Brendan Kearney

Jan 26, 2020

BINF 6203 Genomics

Lab 1: Next Generation Sequencing Quality Control

INTRODUCTION

Analyzing sequenced data is a major component of research and discovery in the field of bioinformatics. Despite major advances in sequencing technology, the genomic data must still undergo quality control before it is ready to be studied. Quality control seeks to lessen the abundance of errors, repeated sequences, and adapters, which can all contribute to suboptimal sequencing data. While trimming data is a necessity in quality control, there are also potential drawbacks with trimming too aggressively, which can lead to equally unusable sets of data. Using genomic data from Illumina, the most common next-gen sequencing platform, trimming techniques were applied in order to simulate how the data would be prepared for further study.

The tools used for cleaning the genomic data in this lab were:

- SRAToolkit - for downloading and retrieving fastq files.
- FastQC - for visualizing the quality of the sequence data.
- Trimmomatic (Version 0.39) - for cleaning both single and paired-end reads. The ability to control the order of steps and the ability to repeat steps makes it a very powerful trimming tool.

In general, the order of trimming processes is first removing adapter regions, removing low quality reads, and then trimming above or below certain sequence lengths. For this lab, four data sets were analyzed and trimmed:

1. **“Ecoli200”** - a paired-end *E. coli* genomic sequencing run
2. **“SRR1391072”** - *V. vulnificus* paired-end transcriptome
3. **“SR109-3B2”** - Paired-end whole genome shotgun data
4. **“ERR3650066”** - Non-coding RNA data

The entire list of inputs and commands can be found in the methods section. The quality score graphs of each dataset before and after each trim can be found in the results section. Other relevant metrics may be included when certain commands are used – e.g., a % adapter content graph before and after using ILLUMINACLIP. An analysis and justification of each of the parameter decisions and the outcomes can be found in the discussion section.

METHODS

The following scripts were run on the default macOS Catalina Bash Unix shell. The code for each of the four datasets (**1-4**) are separated.

E. coli

Source: Download from web

1A. Before trimming

1B. Drop reads below 95bp

```
     java -jar trimmomatic-0.39.jar PE ecoli200_fwd_paired.fastq
ecoli200_rvs_paired.fastq ecoli_out_paired.fastq ecoli_out_fwd.fastq
ecoli_out_rvs.fastq ecoli_out_unpaired.fastq MINLEN:95
```

1C. Remove bases after 84

```
     java -jar trimmomatic-0.39.jar PE ecoli200_fwd_paired.fastq
ecoli200_rvs_paired.fastq ecoli_out_paired.fastq ecoli_out_fwd.fastq
ecoli_out_rvs.fastq ecoli_out_unpaired.fastq CROP:84
```

1D. Drop reads and remove from the end of reads

```
     java -jar trimmomatic-0.39.jar PE ecoli200_fwd_paired.fastq
ecoli200_rvs_paired.fastq ecoli_out_paired.fastq ecoli_out_fwd.fastq
ecoli_out_rvs.fastq ecoli_out_unpaired.fastq CROP:84 MINLEN:95
```

1E. Crop, then drop reads

```
     java -jar trimmomatic-0.39.jar PE ecoli200_fwd_paired.fastq
ecoli200_rvs_paired.fastq ecoli_out_paired.fastq ecoli_out_fwd.fastq
ecoli_out_rvs.fastq ecoli_out_unpaired.fastq CROP:84 MINLEN:95
```

V. vulnificus transcriptome

Source:

```
   fastq-dump --split-files SRR1391072
```

2A. Before trimming

2B. Remove Illumina adapters, provided in directory

```
     java -jar trimmomatic-0.39.jar PE SRR1391072_1.fastq SRR1391072_2.fastq
SRR1391072_1_pair_out.fastq SRR1391072_1_unpair_out.fastq
SRR1391072_2_pair_out.fastq SRR1391072_2_unpair_out.fastq
ILLUMINACLIP:adapters/TruSeq3-PE.fa:2:30:10
```

2C. Remove adapters, inspect read with a 4-base window, remove when quality is below 15

```
     java -jar trimmomatic-0.39.jar PE SRR1391072_1.fastq SRR1391072_2.fastq
SRR1391072_1_pair_out.fastq SRR1391072_1_unpair_out.fastq
```

```
SRR1391072_2_pair_out.fastq SRR1391072_2_unpair_out.fastq
ILLUMINACLIP:adapters/TruSeq3-PE.fa:2:30:10 SLIDINGWINDOW:4:15
```

2D. Remove adapters, perform sliding window cutting, drop reads below 35

```
     java -jar trimmomatic-0.39.jar PE SRR1391072_1.fastq SRR1391072_2.fastq
SRR1391072_1_pair_out.fastq SRR1391072_1_unpair_out.fastq
SRR1391072_2_pair_out.fastq SRR1391072_2_unpair_out.fastq
ILLUMINACLIP:adapters/TruSeq3-PE.fa:2:30:10 SLIDINGWINDOW:4:15 MINLEN:35
```

2E. Remove adapters, only drop reads below 35

```
     java -jar trimmomatic-0.39.jar PE SRR1391072_1.fastq SRR1391072_2.fastq
SRR1391072_1_pair_out.fastq SRR1391072_1_unpair_out.fastq
SRR1391072_2_pair_out.fastq SRR1391072_2_unpair_out.fastq
ILLUMINACLIP:adapters/TruSeq3-PE.fa:2:30:10 MINLEN:35
```

SR109-3B2

Source: Download from web

3A. Before trimming

3B. Perform sliding window 4-base window, quality threshold 25

```
     java -jar trimmomatic-0.39.jar PE SS109-3B2_R1.fastq.gz SS109-
3B2_R2.fastq.gz SS109-3B2_R1_pair_out.fastq SS109-3B2_R1_unpair_out.fastq
SS109-3B2_R2_pair_out.fastq SS109-3B2_R2_unpair_out.fastq SLIDINGWINDOW:4:25
```

3C. Perform sliding window, remove first 5 bases from read

```
     java -jar trimmomatic-0.39.jar PE SS109-3B2_R1.fastq.gz SS109-
3B2_R2.fastq.gz SS109-3B2_R1_pair_out.fastq SS109-3B2_R1_unpair_out.fastq
SS109-3B2_R2_pair_out.fastq SS109-3B2_R2_unpair_out.fastq SLIDINGWINDOW:4:25
HEADCROP:5
```

3D. Perform sliding window, remove first 5 bases, drop reads below 30

```
      java -jar trimmomatic-0.39.jar PE SS109-3B2_R1.fastq.gz SS109-
3B2_R2.fastq.gz SS109-3B2_R1_pair_out.fastq SS109-3B2_R1_unpair_out.fastq
SS109-3B2_R2_pair_out.fastq SS109-3B2_R2_unpair_out.fastq SLIDINGWINDOW:4:25
HEADCROP:5 MINLEN:30
```

ERR3650066

Source:

```
   fastq-dump SRR1763780
```

*Create custom .fa file with this code for adapter sequence:

```
>RNA 3'Adapter (RA3)
TGGAATTCTCGGGTGCCAAGG
```

4A. Before trimming

4B. Trim RNA 3’ adapter

```
     java -jar trimmomatic-0.39.jar SE ERR3650066.fastq ERR3650066_out.fastq
ILLUMINACLIP:adapters/RA3.fa:2:30:10
```

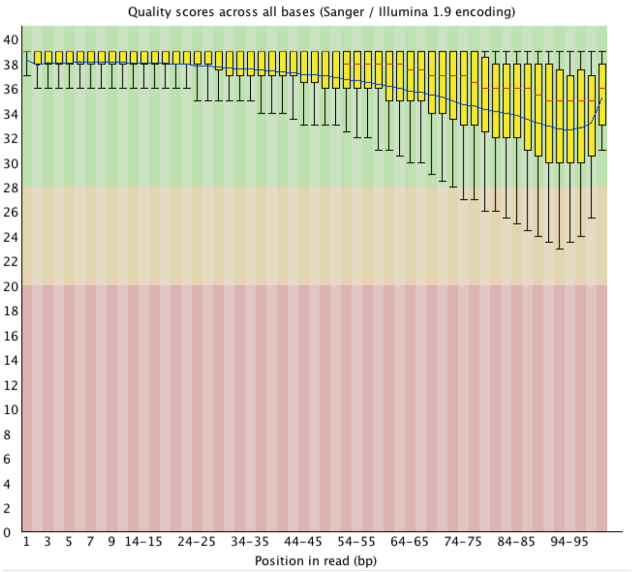
4C. Trim adapter, crop at 30 base pairs

```
     java -jar trimmomatic-0.39.jar SE ERR3650066.fastq ERR3650066_out.fastq
CROP:30
```

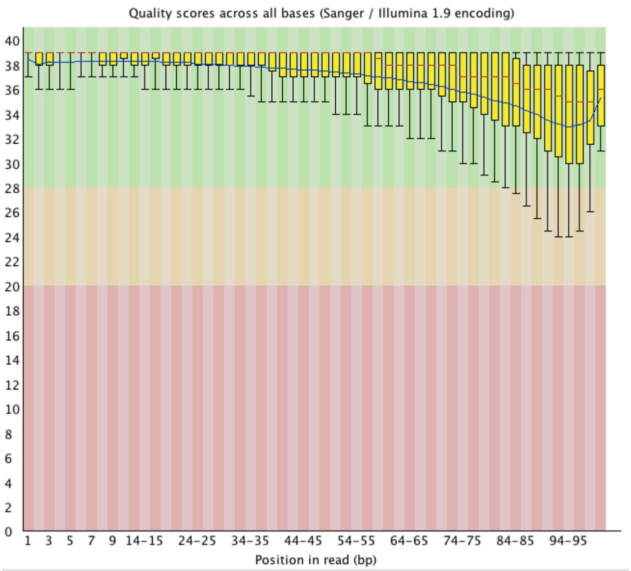
RESULTS

Dataset 1: *Ecoli200*

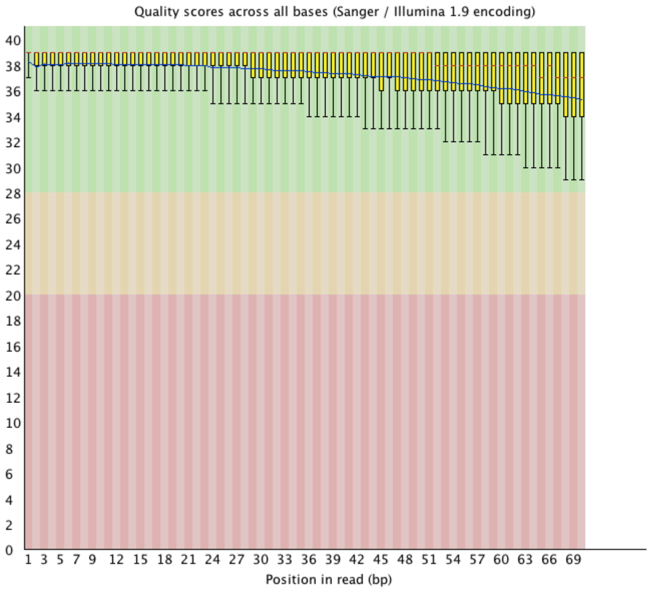
1A – Before trimming



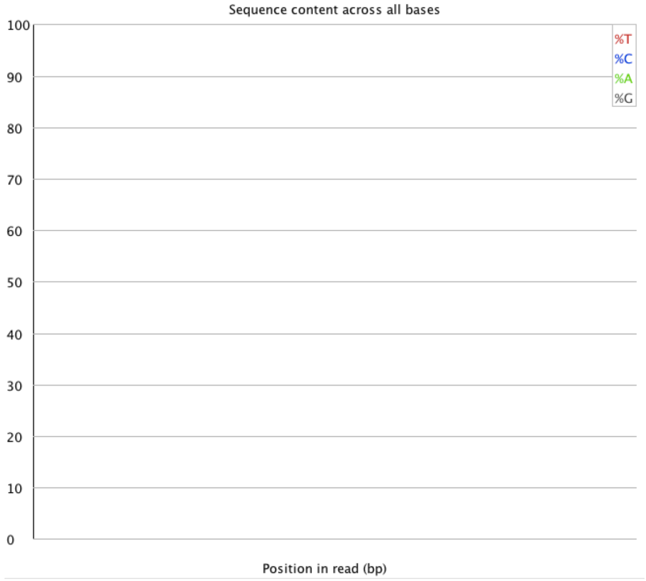
1B – MINLEN 95



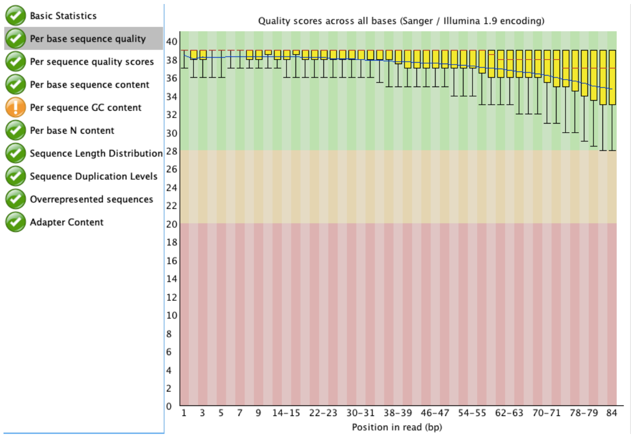
1C – Crop at 70bps



1D – Crop:70 and MINLEN:95 (no surviving pairs)



1E – MINLEN:95 and Crop:70

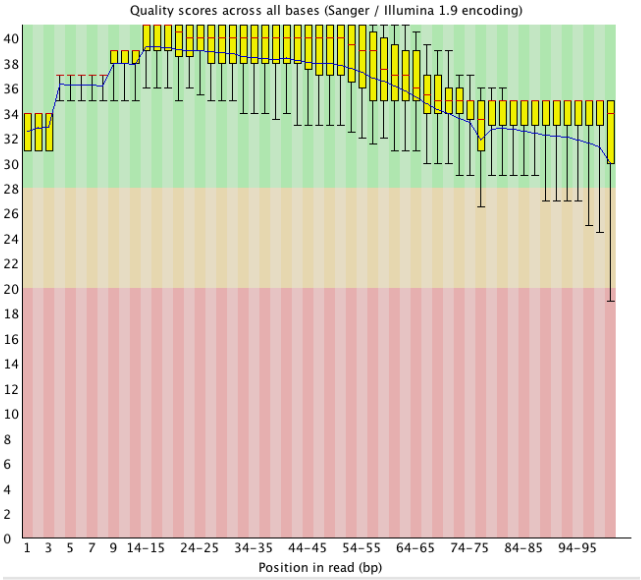

Basic Statistics
Per base sequence quality
Per sequence quality scores
Per base sequence content
Per sequence GC content
Per base N content
Sequence Length Distribution
Sequence Duplication Levels
Overrepresented sequences
Adapter Content
Quality scores across all bases (Sanger / Illumina 1.9 encoding)
40
38
36
34
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
0
1 3 5 7 9 14-15 22-23 30-31 38-39 46-47 54-55 62-63 70-71 78-79 84
Position in read (bp)

Input set	Percentage of reads retained
1A	100
1B	64.43
1C	100
1D	0
1E	64.43

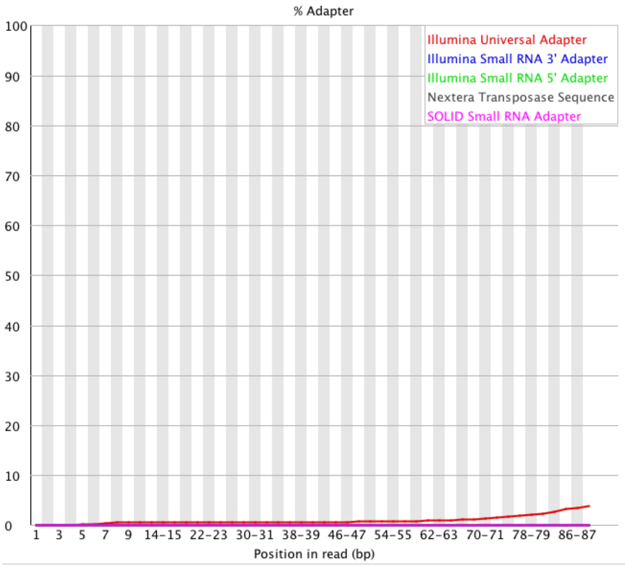
Table 1: Number of reads retained by condition, *E. coli*

Dataset 2: *SRR1391072*

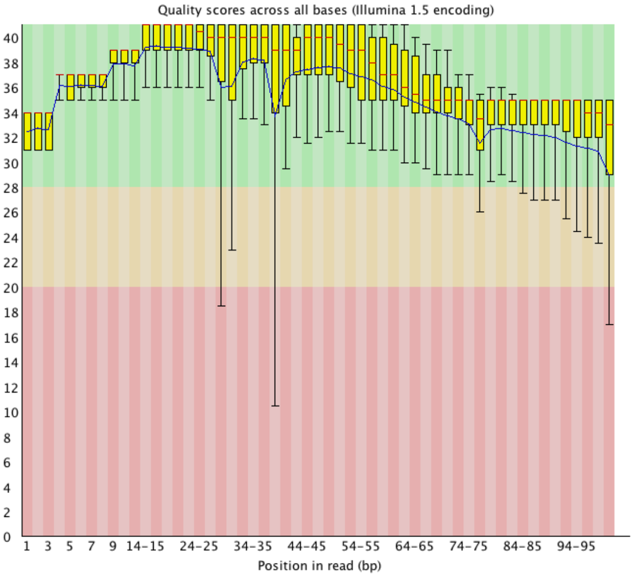
2A – Before trimming



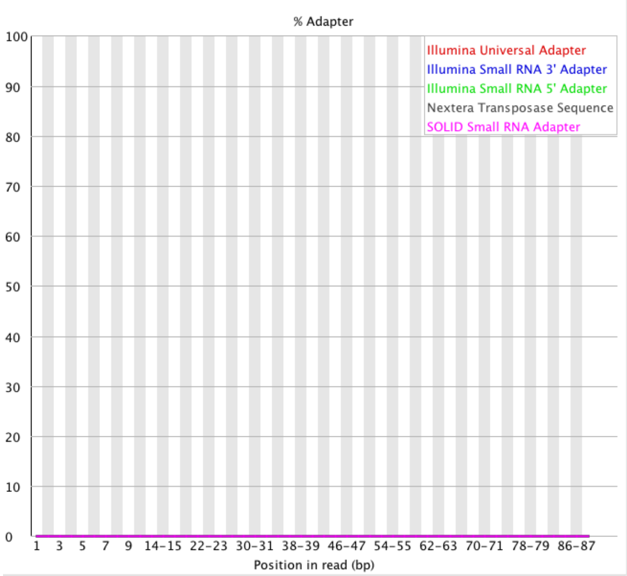
2A – Adapter content



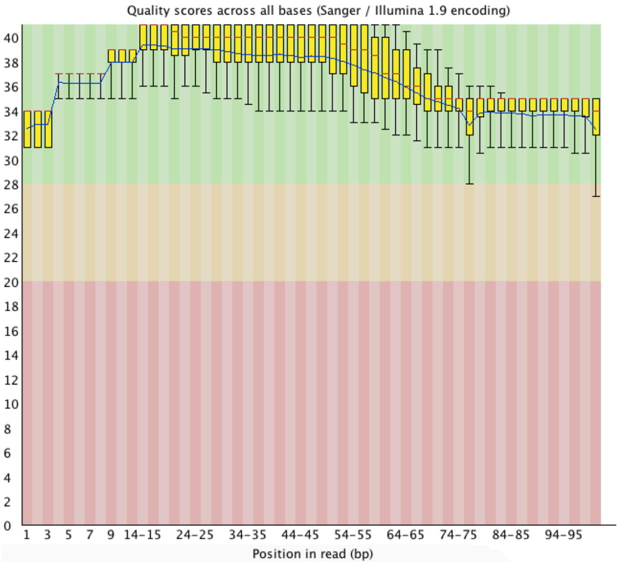
2B – After cutting adapter



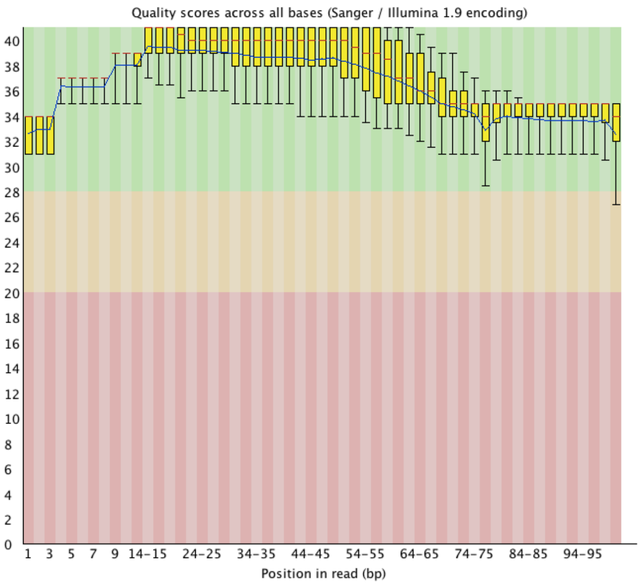
2B – Adapter content after adapter cutting



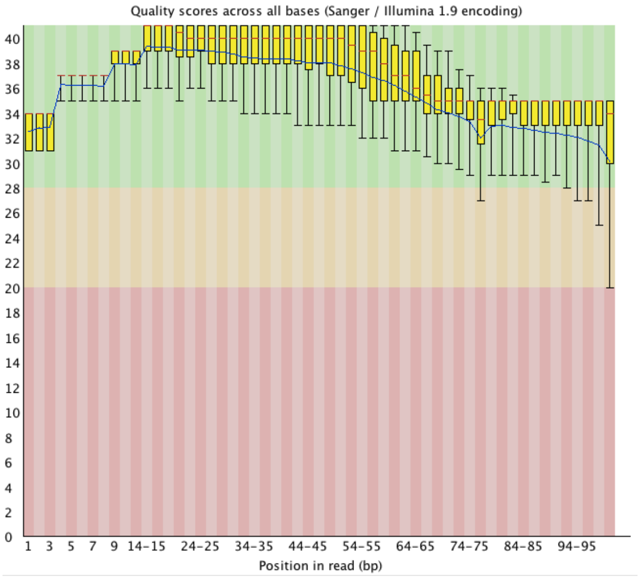
2C – Adapter trim, sliding window 4:15



2D – Adapter trim, sliding window 4:15 and reads below 35bp dropped



2E – Adapter trim, MINLEN35

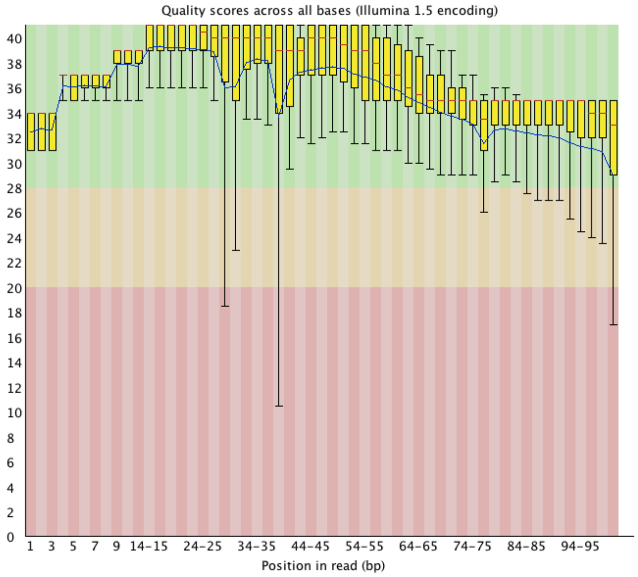


Input set	Percentage of reads retained
2A	100
2B	94.92
2C	93.88
2D	91.54
2E	94.92

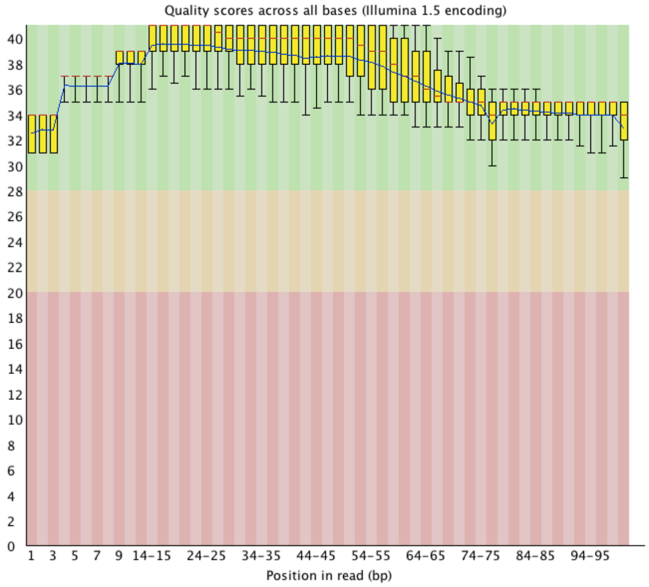
Table 2: Number of reads retained by condition, *V. vulnificus* transcriptome

Dataset 3: SR109-3B2

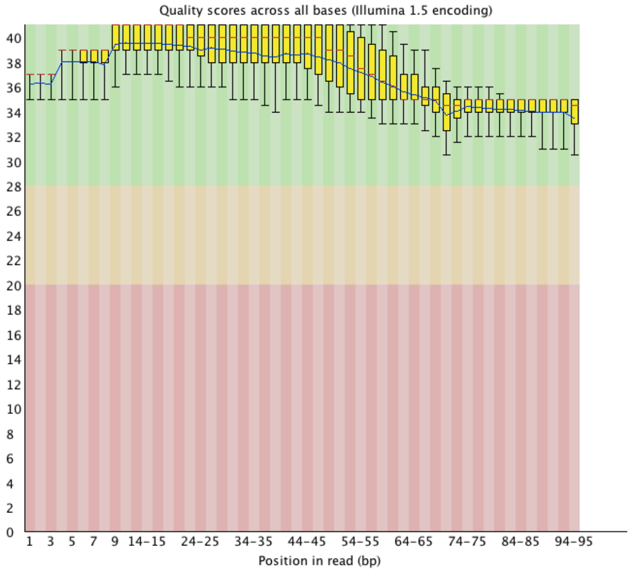
3A – Before trimming



3B – Sliding window 4:25

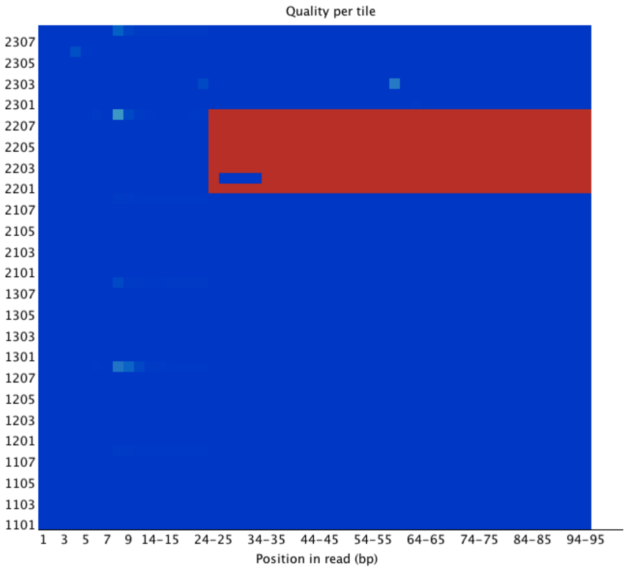


3C – Sliding window 4:25, headcrop: 5

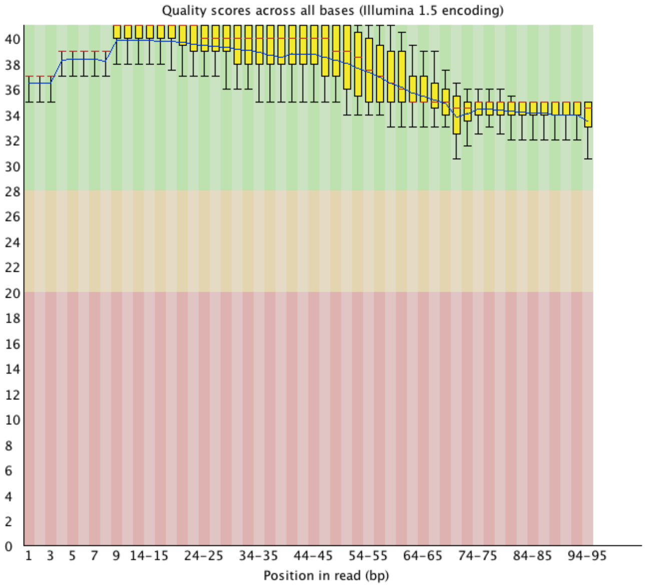


3C – Sliding window 4:25, headcrop 5

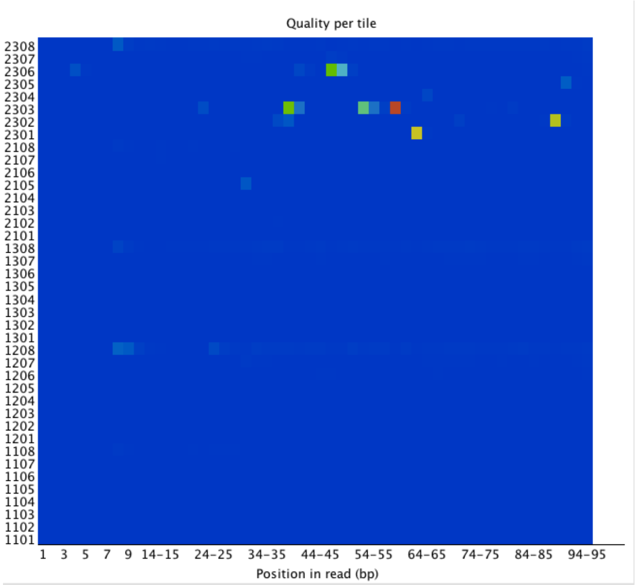
Per-tile sequence quality



3D – Sliding window 4:25, headcrop: 5, MINLEN 35



3D - Sliding window 4:25, headcrop: 5 MINLEN 35, Per-tile sequence quality

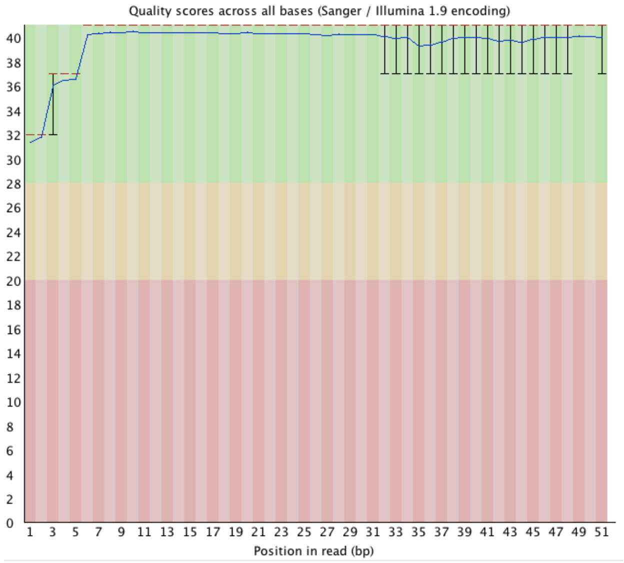


Input set	Percentage of reads retained
3A	100
3B	97.62
3C	97.00
3D	76.55

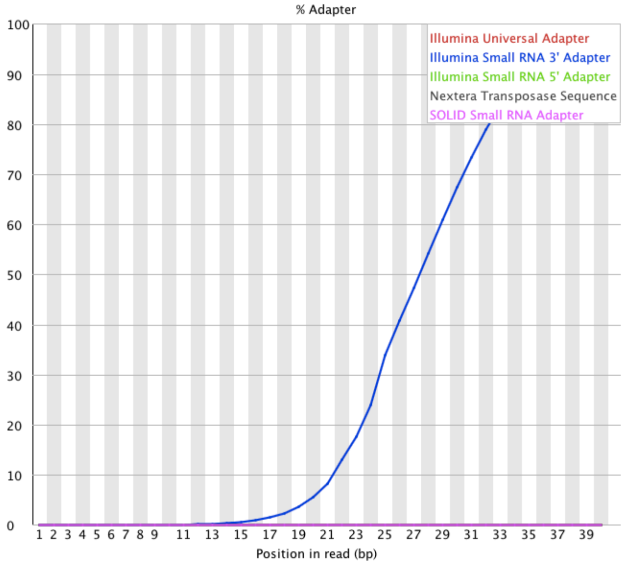
Table 3: Number of reads retained by condition, SR109-3B2

Dataset 4: *ERR3650066*

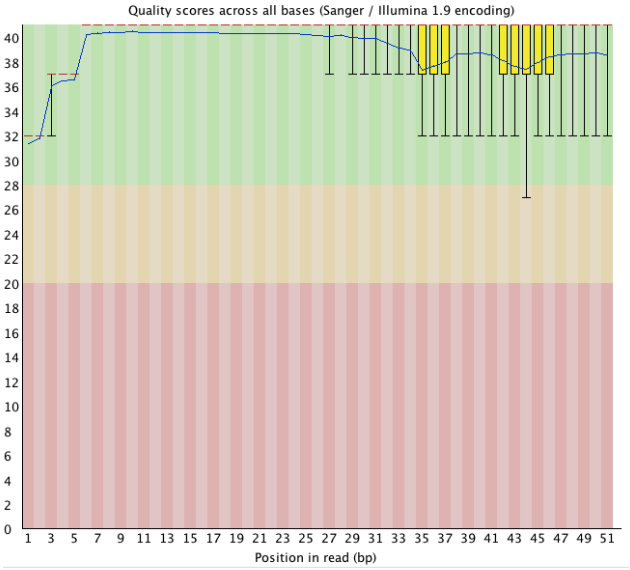
4A – Before trimming



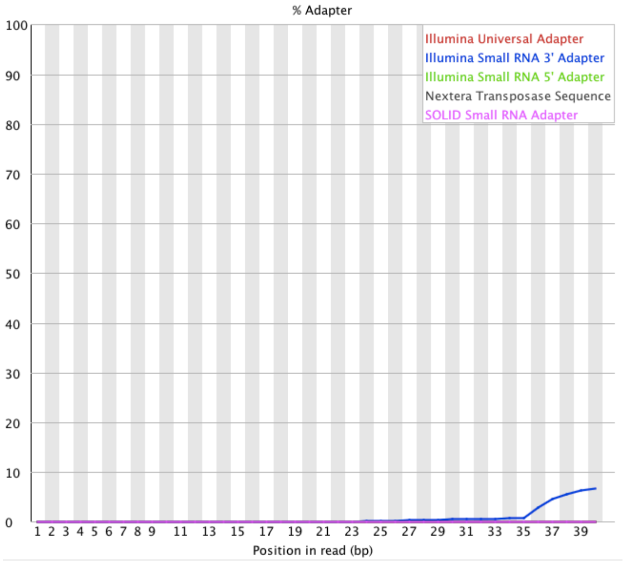
4A – Before trimming



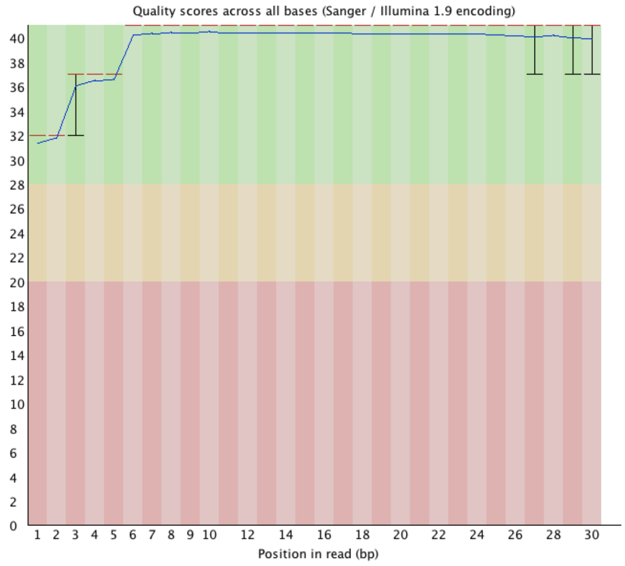
4B – Cut RNA 3' Adapter



4B - Cut RNA 3' Adapter



4C – Cut RNA 3’ Adapter, crop at 30bp



Input set	Percentage of reads retained
4A	100
4B	99.92
4C	100

Table 4: Number of reads retained by condition, ERR3650066

DISCUSSION

For the paired *E. coli* sequencing run, no adapters were detected (**1A**). The MINLEN command was used in an attempt to even the sequence length distribution from about 95-101 (**1B**). This drastically reduced the percentage of reads retained but did not have much impact on the overall quality of reads. A crop command was used to remove higher base pair-positioned reads (**1C**), which did not impact read retainment. It is important to note the order when combining the two commands, as cutting the reads before trimming based on quality leads to zero reads surviving (**1D**). The reverse order is, however, the most optimal trimmed dataset.

In the *V. vulnificus* paired-end transcriptome (**2A**), slight adapter presence was shown. After cutting with ILLUMINACLIP (**2B**), a sliding window command was used to improve read quality (**2C**), which mostly eliminated the extremely low-quality reads. MINLEN:35 combined with the sliding window produced the cleanest trimming (**2D**). For comparison, **2E** contains only the

MINLEN:35 command. In every trimmed dataset, the percentage of reads lost never went over 10%.

In the paired-end shotgun dataset, no adapter sequences were detected (**3A**). A sliding window (**3B**) quality cut was performed, removing most low-quality bases. The HEADCROP command was used to remove the first few low-quality bases (**3C**). Finally, due to extreme deviation per tile after only using a sliding window trim, MINLEN was also used last to remove low length reads (**3D**).

For the ncRNA, the biggest immediate issue was the small RNA 3' adapter (**4A**). With ILLUMINACLIP, the adapter presence was almost entirely removed (**4B**). Only a crop was needed to further reduce low-quality data.

This lab exercise showed the power of clipping software like Trimmomatic, which makes it possible to modify both single and paired-end data with millions of sequences. While it is impossible to trim datasets to perfect quality, this program allows for the removal of more obvious errors that will surely interfere with further analysis. Deciding which exact commands to use for each set is more than open to interpretation, and each trimming on this paper only represents one such option.