RNA-seq Quality Assessment Assignment

Jules Hays*

**Univeristy of Oregon, Knight Campus Graduate Internship Program - Bioinformatics and Genomics Track*

E-mail: jkhay@uoregon.edu

Abstract

Given two demultiplexed file pairs from an RNA sequencing experiment done in 2017, the goal of this assigment is to perform quality control on the sequencing output and further process the data with downstream analyses. In part 1, FastQC is utilized for quality assessments on the sequencing outputs and the results are compared to my own algorithm. In Part 2, the tools Cutadapt and Trimmomatic are utilized to remove adaptor sequences from the reads and trim low quality reads. In Part 3, the reads are aligned to a reference genome with STAR and counts of each gene within the reads are determined with HTSeq-count. From this data, the strandedness of the original RNA-seq library is determined.

The Data

Mouse embryonic fibroblasts were treated with with either FOX and MBLN (n=3), which are alternative splicing factors, FOX or MBNL (n=3 each), or no treatment (n=3). Each of the 12 samples was processed with two replicates, for a total of 24 samples. The RNA was isolated from the cells and prepped for RNA-sequencing with KAPA's Stranded mRNA-Seq kit and selected for 300-400 base pair long cDNA fragments. The libraries were also indexed with dual-indexed adapters. The samples were then normalized and pooled, and sequenced on one lane of a 2 x 101 Illumina HiSeq 4000 run. The files were demultiplexed into the original 24 samples.

The file pairs that will be analyzed throughout this paper are called 16_3D_mbnl_S12_L008 and 6_2D_mbnl_S5_L008, which will be referred to as file pair 16 and 6 throughout the paper. These were both samples treated with only MBNL. Each pair contains a read 1 and a read 2 file. Table 1 shows a summary of the data in each of the files.

Table 1: Summary of the demultiplexed RNA-sequencing output files that this paper will investigate further

File Pair	Read	File Size (MB)	Number of Records	Read Length	Index Sequence
6	1	571	11,028,244	101	NAGCCATG
6	2	653	11,028,244	101	NAGCCATG
16	1	468	8,235,197	101	NCGATCAG
16	2	452	8,235,197	101	NCGATCAG

Part 1 - Read Quality Score Distributions

FastQC Plots

FastQC [1] was used to generate data and plots regarding the quality of the sequenced reads in both read files from the file pairs. Some of the outputs from this tool are plots of the per-base quality score distributions for each file (Figure 1) and plots of the per-base N content for each file (Figure 2). The per base quality score distributions are consistent with the per-base N content plot in all four files analyzed because the small spike in N content at the beginning of the reads corresponds with a lower average quality score at the beginning the reads. The low quality of unknown nucleotide calls at the beginning of a read could be due to mismatched pairing at the ends of the primers in the well during the extension process [2].

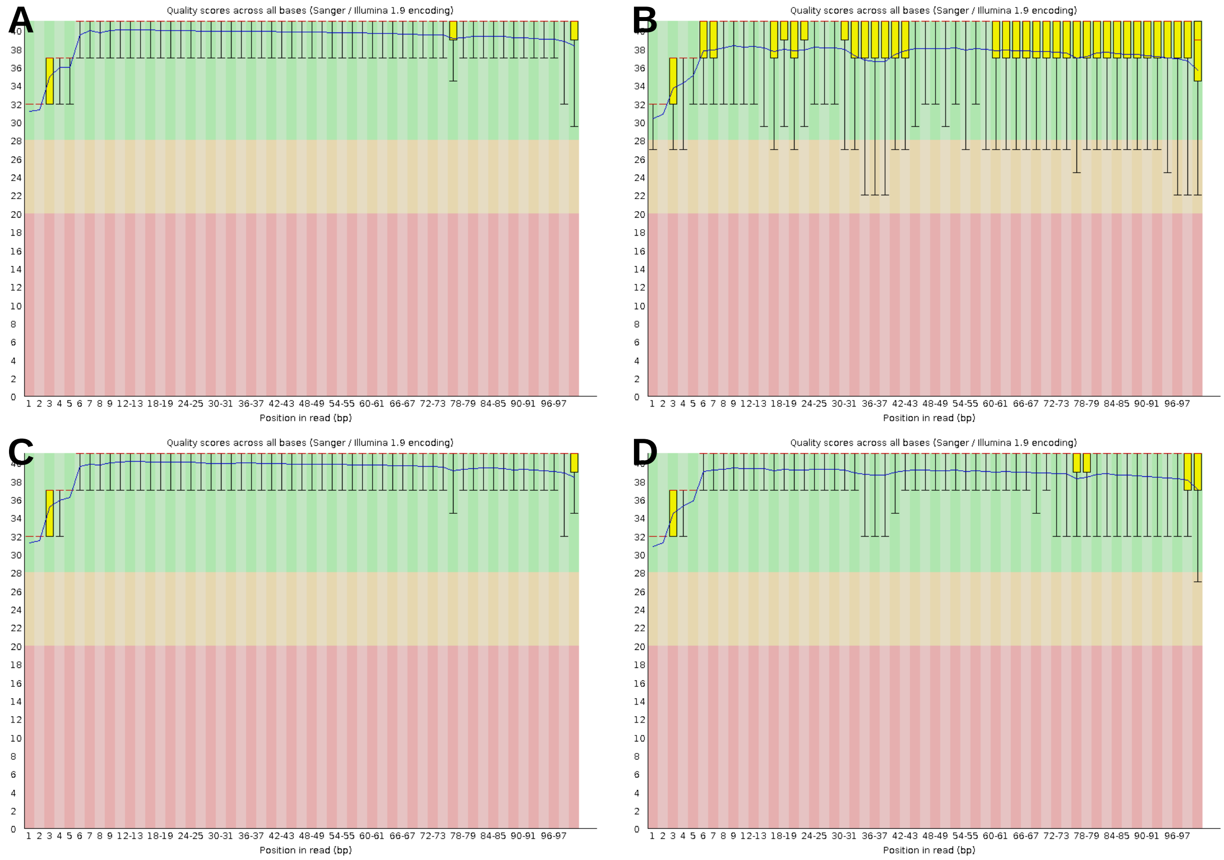


Figure 1: Per-base quality score distributions for (A) pair 6 read 1, (B) pair 6 read 2, (C) pair 16 read 1, and (D) pair 16 read 2. The blue line indicates the mean, the red line indicates the median, the yellow box indicates the interquartile range, and the upper and lower whisker indicate the 10th and 90th percentile quality score for a given base pair position. The plot background divides each plot into regions of very good quality (green), reasonable quality (orange), and poor quality (red).

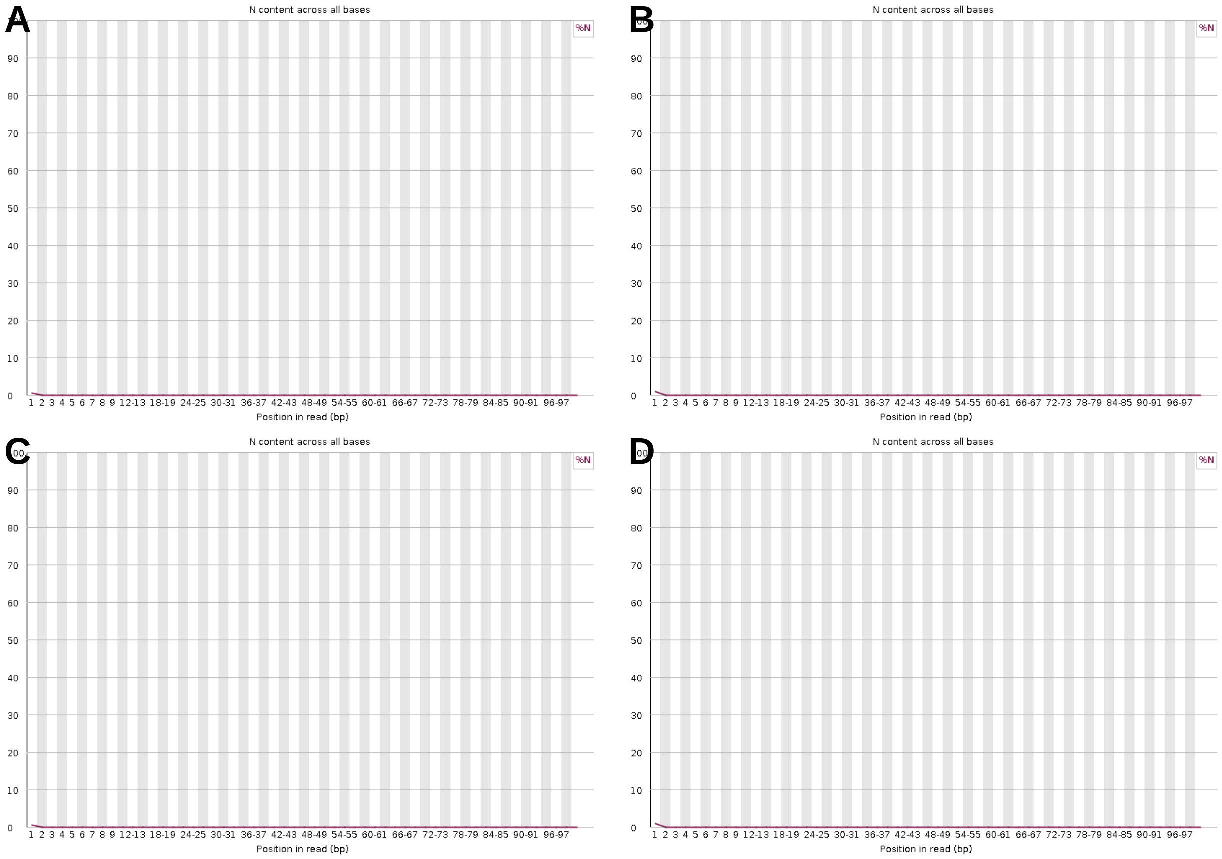


Figure 2: Per-base N content for (A) pair 6 read 1, (B) pair 6 read 2, (C) pair 16 read 1, and (D) pair 16 read 2. The red line indicates the percent of base calls where an N was called for a given position.

Comparison of FastQC Per-Base Quality Plots to my Python Script

The FastQC per-base quality score distributions (Figure 1) were then compared to per-base quality score distributions outputted from my own Python script for each file (Figure 3). One main difference between the two methods is my plots only show the mean quality score per nucleotide position, while the FastQC generated plots give additional information about the distribution including median, interquartile range, 10th and 90th percentiles. Therefore, only the mean can be compared between the two plots. The mean for each nucleotide position looks similar between the two plots, increasing and decreasing in the same locations. Another difference between the two methods is the y-axis of the FastQC plots range from 0 to 30 while mine ranges from 30 to 40. This makes the changes in mean quality score appear less intense in the FastQC plots compared to my script's plot. The FastQC plots also indicate quality level with the background color, which is absent in my plot. Finally, the FastQC plots are binned into two position nucleotide groups on the x-axis, whereas my plots analyze quality per single nucleotide. This binning could potentially cause a loss of data.

In addition to providing a more in-depth analysis of the quality score distribution, FastQC also ran through all four files ten times faster than my Python script (Table 2). The methods used the same percentage of CPU but FastQC uses multithreading so it spread the computation over multiple cores. Overall, FastQC provides a more in-depth analysis of quality distributions while also being faster and more efficient. FastQC might be better in performance than my Python script because FastQC is a Java application [1], and Java code runs faster than Python code because Java does not need go through an interpretation step like Python does. Additionally, FastQC uses multithreading the spread the computation over multiple cores. Finally, FastQC was first released in 2010 so has had many years and likely teams of people to optimize it, whereas my code was developed a couple weeks ago by a single graduate student who is new to bioinformatics. Therefore, it makes sense why FastQC is a better program overall.

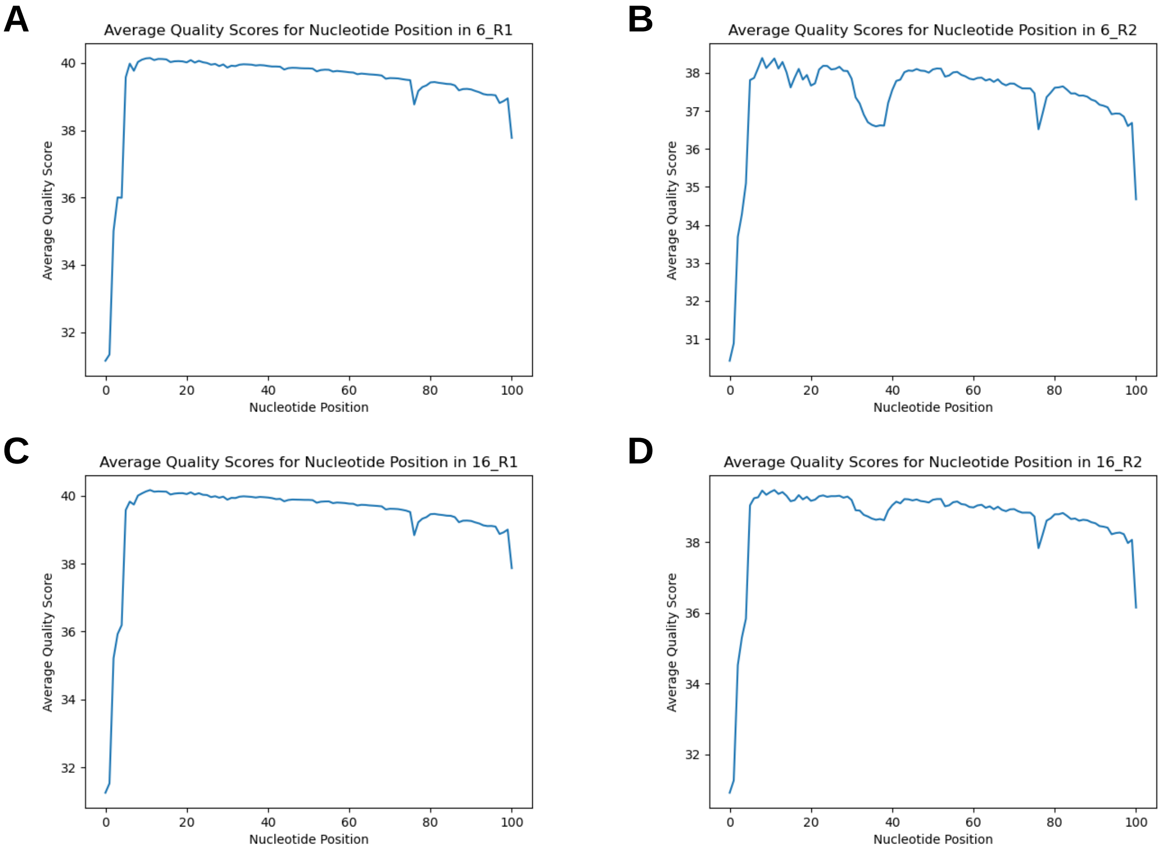


Figure 3: Per-base quality score distributions generated by the author's Python script for (A) pair 6 read 1, (B) pair 6 read 2, (C) pair 16 read 1, and (D) pair 16 read 2. The blue line indicates the mean qualtity score for a given base pair position.

Table 2: Runtime statistics for generating a per-base quality score distribution with FastQC compared to my own script

Method	Run Time (mins:secs)	Number of Cores	CPU Usage (%)
FastQC	2:49	4	99
qual_dist.py	20:08	1	99

Overall Data Quality

Based on the FastQC output for each file, summarized in Table 3, the overall quality of the libraries is high enough to continue with further analyses. Many of the analyses conducted by FastQC passed for all files. First of all, all the reads in all the files are 101 bases long, which is the expected value. The per-base quality scores (Figure 1) for both reads of file pair 16 and read 1 of file pair 6 are in the green range, meaning the quality is very good. For read 2 of pair 6, the mean quality scores are in the green range but the lower quartile and tenth percentile drops into the yellow zone from base pair position 55 and onward, which means the lower quality range of reads in this file are only of reasonable quality. Similarly, the per-sequence quality score plots agree with the per-base quality plots, increasing gradually from a quality score of 30 to 38, then spiking sharply up from 38-40 (Figure 4). This shows that a majority of the reads have an average quality between 38 and 40, which is high The only graph that is different is pair 6 read 2, where the gradual increase starts at a score of 20, and then the graph spikes at a score of 36-40. This means that a majority of the reads for this file are between 36 and 40 and there's some reads below 30, which is consistent with the per base quality for that file. The files will be further processed to improve quality, however, so this quality will improve. In terms of per-base N content, all four files have small increase of less than a couple percentage points in the beginning of the reads, which corresponds with the slightly lower per-base quality in the beginning of the reads, but then the N content tapers off to close to zero percent as the read continues (Figure 2).

Some of the FastQC analyses raised warnings for the files. The per-tile sequence quality plots show the quality scores of sequences in each tile of the Illumina flow cell. All of the files had some low quality tiles near the top of the flow cell (Figure 5). A majority of the tiles, however, are blue, which means high quality. Low quality tiles can be caused by bubbles or smudges on the flow cell, but the reads from these tiles will likely be filtered out when the reads are quality trimmed. There is also some sequence duplication in each of these files but this is to be expected since it is transcriptome data and transcripts can appear multiple times in a sample. The per-base sequence content should be equal for all nucleotides throughout the read, but at the beginning of each read the nucleotide composition is biased, specifically favoring cytosine at base pair position one (Figure 6). This could be due to biases introduced during the random hexamer priming process when generating the cDNA library [3], and if the beginning read is low quality it will likely be trimmed out downstream. Adapter content is low for file pair 6, but spikes from base pair 60 and onward for file pair 16. Since adapters are expected near the 3' end of the read, this spike in adapter content in the later base pair positions is expected. The adapter sequences will be trimmed out during downstream analyses. Finally, there are over-represented sequences in file pair 16 which appear to contain the adapter sequence, and two over-represented sequences from an unknown source in read 2 of file pair 6. Interestingly, the two overrepresented sequences were run through the Blast database, both matching to *Mus musculus* 18S ribosomal RNA (rRNA) genes. This indicates that the the messenger RNA (mRNA) extraction did not fully isolate the mRNA from other types of RNA in this library as this sample was contaminated by rRNA. Despite the warnings on these FastQC analyses, the libraries are high enough quality to continue with further analyses as these further analyses should improve these metrics.

Table 3: Summary of FastQC output for each file

	6		16	
FastQC Output	Read 1	Read 2	Read 1	Read 2
Basic statistics	PASS	PASS	PASS	PASS
Per base sequence quality	PASS	PASS	PASS	PASS
Per tile sequence quality	**FAIL**	**WARN**	**FAIL**	**FAIL**
Per sequence quality scores	PASS	PASS	PASS	PASS
Per base sequence content	**FAIL**	**WARN**	**WARN**	**WARN**
Per sequence GC content	PASS	PASS	PASS	PASS
Per base N content	PASS	PASS	PASS	PASS
Sequence length distribution	PASS	PASS	PASS	PASS
Sequence duplication levels	**FAIL**	**WARN**	**WARN**	**WARN**
Over represented sequences	PASS	**WARN**	**WARN**	**WARN**
Adapter content	PASS	PASS	**WARN**	**WARN**

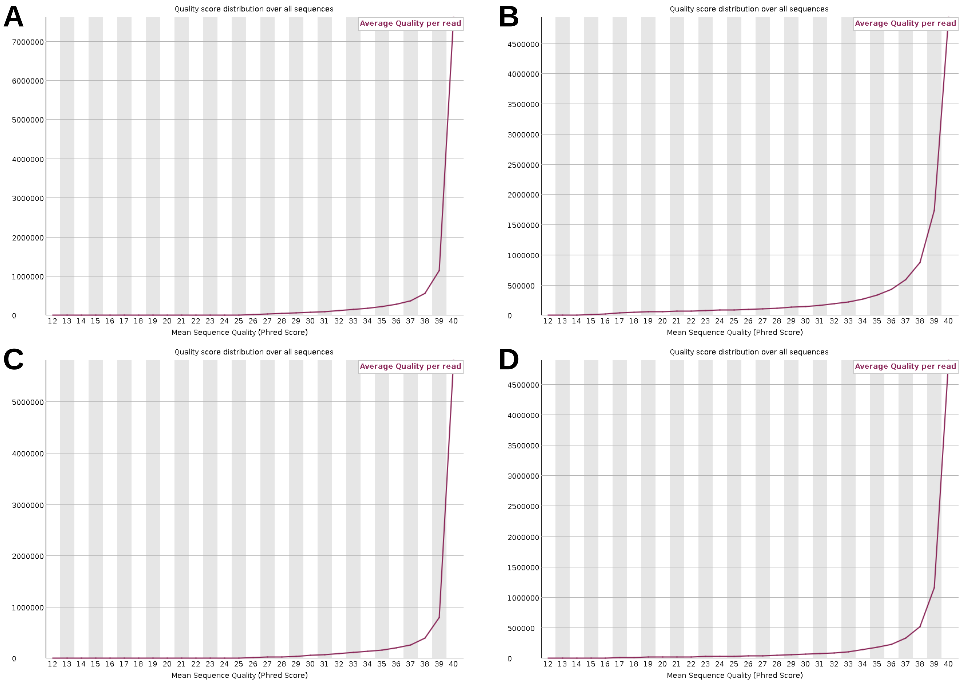


Figure 4: Per-sequence quality distribution for (A) pair 6 read 1, (B) pair 6 read 2, (C) pair 16 read 1, and (D) pair 16 read 2. The red line indicates the frequency of reads with a given mean quality score.

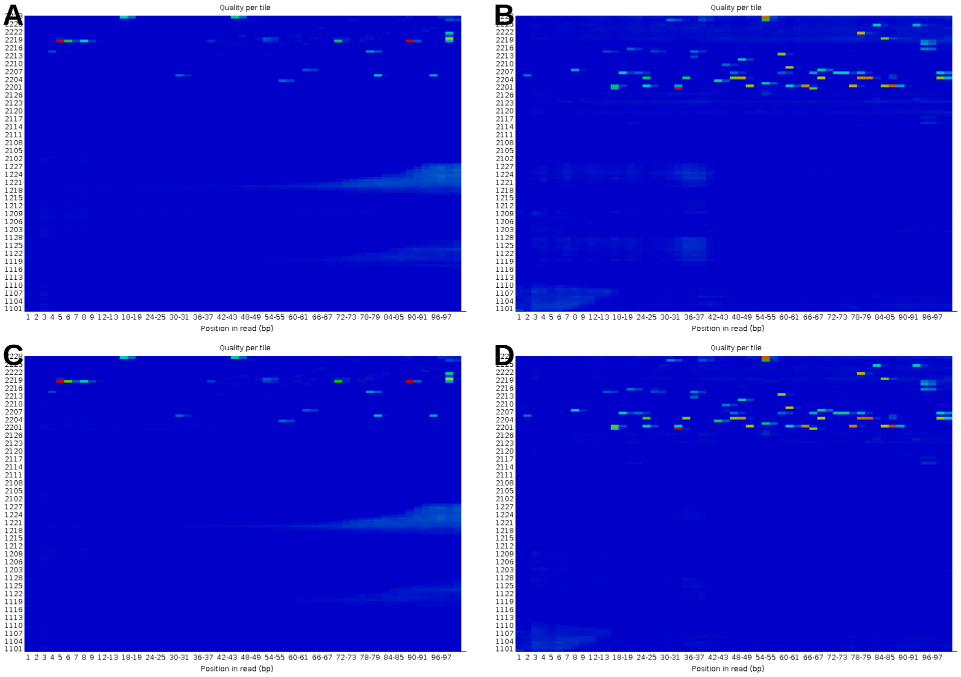


Figure 5: Per-tile sequence quality distribution for (A) pair 6 read 1, (B) pair 6 read 2, (C) pair 16 read 1, and (D) pair 16 read 2. The x-axis depicts the position in the read and the y-axis depicts the flowcell tile that the read came from. Blue indicates high quality and red indicated low quality.

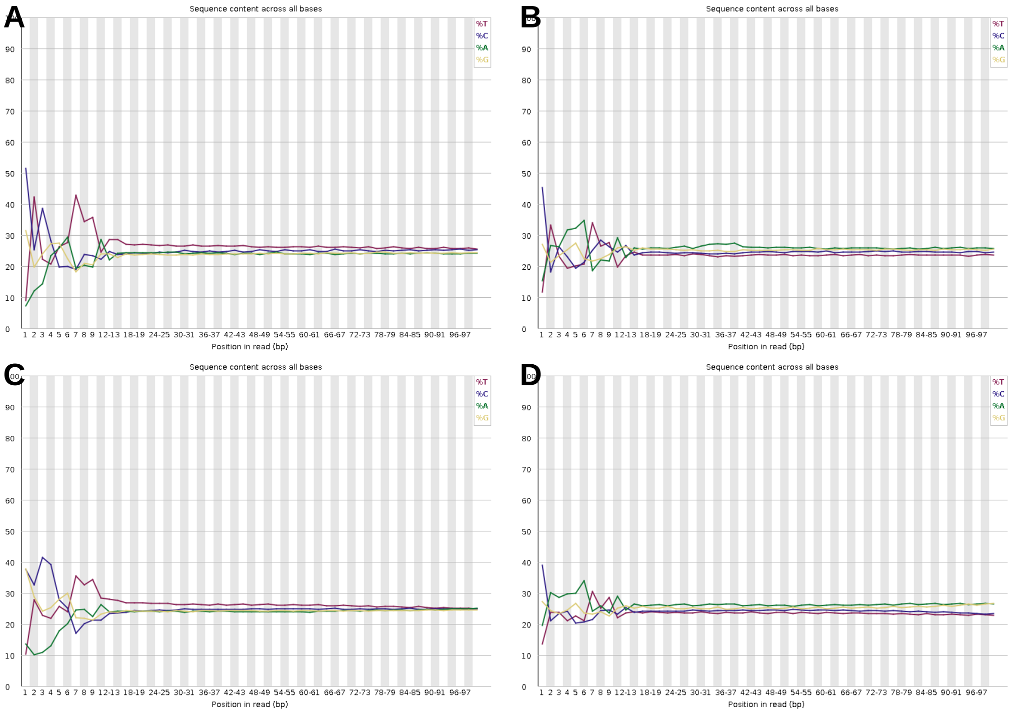


Figure 6: Per-base sequence content distribution for (A) pair 6 read 1, (B) pair 6 read 2, (C) pair 16 read 1, and (D) pair 16 read 2. The colored lines depict the proportion of each base pair at a given nucleotide position. Red is T, blue is C, green is A, yellow is G.

Part 2 - Adaptor Trimming Comparison

Determining Adaptor Sequences

Adapter sequences were determined by first looking at the sample metadata to see the sequencing protocol. Based on the metadata information, the libraries were prepared with KAPA's Stranded mRNA-seq kit. In the information for the kit, it states that the kit is compatible with adapters used in Illumina TruSeq [4]. According to Illumina, the adapter sequences used for TruSeq are:

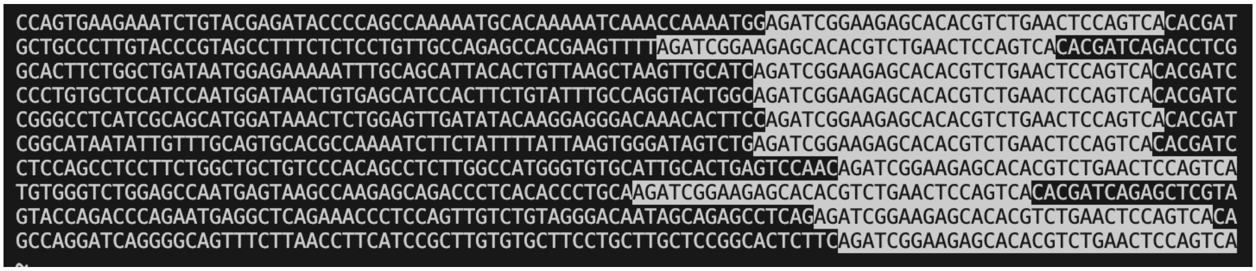
Table 4: Sequences to be used for adapter trimming [5]

	Adaptor Sequences
Read 1:	AGATCGGAAGAGCACACGTCTGAACTCCAGTCA
Read 2:	AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT

To confirm the expected adapter sequences and orientations in the reads, the read 1 adapter sequence was searched for in the read 1 files in each pair, and the read 2 adapter sequence was searched for in the read 2 files. This search was performed using the following command line prompt:

```
zcat <file name> | grep "<ADAPTOR>" | head | less -p "<ADAPTOR>"
```

First, using grep with the adapter sequence returns only reads that contain the adapter sequence. Then, using head returns only the first ten reads, so the entire file of reads is not printed. Finally, using less -p with the adapter sequence highlights the adapter sequence within the read so the position can be visualized. The adapter sequences are expected to be at the 3' end of the read because adapters will only be included in the read if the insert length is less than the read length, causing the polymerase to read beyond the insert into the adapter sequence. For the sample pairs, when adapter sequences did occur in the reads, they appeared towards the 3' end of the read as expected (Figure 7).



```
CCAGTGAAGAAATCTGTACGAGATACCCCAGCCAAAAATGCACAAAAATCAAACCAAAATGGAGATCGGAAGAGCACACGTCTGAACTCCAGTCACACGAT
GCTGCCCTTGTACCCGTAGCCTTTCTCTCCTGTTGCCAGAGCCACGAAGTTTTAGATCGGAAGAGCACACGTCTGAACTCCAGTCACACGATCAGACCTCG
GCACTTCTGGCTGATAATGGAGAAAAATTTGCAGCATTACACTGTTAAGCTAAGTTGCATCAGATCGGAAGAGCACACGTCTGAACTCCAGTCACACGATC
CCCTGTGCTCCATCCAATGGATAACTGTGAGCATCCACTTCTGTATTTGCCAGGTACTGGCAGATCGGAAGAGCACACGTCTGAACTCCAGTCACACGATC
CGGGCCTCATCGCAGCATGGATAAACTCTGGAGTTGATATACAAGGAGGGACAAACACTTCCAGATCGGAAGAGCACACGTCTGAACTCCAGTCACACGAT
CGGCATAATATTGTTTGCAGTGCACGCCAAAATCTTCTATTTTATTAAGTGGGATAGTCTGAGATCGGAAGAGCACACGTCTGAACTCCAGTCACACGATC
CTCCAGCCTCCTTCTGGCTGCTGTCCCACAGCCTCTTGGCCATGGGTGTGCATTGCACTGAGTCCAACAGATCGGAAGAGCACACGTCTGAACTCCAGTCA
TGTGGGTCTGGAGCCAATGAGTAAGCCAAGAGCAGACCCTCACACCCTGCAAGATCGGAAGAGCACACGTCTGAACTCCAGTCACACGATCAGAGCTCGTA
GTACCAGACCCAGAATGAGGCTCAGAAACCCTCCAGTTGTCTGTAGGGACAATAGCAGAGCCTCAGAGATCGGAAGAGCACACGTCTGAACTCCAGTCACA
GCCAGGATCAGGGGCAGTTTCTTAACCTTCATCCGCTTGTGTGCTTCCTGCTTGCTCCGGCACTCTTCAGATCGGAAGAGCACACGTCTGAACTCCAGTCA
```

Figure 7: Highlighted Illumina TruSeq adaptor sequences in ten line of file 16 read 1

Proportion of Reads Trimmed With Cutadapt

Cutadapt was utilized to trim the adapter sequences out of the reads on both file pairs for both the read 1 and read 2 files. File pair 16 was adapter trimmed more extensively than file pair 6 (Table 5), which is reasonable because FastQC determined that file pair 16 had a higher adapter content than file pair 6.

Table 5: Summary of the proportion of reads adaptor trimmed by Cutadapt

File Pair	Read	Number of Reads Trimmed	Percentage of Reads Trimmed
6	1	416,045	3.8%
6	2	502,045	4.6%
16	1	1,002,983	12.2%
16	2	1,069,893	13.0%

Trimmed Read Length Distributions

After filtering adapter sequences, Trimmomatic was utilized to quality trim the reads. The read 2 file in both file pairs was adapter- and quality-trimmed more extensively than the read 1 file, as there is a higher proportion of reads from the read 2 files that were adapter trimmed (Table 4) and a higher frequency of shorter reads in the read 2 files (Figure 8). This is likely because read 2 is lower quality than read 1, as shown in the FastQC analysis (Figure 1), which is caused by a decreased cluster size after the first read is collected and the bridge amplification reoccurs [6]. Additionally, the reagents become old as time progresses in the sequencing reaction, which can decrease quality. This would affect adapter content because as quality degrades, the signal for the inserts within in a well becomes lower, and therefore the reads with shorter inserts that continue into the adapters can dominate the signal and the adapter sequence will be read as the output. Therefore, it is to be expected that read 2 would have lower quality and higher adapter content, and therefore experience heavier adapter and quality trimming.

Re-Running FastQC on the Trimmed Data

Based on the FastQC output for each trimmed file, summarized in Table 6, the overall quality of the libraries improved with adapter and quality trimming. The trimmed files have an improved per-base quality score compared to the original files (Figure 9). For all files, the lower tenth percentile and lower quartile shifted up in quality, which in turn shifts the mean quality up as well. This is especially apparent in file pair 6, where some of the reads extended into the yellow, reasonable quality zone from base position 55 and onward, but in

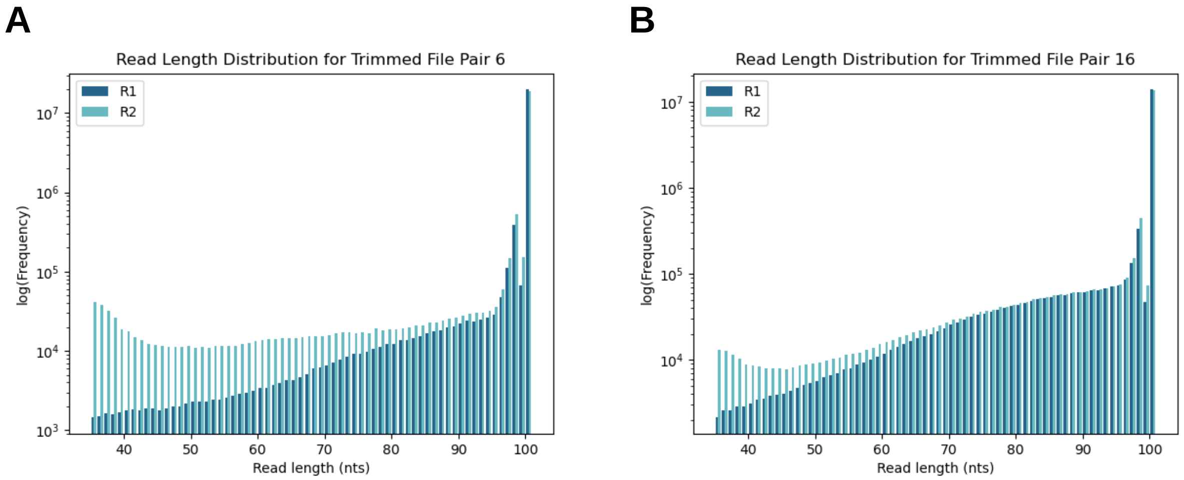


Figure 8: Distribution of read lengths for (A) file pairs 6 and (B) 16 after adaptor trimming with Cutadapt and quality trimming with Trimmomatic. Dark blue is the read 1 file and light blue is the read 2 file. The y-axis is the log base 10 of the frequency of each length.

the trimmed data the spread no longer extends into the reasonable quality zone and is in the good quality zone (green). This improvement is expected because low quality sequences were trimmed out. The trimmed files also have an improved per-sequence quality score, specifically on the lower bound of mean sequence quality (Figure 10). The lower bound of average sequence quality shifted up from 12 to 19 for the read 1 files and from 12 to 16 for the read 2 files. This also shows that the trimming removed lower quality sequences from the files, shifting the mean quality up for the reads as a whole. Corresponding with the increase in per-base and per-sequence quality, the spike in the per-base N content early in the read decreased, showing that the quality filtering cut out any unknown base calls at the beginning of the reads (Figure 11). Finally, adapter content reduced to zero and adapter sequences are no longer over-represented in the reads because the adapters were filtered.

A few of the FastQC analyses still raised a warning with the trimmed files. The per-tile sequence quality plots look nearly identical to the original files, except for a few regions of lighter blue became more darkly shaded, meaning these positions saw a slight increase in quality (Figure 12). The red regions (poor quality), however, still remained red. This may be because the color scale of the per-tile sequence quality plots is relative to the average quality for that base position in a run, so even if a lower quality position improved in quality, it might remain red because the overall mean quality shifted up as well. The per-base sequence content remained nearly identical to the original files, which means the adapter filtering and quality trimming that was performed does not remove nucleotide composition biases at the beginning of a read. The sequence length distribution is no longer consistent at 101 base pairs because, as seen in the read length distribution plots (Figure 8), many of the reads were adapter and quality trimmed so are shorter now. Finally, there are still high duplication levels in the trimmed files, but this is to be expected since a transcript can appear multiple times in a sample. Interestingly, read 2 from pair 6 still has the same over-represented sequences for *Mus musculus* 18S ribosomal RNA (rRNA) genes. This indicates that the trimming process did not remove this contaminant.

Table 6: Summary of FastQC output for each adaptor filtered and quality trimmed files

	6		16	
FastQC Output	Read 1	Read 2	Read 1	Read 2
Basic statistics	PASS	PASS	PASS	PASS
Per base sequence quality	PASS	PASS	PASS	PASS
Per tile sequence quality	**FAIL**	**FAIL**	**FAIL**	**FAIL**
Per sequence quality scores	PASS	PASS	PASS	PASS
Per base sequence content	**FAIL**	**WARN**	**WARN**	**WARN**
Per sequence GC content	PASS	PASS	PASS	PASS
Per base N content	PASS	PASS	PASS	PASS
Sequence length distribution	**WARN**	**WARN**	**WARN**	**WARN**
Sequence duplication levels	**FAIL**	**WARN**	**WARN**	**WARN**
Over represented sequences	PASS	**WARN**	PASS	PASS
Adapter content	PASS	PASS	PASS	PASS

Part 3 - Alignment and Strand-Specificity

Counting Mapped and Unmapped Reads

The trimmed reads were aligned to a mouse genome using STAR, the the number of reads that were mapped and unmapped to the genome are reported (Table 7). A majority

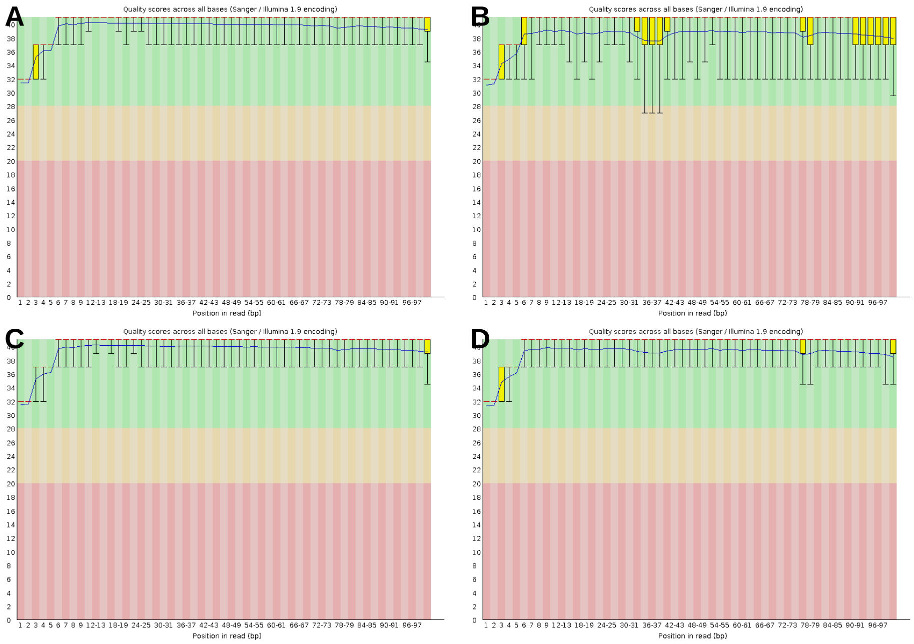


Figure 9: Per-base quality score distributions for trimmed (A) pair 6 read 1, (B) pair 6 read 2, (C) pair 16 read 1, and (D) pair 16 read 2. The blue line indicates the mean qualtity score, the red line indicates the median score, the yellow box indicates the interquartile range of scores, and the upper and lower whisker indicate the 10th and 90th percentile score for a given base pair position. The plot background divides each plot into regions of very good quality (green), reasonable quality (orange), and poor quality (red).

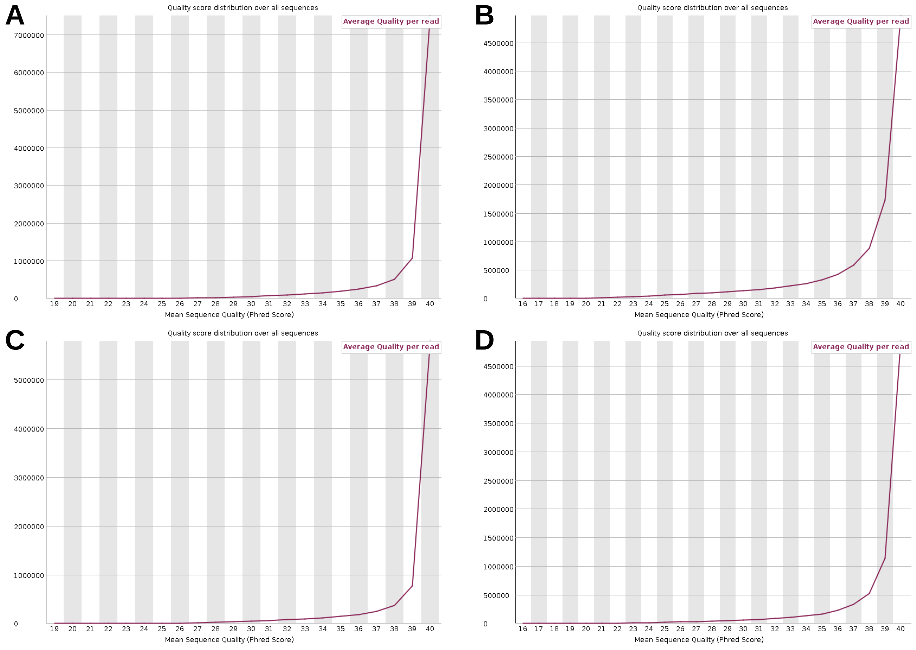


Figure 10: Per-sequence quality distribution for trimmed (A) pair 6 read 1, (B) pair 6 read 2, (C) pair 16 read 1, and (D) pair 16 read 2. The red line indicates the frequency of reads with a given mean quality score.

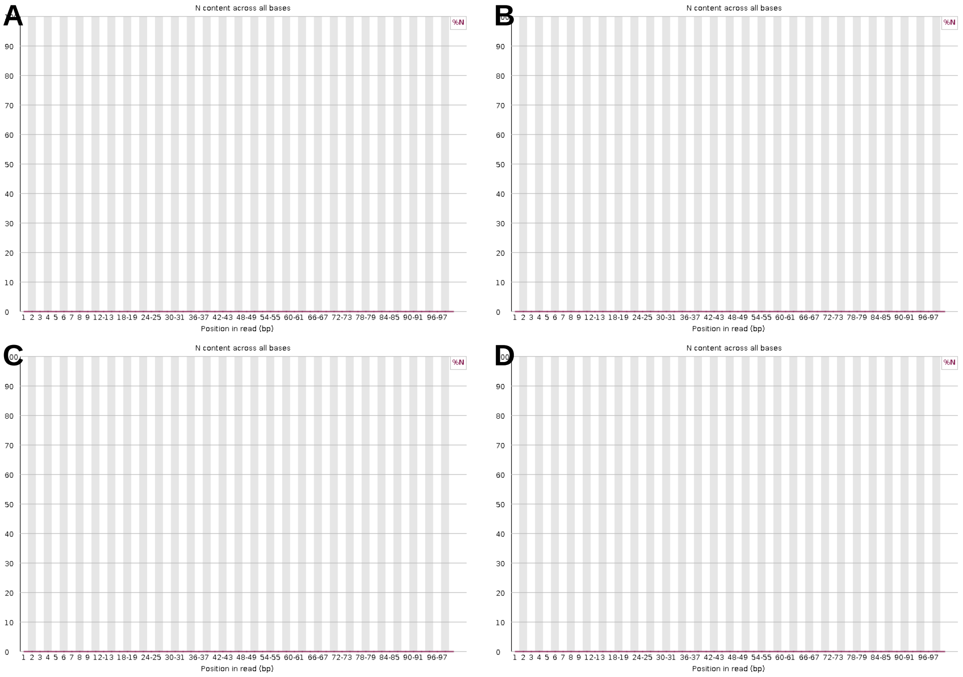


Figure 11: Per-base N content for trimmed (A) pair 6 read 1, (B) pair 6 read 2, (C) pair 16 read 1, and (D) pair 16 read 2. The red line indicates the percent of base calls where an N was called for a given base pair position.

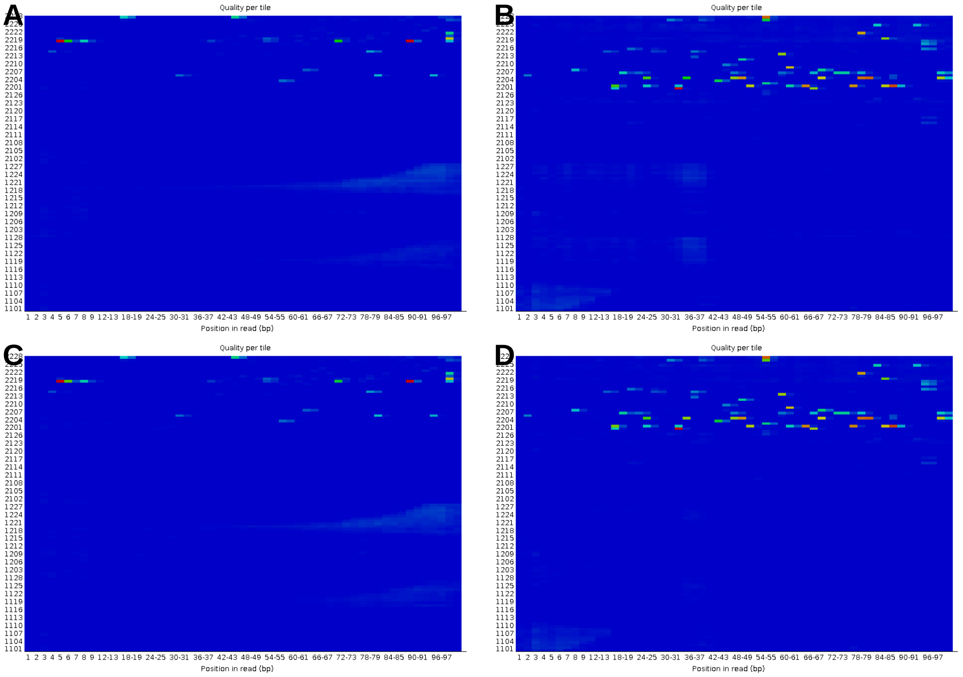


Figure 12: Per-tile sequence quality distribution for trimmed (A) pair 6 read 1, (B) pair 6 read 2, (C) pair 16 read 1, and (D) pair 16 read 2. The x-axis depicts the position in the read and the y-axis depicts the flowcell tile that the read came from. Blue indicated high quality and red indicated low quality.

of the reads were mapped to the genome.

Table 7: Number of mapped and unmapped reads from the STAR alignment output

File Pair	Mapped Reads	Unmapped Reads
6	20,186,291	736,315
16	15,662,583	365,733

*"Mapped" means the read has a primary alignment to the genome. "Unmapped" means the read did not align to any location in the genome.

Evaluating Strandedness of the RNA-Seq Library

The number of reads mapping to features in the mouse genome were counted using Htseq-count. This algorithm was run twice: once where read 1 had to map the the same strand as the feature and read 2 to the opposite strand, and once where read 2 had to map the the same strand as the feature and read 1 to the opposite strand. I propose that these data are strand-specific because there is a large difference in performance depending on which read maps to the feature and which read maps to the opposite strand. When read 2 maps to the feature and read 1 maps to the opposite strand, the percentage of reads mapped to genes was 82% for file pair 6 and 85% for file pair 16 (Table 8). When read 1 maps to the feature and read 2 maps to the opposite strand, file sets 6 and 16 only mapped 3.45% and 4.06% of the reads to genes, respectively. Therefore, since one orientation of mapping the reads to features performed better than another, it can be concluded that the library is strand specific.

Strand-specific RNA-seq libraries are useful because they allow for identification of which strand of cDNA corresponds back the the mRNA template. The template strand is used to transcribe the mRNA, and the non-template strand has the same code as the mRNA. In Illumina stranded sequencing mRNA library prep, read 1 maps to the template strand and read 2 maps to the non-template strand [7]. Therefore, it it is reasonable that in these samples the orientation where read 1 mapped to the opposite strand of the feature and read 2 mapped to the same strand of the feature had a higher percentage of mapped reads, because read 2 contains the same sequence as the original mRNA. It is likely that the small percent of genes mapped in the other orientation was due to chance.

Table 8: Proportion of reads mapped to a feature in the mouse genome

File Pair	stranded = " "	Mapped Reads	Total Reads	Percentage Mapped
6	yes	360978	10461303	3.45%
6	reverse	8607459	10461303	82.28%
16	yes	325575	8014158	4.06%
16	reverse	6872907	8014158	85.76%

*`stranded = yes` means the first read has to map on the same strand as the feature and the second read has to map to the opposite strand. `stranded = reverse` means the second read has to map on the same strand as the feature and the first read has to map to the opposite strand. "Mapped" means the read aligned to one gene in the mouse genome.

Challenge - Analysis of Metadata

One interesting observation is that the indexes contained N in them. File pair 6 should have had `TAGCCATG` as a barcode, but instead had `NAGCCATG`. File pair 16 should have had `ACGATCAG` but instead had `NCGATCAG`. According to Illumina, many demultiplexing softwares treat N as a wild card character [8], so my files could contain barcodes with any character in place of the N. It is unlikely that incorrect indexes were added to these files regardless of the wildcard N because the rest of the index is enough nucleotides away from the other indexes.

Based on the description of the library prep, 300 to 400 is the expected fragment size. Both sample 6 and 16 contained contaminants around size 32 base pairs. These sequences could possibly be primer dimers. A majority of the library in both files was around the 300 to 500 base pair range in length. Sample 16, however, has much less area in this peak compared to sample 6, which could explain why sample 16 has less records than sample 6. Sample 6 had an average fragment size of 441 base pairs and sample 16 had an average fragment size of 353 base pairs. Both of these lengths are within or near the target length range of 300-400 base pairs. Since sample 16 had a shorter average insert length, this could explain why sample 16 had a higher percentage of reads that were adapter trimmed (Table

5). The shorter average insert length would imply that the insert length for more sequences is shorter than the read length so they would run into the adapter sequence. Overall, these libraries were high quality, likely because the fragments that generated them were the correct length with little contamination.

Acknowledgement

Thanks to Leslie, David, Maxine, Lisa, and Jason for teaching me everything I know :)

Supporting Information Available

All python scripts and sbatch scripts with documentation are available at
https://github.com/julhays/QAA/tree/master?tab=readme-ov-file.
Data files are located in the University of Oregon Talapas HPC at the following path:
/projects/bgmp/shared/2017_sequencing/demultiplexed/

References

[1] "Babraham Bioinformatics - FastQC A Quality Control tool for High Throughput Sequence Data," Babraham.ac.uk, 2024. https://www.bioinformatics.babraham.ac.uk/projects/fastqc/ (accessed Sep. 04, 2024).
[2] "WHY are the first few bases of Illumina HiSeq reads of lower quality? - SEQanswers," SEQanswers, Aug. 13, 2015. https://www.seqanswers.com/forum/sequencing-technologies-companies/illumina-solexa/49565-why-are-the-first-few-bases-of-illumina-hiseq-reads-of-lower-quality (accessed Sep. 04, 2024).
[3] K. D. Hansen, S. E. Brenner, and S. Dudoit, "Biases in Illumina transcriptome sequencing caused by random hexamer priming," Nucleic Acids Research, vol. 38, no. 12, pp. e131–e131, Apr. 2010, doi: https://doi.org/10.1093/nar/gkq224.

[4] “KAPA Stranded RNA-Seq Kits,” Sequencing, 2024. https://sequencing.roche.com/us/en/products/gro
stranded-rna-seq-kits.html (accessed Sep. 08, 2024).

[5] “IDT for Illumina–TruSeq DNA and RNA UD Indexes,” Illumina.com, 2023. https://support-docs.illumina.com/SHARE/AdapterSequences/Content/SHARE/AdapterSeq/TruSeq/UDIndexes.htm (accessed Sep. 08, 2024).

[6] ecSeq Bioinformatics, “Why has the reverse read 2 a worse quality than the forward read 1 in Illumina sequencing?,” Ecseq.com, 2020. https://www.ecseq.com/support/ngs/why-has-reverse-read-a-worse-quality-than-forward-read-in-Illumina-sequencing#:˜:text=Both%2C%201)%20a%20
(accessed Sep. 08, 2024).

[7] “Which reads map to each strand in Stranded RNA workflows, and how is strandedness achieved? | Illumina Knowledge,” Illumina.com, 2019. https://knowledge.illumina.com/library-preparation/rna-library-prep/library-preparation-rna-library-prep-reference_material-list/000002238 (accessed Sep. 08, 2024).

[8] “Using an N wildcard in index sequences in different Illumina FastQ generation software | Illumina Knowledge,” Illumina.com, 2019. https://knowledge.illumina.com/software/general/software-general-reference_material-list/000003560 (accessed Sep. 09, 2024).