QAA_report

Christian La France

2022-09-05

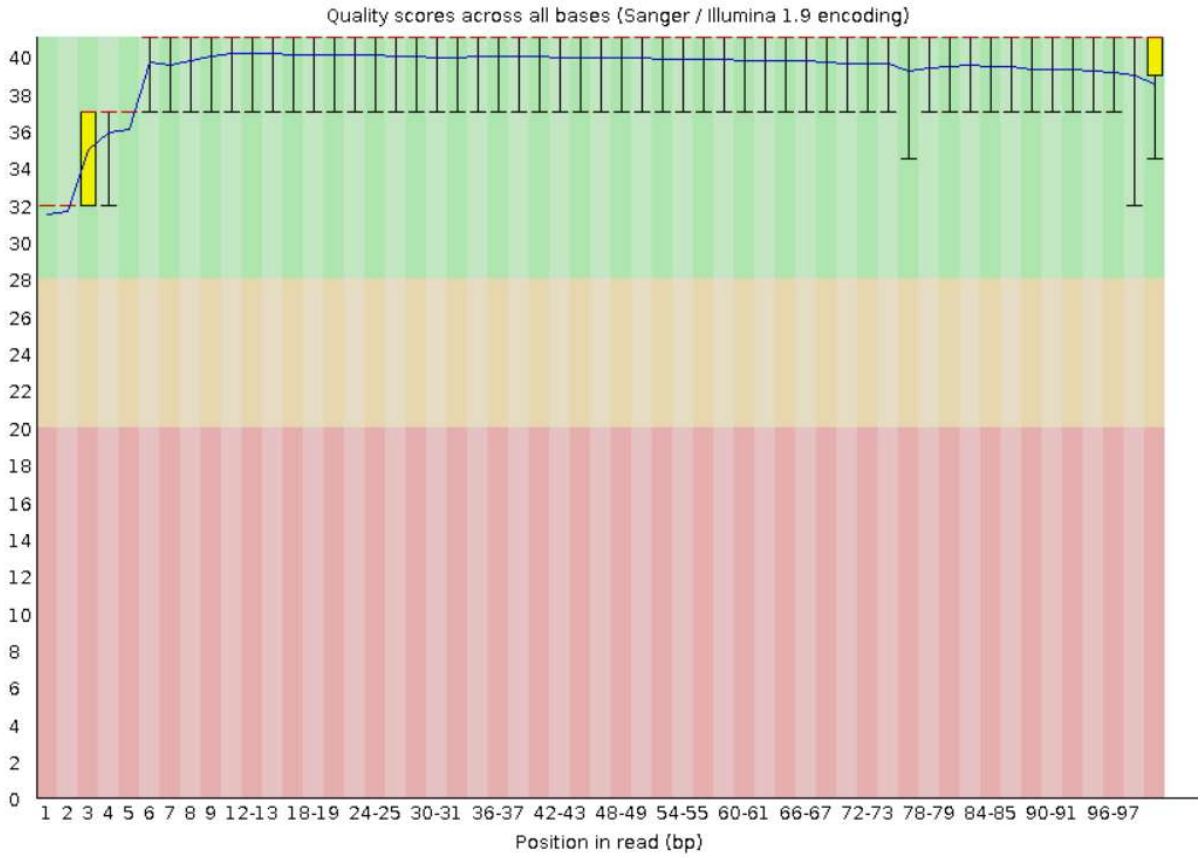
Software/package versions
R: 4.2.1
tidyverse: 1.3.2
ggplot2: 3.3.6
Python: 3.10.5
FastQC: 0.11.5
cutadapt: 4.1
trimmomatic: 0.39
STAR: 2.7.10a
htseq-count: 2.0.2
Ensambl: 107

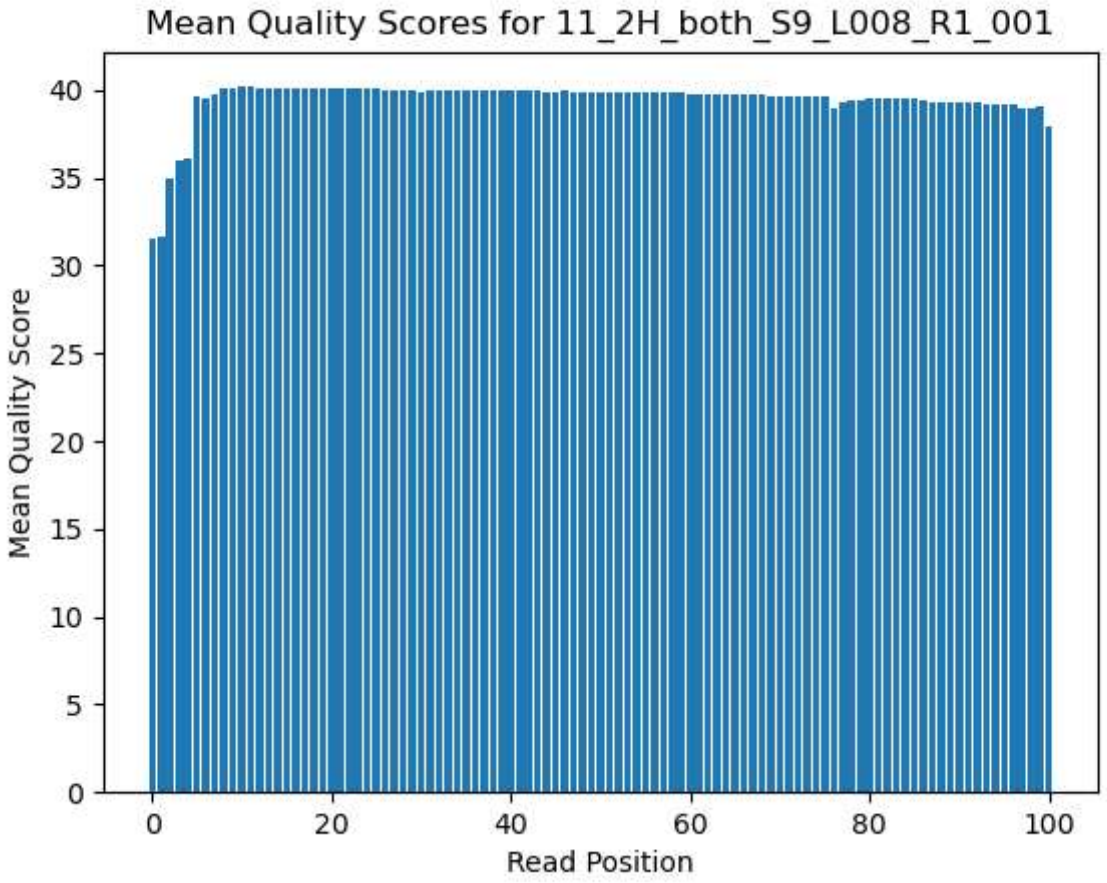
Environment
QAA

Part 1 – Read Quality Score Distributions

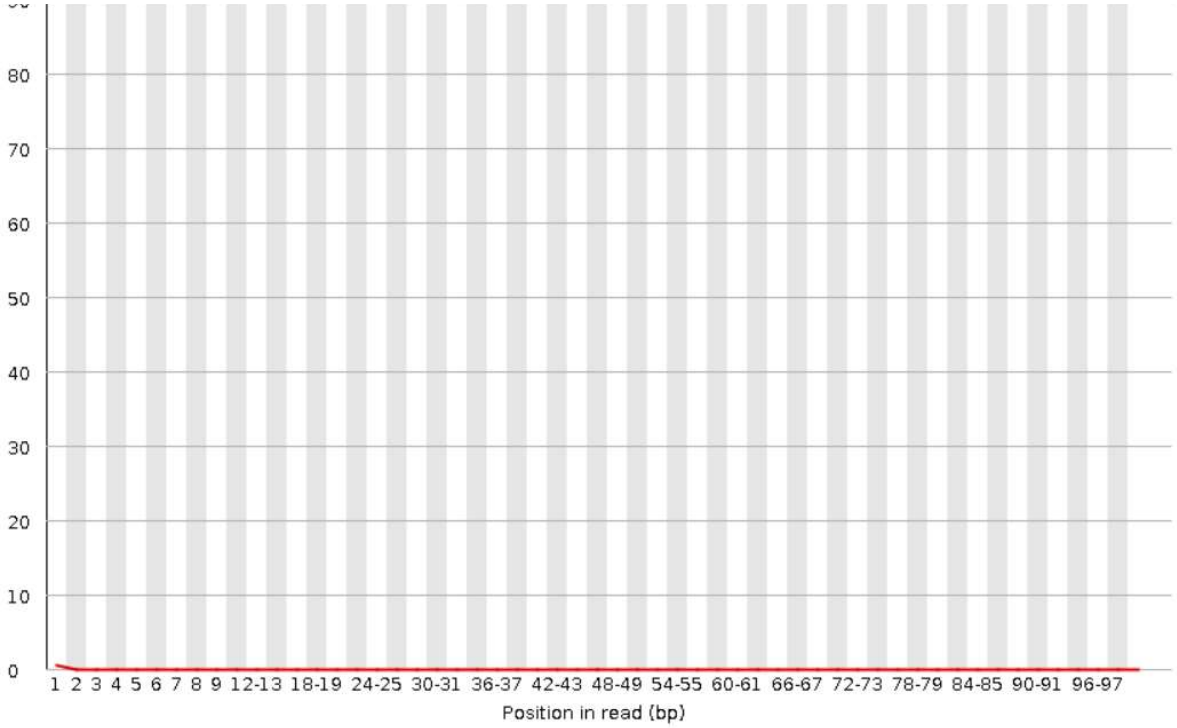
-Plots on the left were generated with `FastQC`. Plots on the right were generated with `mean_phred.py`. These scripts were run on Talapas using the `compare_qc.sh` script.

11_2H_both_S9_L008_R1

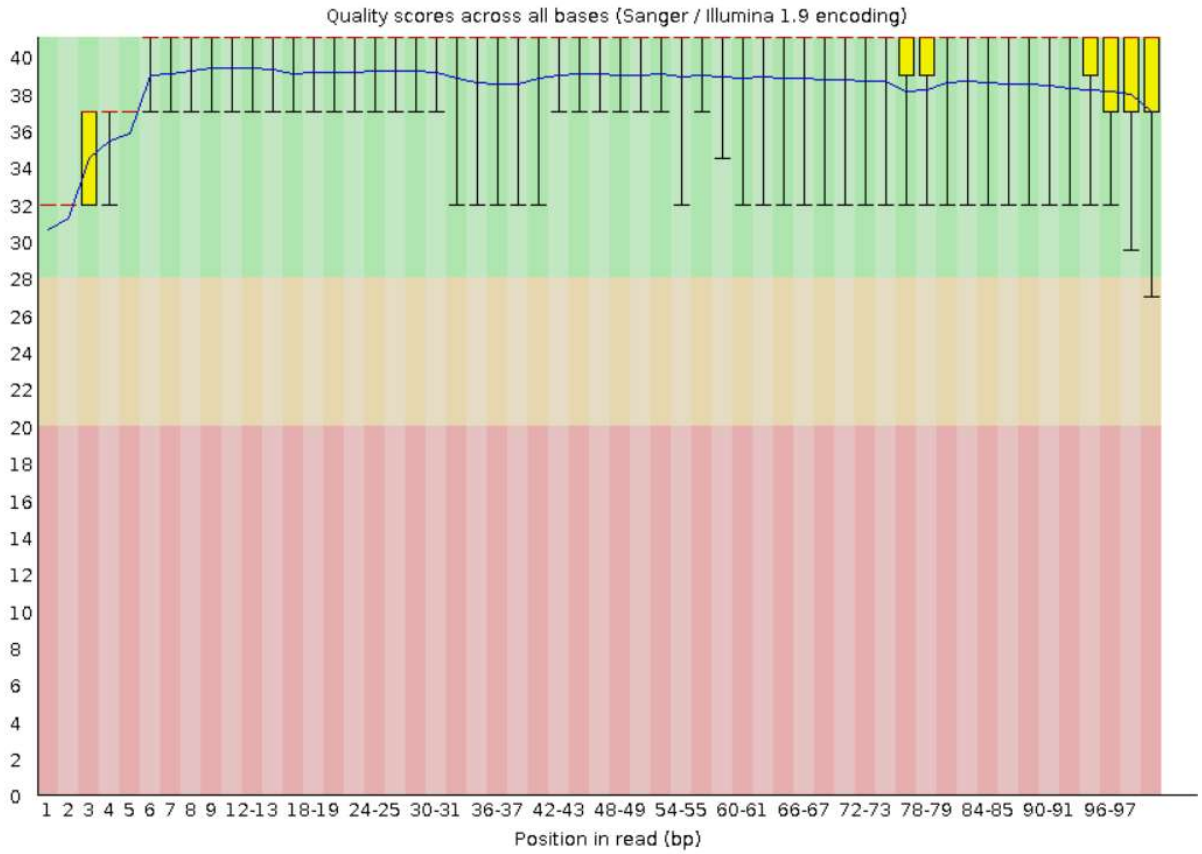

Quality scores across all bases (Sanger / Illumina 1.9 encoding)
40
38
36
34
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
0
1 2 3 4 5 6 7 8 9 12-13 18-19 24-25 30-31 36-37 42-43 48-49 54-55 60-61 66-67 72-73 78-79 84-85 90-91 96-97
Position in read (bp)

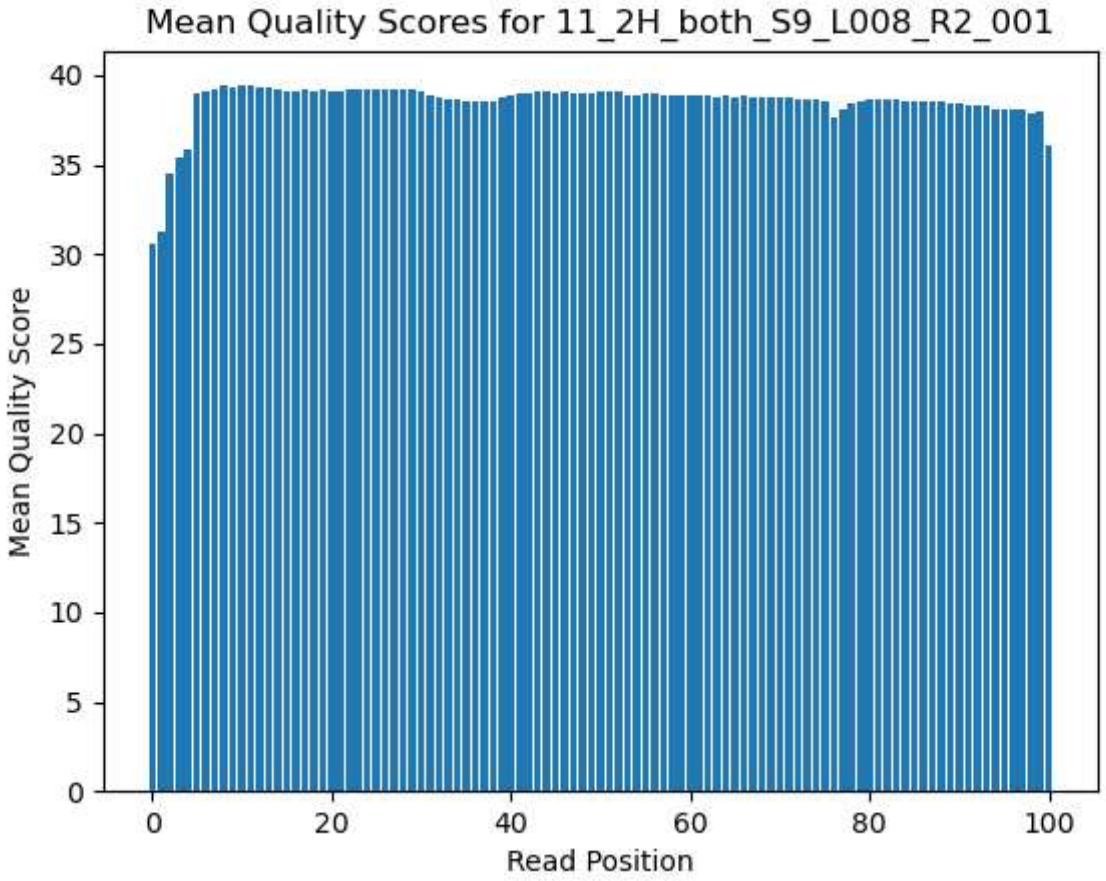

Mean Quality Scores for 11_2H_both_S9_L008_R1_001
Mean Quality Score
40
35
30
25
20
15
10
5
0
0
20
40
60
80
100
Read Position

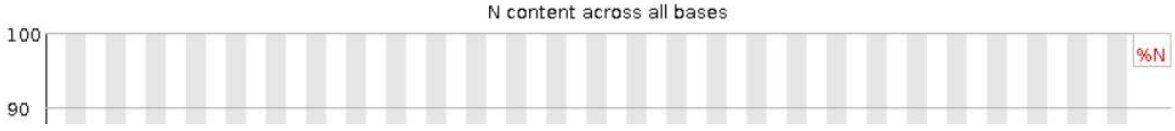

N content across all bases
100
90
%N

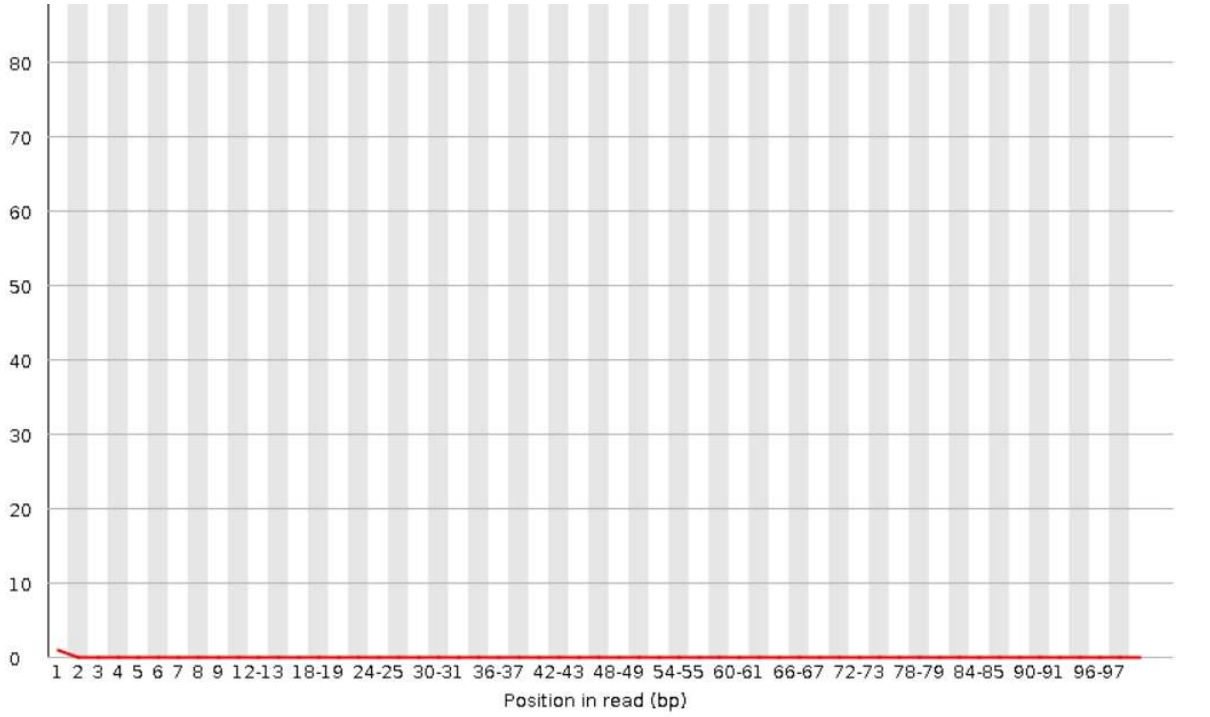


11_2H_both_S9_L008_R2

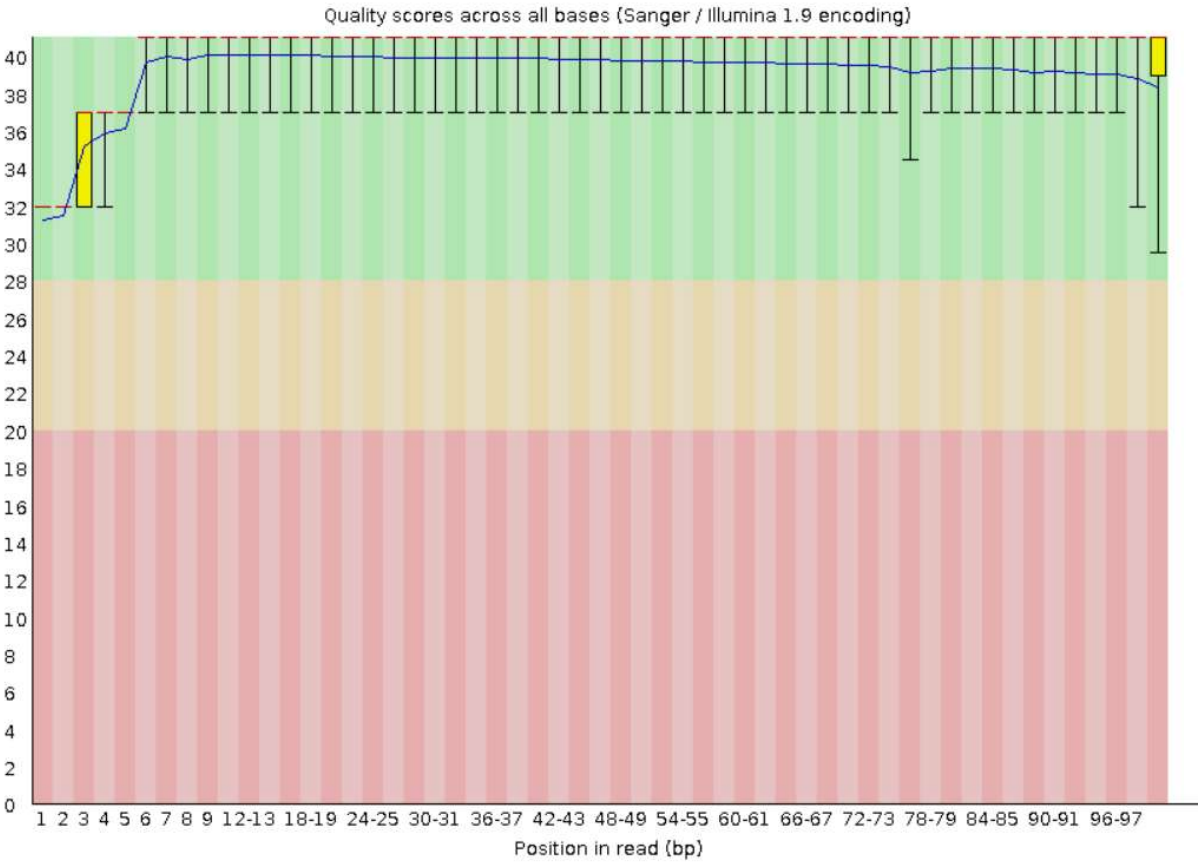

Quality scores across all bases (Sanger / Illumina 1.9 encoding)
40
38
36
34
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
0
1 2 3 4 5 6 7 8 9 12-13 18-19 24-25 30-31 36-37 42-43 48-49 54-55 60-61 66-67 72-73 78-79 84-85 90-91 96-97
Position in read (bp)

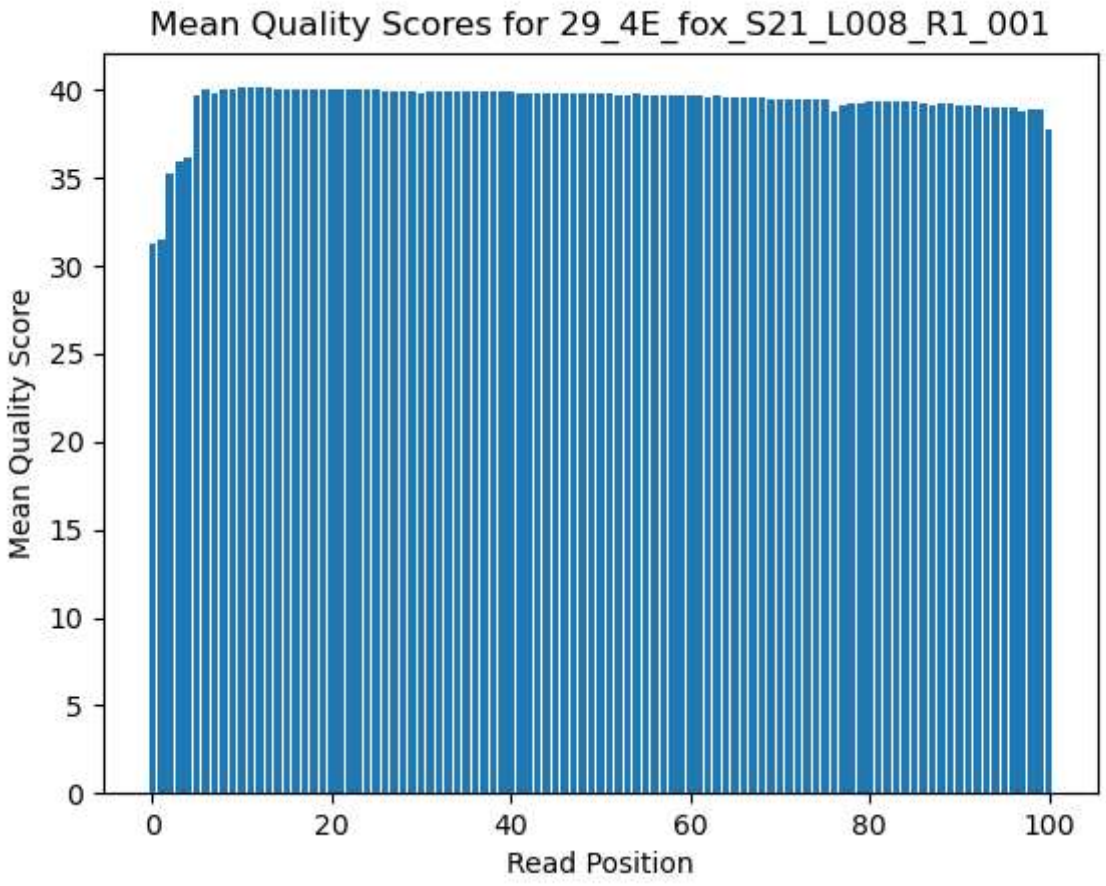

Mean Quality Scores for 11_2H_both_S9_L008_R2_001
Mean Quality Score
40
35
30
25
20
15
10
5
0
0
20
40
60
80
100
Read Position

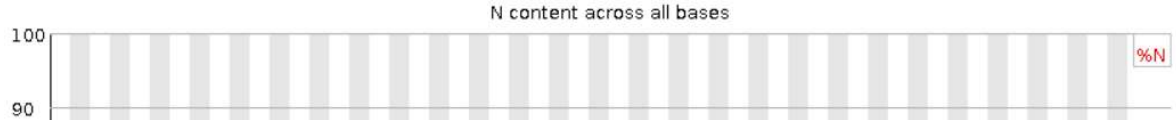

N content across all bases
100
90
%N

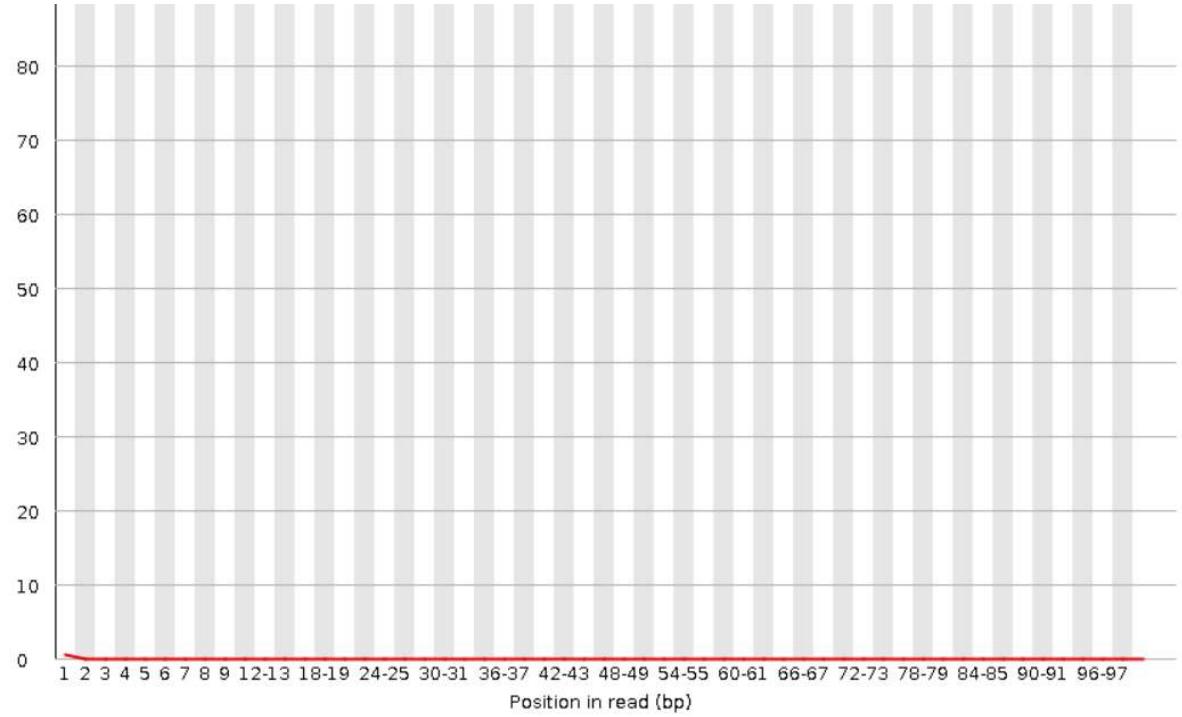


29_4E_fox_S21_L008_R1

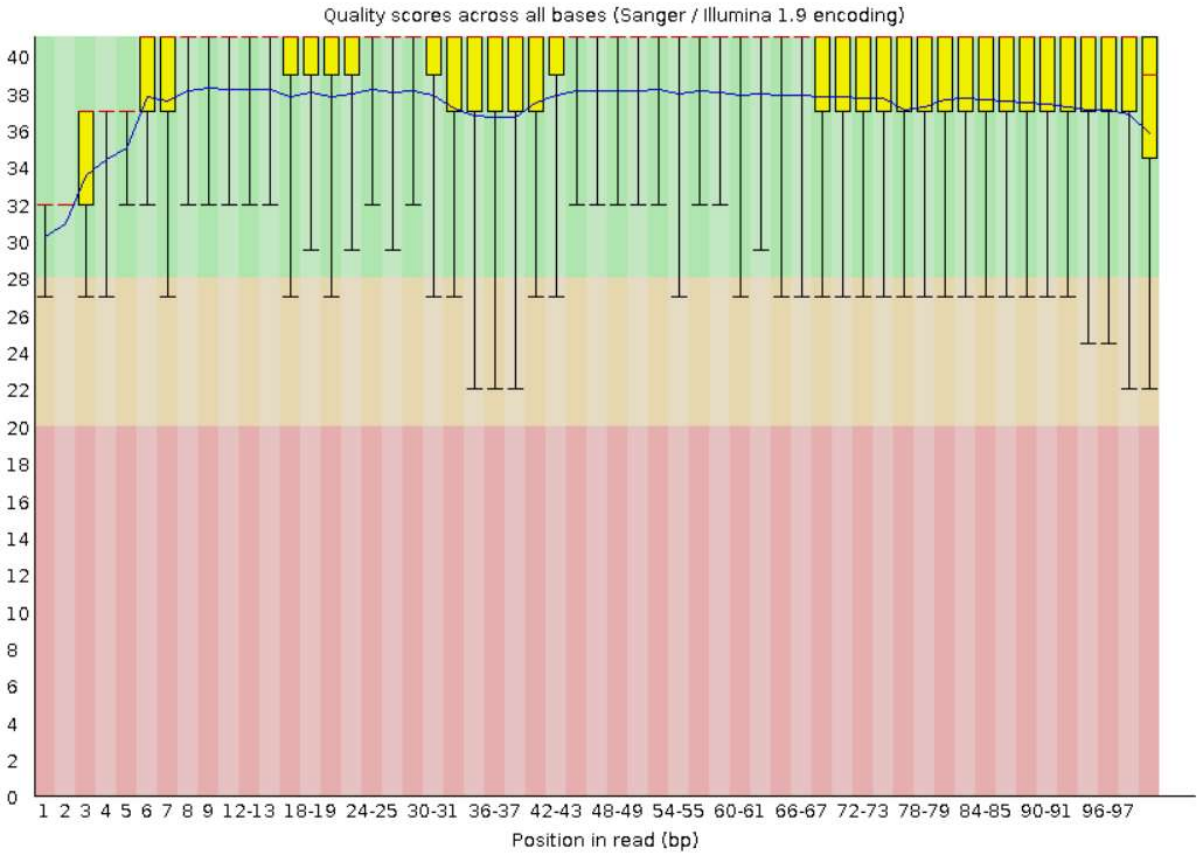

Quality scores across all bases (Sanger / Illumina 1.9 encoding)
40
38
36
34
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
0
1 2 3 4 5 6 7 8 9 12-13 18-19 24-25 30-31 36-37 42-43 48-49 54-55 60-61 66-67 72-73 78-79 84-85 90-91 96-97
Position in read (bp)

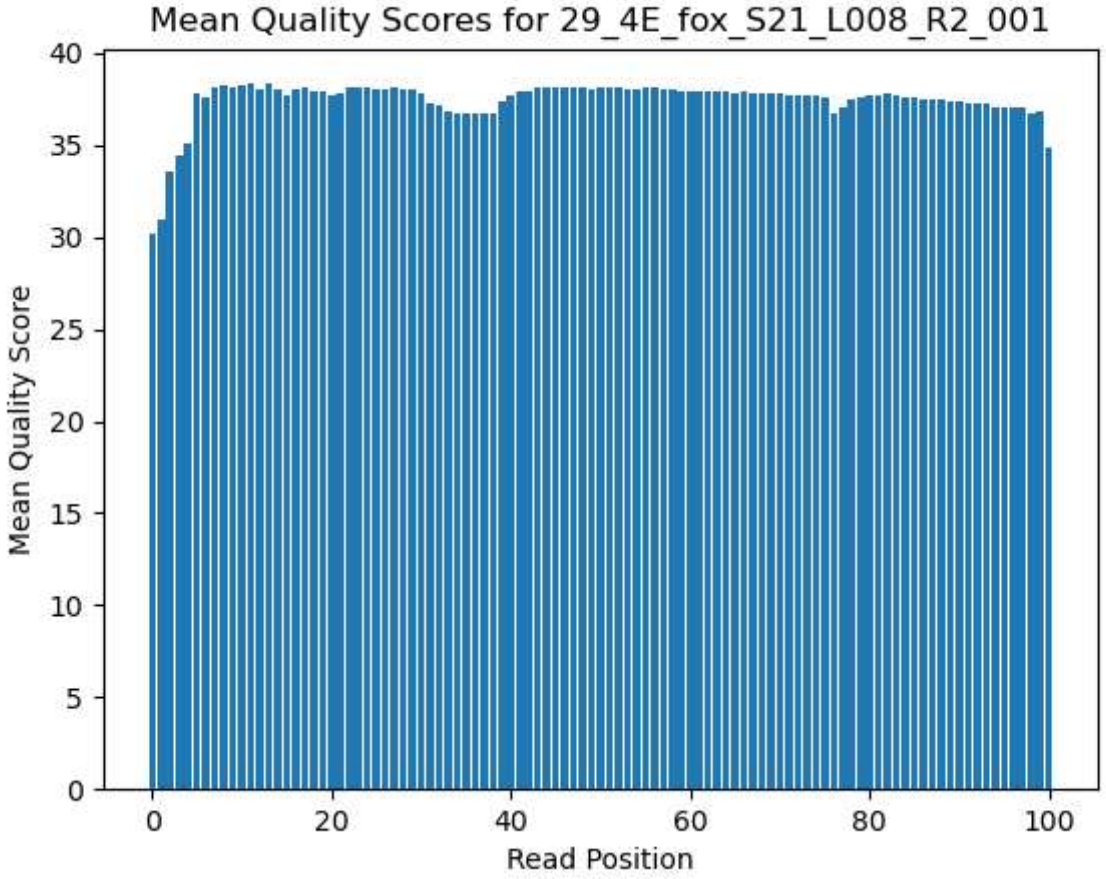

Mean Quality Scores for 29_4E_fox_S21_L008_R1_001
Mean Quality Score
40
35
30
25
20
15
10
5
0
0
20
40
60
80
100
Read Position

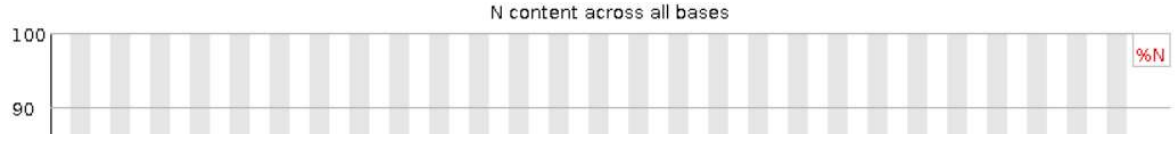

N content across all bases
100
90
%N

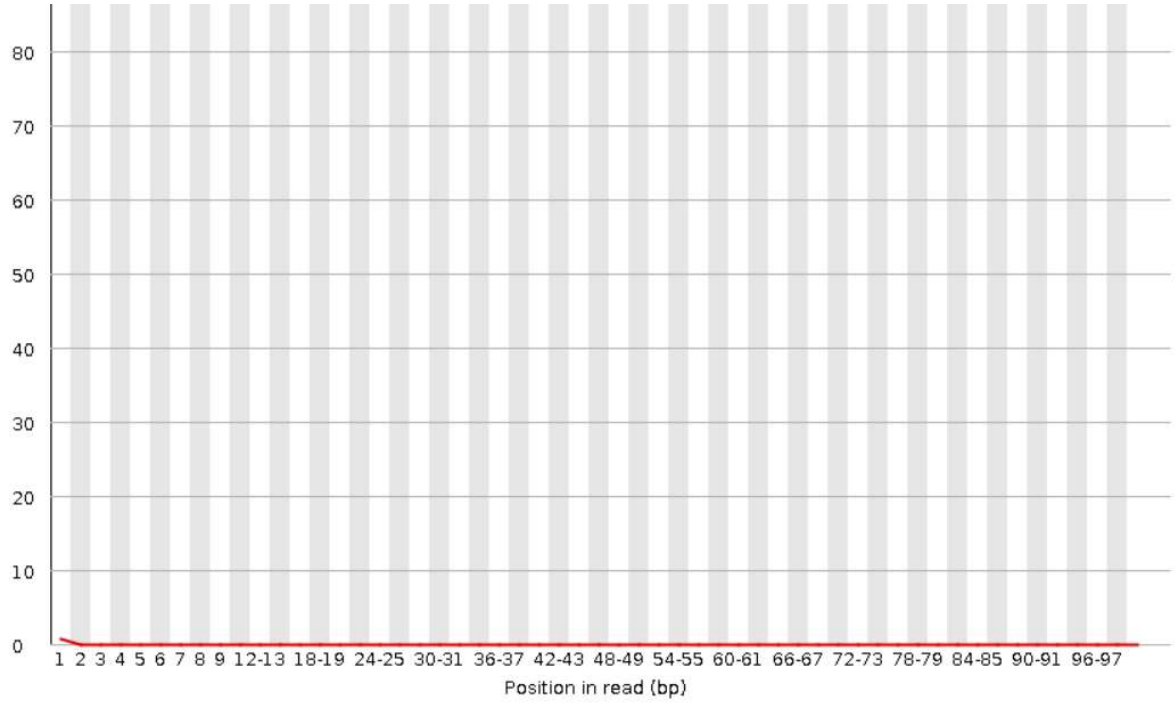


29_4E_fox_S21_L008_R2


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 2 3 4 5 6 7 8 9 12-13 18-19 24-25 30-31 36-37 42-43 48-49 54-55 60-61 66-67 72-73 78-79 84-85 90-91 96-97
Position in read (bp)


Mean Quality Scores for 29_4E_fox_S21_L008_R2_001
Mean Quality Score
40
35
30
25
20
15
10
5
0
0
20
40
60
80
100
Read Position


N content across all bases
100
90
%N



Answers to questions:

1. The per-base N content is consistent with the per base quality scores. There are more N's at the beginning of the sequence, and the beginning of the sequence also has lower quality scores on average.
2. The plots generated from mean_phred.py and FastQC are very similar with no obvious differences. FastQC was much faster, probably due to significant optimization and being written in a faster programming language.
3. The two libraries follow a similar trend, low in the beginning, highest in the middle, and slowly drops off at the end. R2 is lower in general than R1 for both libraries. The 29_4E_fox_S21_L008 library has slighly higher per-base quality scores on average.

Part 2 – Adaptor Trimming Comparison

5. check for adapters:

```
zcat 11_2H_both_S9_L008_R1_001.fastq.gz | grep AGATCGGAAGAGCACACGTCTGAACTCCAGTCA | wc -l
```

Output: 23629

```
zcat 11_2H_both_S9_L008_R2_001.fastq.gz | grep AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT | wc -l
```

Output: 24496

```
zcat 29_4E_fox_S21_L008_R1_001.fastq.gz | grep AGATCGGAAGAGCACACGTCTGAACTCCAGTCA | wc -l
```

Output: 39701

```
zcat 29_4E_fox_S21_L008_R2_001.fastq.gz | grep AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT | wc -l
```

Output: 39929

Searching for the R2 adapters in the R1 reads finds no matches, example:

```
zcat 11_2H_both_S9_L008_R1_001.fastq.gz | grep AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT | wc -l
```

Output: 0

cutadapt commands:

```
cutadapt -a AGATCGGAAGAGCACACGTCTGAACTCCAGTCA \
-A AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT \
-o TRIMMED_11_2H_both_S9_L008_R1_001.fastq \
-p TRIMMED_11_2H_both_S9_L008_R2_001.fastq \
/projects/bgmp/shared/2017_sequencing/demultiplexed/29_4E_fox_S21_L008_R1_001.fastq.gz \
/projects/bgmp/shared/2017_sequencing/demultiplexed/29_4E_fox_S21_L008_R2_001.fastq.gz \
-j 10
```

```
cutadapt -a AGATCGGAAGAGCACACGTCTGAACTCCAGTCA \
-A AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT \
-o TRIMMED_29_4E_fox_S21_L008_R1_001.fastq \
-p TRIMMED_29_4E_fox_S21_L008_R2_001.fastq \
/projects/bgmp/shared/2017_sequencing/demultiplexed/29_4E_fox_S21_L008_R1_001.fastq.gz \
/projects/bgmp/shared/2017_sequencing/demultiplexed/29_4E_fox_S21_L008_R2_001.fastq.gz \
-j 10
```

Output:

File name	Total reads	Reads with adapter	Percent of reads with adapter
TRIMMED_11_2H_both_S9_L008_R1_001.fastq	17,919,193	874,706	4.9%
TRIMMED_11_2H_both_S9_L008_R2_001.fastq	17,919,193	1,016,991	5.7%
TRIMMED_29_4E_fox_S21_L008_R1_001.fastq	4,827,433	361,886	7.5%
TRIMMED_29_4E_fox_S21_L008_R2_001.fastq	4,827,433	400,819	8.3%

6. Trommomatic commands and outputs:

```
trimmomatic PE ../trimmed_fq/TRIMMED_11_2H_both_S9_L008_R1_001.fastq \
../trimmed_fq/TRIMMED_11_2H_both_S9_L008_R2_001.fastq \
QT_11_2H_both_S9_L008_1.fq \
QT_11_2H_both_S9_L008_2.fq \
QT_11_2H_both_S9_L008_3.fq \
QT_11_2H_both_S9_L008_4.fq \
LEADING:3 TRAILING:3 SLIDINGWINDOW:5:15 MINLEN:35
```

Output:
Quality encoding detected as phred33 Input Read Pairs: 17919193 Both Surviving: 17465123 (97.47%) Forward Only Surviving: 431903 (2.41%) Reverse Only Surviving: 13431 (0.07%) Dropped: 8736 (0.05%) TrimmomaticPE:

Completed successfully

```
trimmomatic PE ../trimmed_fq/TRIMMED_29_4E_fox_S21_L008_R1_001.fastq \
../trimmed_fq/TRIMMED_29_4E_fox_S21_L008_R2_001.fastq \
QT_29_4E_fox_S21_L008_1.fq \
QT_29_4E_fox_S21_L008_2.fq \
QT_29_4E_fox_S21_L008_3.fq \
QT_29_4E_fox_S21_L008_4.fq \
LEADING:3 TRAILING:3 SLIDINGWINDOW:5:15 MINLEN:35
```

Output:
Quality encoding detected as phred33 Input Read Pairs: 4827433 Both Surviving: 4571696 (94.70%) Forward Only Surviving: 248101 (5.14%) Reverse Only Surviving: 3371 (0.07%) Dropped: 4265 (0.09%) TrimmomaticPE: Completed successfully

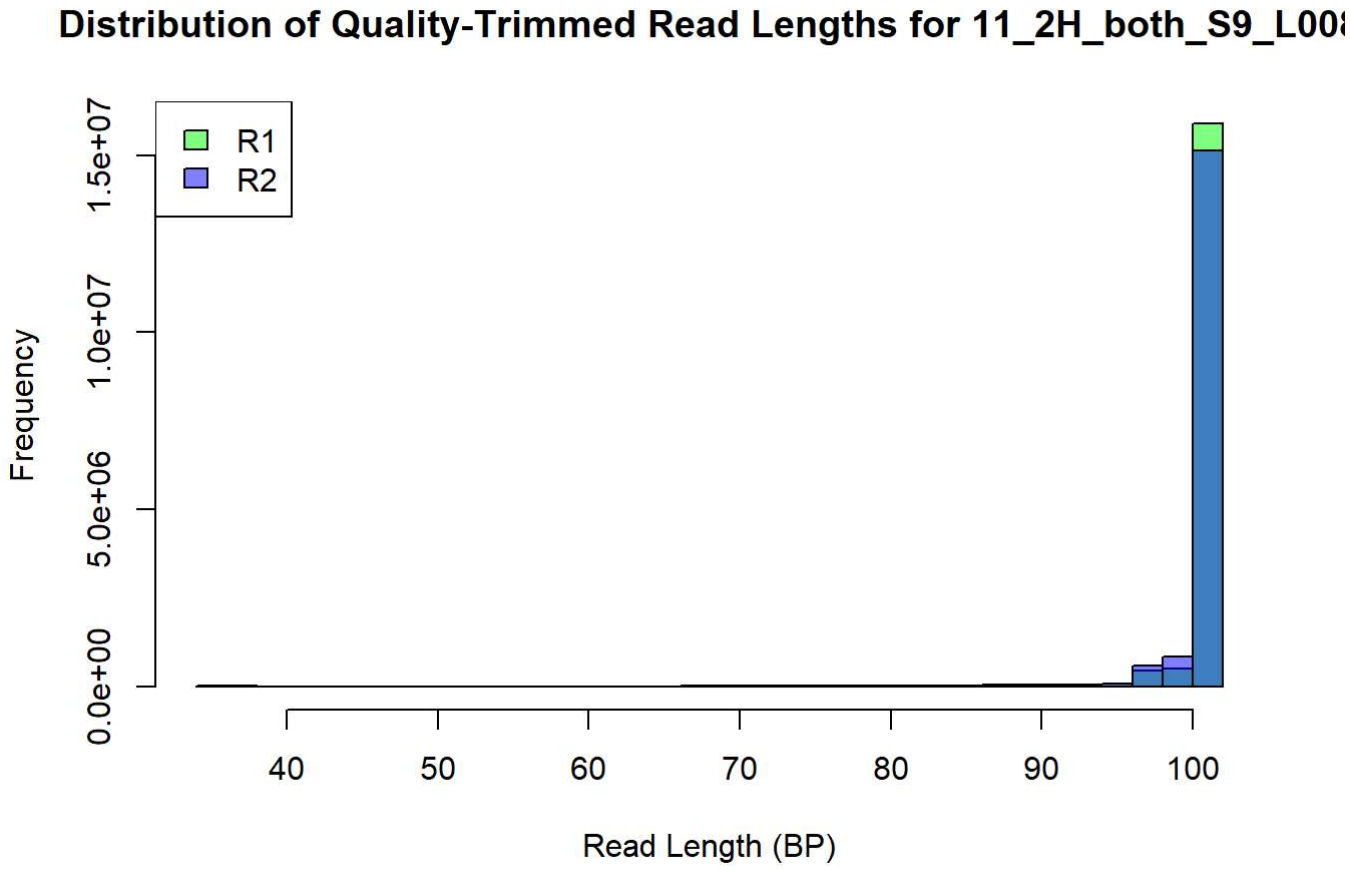
7. Plot:
-find the lengths of each read to plot:

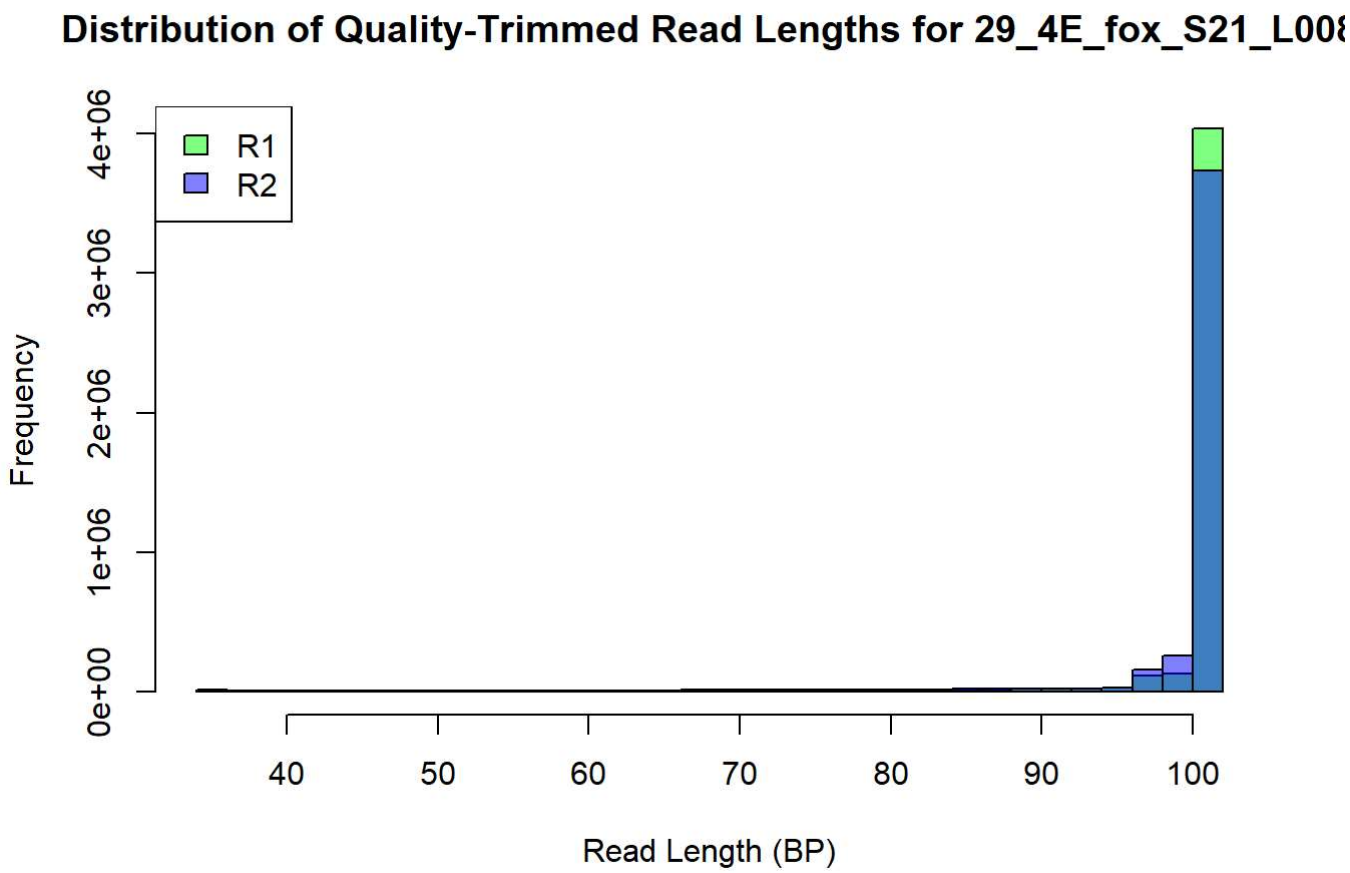
```
cat QT_11_2H_both_S9_L008_R1_FP.fq | sed -n "2~4p" | awk '{print length($0)}' >> QT_11_2H_both_S
9_L008_R1_001_LENS.txt
```

```
cat QT_11_2H_both_S9_L008_R2_RP.fq | sed -n "2~4p" | awk '{print length($0)}' >> QT_11_2H_both_S
9_L008_R2_001_LENS.txt
```

```
cat QT_29_4E_fox_S21_L008_R1_FP.fq | sed -n "2~4p" | awk '{print length($0)}' >> QT_29_4E_fox_S2
1_L008_R1_001_LENS.txt
```

```
cat QT_29_4E_fox_S21_L008_R2_RP.fq | sed -n "2~4p" | awk '{print length($0)}' >> QT_29_4E_fox_S2
1_L008_R2_001_LENS.txt
```


Distribution of Quality-Trimmed Read Lengths for 11_2H_both_S9_L00
R1
R2
Frequency
0.0e+00
5.0e+06
1.0e+07
1.5e+07
40
50
60
70
80
90
100
Read Length (BP)


Distribution of Quality-Trimmed Read Lengths for 29_4E_fox_S21_L00
R1
R2
Frequency
0e+00
1e+06
2e+06
3e+06
4e+06
40
50
60
70
80
90
100
Read Length (BP)

Answers to questions:

7. I expect R1 and R2 to have similar rates of adapter trimming since whether adapters are present is determine by the insert length and if that is lower than the sequencing runs specified. Though since the DNA sits on the flow cell for a long time before R2 is sequenced, it is possible that it has undergone some degredation which might cause some issues. However, I do expect R2 to have more quality trimming than R1, for the same reason. Sitting on the flow cell for a long time will probably cause some degredation.

Part 3 – Alignment and Strand-Specificity

8. Commands for files:

```
wget http://ftp.ensembl.org/pub/release-107/fasta/mus_musculus/dna/Mus_musculus.GRCm39.dna.primary_assembly.fa.gz
```

```
wget http://ftp.ensembl.org/pub/release-107/gtf/mus_musculus/Mus_musculus.GRCm39.107.gtf.gz
```

-Used `align.sh` to generate alignment database and perform alignment.

-Mapped counts generated with `parse_sam.py` :

File name	Mapped count	Percent mapped	Unmapped count	Percent unmapped
mouse_ens107_GRCm39_11_2H_both_Aligned.out.sam	33,855,756	94.468%	1,982,630	5.532%
mouse_ens107_GRCm39_29_4E_fox_Aligned.out.sam	8,995,444	93.17%	659,422	6.83%

10. htseq-count commands:

```
htseq-count --stranded=yes -c 11_2H_both_stranded_counts.tsv -n 10 \
--with-header mouse_ens107_GRCm39_11_2H_both_Aligned.out.sam Mus_musculus.GRCm39.107.gtf
```

```
htseq-count --stranded=yes -c 29_4E_fox_stranded_counts.tsv -n 10 \
--with-header mouse_ens107_GRCm39_29_4E_fox_Aligned.out.sam Mus_musculus.GRCm39.107.gtf
```

```
htseq-count --stranded=reverse -c 11_2H_both_reverse_counts.tsv -n 10 \
--with-header mouse_ens107_GRCm39_11_2H_both_Aligned.out.sam Mus_musculus.GRCm39.107.gtf
```

```
htseq-count --stranded=reverse -c 29_4E_fox_reverse_counts.tsv -n 10 \
--with-header mouse_ens107_GRCm39_29_4E_fox_Aligned.out.sam Mus_musculus.GRCm39.107.gtf
```

11. For both of the files, there are more counts when –stranded=reverse is specified. Based on this and htseq-count –help, the reads are from strand-specific libraries. If it was unstranded, the number of mapped reads would be the same for both parameters.

-Commands used:

```
cat 11_2H_both_stranded_counts.tsv | grep -v '^_' | awk '{sum+=$2} END {print sum}'
```

```
cat 11_2H_both_reverse_counts.tsv | grep -v '^_' | awk '{sum+=$2} END {print sum}'
```

```
cat 29_4E_fox_stranded_counts.tsv | grep -v '^_' | awk '{sum+=$2} END {print sum}'
```

```
cat 29_4E_fox_reverse_counts.tsv | grep -v '^_' | awk '{sum+=$2} END {print sum}'
```

htseq-count results:

File name	Stranded option	Reads mapped to genes
11_2H_both_stranded_counts.tsv	stranded=yes	600,435
11_2H_both_reverse_counts.tsv	stranded=reverse	13,934,885
29_4E_fox_stranded_counts.tsv	stranded=yes	187,153
29_4E_fox_reverse_counts.tsv	stranded=reverse	3,923,907