QAA Bi623

Bailey Knight

2023-09-11

Objective

The objectives of this assignment are to use existing tools for quality assessment and adaptor trimming, compare the quality assessments to those from your own software, and to demonstrate your ability to summarize other important information about this RNA-Seq data set in a high-level report. Questions for this assignment are answered in the discussion sections.

Background

We are given two paired end RNA seq reads from the 2017 bgmp RNAseq run. The files have already been demultiplexed. the Data for all samples can be found in Talapas path "/projects/bgmp/shared/2017_sequencing/demultiplexed/ The specific samples used for this report are: 8_2F_fox_S7_L008 and 14_3B_control_S10_L008. Each has a Read 1 and Read 2 file associated with it.

General Workflow

The high level workflow is as follows:
1. fastqc for quality reports
2. Custom python script comparison to fastqc
3. Trim adapters from reads using cutadapt
4. Trim reads based on quality using trimmomatic
5. Made script to plot read length distribution
6. Make database using STAR software
7. Align reads using STAR
8. Count mapped and unmapped using python script
9. Count genes mapped using htseq count for forward and reverse
10. Count total genes mapped

For a detailed account of the work see my lab notebook on Talapas below:
/projects/bgmp/bailey/bioinfo/Bi623/Assignments/QAA/labnotebook.txt
Ensemble fasta: https://ftp.ensembl.org/pub/release-110/fasta/mus_musculus/dna/Mus_musculus.GRCm39.dna.primary_assembly.fa.gz Ensemble GTG: https://ftp.ensembl.org/pub/release-110/gtf/mus_musculus/Mus_musculus.GRCm39.110.gtf.gz

Part 1- Data Quality Investigation

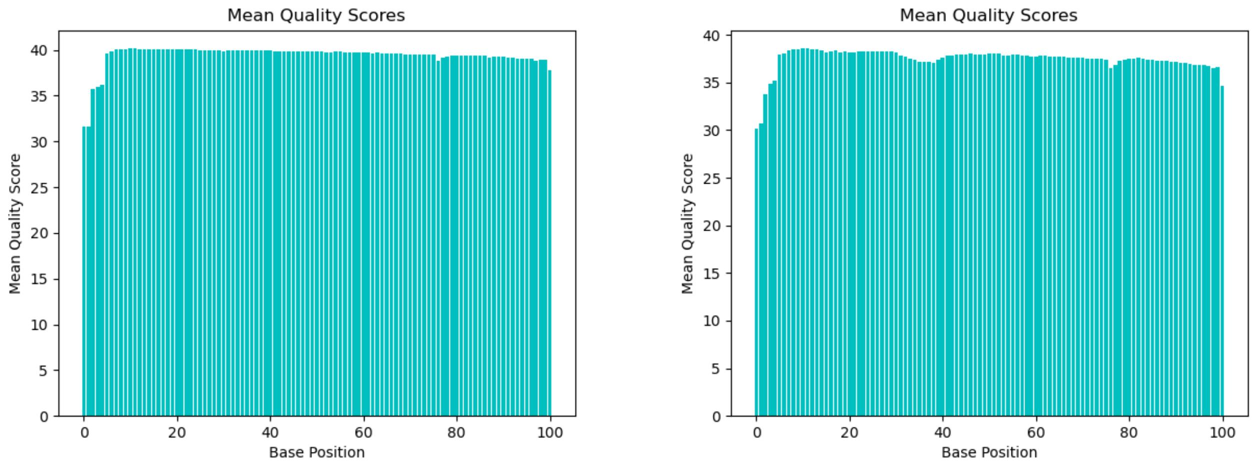

Figure 1: Mean Quality Scores Per base position for each read of 8_2F_fox_S7_L008 Read 1 (left) and Read 2 (right). Plots were generated with custom python script.

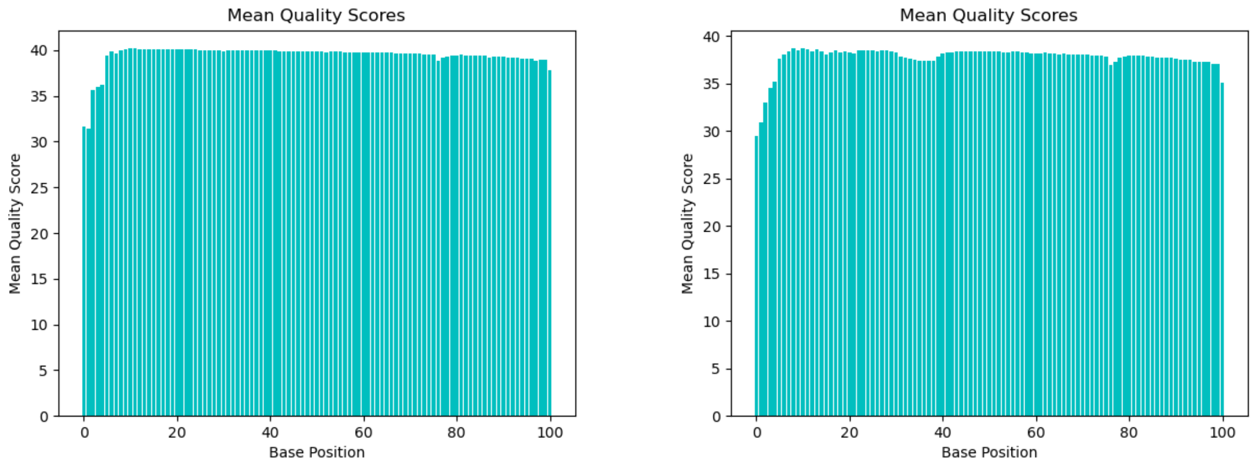

Figure 2: Mean Quality Scores Per base position for each read of 14_3B_control_S10_L008 Read 1 (left) and Read 2 (right). Plots were generated with custom python script.

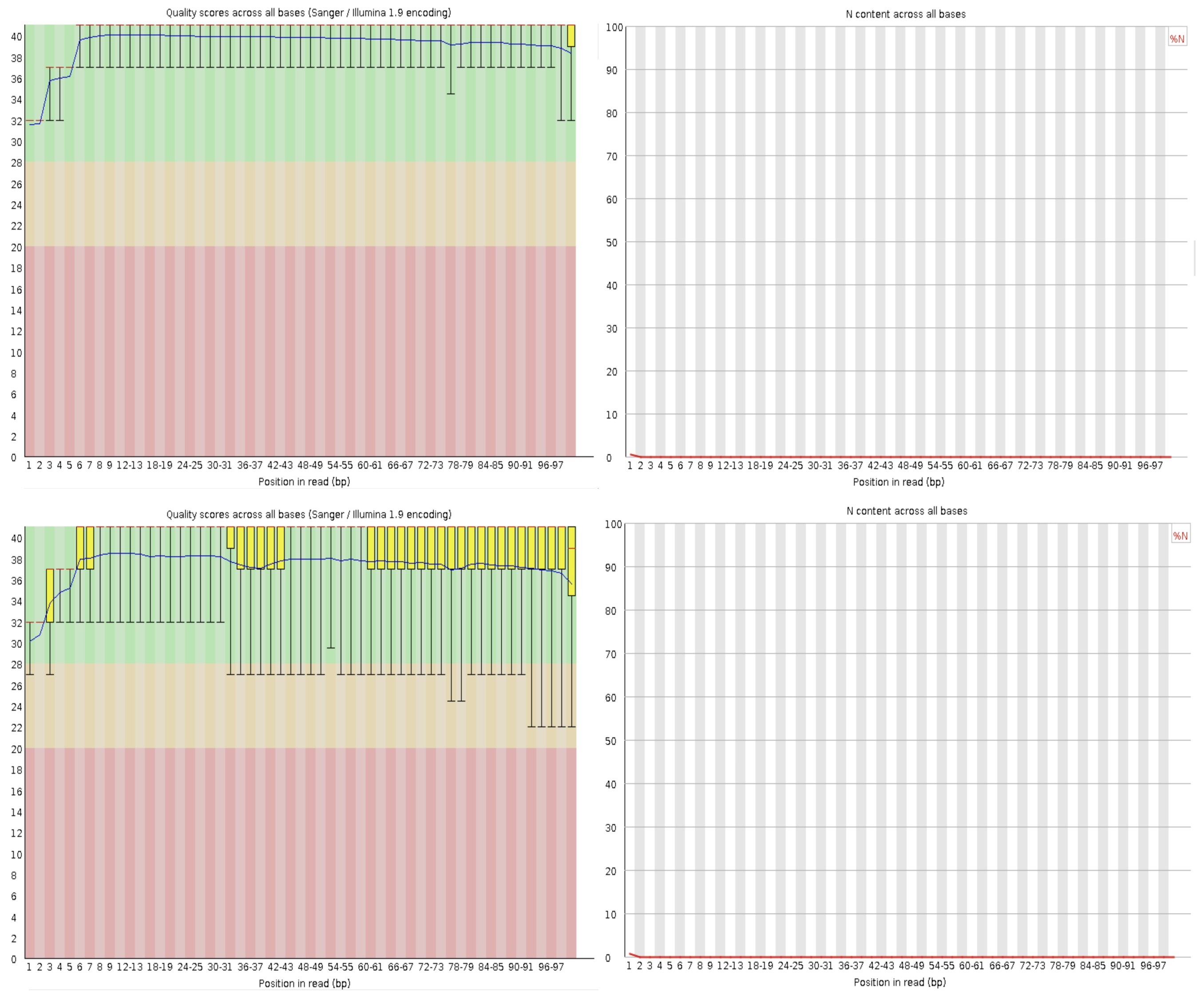

Figure 3: Left figures represent Quality Score Frequency per base position for sample 8_2F_fox_S7_L008 generated by Fastqc. Trend line follows the mean. Error bars represent data between 10th and 90th percentile. Box plots represent interquartile range. Right figures represent average N's found at each read position demonstrating a poor read at that position. Top figures represent Read 1 and bottom figures represent Read 2.

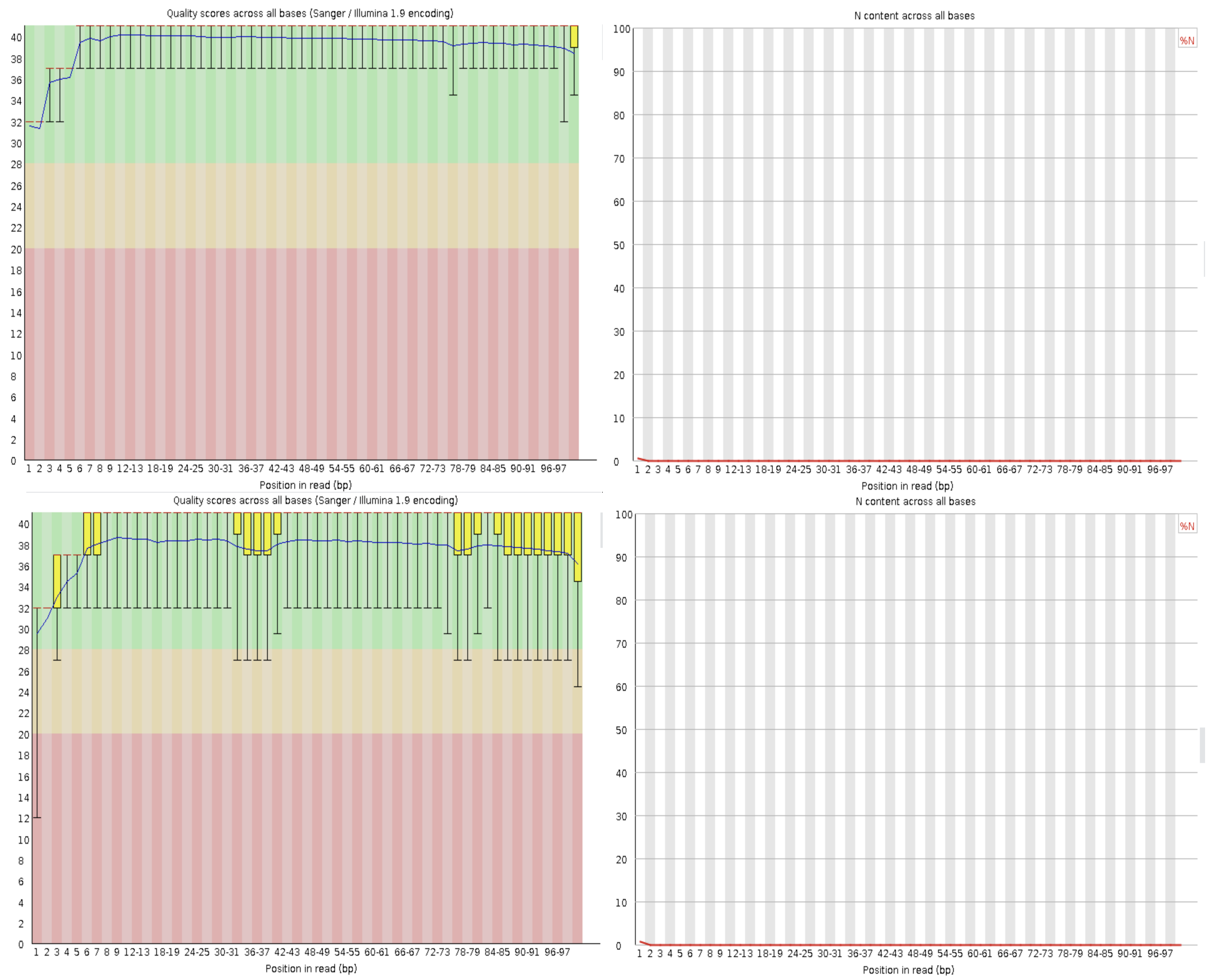

Figure 4: Left figures represent quality Score Frequency per base position for sample 14_3B_control_S10 generated by Fastqc. Trend line follows the mean. Error bars represent data between 10th and 90th percentile. Box plots represent interquartile range. Right figures represent average N's found at each read position demonstrating a poor read at that position. Top figures represent Read 1 and bottom figures represent Read 2.

Part 1 Discussion

14_3B plots shape look similar between the processing technique I coded vs fastqc. The plot type (mine being a histogram) vs fastqc line with error bars is different. The overall trends are pretty much identical with the max and mins occurring in the same spots on graphs. The max was around 40 for both processing techniques for the 14_3b data. The same conclusions can be drawn from the comparisons of the 8_2F data. In terms of processing time, fastqc was significantly faster and produced more plots. It took 3 minutes for the 8_2F_fox_S7 reads and 30 seconds for the 14_3B_control_S10 reads. Compared to my custom script which took 9 minutes and 3 minutes for the samples respectively. Overall due to the quality scores generally being greater than 35 I would recommend this data to be used. Although read data is more variable this is expected considering the nature of the sequencing. Proceed with greater caution with 14_3B_control as the per tile sequence quality had a streaked area for concern on the flow cell. Although this is a cause for concern, considering the overall quality score distribution the data can move forward for preprocessing and mapping. If data has poor results for mapping, it may be worth to filter out more poor quality cells.

Part 2 – Adapter and Quality-based Trimming

Table 1: Adapter Trimming using cutadapt for read pairs of both samples processed and written using cutadapt displayed in both read pairs and total basepairs.

Sample	Reads.Processed	Reads.Written	Percent.Read1.Trimmed	Percent.Read2.Trimmed	Percent.Written
14_3B_control_s10	4440378	4440378	6.0	6.7	100
8_2F_fox_S7	36482601	36482601	5.9	6.6	100

Table 2: Quality Trimming using trimmomatic for read pairs of both samples. Pairs processed using trimmomatic displayed. Less than 1% reads dropped for both read pairs.

Sample	Iput.Read.Pairs	Surviving.Read.Pairs	Percent.Both.Surviving	Percent.Only.Forward	Percent.Only.Reverse	Percent.Dropped
14_3B_control_s10	4440378	4246652	95.64	4.11	0.07	0.18
8_2F_fox_S7	36482601	34791157	95.36	4.47	0.08	0.09

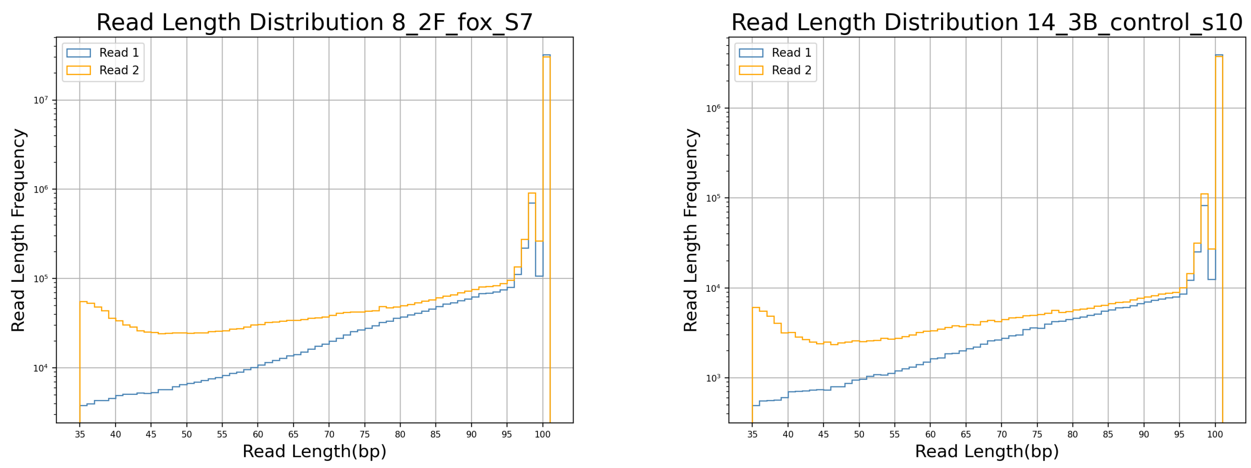

Figure 5: Read Length distributions for two Paired End RNAseq samples, 8_2F_fox_S7 and 14_3B_control_s10. Samples were adapter trimmed with cutadapt. Sequences quality trimmed with Trimmomatic.

Part 2 Discussion

Cutadapt was used followed by Trimmomatic for adapter trimming and quality filtering. Setting for each can be found in labnotebook.txt referenced in "General Workflow" section of the report. Adapter trimming trimmed 6% and 5.9% of sequences for reads 1 and 2 respectively for 14_3B_control_s10. Adapter trimming trimmed 6.7% and 6.6% of sequences for reads 1 and 2 respectively for 8_2F_fox_S7. Trimmomatic quality filtered out a total of 0.09% and 0.18% reads for 8_2F_fox_S7 and 14_3B_control_s10 resepctively. The adapter trimming for both reads was nearly identical for the forward and reverse reads as expected since they are compliments. The quality forward reads were trimmed at a higher rate than the reverse reads by nearly two orders of magnitude. Read 2 had consistently greater frequency of low length reads than Read 1 as expected considering idle time for read 2 on the sequencer while read 1 is being read.

Part 3 Alignment and Mapping

Table3: Percent mapped reads for custom python script. Secondary alignments were not counted as part of the total value for Mapped reads.

Sample	Mapped	UnMapped	Percent.Mapped
14_3B_control_s10	8312388	180916	97.86990
8_2F_fox_S7	67070899	2511415	96.39073

Table 4: Post-mapping analysis of htseq parameters for read counts that map to features. Stranded parameters used were "–stranded=yes" and "–stranded=reverse". Percents represent percent of reads that mapped to features using the parameters previously specified. Command used to generate percentages: awk '{total+=$2} $1~"___" {sum+=$2} END {print (total-sum)/total*100}' filepath.

Sample	Percent.Mapped.htseq.Reverse	Percent.Mapped.htseq.Yes
14_3B_control_s10	86.3475	3.87395
8_2F_fox_S7	80.5948	3.62400

Part 3 Discussion

Mouse Genome fasta and GTF files were acquired from ensemble. Links can be found under the "General Workflow" section. A database was generated using STAR software and the two files mentioned previously. The trimmed and filtered sequences of each sample were then aligned against the database using the STAR software. A SAM file was generated from the alignment that was then run through a custom python script. The script counted mapped and unmapped reads while ignoring any seconday alignments. Htseq was run on the SAM files using "–stranded=yes" and "–stranded=reverse" and mapping percentages were calculated. "stranded–reverse" yielded 80.6% and 86.9% mapped compared to 3.6% and 3.9% mapped for 8_2Freverse and 14_3b_control respectively. This indicates the RNAseq run is paired end and stranded run because the "reverse" option takes read order and orientation into account during comparison.