RNA-seq Quality Assessment Assignment

Juyoung Lee

9/7/2021

Data

2 demultiplexed file pairs:
1. 16_3D_mbnl_S12_L008
2. 21_3G_both_S15_L008

Part 1: Read quality score distributions

1. Use FastQC to produce quality score distribution graphs and per-base N content for R1 and R2.

```
fastqc \
  -o /projects/bgmp/jlee26/bioinformatics/Bi623/qaa \
  -t 2 \
  /projects/bgmp/shared/2017_sequencing/demultiplexed/
  16_3D_mbnl_S12_L008_R1_001.fastq.gz \
  /projects/bgmp/shared/2017_sequencing/demultiplexed/
  16_3D_mbnl_S12_L008_R2_001.fastq.gz

fastqc \
  -o /projects/bgmp/jlee26/bioinformatics/Bi623/qaa \
  -t 2 \
  /projects/bgmp/shared/2017_sequencing/demultiplexed/
  21_3G_both_S15_L008_R1_001.fastq.gz \
  /projects/bgmp/shared/2017_sequencing/demultiplexed/
  21_3G_both_S15_L008_R2_001.fastq.gz
```

Figure 1 - 4 shows the distribution of quality scores and n content at each base for read 1 and read 2. The N content is higher at the first couple of bases, which is consistent with the lower quality score at the beginning of the bases. The N content is higher for read 2 for both demultiplexed file pairs compared to read 1, which is also consistent with lower quality score for read 2 compared to read 1.

16_3D_mbnl_S12_L008 Read 1

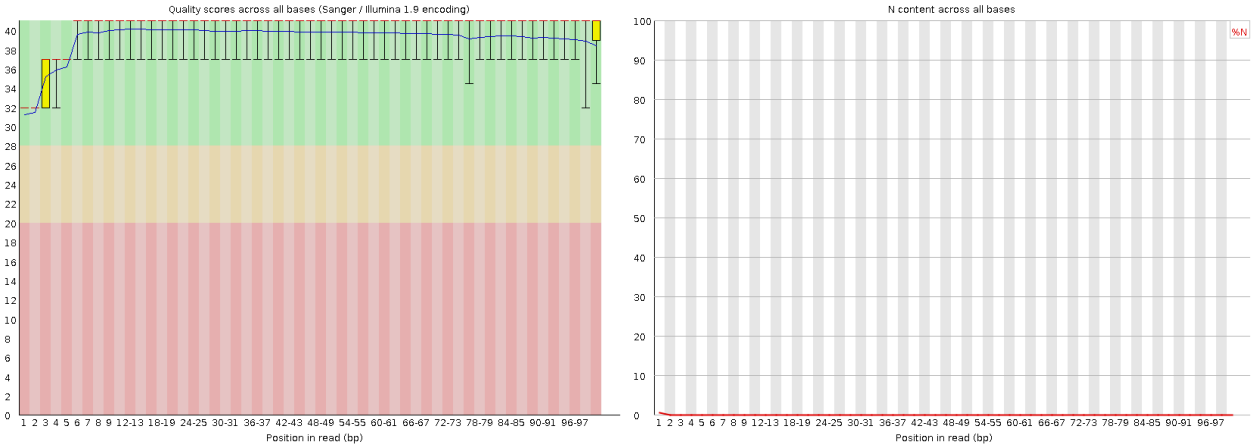

Figure 1. Quality score distribution (left) and per-base n content (right) for 16_3D_mbnl_S12_L008 R1.

16_3D_mbnl_S12_L008 Read 2

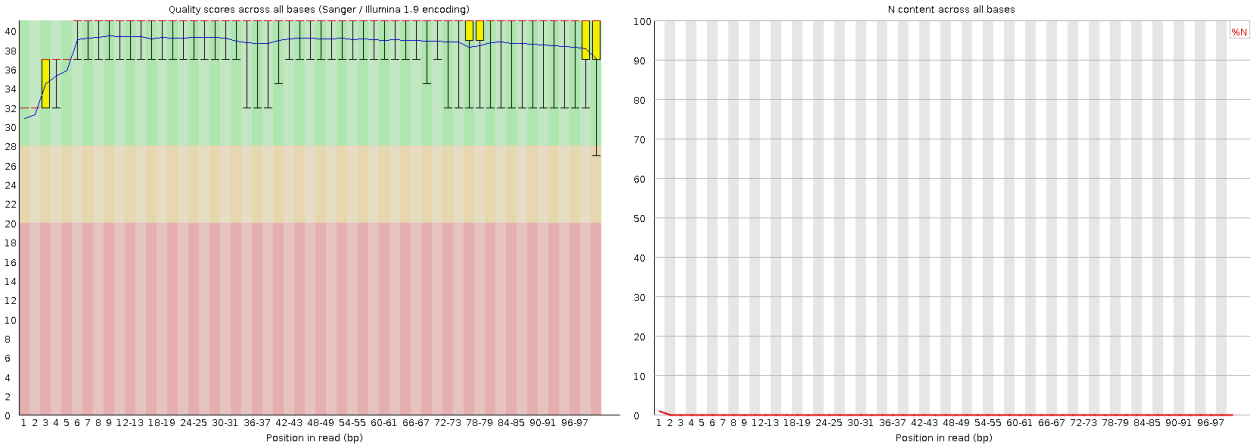

Figure 2. Quality score distribution (left) and per-base n content (right) for 16_3D_mbnl_S12_L008 R2.

21_3G_both_S15_L008 Read 1

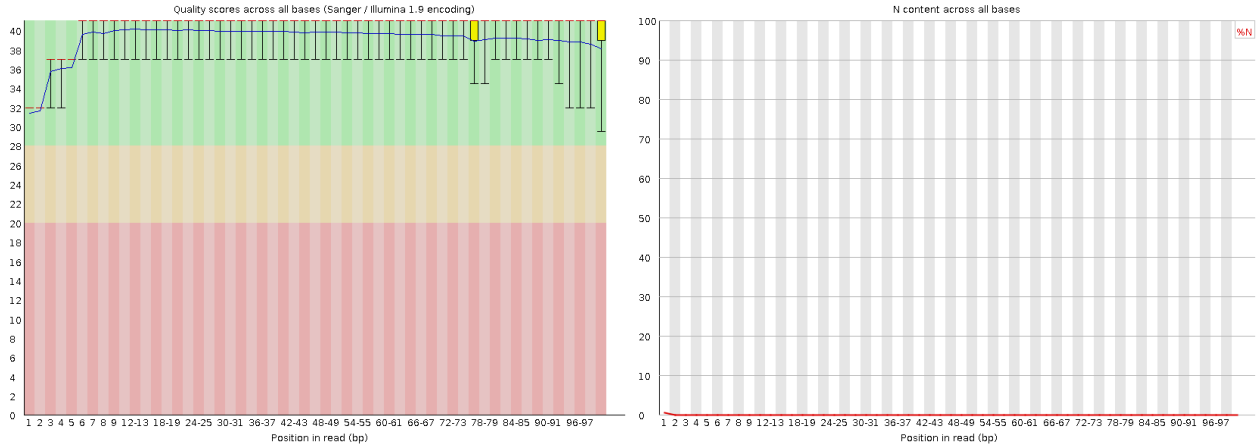

Figure 3. Quality score distribution (left) and per-base n content (right) for 21_3G_both_S15_L008 R1.

21_3G_both_S15_L008 Read 2

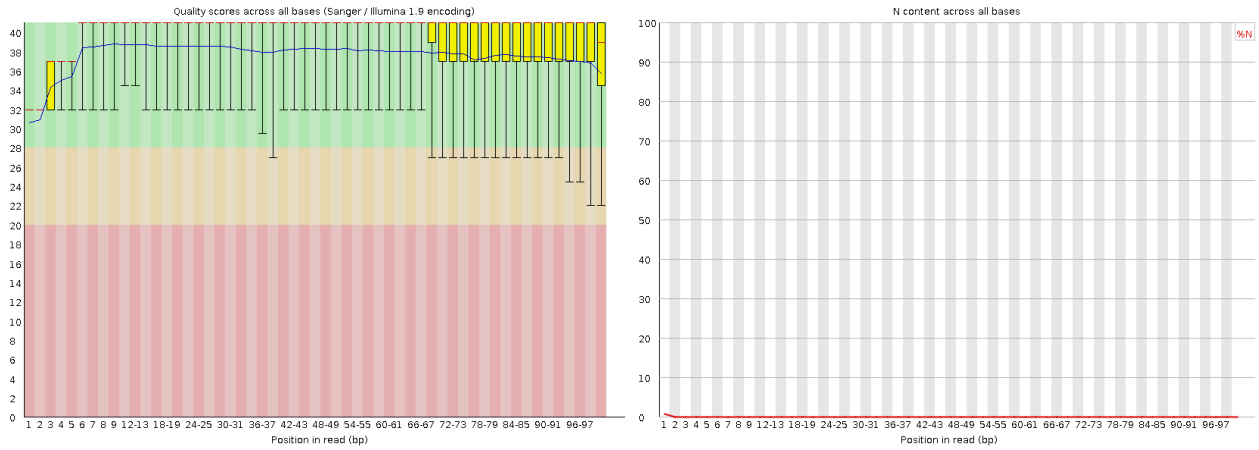

Figure 4. Quality score distribution (left) and per-base n content (right) for 21_3G_both_S15_L008 R2.

2. Use plot script from Bi622 Demultiplex Assignment to create quality score distribution.
   sbatch script:

```
./qaa_p1_q2_demul.py \
  -f /projects/bgmp/shared/2017_sequencing/demultiplexed/
  16_3D_mbnl_S12_L008_R1_001.fastq.gz \
  -r R1 -o 16_3D_mbnl_S12_L008_R1_001

./qaa_p1_q2_demul.py \
  -f /projects/bgmp/shared/2017_sequencing/demultiplexed/
  16_3D_mbnl_S12_L008_R2_001.fastq.gz \
  -r R2 -o 16_3D_mbnl_S12_L008_R2_001

./qaa_p1_q2_demul.py \
```

```
  -f /projects/bgmp/shared/2017_sequencing/demultiplexed/
  21_3G_both_S15_L008_R1_001.fastq.gz \
  -r R1 -o 21_3G_both_S15_L008_R1_001

./qaa_p1_q2_demul.py \
  -f /projects/bgmp/shared/2017_sequencing/demultiplexed/
  21_3G_both_S15_L008_R2_001.fastq.gz \
  -r R2 -o 21_3G_both_S15_L008_R2_001
```

The FastQC quality score distribution plots and my plot script (Fig. 5 and Fig. 6) resulted in the same distribution, however, the graph from FastQC also shows the interquartile range with the color-coded background to show the quality of the distributions.
The run time was also approximately twice as fast for FastQC, which ran for less than 2 minutes, compared to my plot script, which ran for less than 5 minutes. This can be due to the difference in the programming language. FastQC is written in Java, which is relatively faster than my python script since Java is a compiled language.

16_3D_mbnl_S12_L008

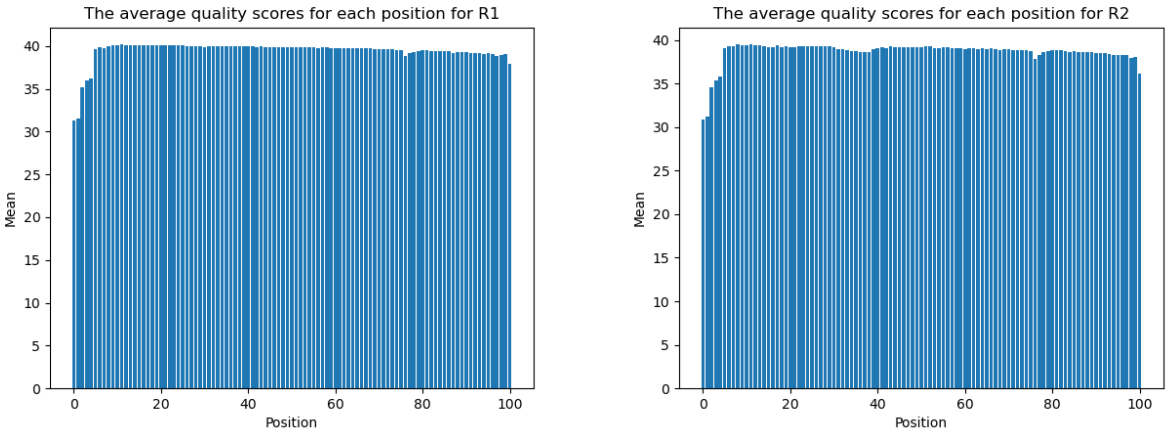

Figure 5. Quality score distribution for 16_3D_mbnl_S12_L008 read 1 (left) and read 2 (right).

21_3G_both_S15_L008

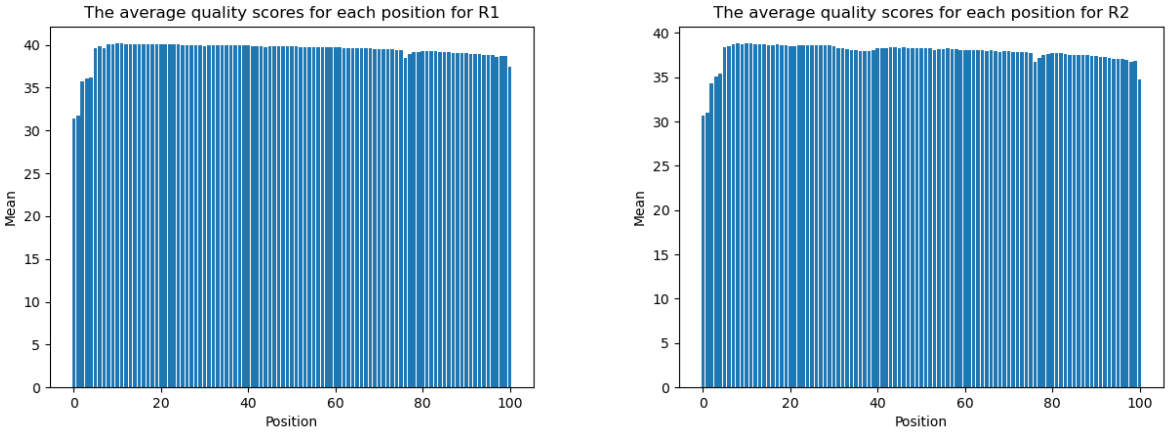

Figure 6. Quality score distribution for 21_3G_both_S15_L008 read 1 (left) and read 2 (right).

3. Overall, the quality of both libraries are good. The quality is expected to be lower for the first couple of positions due to the physical limitations of the sequencing instrument, however, the rest of the bases are relatively high quality. Although the mean quality is lower for read 2 compared to read 1 for both libraries, the quality score is still relatively good. The mean quality score also decreases at position 78 compared to other positions for all distributions and decreases towards the end of the position. However, the quality score is still relatively good, with an average of high thirties.

Part 2: Adaptor trimming comparison

4. Install cutadapt and trimmomatic to a new conda environment.
   script:

```
conda create --name QAA
conda activate QAA
conda install cutadapt
conda install trimmomatic
```

Package Version:
cutadapt: 3.4
trimmomatic: 0.39

5. Use cutadapt to trim adapter sequences.
   script:

```
cutadapt \
  -a AGATCGGAAGAGCACACGTCTGAACTCCAGTCA \
  -A AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT \
  -o /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/
  16_3D_mbnl_S12_L008_R1_001_out1.fastq \
  -p /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/
  16_3D_mbnl_S12_L008_R2_001_out2.fastq \
```

```
  /projects/bgmp/shared/2017_sequencing/demultiplexed/
  16_3D_mbnl_S12_L008_R1_001.fastq.gz \
  /projects/bgmp/shared/2017_sequencing/demultiplexed/
  16_3D_mbnl_S12_L008_R2_001.fastq.gz

cutadapt \
  -a AGATCGGAAGAGCACACGTCTGAACTCCAGTCA \
  -A AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT \
  -o /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/
  21_3G_both_S15_L008_R1_001_out1.fastq \
  -p /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/
  21_3G_both_S15_L008_R2_001_out2.fastq \
  /projects/bgmp/shared/2017_sequencing/demultiplexed/
  21_3G_both_S15_L008_R1_001.fastq.gz \
  /projects/bgmp/shared/2017_sequencing/demultiplexed/
  21_3G_both_S15_L008_R2_001.fastq.gz
```

The FastQC reports overrrepresented sequences, and since adapters would be present in all sequences, the overrepresented sequence would most likely be the adapter. For read 1 of 16_3D_mbnl_S12_L008, the overrepresented sequence was GATCGGAAGAGCACACGTCTGAACTCCAGTCACACGATCA-GATCTCGTAT, and for read 2, the sequence was GATCGGAAGAGCGTCGTGTAGGGAAAGAGT-GTCTGATCGTGTGTAGATCT. When comparing to the provided adapter sequence, part of the adapter sequence is in the overrepresented sequence.
This also occured for the other library, 21_3G_both_S15_L008. The overrepresented sequence for read 1 and read 2 was GATCGGAAGAGCACACGTCTGAACTCCAGTCACGTCCTAAGATCTCGTAT and GATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTCTTAGGACGTGTAGATCT, respectively.
Both libraries used Illumina Universal Adapter, which is shown in the FastQC reports, and the adapter sequences used for read 1 and read 2 was AGATCGGAAGAGCACACGTCTGAACTCCAGTCA and AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT, respectively.
To confirm that the provided adapter sequences were oriented correctly, I used grep to determine how many sequences had the adapter sequence.

There were 8235197 amount of sequence lines for 16_3D_mbnl_S12_L008 R1.

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
16_3D_mbnl_S12_L008_R1_001.fastq.gz | sed -n '2~4p' | wc -l
```

When grepping for R1 adapter sequence, there are 115556 amount of sequence lines with the adapter, which is 1.40% of the total sequence lines.

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
16_3D_mbnl_S12_L008_R1_001.fastq.gz | sed -n '2~4p' | grep
"AGATCGGAAGAGCACACGTCTGAACTCCAGTCA" | wc -l
```

To ensure that the adapter is positioned 5' to 3' direction, I reversed the sequence to: ACTGACCT-CAAGTCTGCACACGAGAAGGCTAGA. There were no matches when grepping for this sequence.

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
16_3D_mbnl_S12_L008_R1_001.fastq.gz | sed -n '2~4p' | grep
"ACTGACCTCAAGTCTGCACACGAGAAGGCTAGA" | wc -l
```

I then checked to determine if the provided R1 adapter sequence was complimented, and I grepped for the complimented sequence: TCTAGCCTTCTCGTGTGCAGACTTGAGGTCAGT. Once again, there were no matches.

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
16_3D_mbnl_S12_L008_R1_001.fastq.gz | sed -n '2~4p' | grep
"TCTAGCCTTCTCGTGTGCAGACTTGAGGTCAGT" | wc -l
```

I finally checked to determine if the provided R1 adapter sequence was reverse complimented, and I grepped for the sequence: TGACTGGAGTTCAGACGTGTGCTCTTCCGATCT. Once again, there were no matches.

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
16_3D_mbnl_S12_L008_R1_001.fastq.gz | sed -n '2~4p' | grep
"TGACTGGAGTTCAGACGTGTGCTCTTCCGATCT" | wc -l
```

Therefore, the provided R1 adapter sequence is correctly oriented for 16_3D_mbnl_S12_L008 R1.
Similar results was shown for 21_3G_both_S15_L008_R1_001 file.

To confirm that the provided R2 adapter sequences were oriented in correctly, I used grep to determine how many sequences had the R2 adapter sequence.
There are 8235197 total sequence lines for 16_3D_mbnl_S12_L008 R2.

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
16_3D_mbnl_S12_L008_R2_001.fastq.gz | sed -n '2~4p' | wc -l
```

When grepping for R2 adapter sequence, there are 115921 amount of sequence lines with the adapter, which is 1.41% of the total sequence lines.

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
16_3D_mbnl_S12_L008_R2_001.fastq.gz | se
d -n '2~4p' | grep "AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT" | wc -l
```

To ensure that the adapter is positioned 5' to 3' direction, I reversed the sequence to: TGTGAGAAAGGGATGTGCTGCGAGAAGGCTAGA. There were no matches when grepping for this sequence.

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
16_3D_mbnl_S12_L008_R2_001.fastq.gz | sed -n '2~4p' | grep
"TGTGAGAAAGGGATGTGCTGCGAGAAGGCTAGA" | wc -l
```

I then checked to determine if the provided R2 adapter sequence was complimented, and I grepped for the complimented sequence: TCTAGCCTTCTCGCAGCACATCCCTTTCTCACA. Once again, there were no matches.

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
16_3D_mbnl_S12_L008_R2_001.fastq.gz | sed -n '2~4p' | grep
"TCTAGCCTTCTCGCAGCACATCCCTTTCTCACA" | wc -l
```

I finally checked to determine if the provided R2 adapter sequence was reverse complimented, and I grepped for the sequence: ACACTCTTTCCCTACACGACGCTCTTCCGATCT. Once again, there were no matches.

```
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/
16_3D_mbnl_S12_L008_R2_001.fastq.gz | sed -n '2~4p' | grep
"ACACTCTTTCCCTACACGACGCTCTTCCGATCT" | wc -l
```

Therefore, the provided R2 adapter sequence is correctly oriented for 16_3D_mbnl_S12_L008 R2. Similar results was shown for 21_3G_both_S15_L008_R2_001 file.

6. Use trimmomatic to trim qualities.
   script:

```
trimmomatic PE \
  -phred33 \
  -trimlog \
  /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/21_3G_both_S15_L008_outlog \
  /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/
  21_3G_both_S15_L008_R1_001_out1.fastq \
  /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/
  21_3G_both_S15_L008_R2_001_out2.fastq \
  -baseout /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/
  21_3G_both_S15_L008.fastq.gz \
  LEADING:3 \
  TRAILING:3 \
  SLIDINGWINDOW:5:15 \
  MINLEN:35

trimmomatic PE \
  -phred33 \
  -trimlog \
  /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/16_3D_mbnl_S12_L008_outlog \
  /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/
  16_3D_mbnl_S12_L008_R1_001_out1.fastq \
  /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/
  16_3D_mbnl_S12_L008_R2_001_out2.fastq \
  -baseout \
  /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/16_3D_mbnl_S12_L008.fastq.gz \
  LEADING:3 \
  TRAILING:3
  SLIDINGWINDOW:5:15 \
  MINLEN:35
```

7. Plot trimmed read length distributions.
   script:

```
#!/usr/bin/env python
import matplotlib.pyplot as plt
import math
import argparse

def get_args():
    parser = argparse.ArgumentParser(description="Files")
    parser.add_argument("-f", "--file", help="Specify the absolute
                        pathway to the outlog from trimmomatic.", required=True)
    parser.add_argument("-o", "--output", help="Specify the name of
                        the output graph name.", required=True)
    return parser.parse_args()
```

```
args = get_args()

counter = 0
seqlen = 101
dict_r1_trimlen_freq = {key: 0 for key in range(102)}
dict_r2_trimlen_freq = {key: 0 for key in range(102)}
#print(dict_r1_trimlen_freq)

with open(args.file, "r") as f:
    for line in f:
        if counter % 2 == 0:
            line = line.strip().split()
            length = int(line[2])
            trimlen = seqlen - length
            dict_r1_trimlen_freq[trimlen] += 1
            #print(line)
        elif counter % 2 == 1:
            line = line.strip().split()
            length = int(line[2])
            trimlen = seqlen - length
            dict_r2_trimlen_freq[trimlen] += 1
        counter += 1

x = range(102)
plt.figure()
plt.bar(x, dict_r1_trimlen_freq.values(), alpha=0.5, label="R1")
plt.bar(x, dict_r2_trimlen_freq.values(), alpha=0.5, label="R2")
plt.legend()
plt.yscale("log")
plt.title(f"Trimmed Read Length Distribution for R1 & R2")
plt.xlabel("Trimmed Length (number of nucleotides)")
plt.ylabel("Frequency (Log10)")
plt.savefig(f"{args.output}.png")
```

The reads in both libraries were trimmed for adapters and poor qualities, and the distribution of the trimmed length is shown in Figure 7. Read 2 is expected to be trimmed more frequently and at longer lengths. This is due to the sequencing process. When the instrument is sequencing the DNA, the sample sits in the flow cell for long periods of time, which slowly degrades the sample and decreases the quality of the reads. Therefore, read 2 have lower quality reads, which increases the frequency and the amount of trimming.

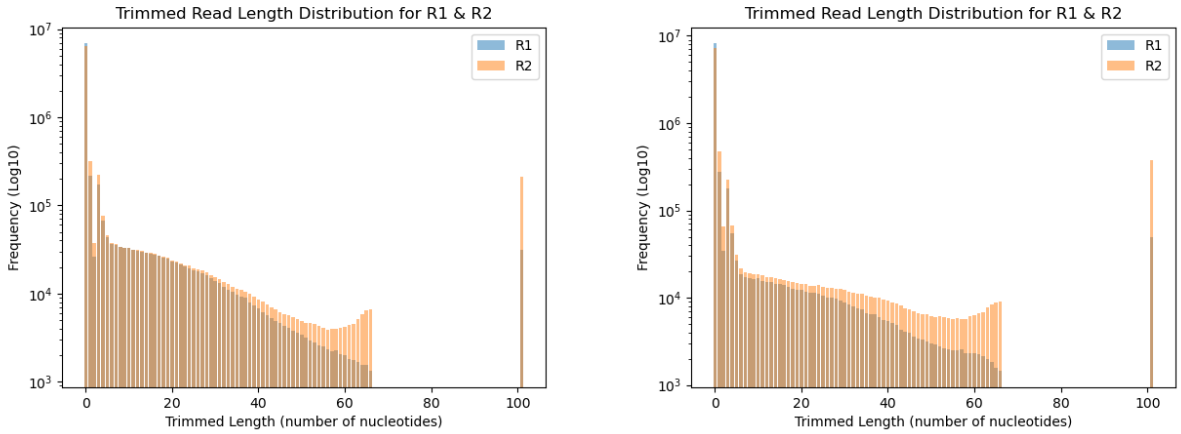

Figure 7. Trimmed read length distribution for 16_3D_mbnl_S12_L008 (left) and 21_3G_both_S15_L008 (right).

Part 3: Alignment and strand-specificity

8. Install software
   script:

```
conda install star
conda install numpy
conda install pysam
conda install matplotlib
pip install HTSeq
```

Software Version:
star: 2.7.9a
numpy: 1.21.2
pysam: 0.16.0.1
matplotlib: 3.4.3
HTSeq-count: 0.13.5

9. Download mouse genome fasta files. script:

```
wget http://ftp.ensembl.org/pub/release-104/fasta/mus_musculus/dna/
Mus_musculus.GRCm39.dna.primary_assembly.fa.gz

wget http://ftp.ensembl.org/pub/release-104/gtf/mus_musculus/
Mus_musculus.GRCm39.104.gtf.gz
```

Build database from mouse genome fasta file:/ script:

```
STAR --runThreadN 8 \
    --runMode genomeGenerate \
    --genomeDir /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/
    Mus_musculus.GRCm39.dna.ens104.STAR_2.7.9a \
```

```
    --genomeFastaFiles /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/mus/
    Mus_musculus.GRCm39.dna.primary_assembly.fa \
    --sjdbGTFfile /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/mus/
    Mus_musculus.GRCm39.104.gtf
```

Align reads to the geome database:
script:

```
#16_3D_mbnl_S12_L008
STAR --runThreadN 8 \
    --runMode alignReads \
    --outFilterMultimapNmax 3 \
    --outSAMunmapped Within KeepPairs \
    --alignIntronMax 1000000 \
    --alignMatesGapMax 1000000 \
    --readFilesCommand zcat \
    --readFilesIn /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/
    16_3D_mbnl_S12_L008_1P.fastq.gz
    /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/16_3D_mbnl_S12_L008_2P.fastq.gz \
    --genomeDir /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/
    Mus_musculus.GRCm39.dna.ens104.STAR_2.7.9a \
    --outFileNamePrefix Mus_musculus.ens104_16_3D_mbnl_S12_L008

#21_3G_both_S15_L008
/usr/bin/time -v STAR --runThreadN 8 \
    --runMode alignReads \
    --outFilterMultimapNmax 3 \
    --outSAMunmapped Within KeepPairs \
    --alignIntronMax 1000000 \
    --alignMatesGapMax 1000000 \
    --readFilesCommand zcat \
    --readFilesIn /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/
    21_3G_both_S15_L008_1P.fastq.gz /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/
    21_3G_both_S15_L008_2P.fastq.gz \
    --genomeDir /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/
    Mus_musculus.GRCm39.dna.ens104.STAR_2.7.9a \
    --outFileNamePrefix Mus_musculus.ens104_21_3G_both_S15_L008
```

10. Report number of mapped and unmapped reads.
    Table 1 and 2 shows the number of mapped and unmapped reads for 16_3D_mbnl_S12_L008 and 21_3G_both_S15_L008.

Table 1: Number of mapped and unmapped reads to genome from 16_3D_mbnl_S12_L008

	count
mapped	15662612
unmapped	365754

Table 2: Number of mapped and unmapped reads to genome from 21_3G_both_S15_L008

	count
mapped	17061136
unmapped	645494

11. Use HTSeq-count to count reads.
    script:

```
#16_3D_mbnl_S12_L008, stranded=yes
htseq-count \
  --stranded=yes \
  -o /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/htseq/
  Mus_musculus.ens104_16_stranded \
  Mus_musculus.ens104_16_3D_mbnl_S12_L008Aligned.out.sam \
  /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/mus/Mus_musculus.GRCm39.104.gtf
  > /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/
  Mus_musculus.ens104_16_stranded.output

#16_3D_mbnl_S12_L008, stranded=no
htseq-count \
  --stranded=no \
  -o /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/htseq/
  Mus_musculus.ens104_16_unstranded \
  Mus_musculus.ens104_16_3D_mbnl_S12_L008Aligned.out.sam/
  projects/bgmp/jlee26/bioinformatics/Bi623/qaa/mus/Mus_musculus.GRCm39.104.gtf
  > /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/
  Mus_musculus.ens104_16_unstranded.output

#21_3G_both_S15_L008, stranded=yes
htseq-count \
  --stranded=yes \
  -o /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/htseq/
  Mus_musculus.ens104_21_stranded \
  Mus_musculus.ens104_21_3G_both_S15_L008Aligned.out.sam/
  projects/bgmp/jlee26/bioinformatics/Bi623/qaa/mus/Mus_musculus.GRCm39.104.gtf
  > /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/
  Mus_musculus.ens104_21_stranded.output

#21_3G_both_S15_L008, stranded=no
htseq-count \
  --stranded=no \
  -o /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/htseq/
  Mus_musculus.ens104_21_unstranded \
  Mus_musculus.ens104_21_3G_both_S15_L008Aligned.out.sam/
  projects/bgmp/jlee26/bioinformatics/Bi623/qaa/mus/Mus_musculus.GRCm39.104.gtf
  > /projects/bgmp/jlee26/bioinformatics/Bi623/qaa/
  Mus_musculus.ens104_21_unstranded.output
```

12. Determine strandedness of RNA-seq libraries.
    script:

```
cat Mus_musculus.ens104_16_stranded.output | grep -v '^__' |
awk '{sum+=$2} END {print sum}'
# returned 320966
cat Mus_musculus.ens104_16_unstranded.output | grep -v '^__' |
awk '{sum+=$2} END {print sum}'
# returned 6724245

awk '{sum+=$2} END {print sum}' Mus_musculus.ens104_16_stranded.output
# returned 8014183
awk '{sum+=$2} END {print sum}' Mus_musculus.ens104_16_unstranded.output
# returned 8014183



cat Mus_musculus.ens104_21_stranded
.output | grep -v '^__' | awk '{sum+=$2} END {print sum}'
# returned 327477
 cat Mus_musculus.ens104_21_unstrand
ed.output | grep -v '^__' | awk '{sum+=$2} END {print sum}'
# returned 6979158
awk '{sum+=$2} END {print sum}' Mus_musculus.ens104_21_stranded.output
# returned 8853315
 awk '{sum+=$2} END {print sum}' Mus_musculus.ens104_21_unstranded.output
# returned 8853315
```

Table 3 shows the percent reads that were mapped to a feature. When stranded option is turned off, there is high percentage of reads mapped. This is because reads can be matched to the feature despite the orientation of the reads. However, when stranded is turned on, there is drastic decrease in reads mapped. This is because the strands need to be in the same orientation as the feature to mapped. Since the percent reads that were mapped decreased when the stranded was turned on, the RNA-seq must be not strand-specific for both libraries, since less than or equal to 4.00% of reads were mapped when stranded option was on. If these libraries were strand specific, we should see high number of reads mapped to the feature, such as values approximately 80%, even with stranded option turned on.

Table 3: Percent of reads mapped to a feature with stranded option using HTSeq-count

	–stranded	Mapped Reads	Total Number of Reads	% Reads Mapped to a Feature
16_3D_mbnl_S12_L008 Y	Yes	320966	8014183	4.00
16_3D_mbnl_S12_L008 N	No	6724245	8014183	83.9
21_3G_both_S15_L008 Y	Yes	327477	8853315	3.70
21_3G_both_S15_L008 N	No	6979158	8853315	78.8