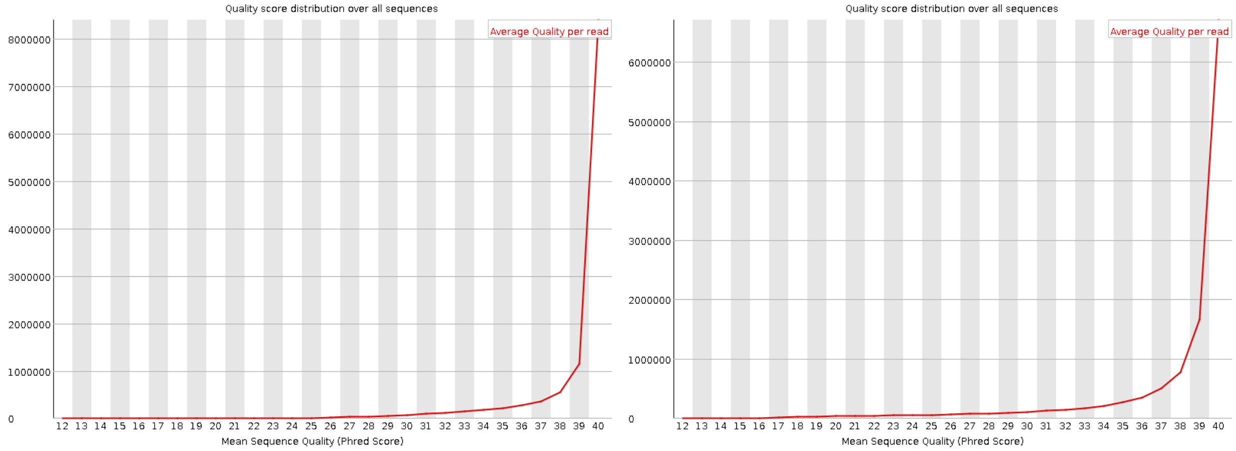

Figure 4: Quality scores per read. Most reads have quality scores 30 and higher.

Per base sequence content was flagged by the FastQC. The first 8 bases did not have balanced base diversity. That could be attributed to worse sequencing quality at the beginning of a read, and wrong base calls. This problem will be solved later by the downstream quality trimming by trimmomatic.

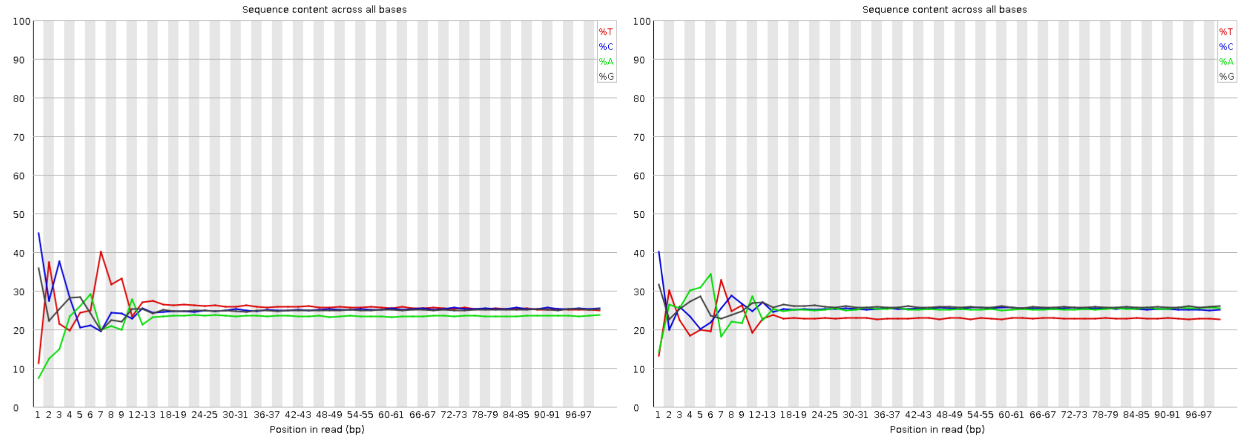

Figure 5: Per base sequence content. FastQC flagged this metric most likely because the beginning of each read don't have uniform base distribution

Duplication rate represents the number of same RNA molecules which includes levels of differential gene expression and PCR replicates. There is expected distribution of replicates where some transcripts present in smaller quantities, and some transcripts presented in larger quantities. Read 2 has slightly smaller number of duplicates because of the worse overall quality of the read 2.

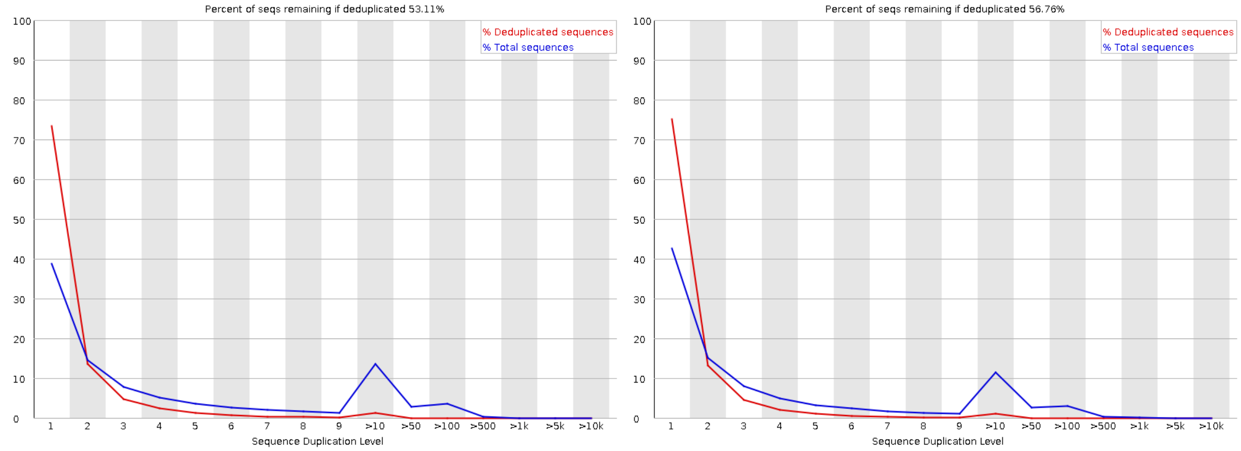

Figure 6: Duplication rates, read 1 and read 2. Read 2 has a slightly lower duplication rate bacause of lower quality scores less duplicates were detected.

K-mer composition shows that there are 2 k-mers that are over represented in the sequencing data. We will have to keep this in mind while doing the downstream analysis. Read 2 has higher number of unique k-mers which is in line with the lower quality scores for read 2.

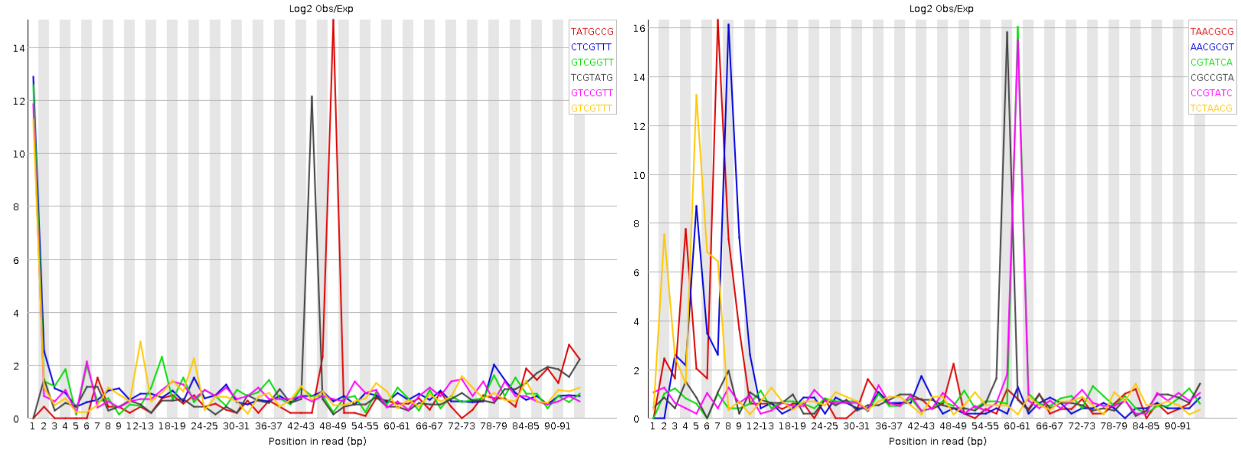

Figure 7: k-mer profile per read. There are spikes for 2 k-mers overrepresented in the sequencing data

Now I am going to briefly go over other graphs which were outputted by FastQC. I am not going to include them in this report because they are less relevant for the overall sequencing data quality assessment. All of the reads had length 101bp which is expected based on the sequencing kit used for this experiment. There was slight increase of N's at the beginning of some reads, however this will be fixed by quality trimming downstream. There were no over represented sequences in both reads sequencing data. Finally, the GC content was normally distributed with the mean around 50% of GC which is acceptable by Illumina standards.

Conclusions about "both" condition sequencing data quality.

Based on the graphs I presented above, I think that the sequencing data for the read and read 2, both condition are acceptable for the downstream RNA seq analysis. Most of the quality scores are 30 and higher. Additionally, the reads that have overall lower quality scores will be quality trimmed or removed with the downstream data processing.

Read quality assessment for forward and reverse reads, "control" condition.

The FastQC output for read 1 and read 2, control condition were very similar to the "both" condition and thus I am going to primarily focus of the quality per position and tile graphs and briefly mention all of the other metrics without presenting the graphs in this report. The graphs could be found in the git repository.

Quality per position is good for the control condition samples. Illumina guidelines say that acceptable quality scores are 30 and higher, and most of positions are in range between 31 and 39 for the read 1 and 30-38 for the read 2. The read 2 has predictably lower quality for the reasons, I described earlier.

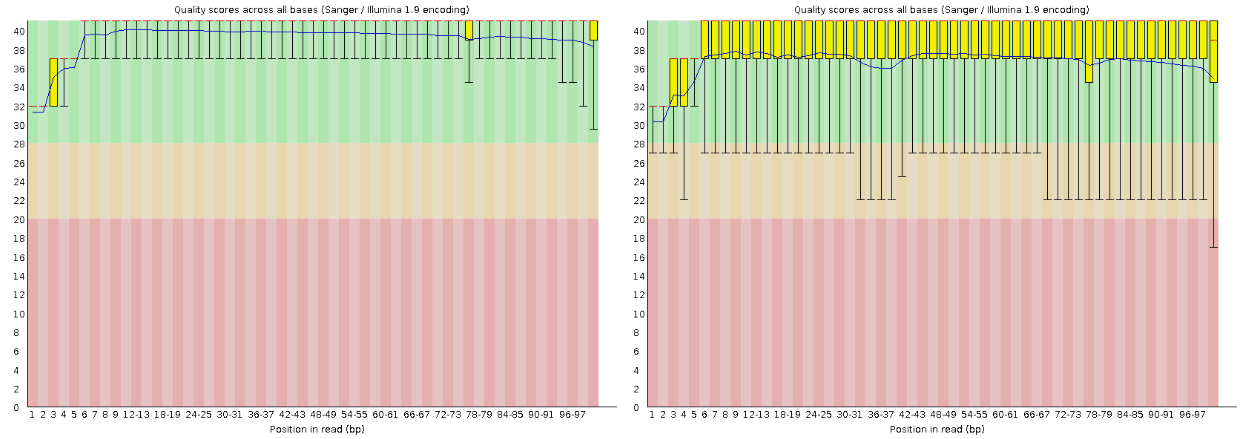

Figure 8: Quality score distributions by position on the read, both condition produced by FastQC. Read 1 shows better overall quality per base position than Read 2

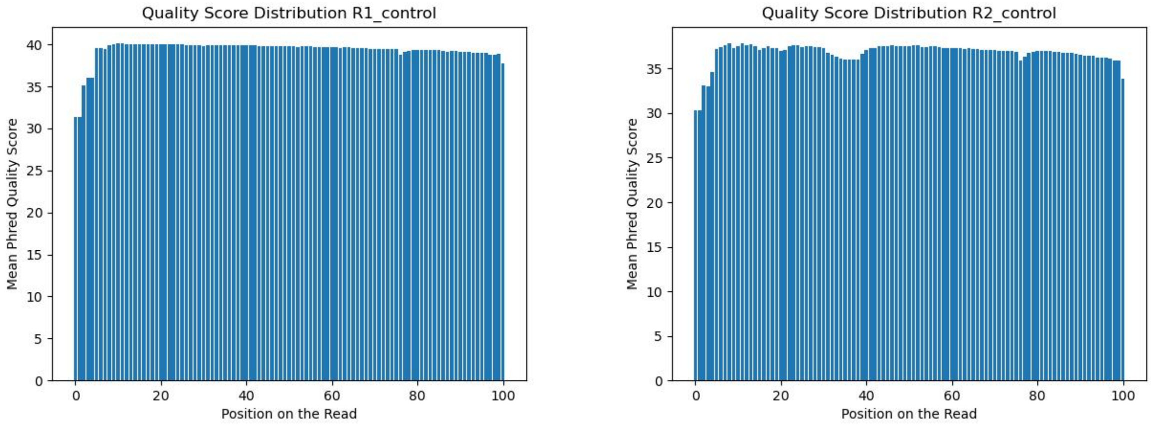

Figure 9: Quality score distributions by position on the read, both condition produced by my script. Read 1 shows better overall quality per base position than Read 2

Quality assessment per tile shows overall uniform sequencing quality except some tiles in read 1 and read 2. Additionally, there was potentially a bubble durind the read 1 cycles which doesn't seem to affect the overall data quality.

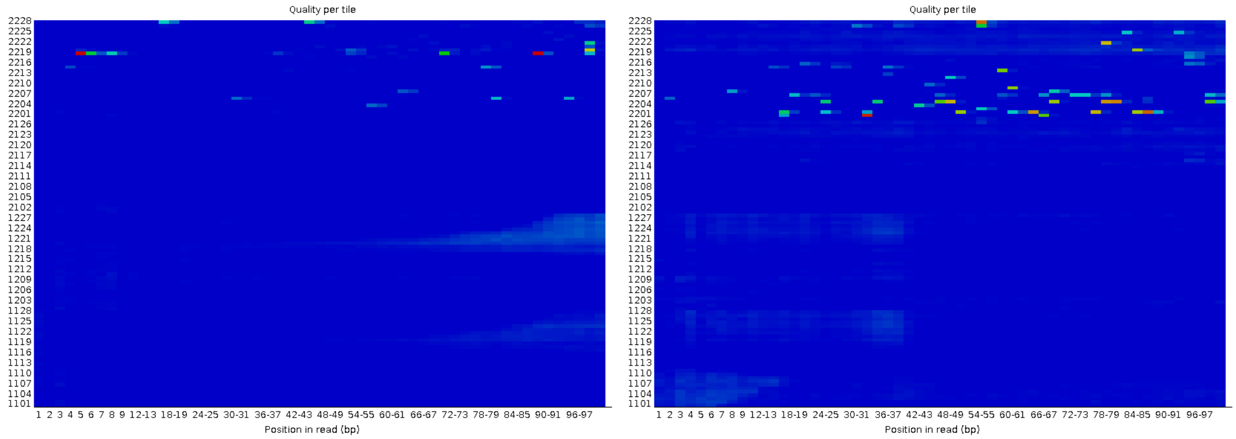

Figure 10: Quality assessment per tile on a flowcell for read 1 (left graph) and read 2 (right graph). Read 1 contains possibly a bubble in the middle of the view which is represented by lighter blue color.

Per sequence quality score graphs for both read 1 and read 2 show that most of the reads have average quality scores higher that 30, and there are very small counts of reads that have average quality smaller than 29. Per base sequence content graph shows similar to the 'both' condition phenomenon when the first 8-10 bases of each sequence don't have balanced base types. Per base N content graph shows slight increase of Ns at the very beginning of a read which most likely contribute to missed base calls and noise represented in the per base sequence content graph. Per sequence QC content is distributed normally with a peak at 50% which is ideal for the high diversity libraries. Sequence duplication levels represent various levels of duplication which is normal for RNAseq data.

Conclusions about "control" condition sequencing data quality.

Overall data quality per position and per read is acceptable by Illumina standards with all positions average quality score and most read average quality scores being higher than 30. Most of the tiles on the flowcell we evenly sequencesd, and otherwise, sequencing data quality FastQC output looks normal.

Part 2 – Adaptor trimming comparison

I used cutadapt to trim the adapters. Cutadapt takes forward and reverse file for each sample, find reads that contain adapter sequences, and trims them. If the resulting read is too short, this read and it's pair will be removed by the paired end cutadapt.

Summary table of the cutadapt output.

sample	read 1 removed(%)	read 2 removed (%)
both	5.3	6.1
control	3.2	3.9

After cutadapt, I ran trimmomatic which trims reads based on a quality score measured by a sliding window. If a read becomes shorter than 35bp it is removed and it's pair is put into the unpaired read folder. Since the FastQC and my python script outputs showed that the read 2 in both conditions haad overall worse quality than read 1, I expected it to be trimmed more than read 1. The graphs below depicting read lengths distributions confirm my prediction: although most length counts for both read 1 and read 2 are close to 101bp, read 2 has more counts than read 1 for the shorter reads.

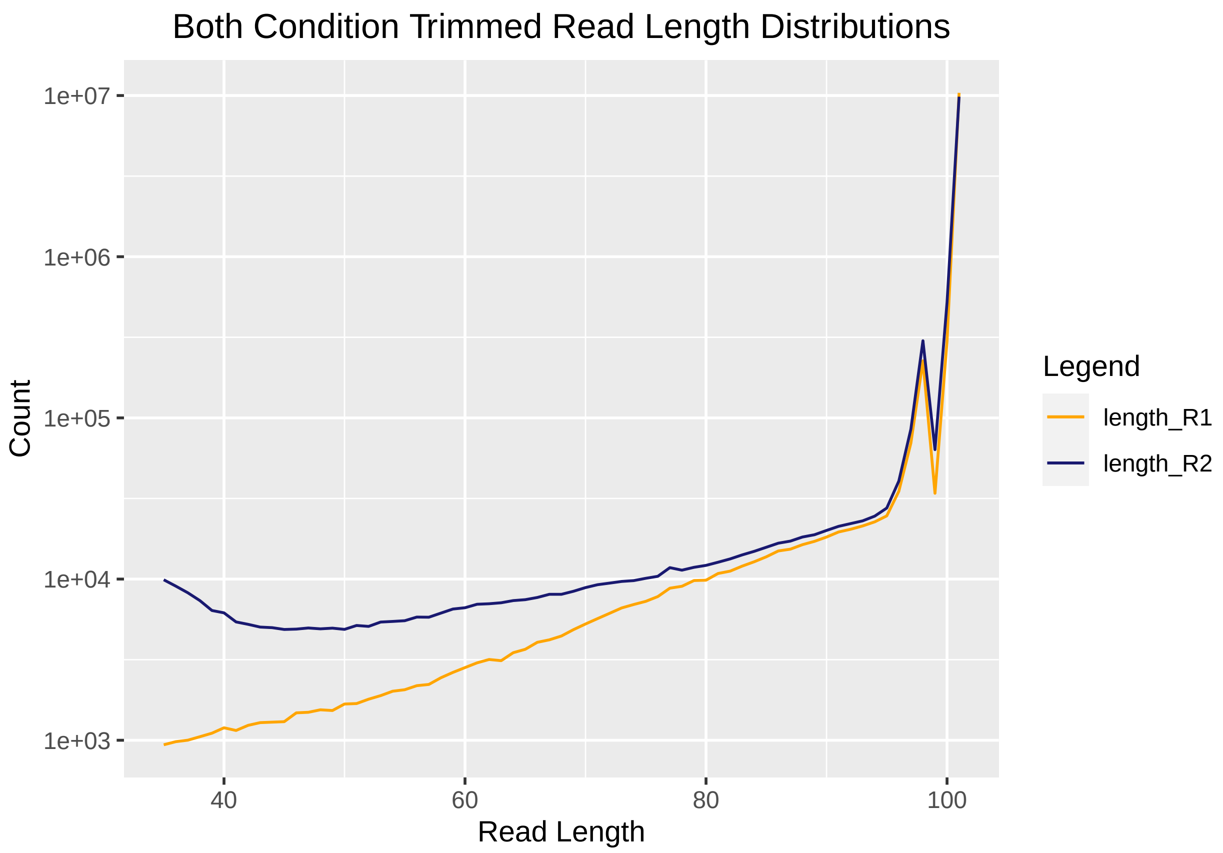

Figure 11: Read length distribution after trimming reads with cutadapt and trimmomatic for the both condition. Read 2 has higher counts for the lowed read lengths which means that it was trimmed more aggressively

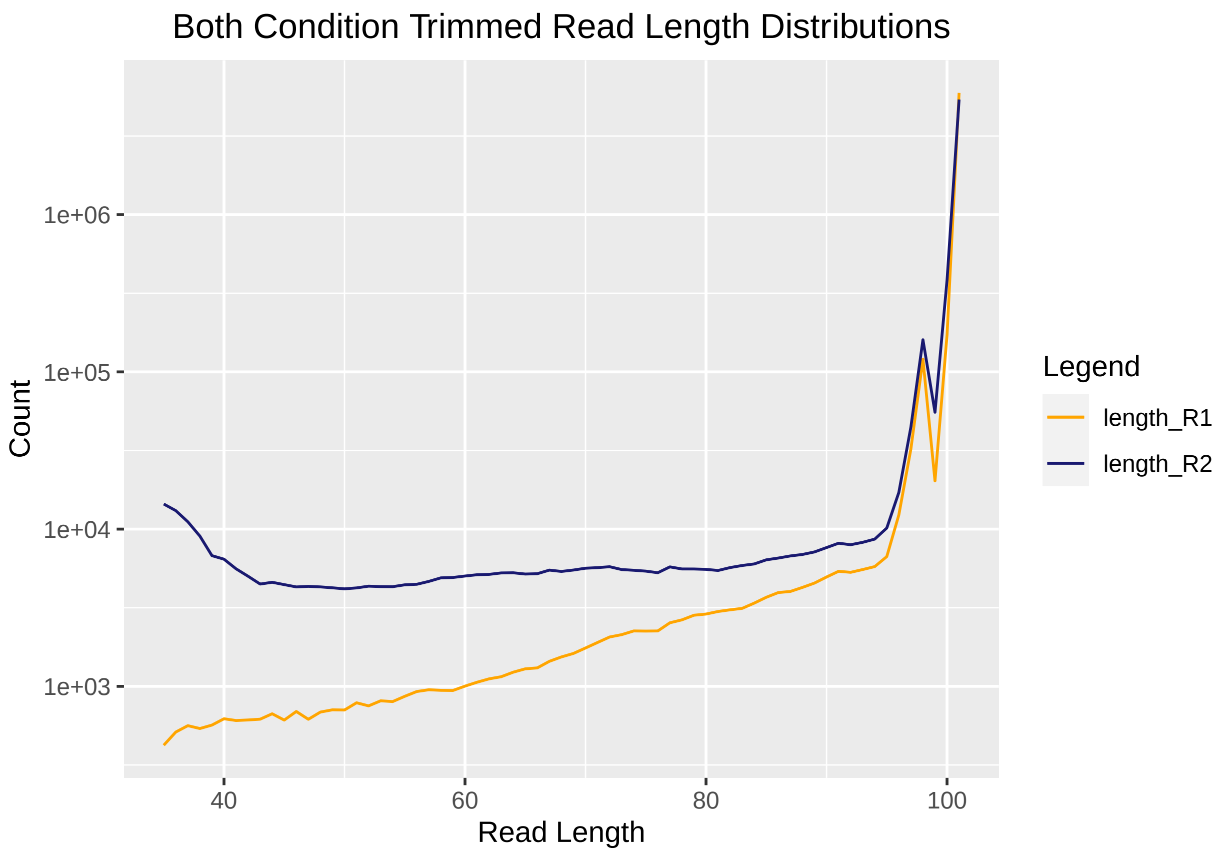

Figure 12: Read length distribution after trimming reads with cutadapt and trimmomatic for the control condition. Read 2 has higher counts for the lowed read lengths which means that it was trimmed more aggressively

Part 3 – Alignment and strand-specificity

After creating STAR database using a mouse genome Mus musculus DNA primary assembly, release-110 and the GTF files for the same assembly release, I aligned my data to the resulting database.

Summary of mapped and unpapped reads.

sample	count mapped reads	count unmapped reads
both	22404319	533613
control	12359959	496079

I ran htseq to count the number of reads mapped to each feature and determine strandedness of the kit.

Summary of read percents mapped to features

sample	(%) of reads mapped stranded yes	(%) of reads mapped reverse
both	3.9	86.4
control	3.7	81.8

The percent distribution between the outputs obtained from the stranded=yes mode and stranded reverse tells us that the kit used for the library prep was stranded and that the second strand on the cDNA was digested during the library prep process. If the kit was not stranded, we would have expected to see approximately equal distribution of percent mapped in standed yes and reverse mode. It happens because if the kit is not stranded, during the library prep we produce copies of both strands and the information about the original strand dirrectionality is not preserved. Oppositely, if the kit was stranded either first or second strand of the cDNA would be preserved. As a result either stranded yes which counts reads that align to a feature directly or stranded reverse which counts reads that are reverse to the feature will have most of the counts. In our case, most reads are aligning in the stranded reverse mode which indicates that the kit was indeed stranded and the second strand of the cDNA was removed during the library prep.