QAA_Report

Temi Adewunmi

2023-09-12

Contents

This assignment aims to use existing tools for quality assessment and adaptor trimming, compare the quality assessments to those from your own software, and create a summary of my findings. The files I used for this analysis and report are 11_2H_both_S9_L008_R1_001.fastq.gz, 11_2H_both_S9_L008_R2_001.fastq.gz and 14_3B_control_S10_L008_R1_001.fastq.gz,

14_3B_control_S10_L008_R2_001.fastq.gz. Further down the line, these will be called 11_2H_R1, 11_2H_R2 and 14_3B_R1, 14_3B_R2, respectively.

Part 1: Quality Score Distribution across all Files

First, we used fastqc, which generates a report that provides an overview of possible quality control metrics that can be used on our data. We then used our previously created code to generate plots that look at the mean score at each base pair.

11_2H

Figure 1 and **Figure 2** contain the quality scores across all the bases and N content across all bases for Read 1 and Read 2 of the 11_2H dataset. The 3rd figure holds the mean quality score content for Read 1 and Read 2, respectively. When comparing the fastqc-generated plots to ours, we see that the fastqc quality score plots have interquartile ranges while the ones we generated do not. Also, on the x-axis of the fastqc plot, the positions go in ranges, whereas in the self-generated plot, it does not group up the positions. Also, it took a shorter time for fastqc to generate the reports. The times it took to run for all files across the two methods are below.

Table 1: 11_2H Run Times for Fastqc and Demultiplex

	Reads	Fastqc	Demultiplex
User_Time1	Read 1	75.33	250.55
User_Time2	Read 2	75.28	238.36
System_Time1	Read 1	3.76	2.50
System_Time2	Read 2	3.97	2.44
Elapsed_Time1	Read 1	1.00	4.00
Elapsed_Time2	Read 2	1.00	3.00

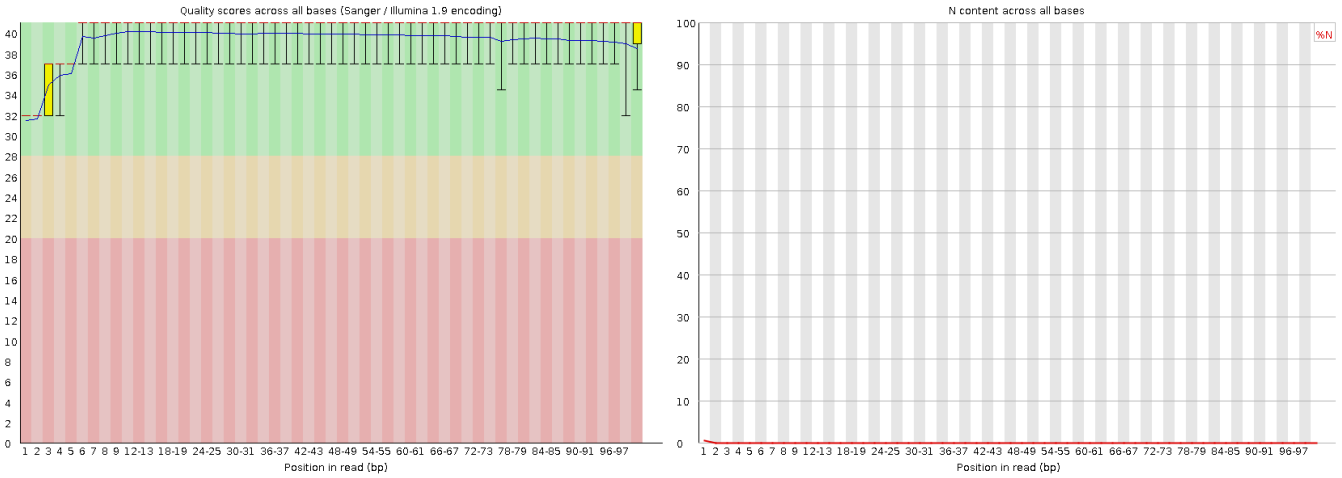

Figure 1: 11-2H-R1 fastqc per base mean quality score

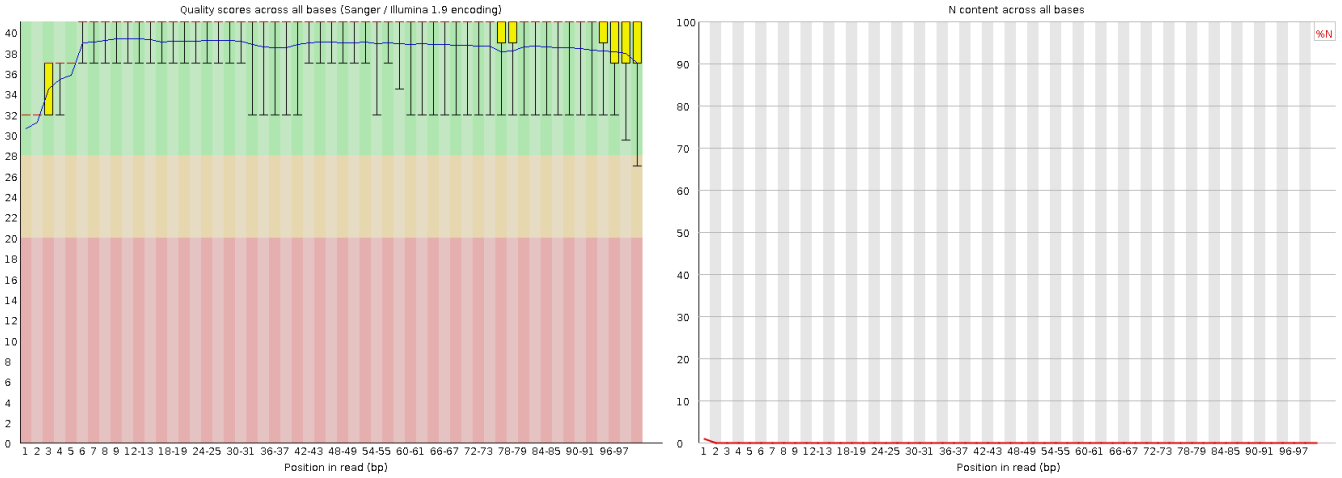

Figure 2: 11-2H-R2 fastqc per base mean quality score

When looking at the generated plots, it is safe to see that the overall quality of the reads is good. In the fastqc-generated plots, the lowest quality score value is about 32 for both reads and in the plots we generated, the lowest quality score value is the same, i.e. around 32. As we look at more sequences in all the plots, the quality score vastly improves, reaching stable around the 40-41 range, which is a very good quality score value for most reads. Also, the N base content in Figures 1 and 2B shows that there are practically 0 unknown nucleotides in both reads, which matches the quality score plots. So, the quality of our reads is high enough for it to be used for further analysis.

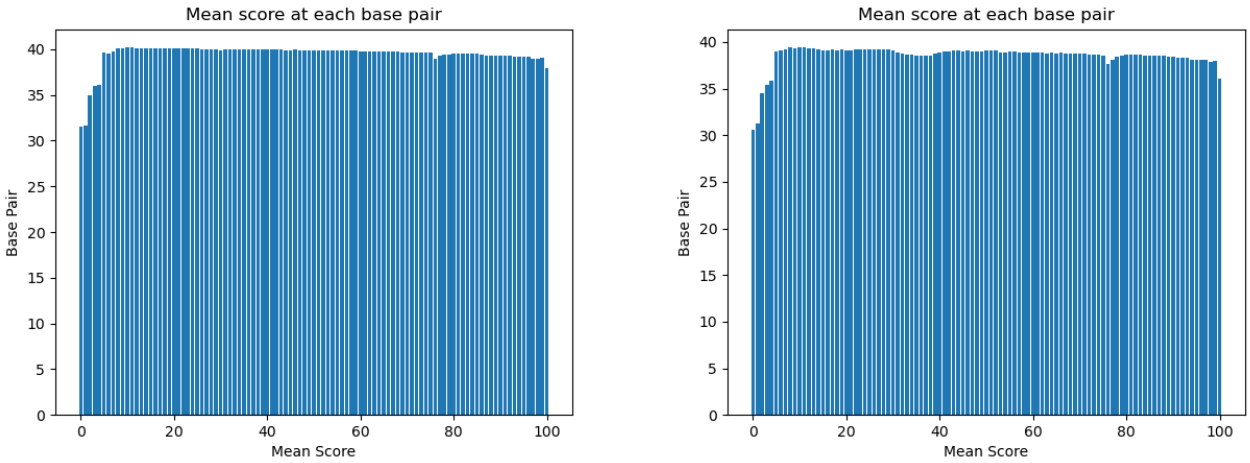

Figure 3: 11-2H Mean Score per base mean quality score. Read 1 and Read 2 respectively

14_3B

The plots generated for the 14_3B samples have the same differences as 11_2H above, as they were generated the same way. However, for Read 1, the fastqc plots show much lower quality score ranges on the interquartile plots than in 11_2H. The N base count plot here also shows barely any unknown nucleotides in our sequences. The run times for both plot generation processes are below.

Table 2: 14_3B Run Times for Fastqc and Demultiplex

	Reads	Fastqc	Demultiplex
User_Time1	Read 1	20.76	58.70
User_Time2	Read 2	21.46	58.48
System_Time1	Read 1	1.11	2.14
System_Time2	Read 2	1.18	2.29
Elapsed_Time1	Read 1	21.50	58.92
Elapsed_Time2	Read 2	22.15	58.43

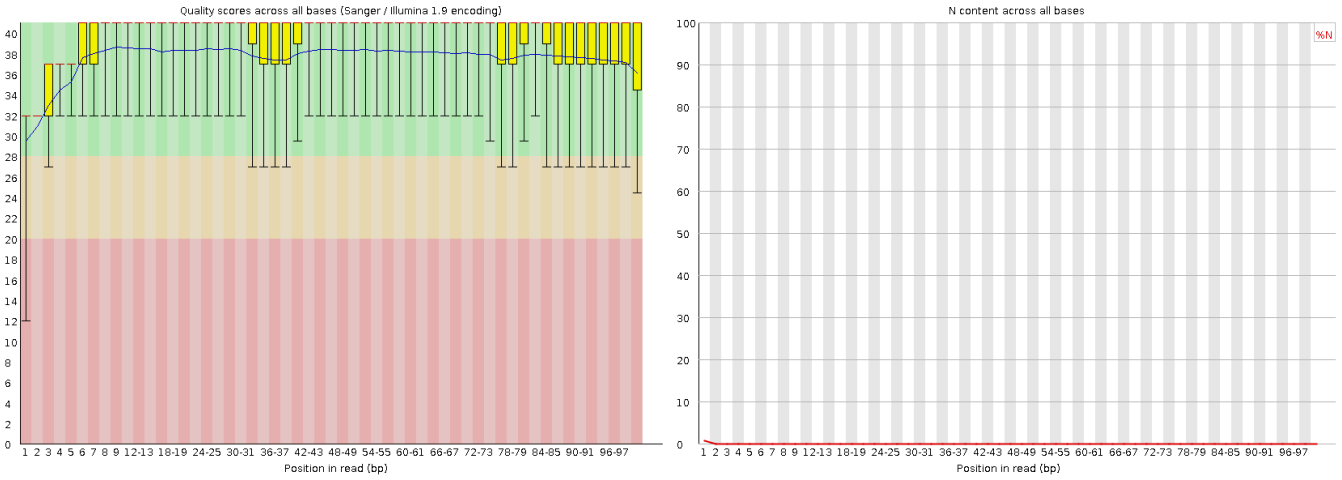

Figure 4: 14-3B-R2 fastqc per base mean quality score

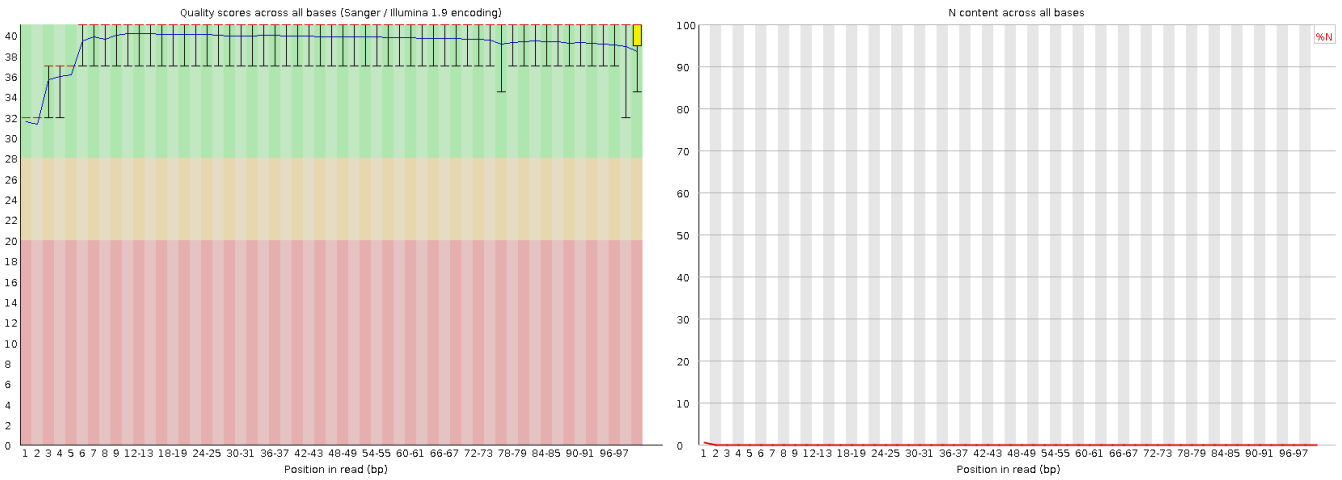

Figure 5: 14-3B-R1 fastqc per base mean quality score

When looking at the overall quality of our sequences, we see that the sequences in Read 2 have a lower quality than those in Read 1 in our generated plot. However, both reads still have good scores and are high-quality enough for further analysis.

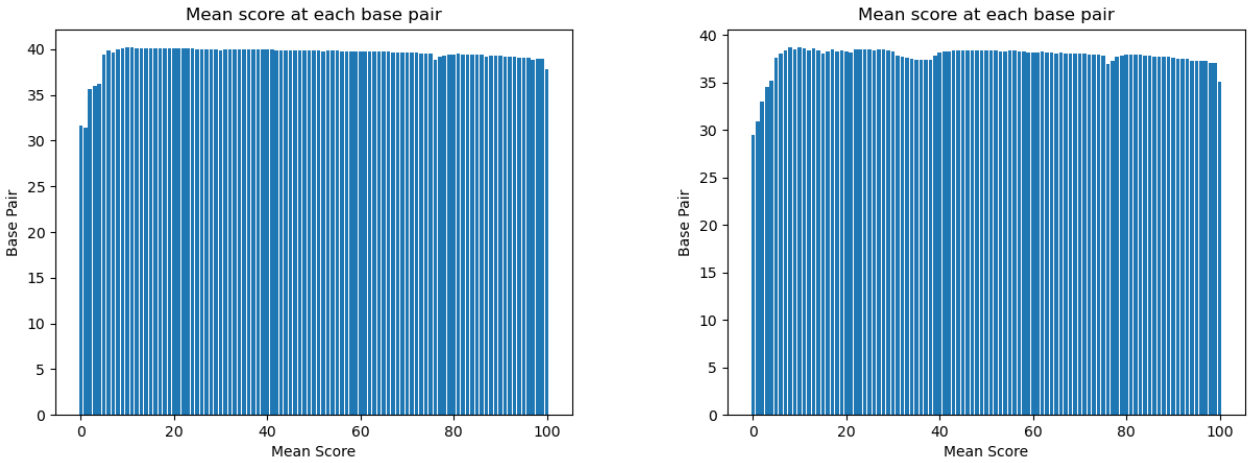

Figure 6: 14-3B Mean Score per base mean quality scoreRead 1 and Read 2 respectively

Part 2: Adaptor Trimming and Quality Checks

After ensuring that the reads were good enough for further analysis, I found the adaptors that needed to be trimmed from the sequences. These adapters were found from an illumina website and the same adaptors were used for both datasets.

To get the illumina adapters I looked here: https://knowledge.illumina.com/library-preparation/general/library-preparation-general-reference_material-list/000001314

Read 1: AGATCGGAAGAGCACACGTCTGAACTCCAGTCA

Read 2: AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT

In order to confirm that these were the correct adaptors, I used UNIX commands to check the sequences.

The commands return lines where the adapter exists in the file, so it is confirmed that the adapters are there. It also confirmed that the Read 1 adapters were only in the R1 files, and the Read 2 adapters were only in the R2 files.

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/11_2H_both_S9_L008_R1_001.fastq.gz
| grep 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCA'
```

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/11_2H_both_S9_L008_R2_001.fastq.gz
| grep 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCA'
```

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/14_3B_control_S10_L008_R2_001.fastq.gz
| grep 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCA'
```

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/14_3B_control_S10_L008_R1_001.fastq.gz
| grep 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCA'
```

After identifying the correct adaptors. `CUTADAPT` was used to remove the adapters and it resulted in this proportion reads being trimmed:

Table 3: Proportion of Trimmed reads

Samples	Read1	Read2
11_2H	4.9%	5.7%
14_3B	6.0%	6.7%

Following the adaptor trimming, `TRIMMOMATIC` was used to do more quality trimming. Which resulted in some of the sequences in our reads being shorter. The length of the reads after trimming can be seen in **Figure 7**. When looking at the plots of sequences across the samples, we see that the sequences in Read 2 are trimmed more extensively than those in Read 1. This is because Read 2 has a lower quality score distribution, which we can see from our figures above and with the parameters set for Trimmomatic, the package cuts low-quality base pairs in at the beginning and end, as well as any base pairs that have an average quality score below 15. So, we end up with shorter-length sequences for Read 2.

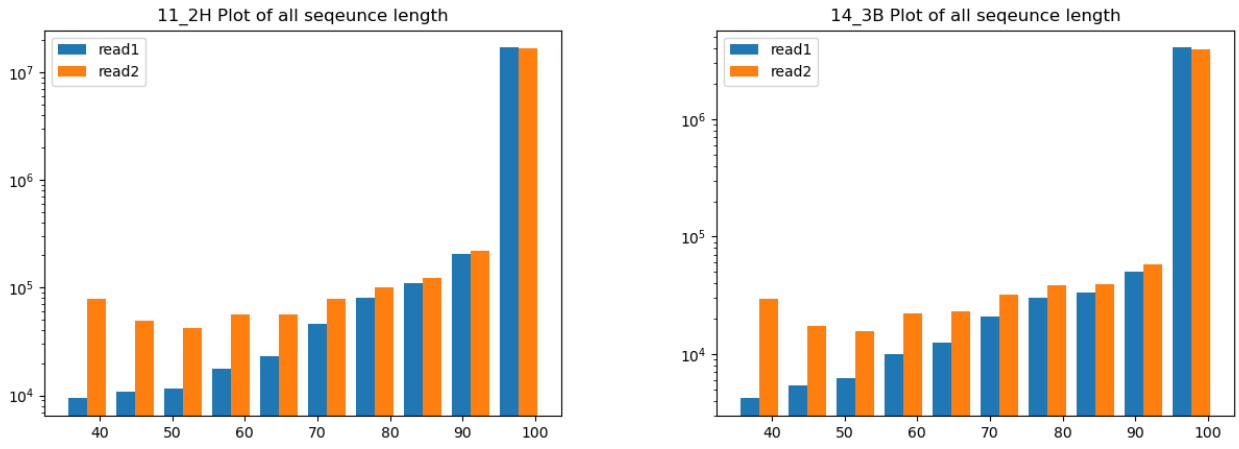

Figure 7: Read Lengths for Both sets samples

Part 3: Alignment and strand-specificity

After ensuring our samples have been appropriately filtered and trimmed, the next step would be to perform Mapping and Alignment. This was done using the STAR aligner and scripts written in a previous assignment.

The files used to create the reference genome were `Mus_musculus.GRCm39.dna_sm.primary_assembly.fa.gz`, `Mus_musculus.GRCm39.110.gtf.gz`, both downloaded from Ensembl. One of the outputs from this alignment was a SAM file. Before further analysis, we used previously written code to get the number of Mapped and Unmapped reads from our SAM file.

Table 4: Mapped and Unmapped Reads per Sample

Samples	Mapped	Unmapped
11_2H	33637672	1293554
14_3B	8312388	180916

For the next step, we used `htseq-count` on the SAM file to count reads that map to features. We ran it with 2 options –stranded=yes and –stranded=reverse and were able to determine whether our data was strand-specific or not.

Table 5: 11_2H Htseq Results

Stranded	Mapped	Total_Reads	Percent_Map
Yes	591682	17465613	3.38%
Reverse	13819087	17465613	79.12%

Table 6: 14_3B Htseq Results

Stranded	Mapped	Total_Reads	Percent_Map
Yes	164513	4246652	3.87%
Reverse	3666879	4246652	86.35%

Within HTSeq, the 'stranded' parameter presents a trio of choices. When employed without specification, it indicates whether a given read has aligned with any feature, regardless of whether it aligns on the same strand or its opposite counterpart. Opting for 'stranded=yes' as the parameter suggests that for paired-end reads to be tallied, the first read must align with the same strand as the feature, while the second read should align with the opposite strand. Conversely, the utilization of 'stranded=reverse' as the parameter inverts the criteria compared to 'stranded=yes.'

Looking at the tables above, we see an examination of the files using 'stranded=yes' unveiled that only 3.28% (11_2H) and 3.87% (14_3B) of the reads mapped accordingly. Conversely, 'stranded=reverse' yielded a substantially higher percentage, with 79.12% (11_2H) and 86.35% (14_3B) of reads aligning in a stranded manner. These findings lead me to conclude that the RNA-seq is indeed designed to be strand-specific.