QAA

Natalie Elphick

9/4/2021

Part 1 – Read quality score distributions

FastQC bash commands :

```
fastqc --extract -o /projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA
/projects/bgmp/shared/2017_sequencing/
  demultiplexed/32_4G_both_S23_L008_R1_001.fastq.gz
/projects/bgmp/shared/2017_sequencing/
  demultiplexed/32_4G_both_S23_L008_R2_001.fastq.gz

fastqc --extract -o /projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA
/projects/bgmp/shared/2017_sequencing/
  demultiplexed/3_2B_control_S3_L008_R1_001.fastq.gz
/projects/bgmp/shared/2017_sequencing/
  demultiplexed/3_2B_control_S3_L008_R2_001.fastq.gz
```

qscore_plot.py

```
#!/usr/bin/env python
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import gzip
import argparse

#Get required variables
parser = argparse.ArgumentParser(description="TBD")
parser.add_argument("-r1", "--r1_filename", help="filename for read1", required=True)
parser.add_argument("-r2", "--r2_filename", help="file name for read2", required=True)
parser.add_argument("-sn", "--sample_name",
help="Sample label to append to output file names", required=True)
args = parser.parse_args()

read1=str(args.r1_filename)
read2=str(args.r2_filename)

sample_name=str(args.sample_name)
# indexes = "/projects/bgmp/shared/2017_sequencing/indexes.txt"

def populate_list(fh,read_length):
```

```
    '''Takes in a fastq file with reads of length 101 and
     populates a numpy array with all of the quality scores for each base. '''

    qlist = np.zeros([read_length])
    k=1
    for line in fh:
        if k%4 ==0:
            for i,qscore in enumerate(line.strip()):
                qlist[i]+=qscore-33

        k+=1
    counter= k-1
    return qlist,counter

def plot_mean(mean_array,read_length,fh_name):
    '''Takes in a numpy array of means and graphs the distribution,
    saving the file to a given name'''
    x = range(0,read_length)
    y = mean_array
    plt.plot(x, y)
    plt.xlabel('Base', fontsize=15)
    plt.ylabel('Mean Quality Score', fontsize=15)
    plt.savefig("mean_distribution_"+str(fh_name)+".png")
    plt.close()

r1_f = gzip.open(read1)
r2_f = gzip.open(read2)

L_r1, c_r1 = populate_list(r1_f,101)
L_r2, c_r2 = populate_list(r2_f,101)

mean_r1 = L_r1/(c_r1/4)
mean_r2 = L_r2/(c_r2/4)

plot_mean(mean_r1,101,str(sample_name +" "+"read_1"))
plot_mean(mean_r2,101,str(sample_name +" "+"read_2"))

r1_f.close()
r2_f.close()
```

qscore_plot_wrapper.sh

```
#!/bin/bash

#SBATCH --partition=bgmp        ### Partition (like a queue in PBS)
#SBATCH --job-name=qscore_plot   ### Job Name
#SBATCH --output=qscore_plot_%j.out       ### File in which to store job output
#SBATCH --error=qscore_plot_%j.err ### File in which to store job error messages
#SBATCH --time=0-12:01:00       ### Wall clock time limit in Days-HH:MM:SS
#SBATCH --nodes=1              ### Number of nodes needed for the job
#SBATCH --ntasks-per-node=8    ### Number of tasks to be launched per node
#SBATCH --account=bgmp      ### Account used for job submission
```

```
conda activate bgmp_py39

/usr/bin/time -v python3 ./qscore_plot.py -r1 /projects/bgmp/shared/
  2017_sequencing/demultiplexed/3_2B_control_S3_L008_R1_001.fastq.gz \
-r2 /projects/bgmp/shared/
  2017_sequencing/demultiplexed/3_2B_control_S3_L008_R2
_001.fastq.gz \
-sn "3_2B_control"

/usr/bin/time -v python3 ./qscore_plot.py -r1 /projects/bgmp/shared/
  2017_sequencing/demultiplexed/32_4G_both_S23_L008_R1_001
.fastq.gz \
-r2 /projects/bgmp/shared/
  2017_sequencing/demultiplexed/32_4G_both_S23_L008_R2
_001.fastq.gz \
-sn "32_4G_both"
```

```
plot1 <- readPNG('3_2B_control_S3_L008_R1_001_fastqc/Images/per_base_quality.png')
plot2 <- readPNG('mean_distribution_3_2B_control read_1.png')

plot3 <- readPNG('3_2B_control_S3_L008_R2_001_fastqc/Images/per_base_quality.png')
plot4 <- readPNG('mean_distribution_3_2B_control read_2.png')

plot5 <- readPNG('32_4G_both_S23_L008_R1_001_fastqc/Images/per_base_quality.png')
plot6 <- readPNG('mean_distribution_32_4G_both read_1.png')

plot7 <- readPNG('32_4G_both_S23_L008_R2_001_fastqc/Images/per_base_quality.png')
plot8 <- readPNG('mean_distribution_32_4G_both read_2.png')
```

```
grid.arrange(rasterGrob(plot1),
             rasterGrob(plot2),
             rasterGrob(plot3),
             rasterGrob(plot4),
             ncol=2,
             top="3_2B control")
```

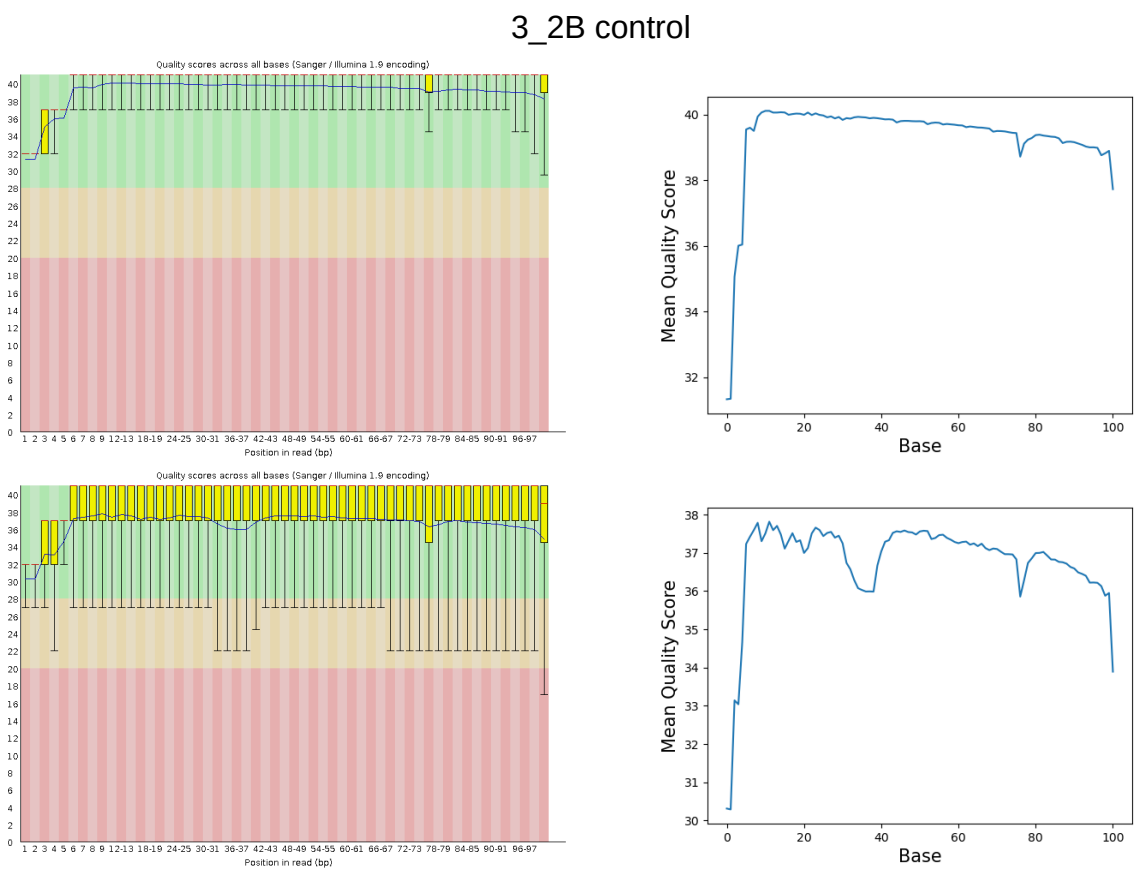

Figure 1: FASTQC plots vs plots generated by qscore_plot.py.

```
grid.arrange(rasterGrob(plot5),
             rasterGrob(plot6),
             rasterGrob(plot7),
             rasterGrob(plot8),
             ncol=2,
             top="32_4G both")
```

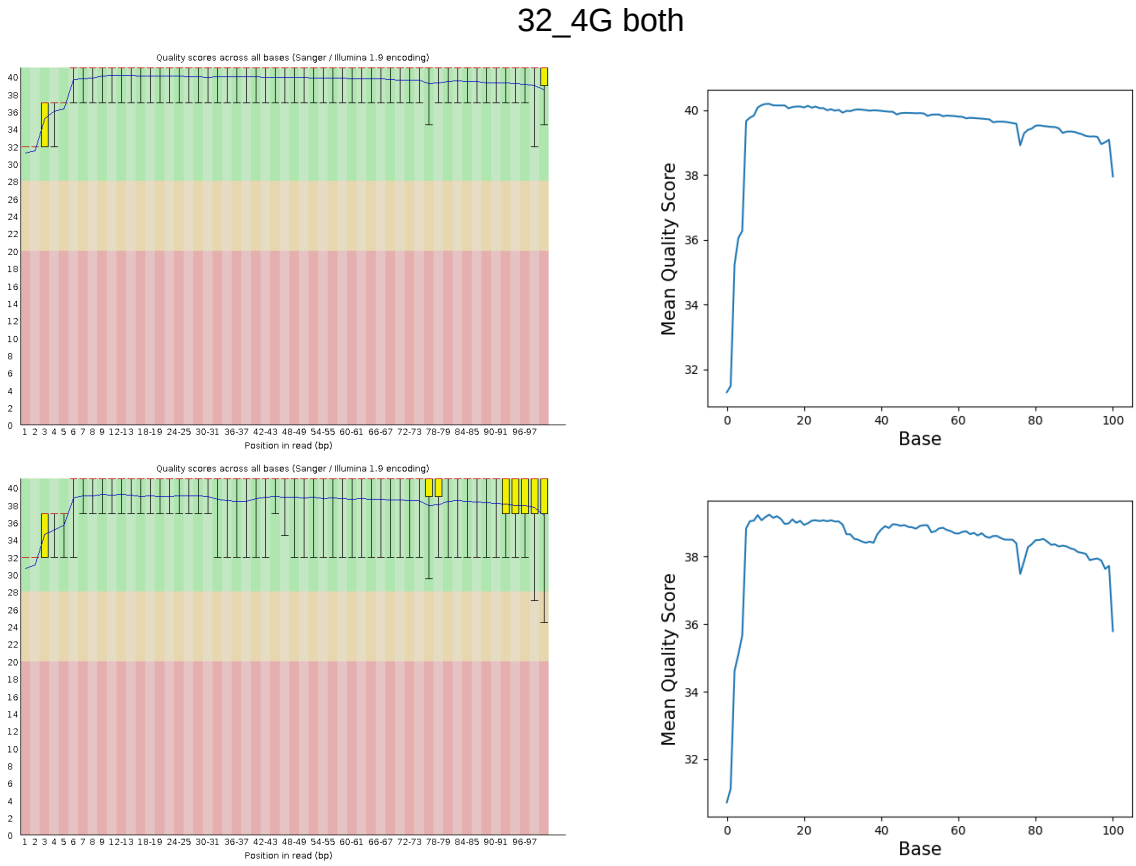

Figure 2: FASTQC plots vs plots generated by qscore_plot.py.

- The FastQC plots contain box whisker plots for each position, the yellow box being the interquartile range, the red line being the median and upper and lower whiskers representing the 10% and 90% points. The plots generated by qscore_plot.py only plot the mean q score and have a wider scale for the y axis which makes the reads look worse than they are. FastQC was faster than qscore_plot.py, because it has been optimized for this task.
- The overall quality of both libraries looks very good, both contain low levels of adapter content and good per base sequence quality.

Part 2 – Adaptor trimming comparison

Create QAA conda environment:

```
conda create --name QAA
conda activate QAA
conda install cutadapt=3.4
conda install trimmomatic=0.39
```

Run cutadapt on all 4 files using cut_adapters.sh:

```
#!/bin/bash

#SBATCH --partition=bgmp        ### Partition (like a queue in PBS)
#SBATCH --job-name=cut_adapt   ### Job Name
#SBATCH --output=cut_adapt_%j.out ### File in which to store job output
#SBATCH --error=cut_adapt_%j.err  ### File in which to store job error messages
#SBATCH --time=0-12:01:00       ### Wall clock time limit in Days-HH:MM:SS
#SBATCH --nodes=1              ### Number of nodes needed for the job
#SBATCH --ntasks-per-node=8    ### Number of tasks to be launched per node
#SBATCH --account=bgmp      ### Account used for job submission



conda activate QAA

/usr/bin/time -v cutadapt -j 8 -a AGATCGGAAGAGCACACGTCTGAACTCCAGTCA \
-A AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT \
-o /projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/
  Adapter_trimmed/3_2B_control.1.fastq \
-p /projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/
  Adapter_trimmed/3_2B_control.2.fastq \
/projects/bgmp/shared/2017_sequencing/demultiplexed/
  3_2B_control_S3_L008_R1_001.fastq.gz \
/projects/bgmp/shared/2017_sequencing/demultiplexed/
  3_2B_control_S3_L008_R2_001.fastq.gz

/usr/bin/time -v cutadapt -j 8 -a AGATCGGAAGAGCACACGTCTGAACTCCAGTCA \
-A AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT \
-o /projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/
  Adapter_trimmed/32_4G_both.1.fastq \
-p /projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/
  Adapter_trimmed/32_4G_both.2.fastq \
/projects/bgmp/shared/2017_sequencing/demultiplexed/
  32_4G_both_S23_L008_R1_001.fastq.gz \
/projects/bgmp/shared/2017_sequencing/demultiplexed
/32_4G_both_S23_L008_R2_001.fastq.gz
```

Proportion of read 1 and read 2 trimmed for both sets of files:

File	% of bp trimmed
3_2B_control Read 1	%0.24
3_2B_control Read 2	%0.27
32_4G_both Read 1	%0.63
32_4G_both Read 2	%0.66

3__2B__control

Read 1

Adapters at the begining of the reads =0

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
  3_2B_control_S3_L008_R1_001.fastq.gz |
  grep -E -c "^AGATCGGAAGAGCACACGTCTGAACTCCAGTCA"
```

Adapters anywhere in the reads = 7659

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
  3_2B_control_S3_L008_R1_001.fastq.gz |
  grep -E -c "AGATCGGAAGAGCACACGTCTGAACTCCAGTCA"
```

Adapters at the end of the reads = 27494036

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
  3_2B_control_S3_L008_R1_001.fastq.gz |
  grep -E -c "$AGATCGGAAGAGCACACGTCTGAACTCCAGTCA"
```

Read 2

Adapters at the begining of the reads =0

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
  3_2B_control_S3_L008_R2_001.fastq.gz |
  grep -E -c "^AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT"
```

Adapters anywhere in the reads = 8157

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
  3_2B_control_S3_L008_R2_001.fastq.gz |
  grep -E -c "AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT"
```

Adapters at the end of the reads = 27494036

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
  3_2B_control_S3_L008_R2_001.fastq.gz |
  grep -E -c "$AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT"
```

32__4G__both

Read 1

Adapters at the begining of the reads =0

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
  32_4G_both_S23_L008_R1_001.fastq.gz |
  grep -E -c "^AGATCGGAAGAGCACACGTCTGAACTCCAGTCA"
```

Adapters anywhere in the reads = 7659

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
  32_4G_both_S23_L008_R1_001.fastq.gz |
  grep -E -c "AGATCGGAAGAGCACACGTCTGAACTCCAGTCA"
```

Adapters at the end of the reads = 27494036

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
  32_4G_both_S23_L008_R1_001.fastq.gz |
  grep -E -c "$AGATCGGAAGAGCACACGTCTGAACTCCAGTCA"
```

Read 2

Adapters at the begining of the reads =0

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
  32_4G_both_S23_L008_R2_001.fastq.gz |
  grep -E -c "^AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT"
```

Adapters anywhere in the reads = 8157

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
  32_4G_both_S23_L008_R2_001.fastq.gz |
  grep -E -c "AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT"
```

Adapters at the end of the reads = 27494036

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
  32_4G_both_S23_L008_R2_001.fastq.gz |
  grep -E -c "$AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT"
```

Run trimmomatic.sh on the adapter trimmed files:

```
#!/bin/bash

#SBATCH --partition=bgmp        ### Partition (like a queue in PBS)
#SBATCH --job-name=trimmomatic   ### Job Name
```

```
#SBATCH --output=trimmomatic_%j.out         ### File in which to store job output
#SBATCH --error=trimmomatic_%j.err          ### File in which to store job error messages
#SBATCH --time=0-12:01:00       ### Wall clock time limit in Days-HH:MM:SS
#SBATCH --nodes=1               ### Number of nodes needed for the job
#SBATCH --ntasks-per-node=8     ### Number of tasks to be launched per node
#SBATCH --account=bgmp      ### Account used for job submission


conda activate QAA

/usr/bin/time -v java -jar /projects/bgmp/nelphick/miniconda3/envs/QAA/share
/trimmomatic/trimmomatic.jar \
PE -threads 8 \
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Adapter_trimmed/
  3_2B_control.1.fastq \
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Adapter_trimmed/
  3_2B_control.2.fastq \
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Quality_trimmed/
  3_2B_control_1_paired.fq.gz \
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Quality_trimmed/
  3_2B_control_1_unpaired.fq.gz \
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Quality_trimmed/
  3_2B_control_2_paired.fq.gz \
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Quality_trimmed/
  3_2B_control_2_unpaired.fq.gz \
LEADING:3 TRAILING:3 SLIDINGWINDOW:5:15 MINLEN:35


/usr/bin/time -v java -jar /projects/bgmp/nelphick/miniconda3/envs/QAA/share
/trimmomatic/trimmomatic.jar \
PE -threads 8 \
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Adapter_trimmed/
  32_4G_both.1.fastq \
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Adapter_trimmed/
  32_4G_both.2.fastq \
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Quality_trimmed/
  32_4G_both_1_paired.fq.gz \
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Quality_trimmed/
  32_4G_both_1_unpaired.fq.gz \
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Quality_trimmed/
  32_4G_both_2_paired.fq.gz \
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Quality_trimmed/
  32_4G_both_2_unpaired.fq.gz \
LEADING:3 TRAILING:3 SLIDINGWINDOW:5:15 MINLEN:35
```

Unix commands to get distribution of counts for each quality trimmed file:

```
zcat /projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Quality_trimmed/
  3_2B_control_1_paired.fq.gz | awk '{if(NR%4==2) print length($1)}' | sort |
  uniq -c > 3_2B_control_1_lengths.txt


zcat /projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Quality_trimmed/
```

```
  3_2B_control_2_paired.fq.gz | awk '{if(NR%4==2) print length($1)}' | sort |
  uniq -c > 3_2B_control_2_lengths.txt

zcat /projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Quality_trimmed/
  32_4G_both_1_paired.fq.gz | awk '{if(NR%4==2) print length($1)}' | sort |
  uniq -c > 32_4G_both_1_lengths.txt

zcat /projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Quality_trimmed/
  32_4G_both_2_paired.fq.gz | awk '{if(NR%4==2) print length($1)}' | sort |
  uniq -c > 32_4G_both_2_lengths.txt
```

```
# Read in and fix the frequency column formmatting
read_delim("3_2B_control_1_lengths.txt",col_names = F, delim = " ") %>%
  mutate(X1 = as.numeric(gsub(" ", "", X1)),Read="Read 1") %>%
  rename(freq = X1, len_bp= X2)-> R1_3_2B_control

read_delim("3_2B_control_2_lengths.txt",col_names = F, delim = " ") %>%
  mutate(X1 = as.numeric(gsub(" ", "", X1)),Read="Read 2") %>%
  rename(freq = X1, len_bp= X2)-> R2_3_2B_control

read_delim("32_4G_both_1_lengths.txt",col_names = F, delim = " ") %>%
  mutate(X1 = as.numeric(gsub(" ", "", X1)),Read="Read 1") %>%
  rename(freq = X1, len_bp= X2)-> R1_32_4G_both

read_delim("32_4G_both_2_lengths.txt",col_names = F, delim = " ") %>%
  mutate(X1 = as.numeric(gsub(" ", "", X1)),Read="Read 2") %>%
  rename(freq = X1, len_bp= X2)-> R2_32_4G_both
```

```
rbind(R1_3_2B_control,R2_3_2B_control)%>%
  ggplot(aes(x= len_bp, fill = Read))+
  geom_bar(aes(y= freq),stat = "identity",position = position_dodge())+
  xlab(label = "Length of fragments") +
  ylab(label = "log2 scaled frequency") +
  labs(title = "3_2B control")+
  theme(plot.title = element_text(hjust = 0.5))+
  scale_y_continuous(trans="log2")
```

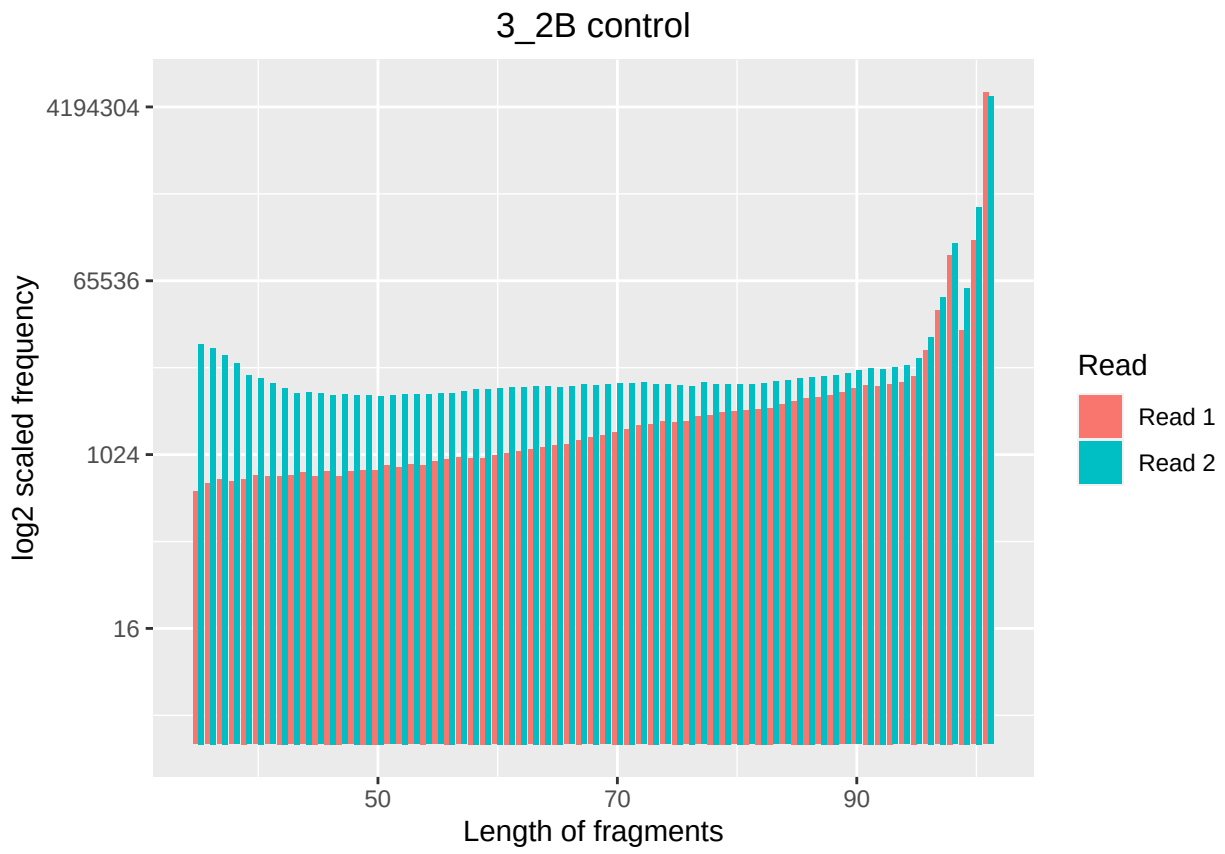

Figure 3: 3_2B control read length distribution after quality trimming.

```
rbind(R1_32_4G_both,R2_32_4G_both)%>%
  ggplot(aes(x= len_bp, fill = Read))+
  geom_bar(aes(y= freq),stat = "identity",position = position_dodge())+
  xlab(label = "Length of fragments") +
  ylab(label = "log2 scaled frequency") +
  labs(title = "32_4G both")+
  theme(plot.title = element_text(hjust = 0.5))+
  scale_y_continuous(trans="log2")
```

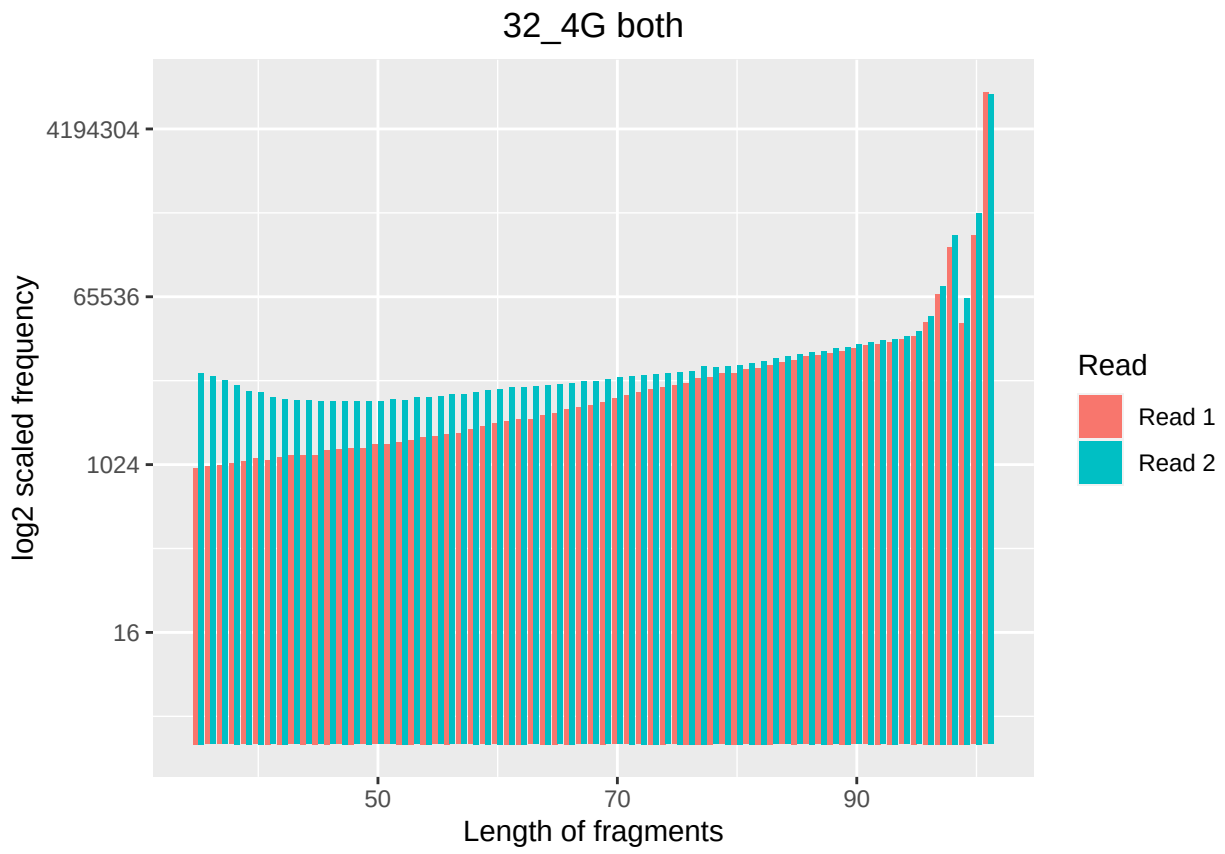

Figure 4: 32_4G both read length distribution after quality trimming.

- As expected, read 2 is adapter and quality trimmed at a higher rate than read1. Resulting in a higher frequency of shorter lengths for read 2.

Part 3 – Alignment and strand-specificity

Install libraries for alignment step

```
conda install star
conda install numpy
conda install pysam
conda install matplotlib

pip install HTseq
```

Generate genome database using STAR:
generate_genome_db.sh

```
#!/bin/bash

#SBATCH --partition=bgmp        ### Partition (like a queue in PBS)
```

```
#SBATCH --job-name=genome_generate   ### Job Name
#SBATCH --output=genome_generate_%j.out   ### File in which to store job output
#SBATCH --error=genome_generate_%j.err  ### File in which to store job error messages
#SBATCH --time=0-01:01:00       ### Wall clock time limit in Days-HH:MM:SS
#SBATCH --nodes=1              ### Number of nodes needed for the job
#SBATCH --ntasks-per-node=8     ### Number of tasks to be launched per node
#SBATCH --account=bgmp      ### Account used for job submission

conda activate QAA

/usr/bin/time -v STAR --runThreadN 8 \
--runMode genomeGenerate \
--genomeDir ./Mus_musculus.GRCm39.ens104.STAR_2.7.9a \
--genomeFastaFiles /projects/bgmp/nelphick/bioinfo/Bi623/Assignments/
  QAA/Mus/Mus_musculus.GRCm39.dna_sm.primary_assembly.fa \
--sjdbGTFfile /projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Mus/
  Mus_musculus.GRCm39.104.gtf
```

Align read files using STAR:

align.sh

```
#!/bin/bash

#SBATCH --partition=bgmp        ### Partition (like a queue in PBS)
#SBATCH --job-name=align_mus  ### Job Name
#SBATCH --output=align_mus_%j.out         ### File in which to store job output
#SBATCH --error=align_mus_%j.err          ### File in which to store job error messages
#SBATCH --time=0-01:01:00       ### Wall clock time limit in Days-HH:MM:SS
#SBATCH --nodes=1              ### Number of nodes needed for the job
#SBATCH --ntasks-per-node=8     ### Number of tasks to be launched per node
#SBATCH --account=bgmp      ### Account used for job submission
conda activate QAA

/usr/bin/time -v STAR --runThreadN 8 --runMode alignReads \
 --outFilterMultimapNmax 3 \
 --outSAMunmapped Within KeepPairs \
 --alignIntronMax 1000000 --alignMatesGapMax 1000000 \
 --readFilesCommand zcat \
 --readFilesIn   /projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/
  Quality_trimmed/3_2B_control_1_paired.fq.gz
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Quality_trimmed/
  3_2B_control_2_paired.fq.gz\
 --genomeDir /projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/
  Mus_musculus.GRCm39.ens104.STAR_2.7.9a \
 --outFileNamePrefix 3_2B_control_Mus_align/3_2B_control_Mus_align_

/usr/bin/time -v STAR --runThreadN 8 --runMode alignReads \
 --outFilterMultimapNmax 3 \
 --outSAMunmapped Within KeepPairs \
 --alignIntronMax 1000000 --alignMatesGapMax 1000000 \
 --readFilesCommand zcat \
 --readFilesIn   /projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/
  Quality_trimmed/32_4G_both_1_paired.fq.gz
```

```
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Quality_trimmed/
  32_4G_both_2_paired.fq.gz\
 --genomeDir /projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/
  Mus_musculus.GRCm39.ens104.STAR_2.7.9a \
 --outFileNamePrefix 32_4G_both_Mus_align/32_4G_both_Mus_align_
```

Count mapped reads of the aligned files using mapped_count.py

mapped_count_wrapper.sh

```
#!/bin/bash

#SBATCH --partition=bgmp        ### Partition (like a queue in PBS)
#SBATCH --job-name=mapped_count   ### Job Name
#SBATCH --output=mapped_count_%j.out         ### File in which to store job output
#SBATCH --error=mapped_count_%j.err          ### File in which to store job error messages
#SBATCH --time=0-01:01:00       ### Wall clock time limit in Days-HH:MM:SS
#SBATCH --nodes=1              ### Number of nodes needed for the job
#SBATCH --ntasks-per-node=8     ### Number of tasks to be launched per node
#SBATCH --account=bgmp      ### Account used for job submission


/usr/bin/time -v python3 ./mapped_count.py -f
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/3_2B_control_Mus_align/
  3_2B_control_Mus_align_Aligned.out.sam \
-o 3_2B_control_mapped_stats.txt

/usr/bin/time -v python3 ./mapped_count.py -f
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/32_4G_both_Mus_align/
  32_4G_both_Mus_align_Aligned.out.sam \
-o 32_4G_both_mapped_stats.txt
```

mapped_count.py

```
#!/user/bin/env python

import argparse


#Get required variables
parser = argparse.ArgumentParser(description="Count unmapped and mapped reads
                                  using alligned SAM file")

parser.add_argument("-f", "--filename", help="Name of fa file", required=True)
parser.add_argument("-o", "--output_filename", help="K-mer size", required=True)
args = parser.parse_args()

file_name=str(args.filename)
output_filename =str(args.output_filename)

file = open(file_name,"r")
out_file = open(output_filename, "x")
```

```
mapped_counter=0
unmapped_counter=0
lc=0

while True:
    line = file.readline().strip()

    if line == "":
        break
    if not line.startswith("@"): #check for qname
        lc+=1
        line_items =line.split()

        bit_flag = line_items[1]

        if (int(bit_flag) & 4) !=4 and (int(bit_flag) & 256) !=256: #is it mapped?
           mapped_counter+=1
        else:
            if (int(bit_flag) & 256) !=256:
                unmapped_counter+=1




#write out results
out_file.write(file_name+"\n")
out_file.write("Number of mapped reads: " + str(mapped_counter) +"\n")
out_file.write("Number of unmapped reads: " + str(unmapped_counter)+"\n")
out_file.write("Number of reads: " + str(mapped_counter+unmapped_counter)+"\n")
out_file.write("Number of lines: " + str(lc)+"\n")


out_file.close()
file.close()
```

3_2B_control_Mus_align_Aligned.out.sam

Mapped reads: 12359960
Unmapped reads: 496078

32_4G_both_Mus_align_Aligned.out.sam

Mapped reads: 22404322
Unmapped reads: 533612

Run htseq-count on the aligned sam files

htseq_count.sh

```
#!/bin/bash

#SBATCH --partition=bgmp        ### Partition (like a queue in PBS)
#SBATCH --job-name=htseq_count   ### Job Name
#SBATCH --output=htseq_count_%j.out         ### File in which to store job output
#SBATCH --error=htseq_count_%j.err          ### File in which to store job error messages
#SBATCH --time=0-12:01:00       ### Wall clock time limit in Days-HH:MM:SS
#SBATCH --nodes=1               ### Number of nodes needed for the job
#SBATCH --ntasks-per-node=8     ### Number of tasks to be launched per node
#SBATCH --account=bgmp      ### Account used for job submission

conda activate QAA

/usr/bin/time -v htseq-count --stranded=yes \
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/
  3_2B_control_Mus_align/3_2B_control_Mus_align_Aligned.out.sam \
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Mus/
  Mus_musculus.GRCm39.104.gtf > 3_2B_control_htseq_coun_stranded.txt

/usr/bin/time -v htseq-count --stranded=no \
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/3_2B_control_Mus_align/
  3_2B_control_Mus_align_Aligned.out.sam \
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Mus/
  Mus_musculus.GRCm39.104.gtf > 3_2B_control_htseq_coun_unstranded.txt

/usr/bin/time -v htseq-count --stranded=yes \
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/
  32_4G_both_Mus_align/32_4G_both_Mus_align_Aligned.out.sam \
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Mus/
  \Mus_musculus.GRCm39.104.gtf > 32_4G_both_htseq_coun_stranded.txt

/usr/bin/time -v htseq-count --stranded=no \
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/
  32_4G_both_Mus_align/32_4G_both_Mus_align_Aligned.out.sam \
/projects/bgmp/nelphick/bioinfo/Bi623/Assignments/QAA/Mus/
  Mus_musculus.GRCm39.104.gtf > 32_4G_both_htseq_coun_unstranded.txt
```

Summarise htseq-count results

Number of reads that mapped to a feature:

```
awk '/^E/{a[FILENAME]+=$2} END{for(f in a) print f, a[f]}' 32_4G_both_htseq_coun_*
32_4G_both_htseq_coun_unstranded.txt 9611948
32_4G_both_htseq_coun_stranded.txt 433654

awk '/^E/{a[FILENAME]+=$2} END{for(f in a) print f, a[f]}' 3_2B_control_htseq_coun_*
3_2B_control_htseq_coun_stranded.txt 234894
3_2B_control_htseq_coun_unstranded.txt 5076414
```

Total number of reads

```
awk '{a[FILENAME]+=$2} END{for(f in a) print f, a[f]}' 32_4G_both_htseq_coun_*
```

```
32_4G_both_htseq_coun_unstranded.txt 11468967
32_4G_both_htseq_coun_stranded.txt 11468967

awk '{a[FILENAME]+=$2} END{for(f in a) print f, a[f]}' 3_2B_control_htseq_coun_*
3_2B_control_htseq_coun_stranded.txt 6428019
3_2B_control_htseq_coun_unstranded.txt 6428019
```

Mapped percentages

```
32_4G_both_htseq_coun_stranded.txt %3.78
32_4G_both_htseq_coun_unstranded.txt %83.81

3_2B_control_htseq_coun_stranded.txt %3.65
3_2B_control_htseq_coun_unstranded.txt %78.97
```

- Both sets of fastq files are most likely non-strand specific because a much lower percentage of features mapped using the –stranded=yes option (32_4G_both=%3.78,3_2B_control=%3.65) compared to the –stranded=no option (32_4G_both=%83.81,3_2B_control=%78.97).