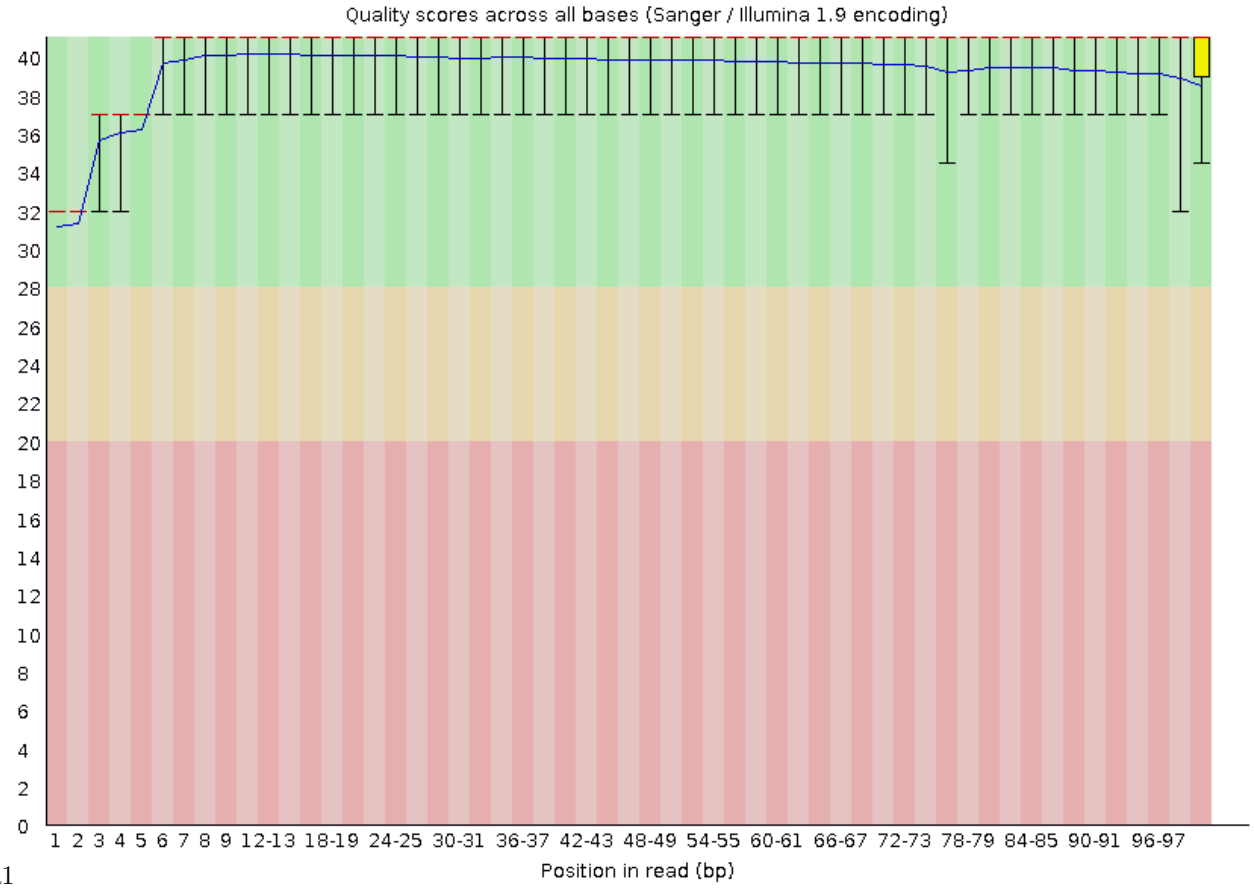
# QAA_report.pdf

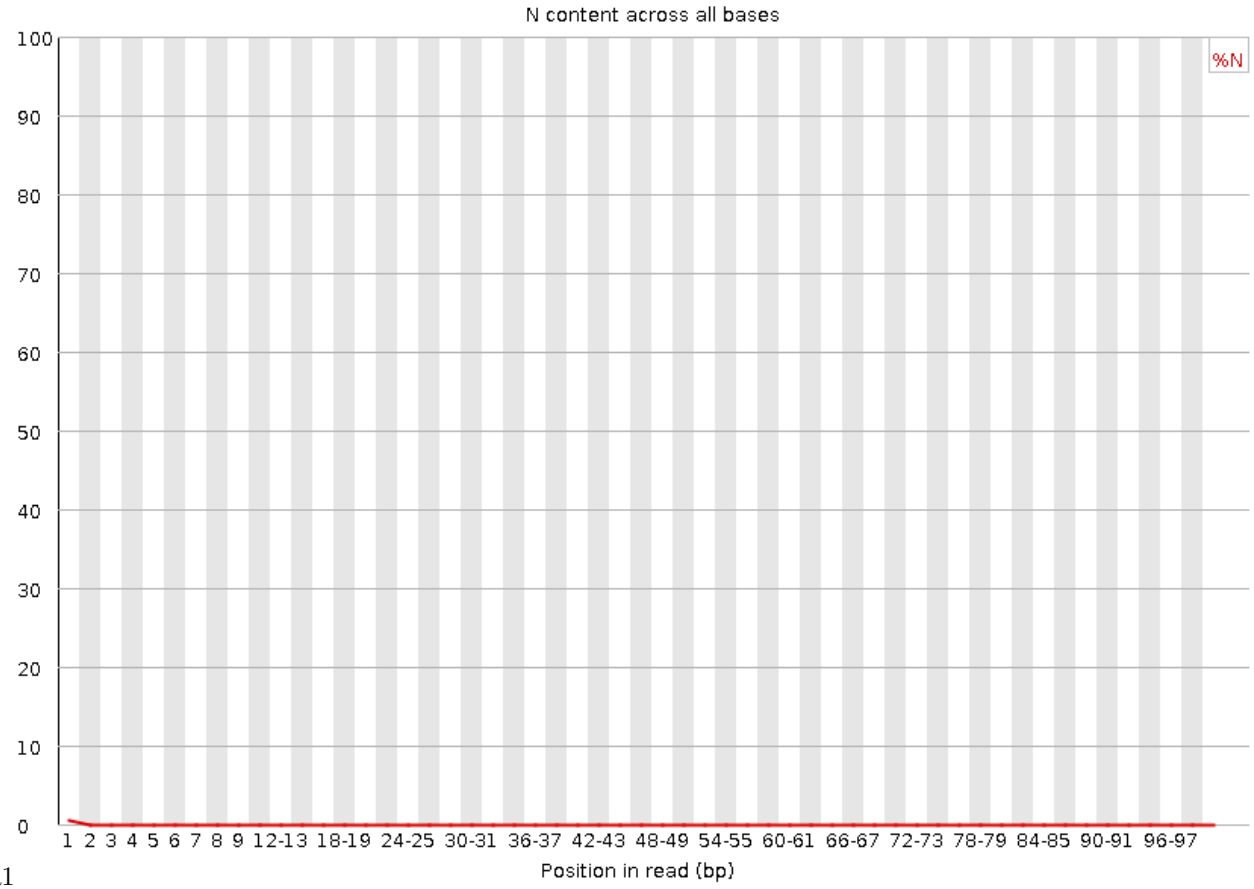
2022-09-07

QAA REPORT

GRAPHS FOR FILE 31_4F_fox_S22_L008: Forward

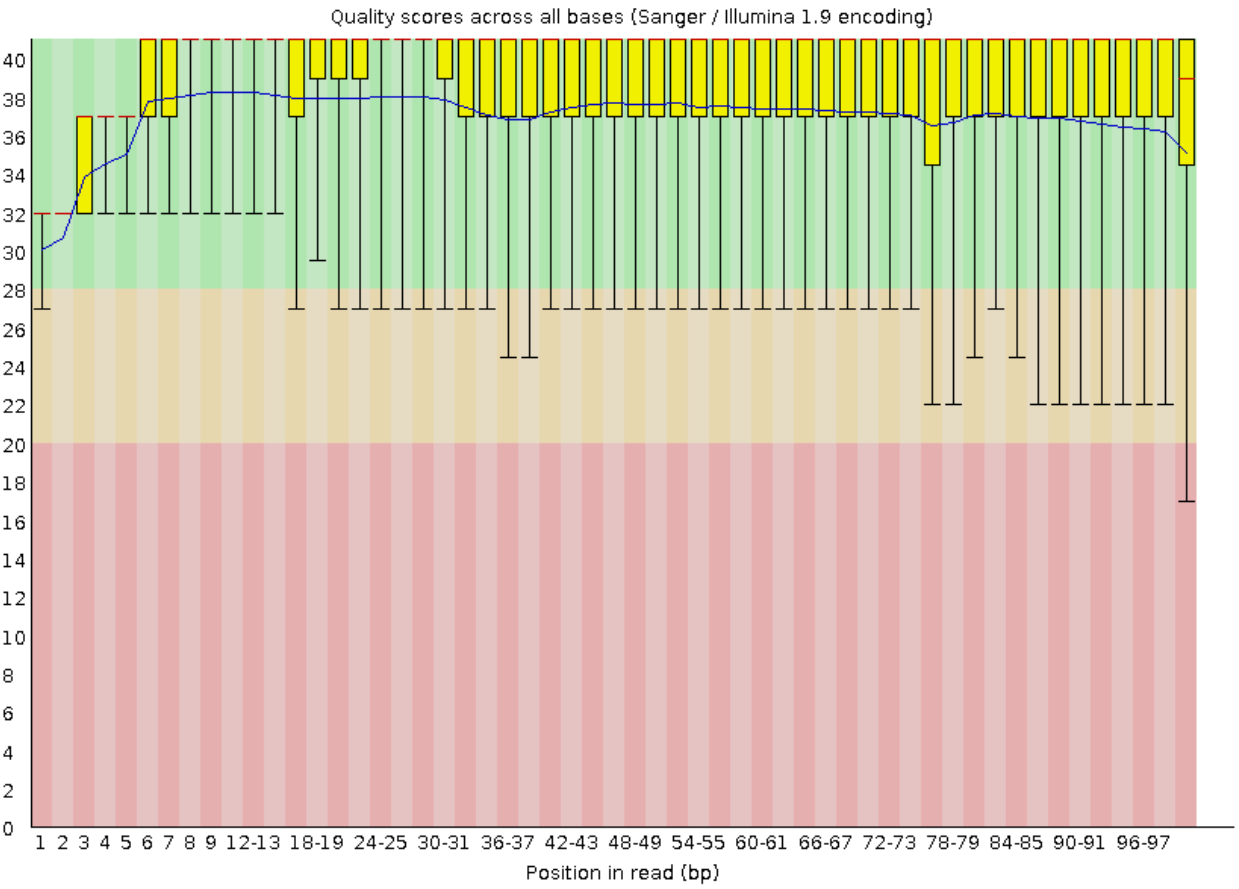



R1

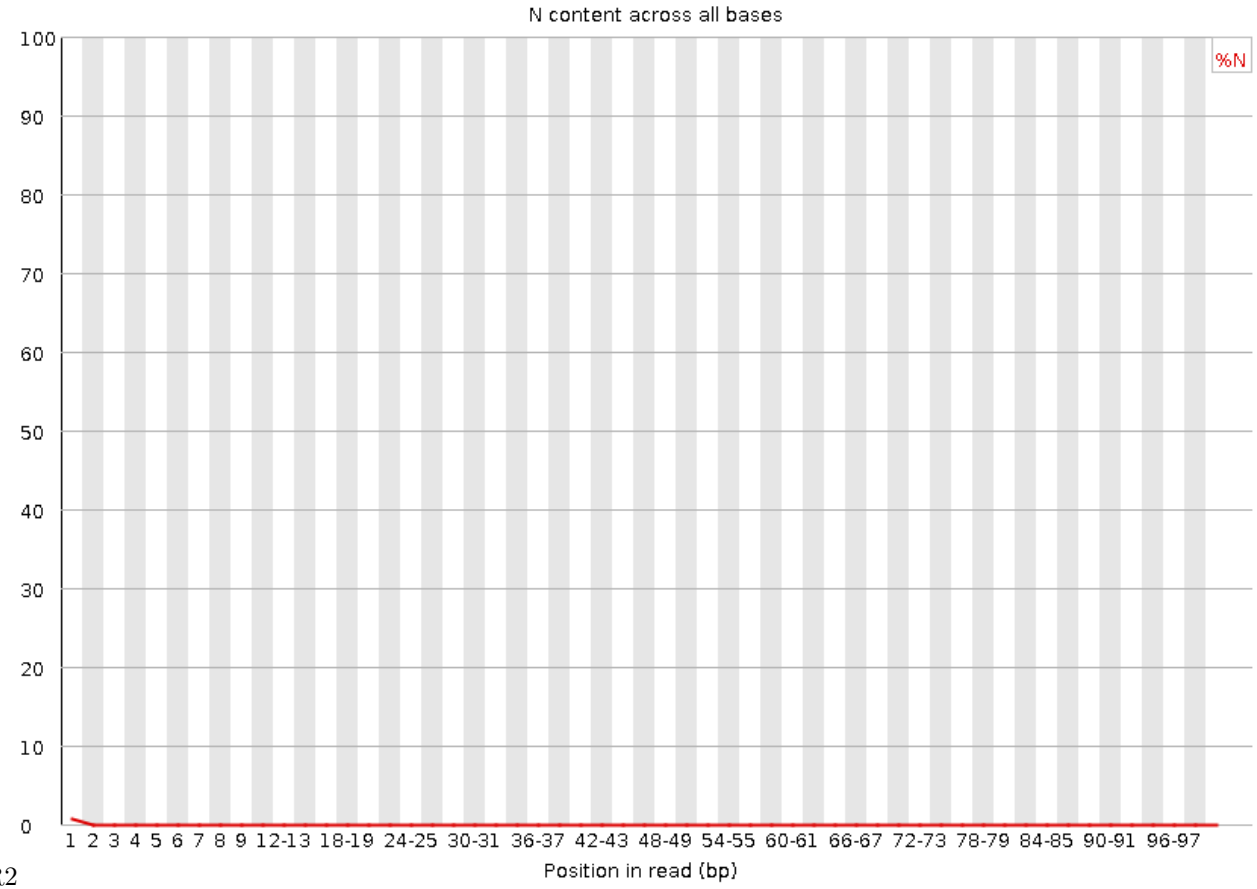



R1

FILE 31_4F_fox_S22_L008: Reverse

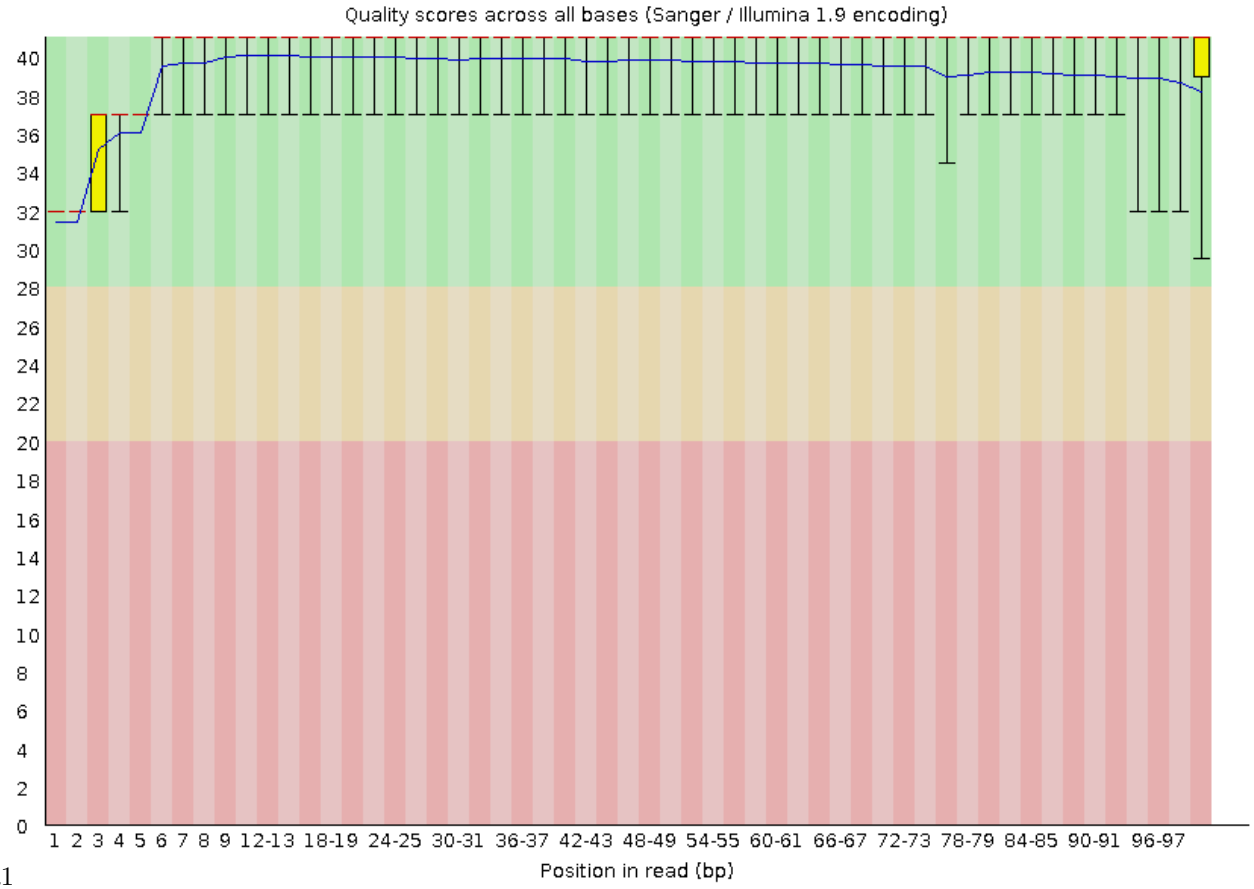

Quality scores across all bases (Sanger / Illumina 1.9 encoding)
40
38
36
34
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
0
1 2 3 4 5 6 7 8 9 12-13 18-19 24-25 30-31 36-37 42-43 48-49 54-55 60-61 66-67 72-73 78-79 84-85 90-91 96-97
Position in read (bp)

R2

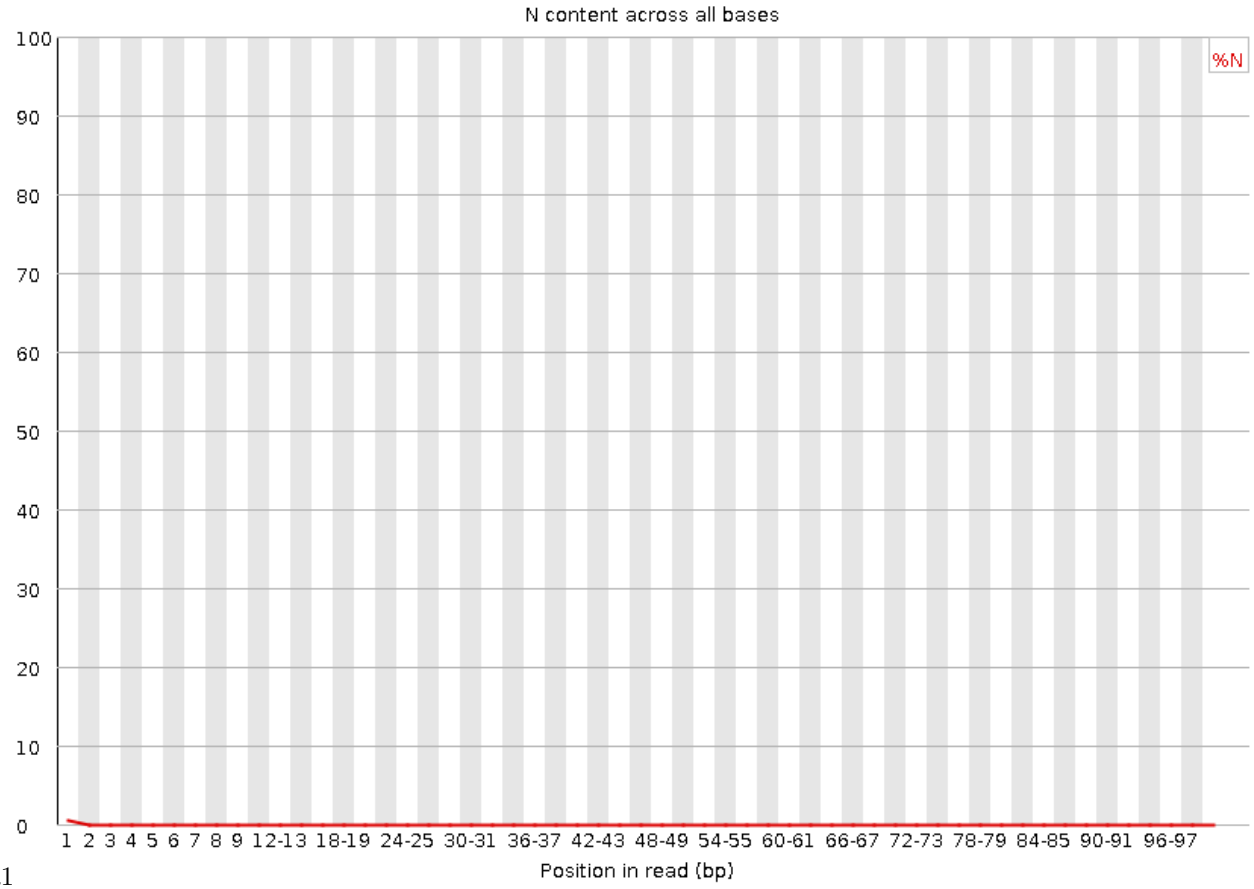



R2

10_2G_both_S8_L008: Forward

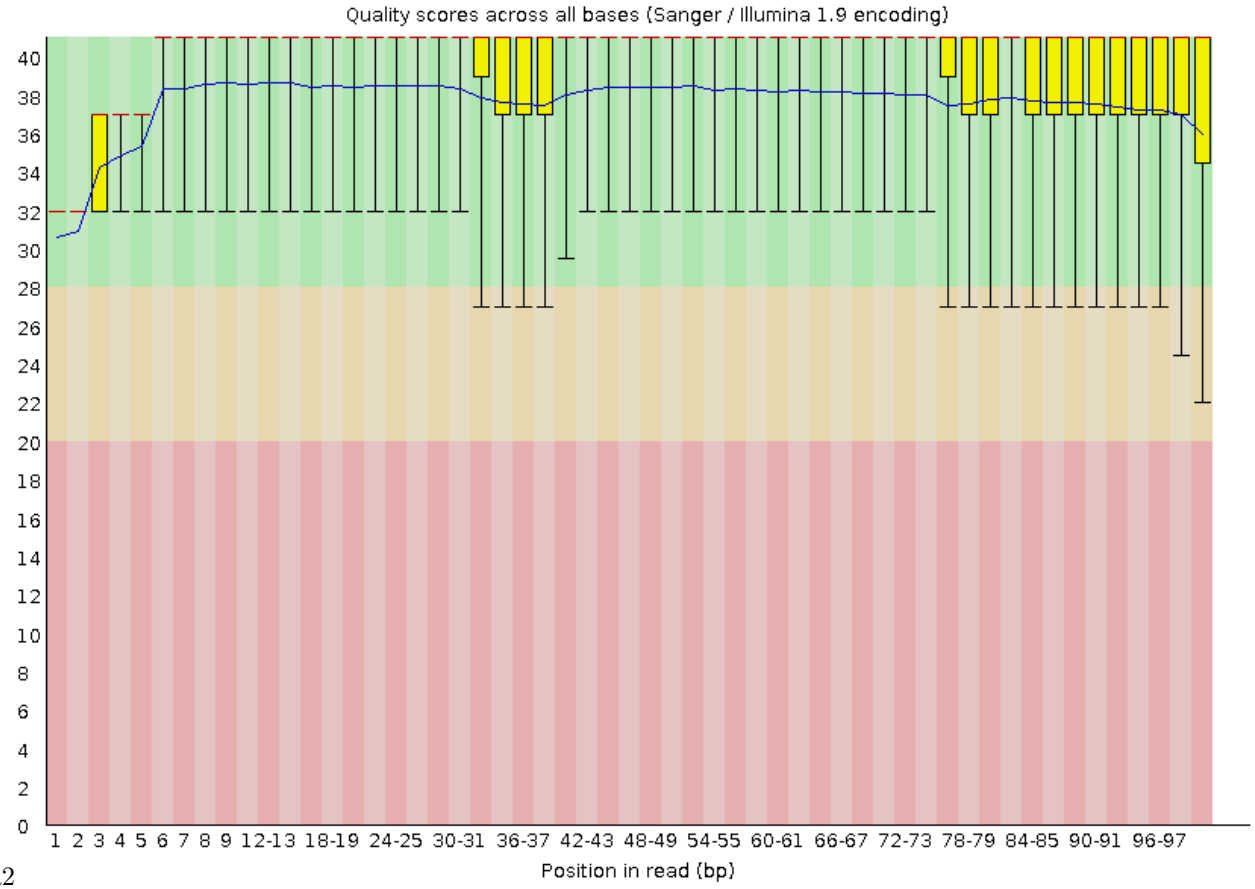

Quality scores across all bases (Sanger / Illumina 1.9 encoding)
40
38
36
34
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
0
1 2 3 4 5 6 7 8 9 12-13 18-19 24-25 30-31 36-37 42-43 48-49 54-55 60-61 66-67 72-73 78-79 84-85 90-91 96-97
Position in read (bp)

#R1

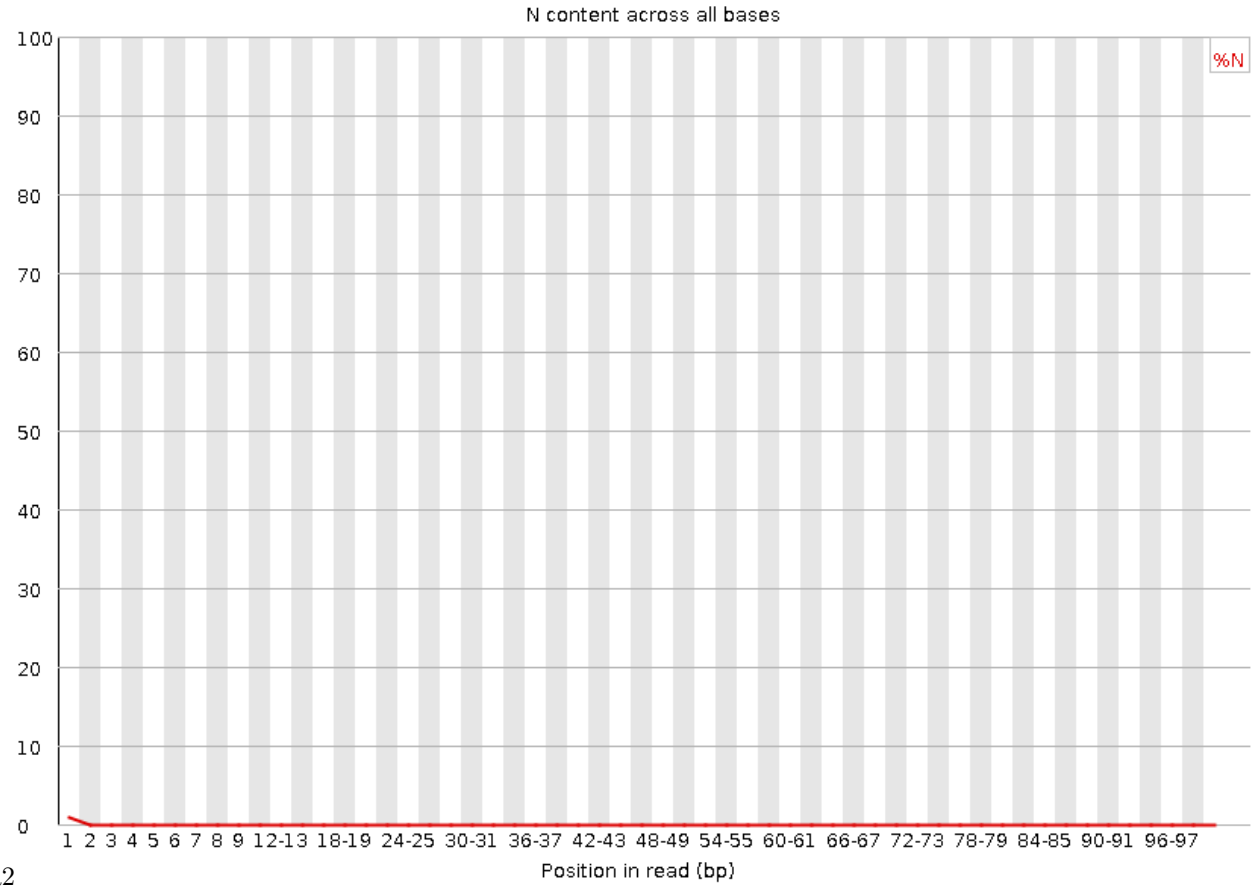



#R1

10_2G_both_S8_L008: Reverse


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)

#R2




#R2

#Question 1.1 The quality scores and per base n content are consistent across the both reads for both files. The quality scores for the R2(reverse compliment) reads are lower then the R1(forward) reads but this is to be expected since the R2 reads are sequenced later and sit on the sequencer long leading to sample degradation.

#Question 1.2 Fastqc is significantly faster than running that my python script. This is likely because Fastqc is a streamlined program designed to elimante the inefficiencies that my code likely has. Presumably FastQC is limiting the amount of extra loops and other inefficient search methods used in its code so that it runs as fast as possible.

#Question 1.3 The quality of both libraries is good. The forward reads are high quality. The reverse compliment reads are lower quality then would be ideal in cetain places with the scores dropping below 28.

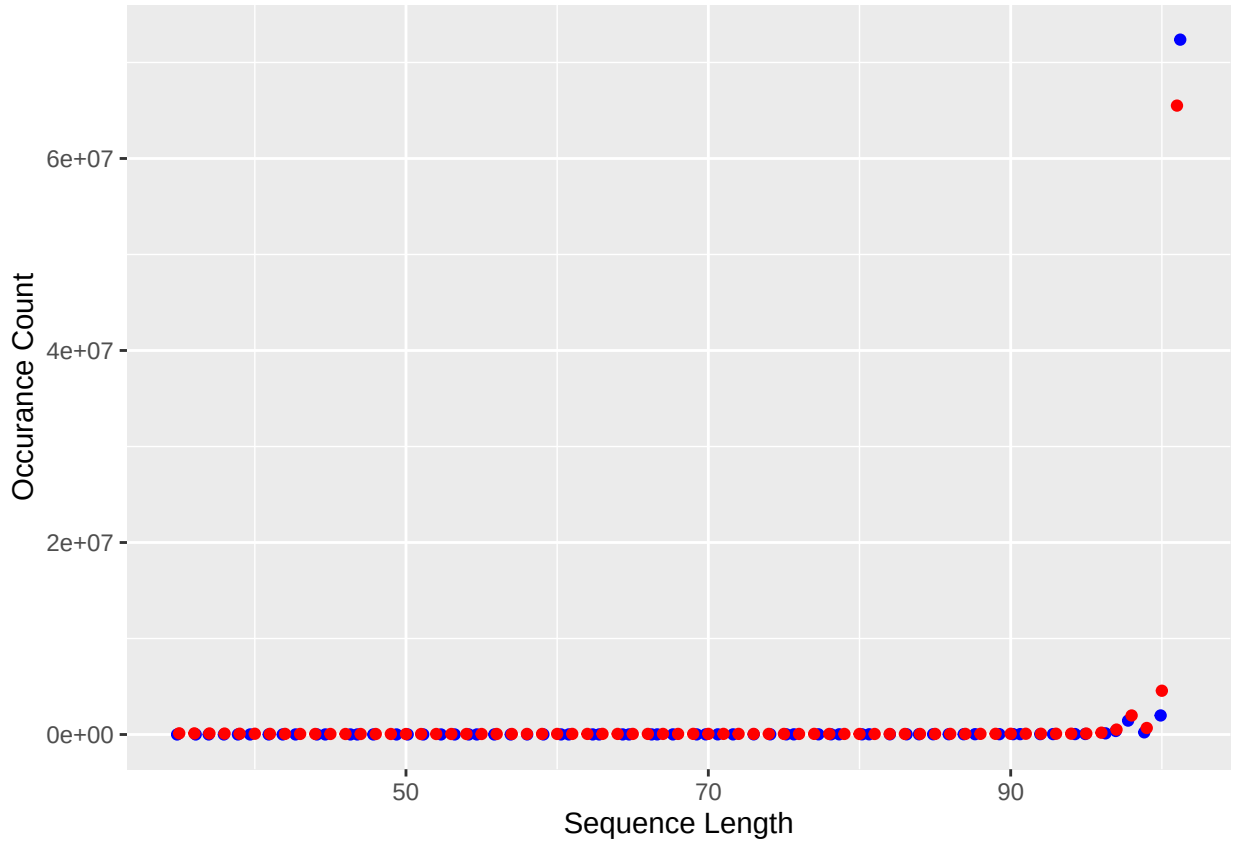
```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## v purrr   0.3.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

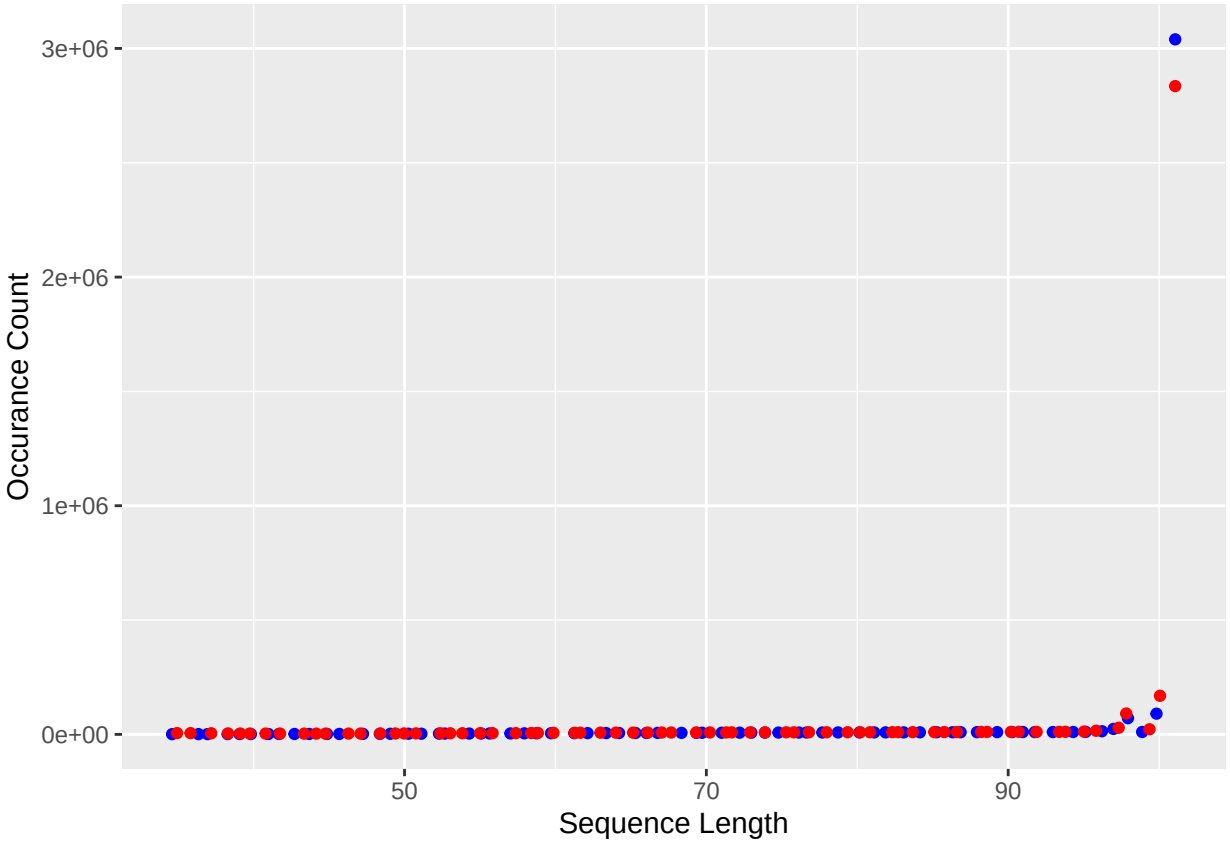
```
df_10_R1 <- read.table("C:\\Users\\ano\\Desktop\\BGMP R Folder\\qaa r data\\dist_10_R1_paired.tsv")
df_10_R2 <- read.table("C:\\Users\\ano\\Desktop\\BGMP R Folder\\qaa r data\\dist_10_R2_paired.tsv")
df_30_R1 <- read.table("C:\\Users\\ano\\Desktop\\BGMP R Folder\\qaa r data\\dist_31_R1_paired.tsv")
df_30_R2 <- read.table("C:\\Users\\ano\\Desktop\\BGMP R Folder\\qaa r data\\dist_31_R2_paired.tsv")


ggplot(df_10_R1, aes(V2, V1)) +
  geom_point(color = 'blue', position = position_jitter()) +
  geom_point(data = df_10_R2, color = 'red', position = 'dodge') +
  labs(x = "Sequence Length", y = "Occurance Count")
```

```
## Warning: Width not defined. Set with `position_dodge(width = ?)`
```




```
ggplot(df_30_R1, aes(V2, V1)) +
  geom_point(color = 'blue', position = position_jitter()) +
  geom_point(data = df_30_R2, color = 'red', position = position_jitter()) +
  labs(x = "Sequence Length", y = "Occurance Count")
```




The graphs show nearly identical distributions for both the forward and reverse reads. There are slightly more long forward reads, this is likely because the reverse reads have to sit on the sequencer longer and so they degrade to shorter lengths, this is expected.

#Question 3.10. STAR & PS8 Script 31_4F has 6969880 (75.49%) Mapped and 226320 Unmapped Reads(24.52) 10-2g has 152719495 (98.49%) Mapped and Unmapped 2332498 (1.51%)

#Question 3.12

HTSEQ 31_4f TOTAL 3597908 Mapped Forward 186062 (5.2%) Mapped Reverse 2957496(82.2%),

This sample is stranded since if it was unstranded we would see the same (or close to same) amount of mapped genes on both the forward and reverse files. Since there is a high degree of difference between the forward mapped genes (5.2%) and reverse mapped genes (82.2%) we know that there are genes in the reverse file that we mapped that we did not map in the forward mapping meaning that the sample is strand specific.

10_2g TOTAL 77520903 Mapped Forward Mapped Reverse 67365786

As above, this sample is also stranded since Mapped forward genes are only % and the mapped reverse genes are % percent of the total.

| Sample | Total | Mapped# | Mapped % |
|---|---|---|---|
| 31_4f Forward | 3597908 | 186062 | 5.2% |
| 31_4f Reverse | 3597908 | 2957496 | 82.2% |
| 10_2g Forward | are neat | $1 | |
| 10_2g Reverse | are neat | $1 | |