QAA

Ross Ellwood

2023-09-15

Introduction

The objective of this report is use various quality control and adapter trimming software, compared to my own software to assess information about a RNA-Seq dataset. The data is being compared to a mouse (*Mus musculus*) reference genome. The input samples I will be using are:
- 15_3C_mbnl_S11_L008_R1_001.fastq.gz
- 15_3C_mbnl_S11_L008_R2_001.fastq.gz
- 24_4A_control_S18_L008_R1_001.fastq.gz
- 24_4A_control_S18_L008_R2_001.fastq.gz

Part 1

In this section we will use FastQC software, as well as my own software, to assess the quality of the data and to determine it it is sufficient for further analysis.

The following plots were generated using FastQC and depict both the per-base quality score distributions as well as per-base N content for each sample and read.

15_3C_mbnl_S11_L008_R1_001:

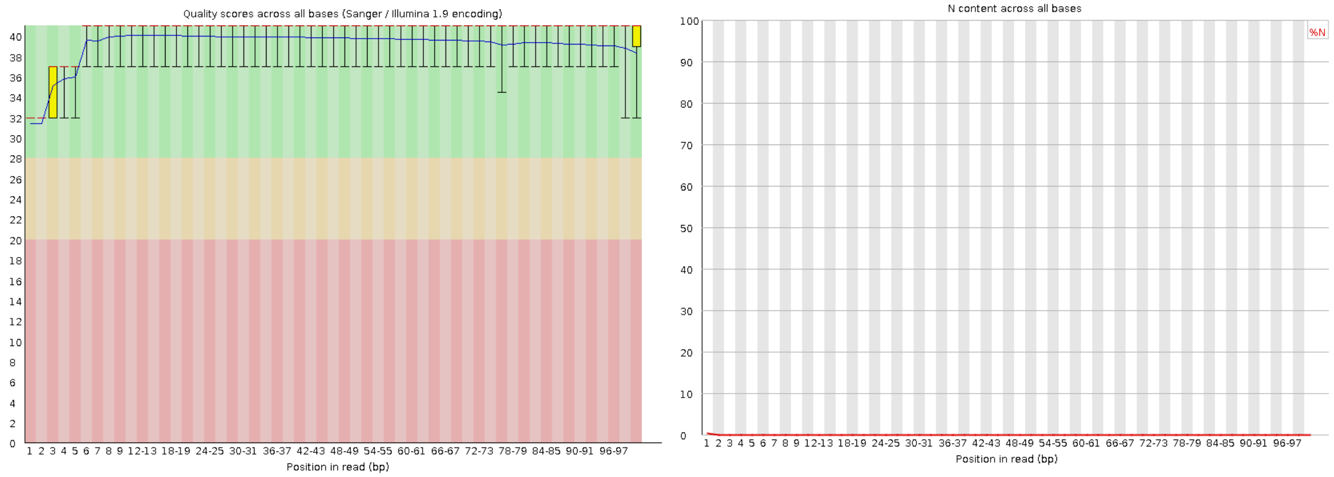

Figure 1: These two plots depict per-base quality score on the left and per-base N content on the right for the 15_3C_mbnl_S11_L008_R1_001 sample.

15_3C_mbnl_S11_L008_R2_001:

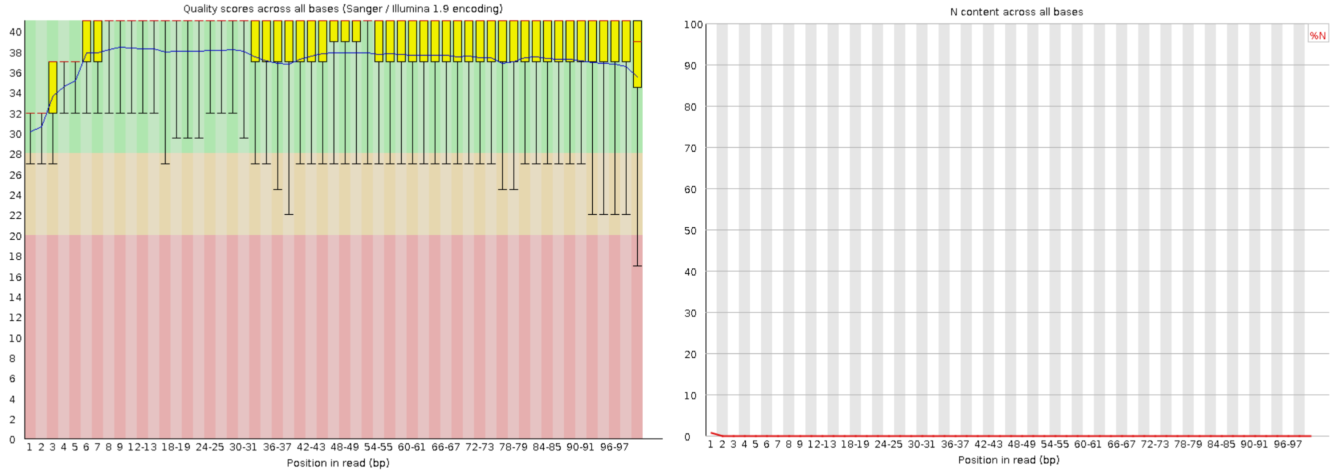

Figure 2: These two plots depict per-base quality score on the left and per-base N content on the right for the 15_3C_mbnl_S11_L008_R2_001 sample.

24_4A_control_S18_L008_R1:

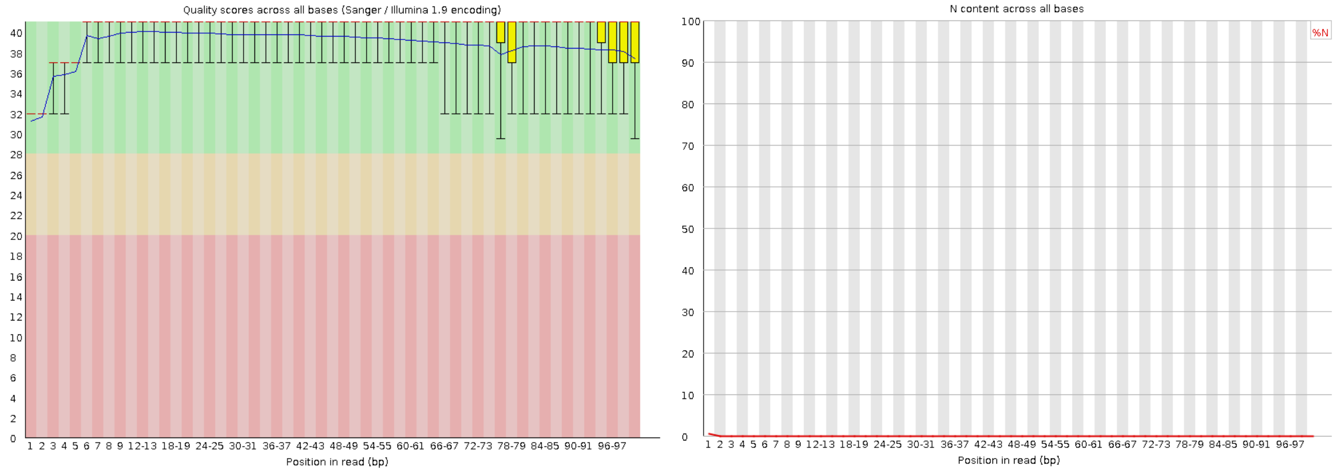

Figure 3: These two plots depict per-base quality score on the left and per-base N content on the right for the 24_4A_control_S18_L008_R1 sample.

24_4A_control_S18_L008_R2_001:

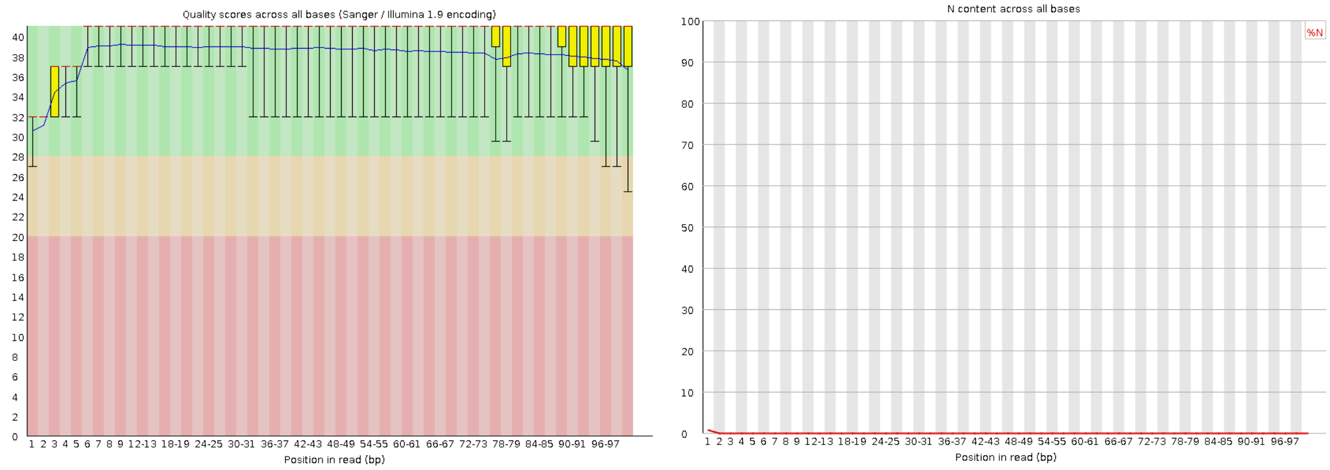

Figure 4: These two plots depict per-base quality score on the left and per-base N content on the right for

the 24_4A_control_S18_L008_R2_001 sample.

Overall, it appears that the per-base N content plots are consistent with the per-base quality score plots. For each sample, the quality score appears to be very high, often in upper 30s, and as a result, the N content is very low the whole time. These two graphs are very much related and consistent.

Next, I will utilize my own software to produce quality score distribution plots.

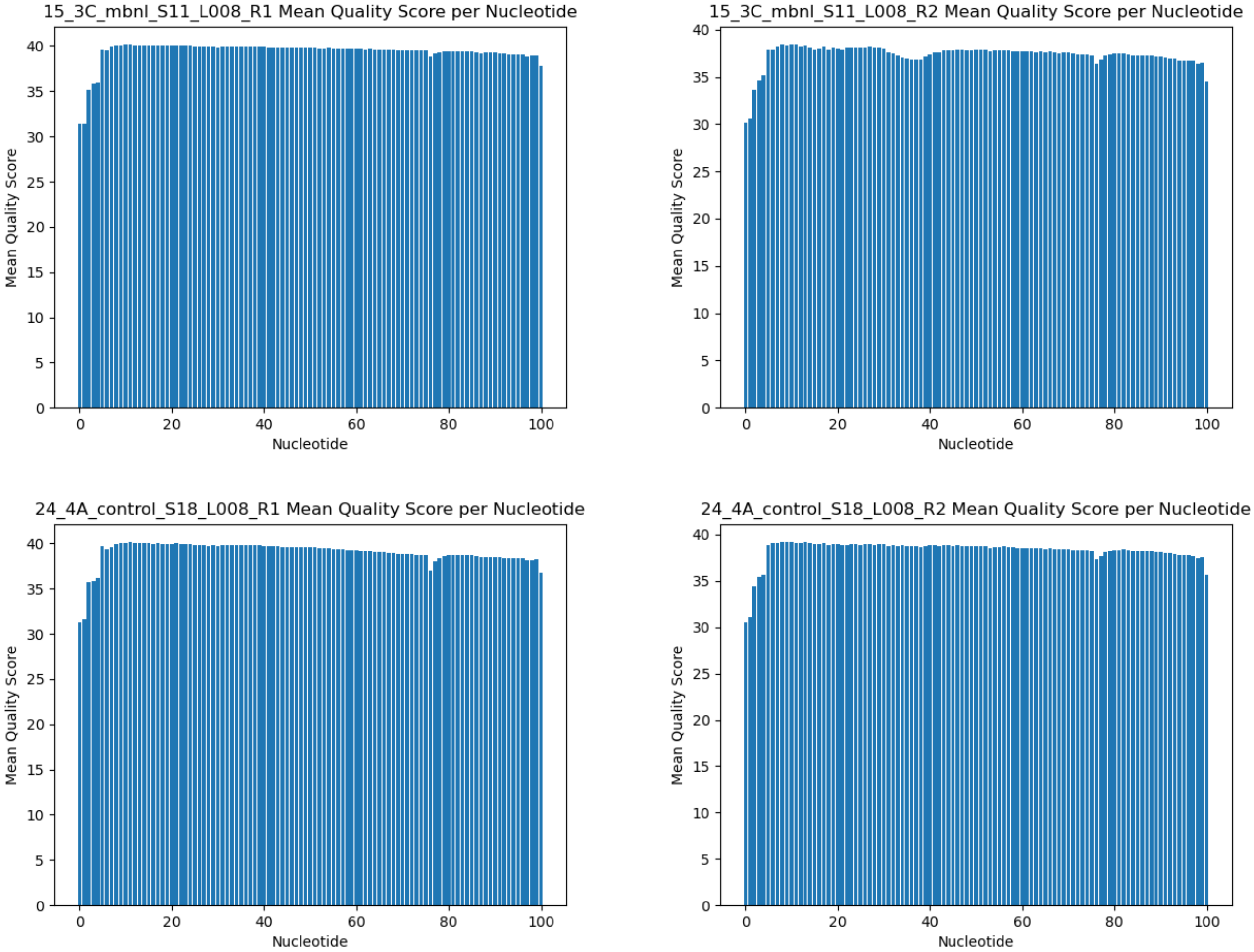

Figure 5: These plots were made with my own software to produce quality score distributions plots for each sample and each read.

When looking at these graphs, the first visible thing is the difference between read 1 and read 2. For both of the samples, the read 1 tends to have a higher quality score per nucleotide than the read 2. This makes sense that read 1 will have a higher quality because it is the first thing to be sequenced. Whereas when read 2 is sequenced, it has been on the sequence for a longer amount of time, so more mistakes are likely to have occurred. When comparing the two samples, their mean quality score seems to be very similar, and there is no visible difference between the two. Fortunately, the plots I have generated follow a seemingly identical trend as the plots generated by FastQC. This also makes sense due to the use of the same input data and a similar algorithm. Additionally, it appeared that the FastQC software ran more quickly than my algorithm. This is likely because FastQC has more of a powerful pipeline and is made to be more efficient than my code.

Overall Data Quality: Overall in the data, we can see a very high average quality score, although we often see lower quality scores in the beginning and and near the end of the read. With RNA-Seq, it is common to see a lower quality score in the first several nucleotide positions. Additionally, due to fragmentation during library prep, it is common we will see a 3' bias meaning that towards the end of the read, we will see a very slight decline in mean quality score per nucleotide. With all of this in mind, the data follows normal RNA-Seq trends and has very high quality scores overall. So, the data has high enough quality to use for further analysis.

Part 2

Next, we cut adapter sequences using cutadapt. The read 1 adapter is: AGATCGGAAGAGCACACGTCTGAACTCCAGTCA and the read 2 adapter is: AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT. From this we determined that the proportion of reads that are trimmed is for the 15_3C_mbnl_S11_L008 sample is 5.4% (417810 times) for read 1 and 6.1% (477359 times) for read 2. On the other hand, for the 24_4A_control_S18_L008 sample, the proportion of reads that trimmed for read 1 is 3.2% (335742 times) and for read 2 is 4% (417709 times).

For this distribution, I used a UNIX coommand to confirm these are the true adapters. Here is an example command I used to confirm this:

```
zcat 24_4A_control_S18_L008_R1_001.fastq.gz | grep "AGATCGGAAGAGCACACGTCTGAACTCCAGTCA" | wc -l
```

The output of this command told me that this adapter occurs 8025 times, meaning it is the correct adapter.

Trimmed Read Length Distribution

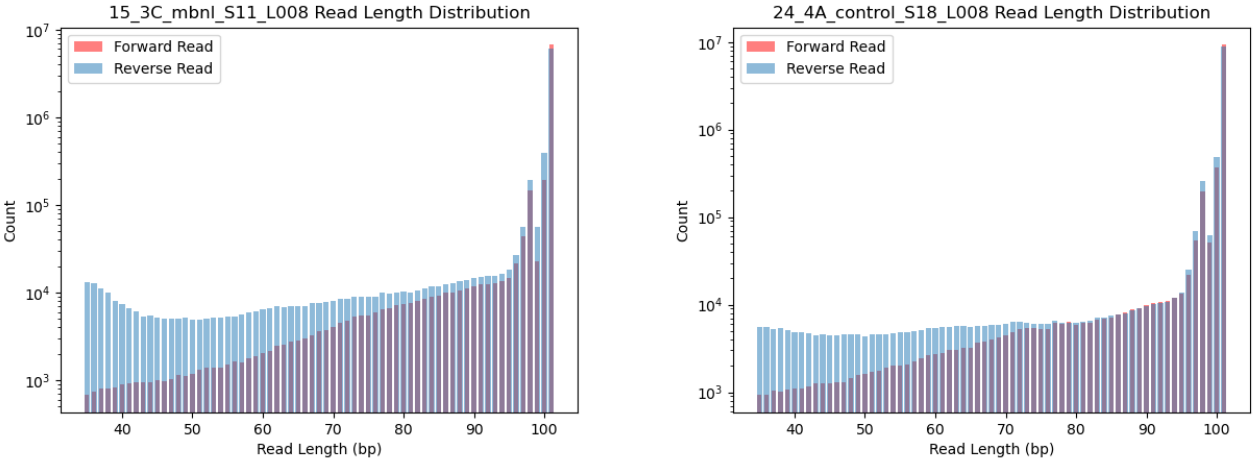

Figure 6: These plots show both of the samples and have the forward read in red and reverse read in blue. In general we can tell that the reverse reads have been adapter-trimmed more frequently.

To generate datasets for these two plots, I used various UNIX commands to sort the data. Here is the specific command used for read 2 of one of the samples:

```
zcat 15_3C_mbnl_S11_L008_R2_001.cut.trimmed.fastq.gz | sed -n ’2~4p’
| awk ’{print length($0)}’ | sort | uniq -c |
sort -n > ../readLenDist15_3C_mbnl_S11_L008_R2.txt
```

In both of these plots, we can see that the reverse read will have a greater abundance of shorter read sizes, although both the forward and reverse reads have a similar abundance of larger read sizes. From this, we can expect that reverse reads are adapter-trimmed at a greater rate. This is because the reverse reads show a greater distribution of short reads meaning that more of the reverse reads were trimmed. The reverse read will be adapter-trimmed more frequently because the reverse read will tend to have more mistakes and tend to have more adapters that need to be trimmed.

Part 3

Now we will report the number of mapped and unmapped reads from each of the two .sam files.

Using my own software:

Samples	Mapped	Unmapped	total	Percentage of Read Mapped
15_3C_mbnl_S11_L008	14436368	400406	14836774	97.30126
24_4A_control_S18_L008	19780620	710244	20490864	96.53385

Figure 7: This table shows both samples and the amount of reads that are mapped and unmapped, as well as their percentages.

Using htseq Now we will count the reads that map to features using the software htseq. We will specify whether or not we used the htseq paramter of stranded=yes or stranded=reverse, or forward or reverse, respectively.

Samples	Forward	Reverse
15_3C_mbnl_S11_L008	3.66%	83.11%
24_4A_control_S18_L008	3.42%	81.76%

Figure 8: This table shows each sample and the percentages of reads that mapped to the forward and reverse strand.

Using htseq, we included the $--stranded$ argument to determine whether the data is from a strand-specific assay. There are three options we could have chosen: $--stranded =< yes, no, stranded >$ and we chose to just use *yes* and *stranded*. Our results showed that for both of the samples, a small percentage (~3%) fell in the *yes* category and a large percentage are in the *reverse* category (~80%).

For the 15_3C_mbnl_S11_L008 sample, I propose that these data are strand-specific because 83.11% of the reads are on the reverse strand. This is because the majority of reads belong to the reverse-strand making it a stranded library. Similarly, for the 24_4A_control_S18_L008 sample, I propose that these data are also strand-specif because 81.76% of the reads are on the reverse strand, implying it is strand-specific.