QAA Final Report

Ike Sanderson

2023-09-14

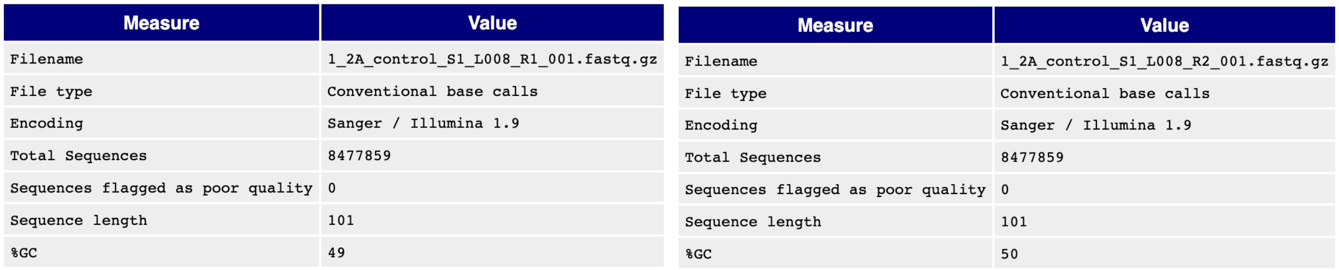
Contents

QAA Report

Part 1A:Read quality score distributions: Fast QC Analysis of 1__2A

Introduction:

I ran FastQC version 0.11.5 on the two pairs of reads I analyzed for this assignment. The FastQC output is presented below



Measure	Value
Filename	1_2A_control_S1_L008_R1_001.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	8477859
Sequences flagged as poor quality	0
Sequence length	101
%GC	49

Measure	Value
Filename	1_2A_control_S1_L008_R2_001.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	8477859
Sequences flagged as poor quality	0
Sequence length	101
%GC	50

Figure 1: 1-2A Basic Statistics. Of note in this report is the read length at 101. GC content is about .50, which is within the normal range for mouse (avg .51).

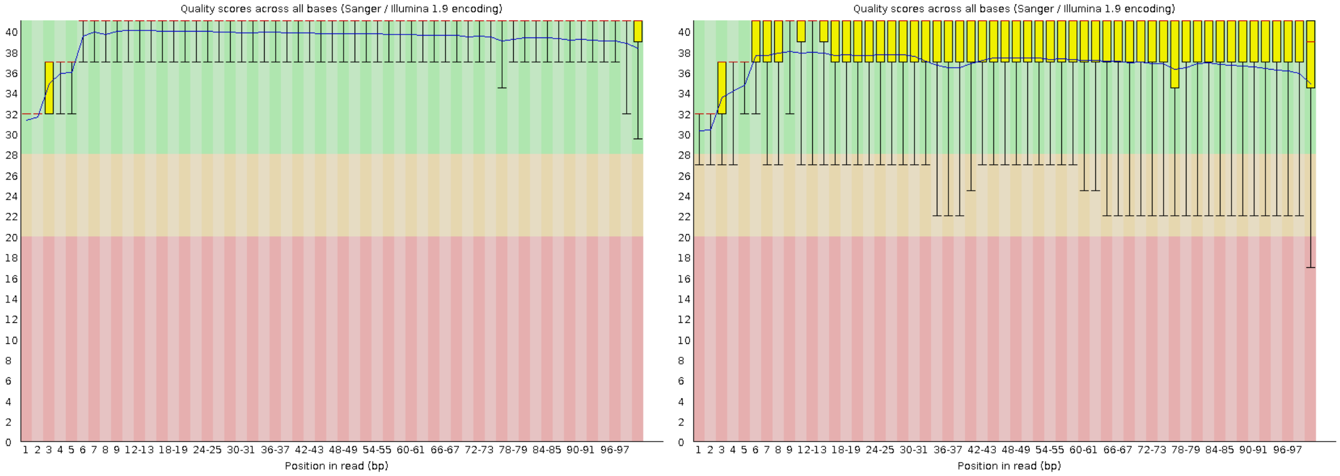


Figure 2: 1-2A Per base sequence quality, Read 1 (left) and read 2 (right) per base sequence quality.This crucial diagram tells me that Read 1 quality is all good, while Read 2 quality dips into the medium level. Because the quality does not drop into the low quality region, I am still confident that the reads will yield meaningful data. The average score (blue line) is good for both reads.

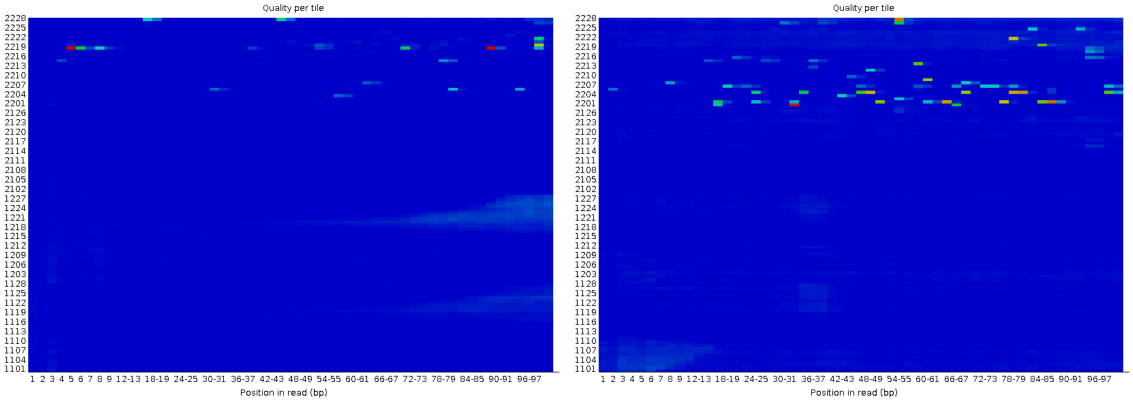


Figure 3: 1-2A Per tile sequence quality, Read 1 (left) and read 2 (right) per tile sequence quality. Note the lighter area on the read 1 tile, halfway down on the right side. This may indicate irregularities in the lane during the reaction. Because the color indication is not extreme, I do not think this is a cause for concern. Note the increased cells of color change on the read 2 tile compared to the read 1 tile. These indicate decreased quality in multiple reads, which is echoed by other output in this report (see per base quality, above)

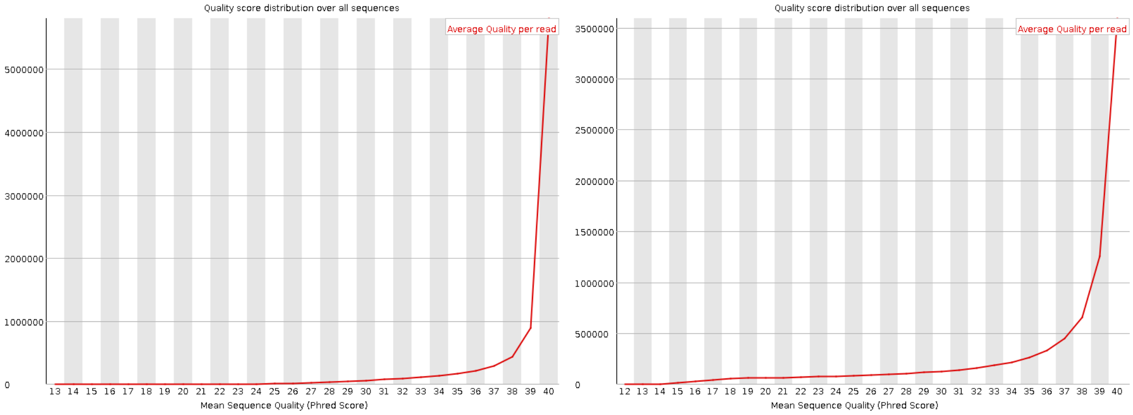


Figure 4: 1-2A Per sequence quality scores, Read 1 (left) and read 2 (right) per sequence quality scores. These figures show overall high scores.

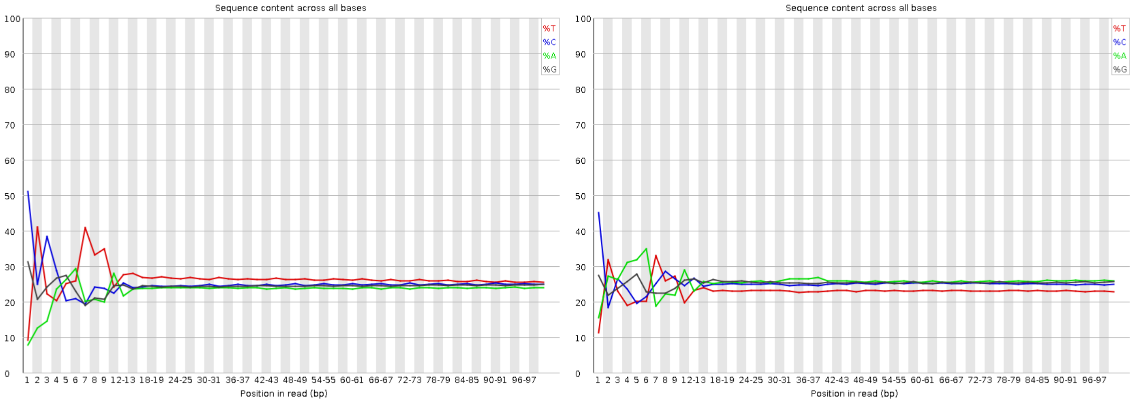


Figure 5: 1-2A Per base sequence content, Read 1 (left) and read 2 (right) per base sequence content. If these reads were all random bases then the lines would all be roughly parallel. The fact that they are not parallel at the beginning of the sequences tells me that there is somthing non-random about the start. Perhaps these all have a similar sequence such as regulatory spots in a 5'UTR.

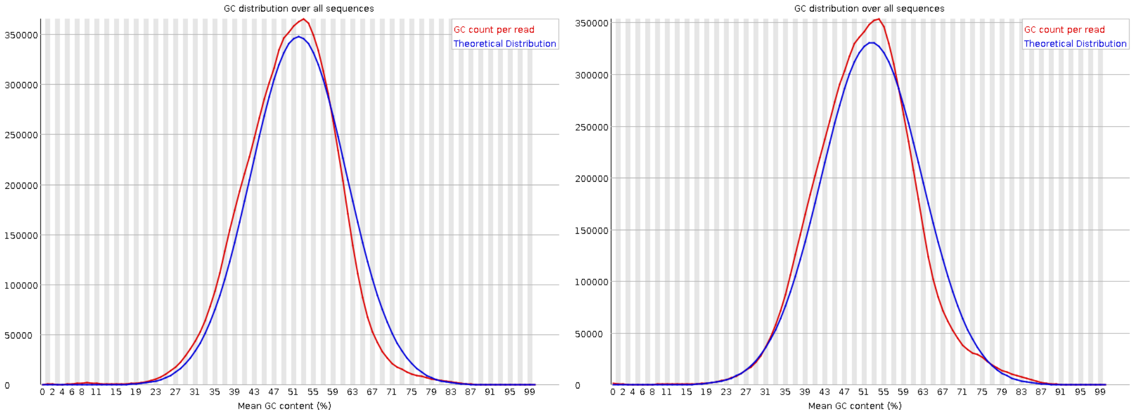


Figure 6: 1-2A Per sequence GC content, Read 1 (left) and read 2 (right) per sequence GC content. As noted above, the GC content in these reads conforms to the expected norm.

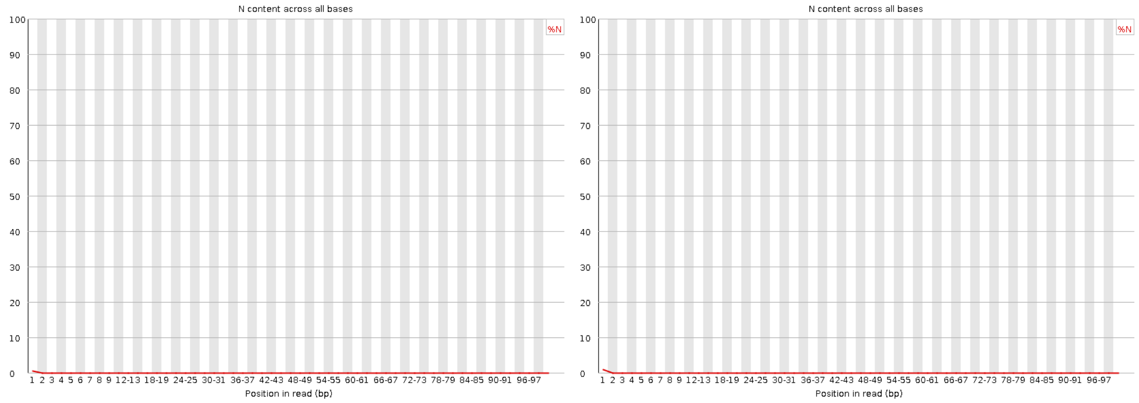


Figure 7: 1-2A Per base N content, Read 1 (left) and read 2 (right) per base N content. These reads have almost no Ns resulting from unknown base calls. This gives further confidence that the reads can be used for further analysis.

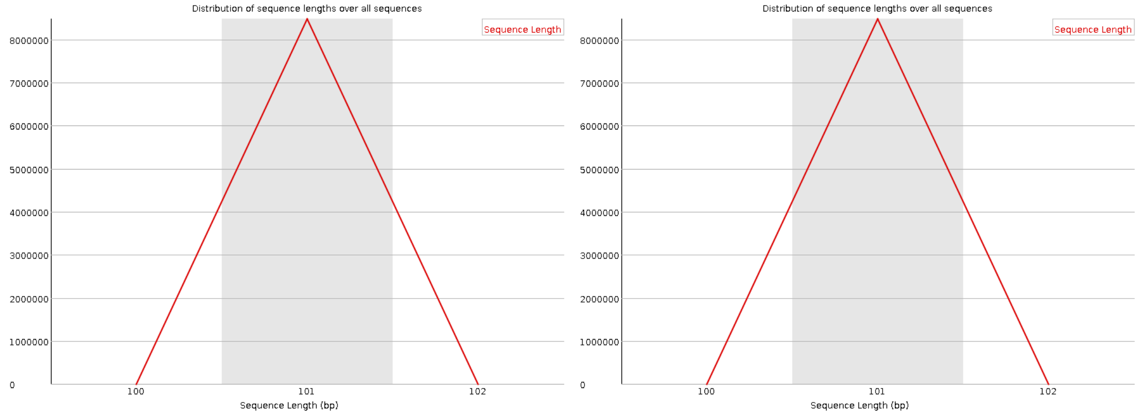


Figure 8: 1-2A Sequence length distribution, Read 1 (left) and read 2 (right) sequence length distribution.These reads all start out 101 bp.

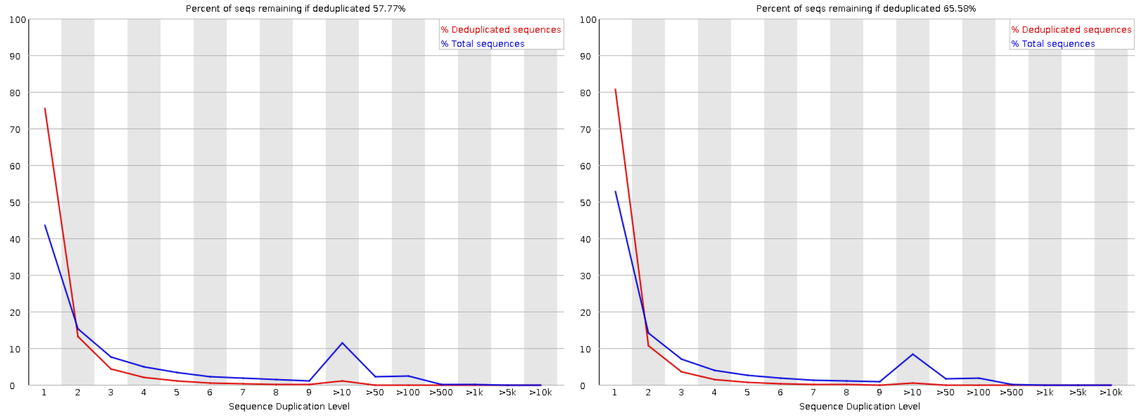


Figure 9: 1-2A Sequence duplication levels, Read 1 (left) and read 2 (right) sequence duplication levels. There is only a little bit of sequence duplication in these reads. No alert was raised by the module, which would have warned of too great a degree of duplication.

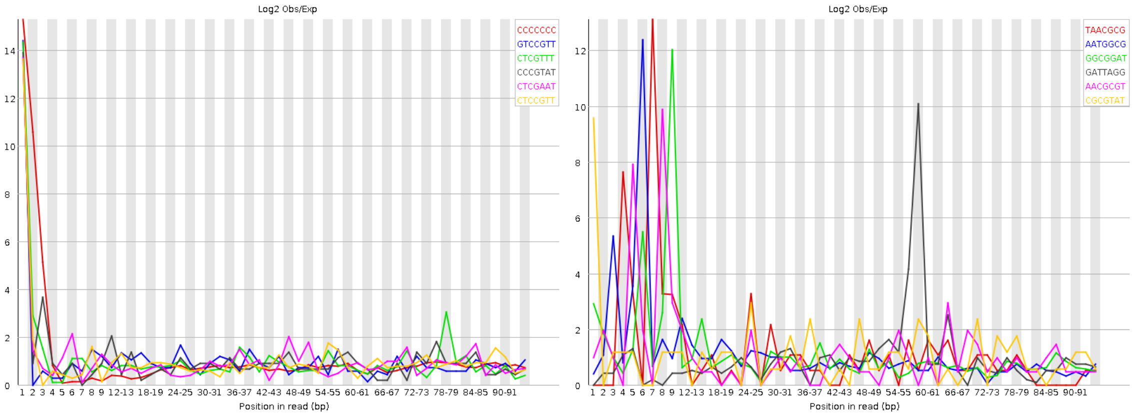
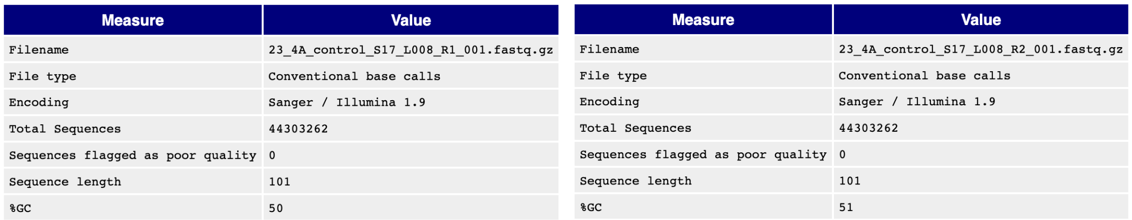


Figure 10: 1-2A Kmer content, Read 1 (left) and read 2 (right) Kmer content. There are a few spikes in 5-mer enrichment but not so great that an alert was raised (failure threshold at 10x). The ones at the start might indicate more 5'UTR features. I don't really have an explanation for Read 2's spike at bp 60.

Part 1B:Read quality score distributions: Fast QC Analysis of 23_4A



Measure	Value
Filename	23_4A_control_S17_L008_R1_001.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	44303262
Sequences flagged as poor quality	0
Sequence length	101
%GC	50

Measure	Value
Filename	23_4A_control_S17_L008_R2_001.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	44303262
Sequences flagged as poor quality	0
Sequence length	101
%GC	51

Figure 11: 23-4A Basic Statistics, Summary information. Of note in this report is the read length at 101. GC content is about .51, which is right at the normal range for mouse (avg .513).

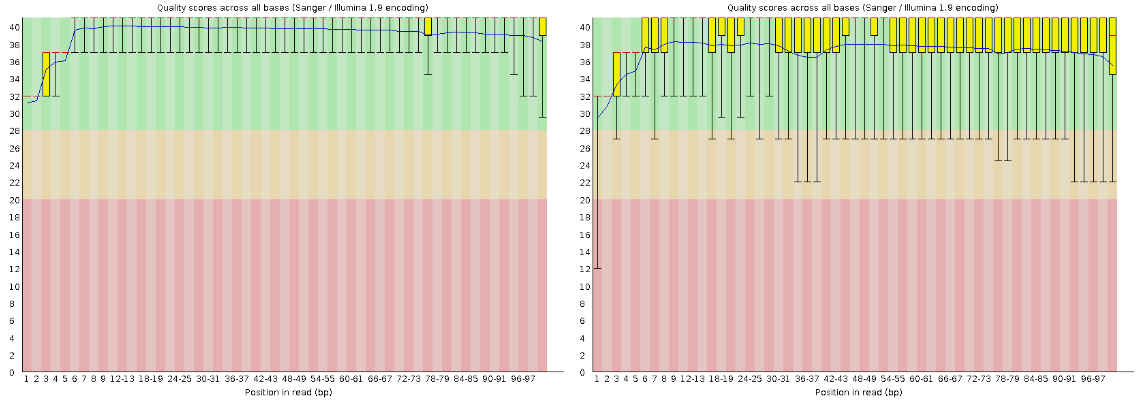


Figure 12: 23-4A Per base sequence quality, Read 1 (left) and read 2 (right) per base sequence quality.This diagram tells me that Read 1 quality is good. Read 2 quality is also good, though the 90th percentile range of many read quality scores dip into the medium level. Because the average quality does not drop into the low quality region, I am still confident that the reads will yield meaningful data.

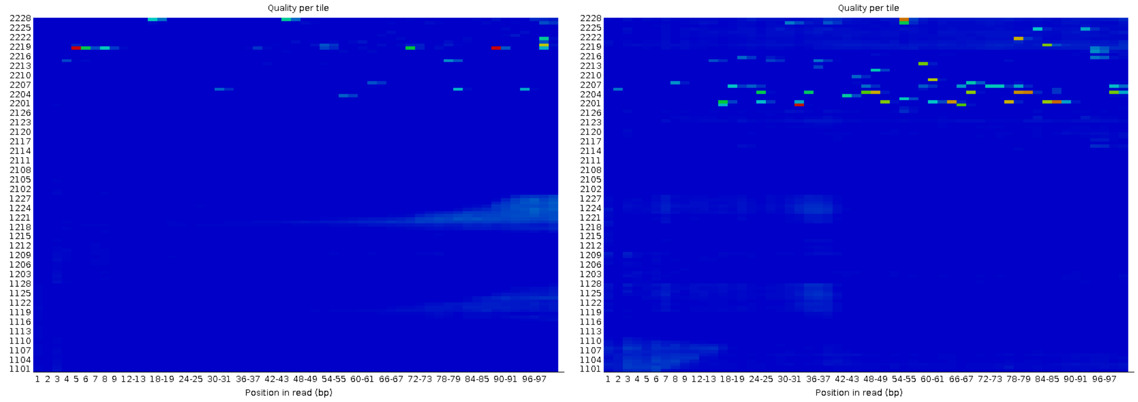


Figure 13: 23-4A Per tile sequence quality, Read 1 (left) and read 2 (right) per tile sequence quality. Much like the tile quality report above: note the lighter area on the read 1 tile, halfway down on the right side. This may indicate irregularities in the lane during the reaction. Because the color indication is not extreme, I do not think this is a cause for concern. Note the increased cells of color change on the read 2 tile compared to the read 1 tile. These indicate decreased quality in multiple reads, which is echoed by other output in this report (see per base quality, above)

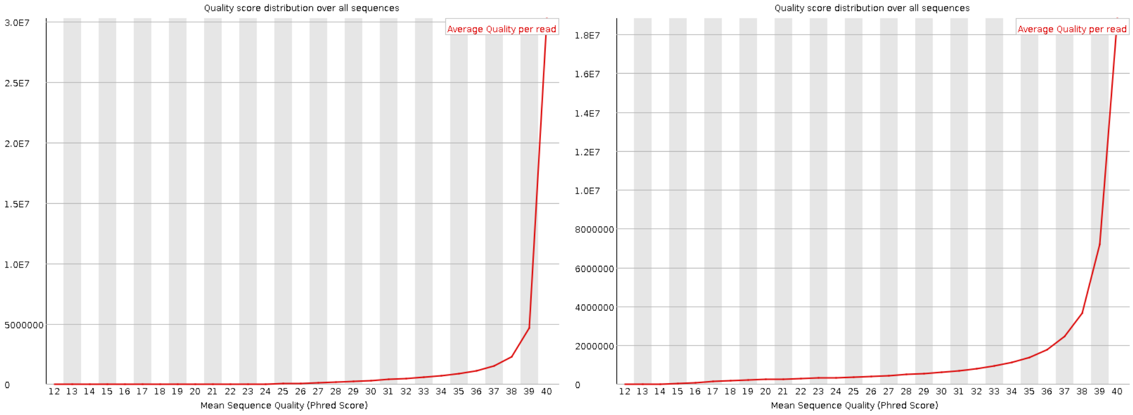


Figure 14: 23-4A Per sequence quality scores, Read 1 (left) and read 2 (right) per sequence quality scores. These figures show overall high scores.

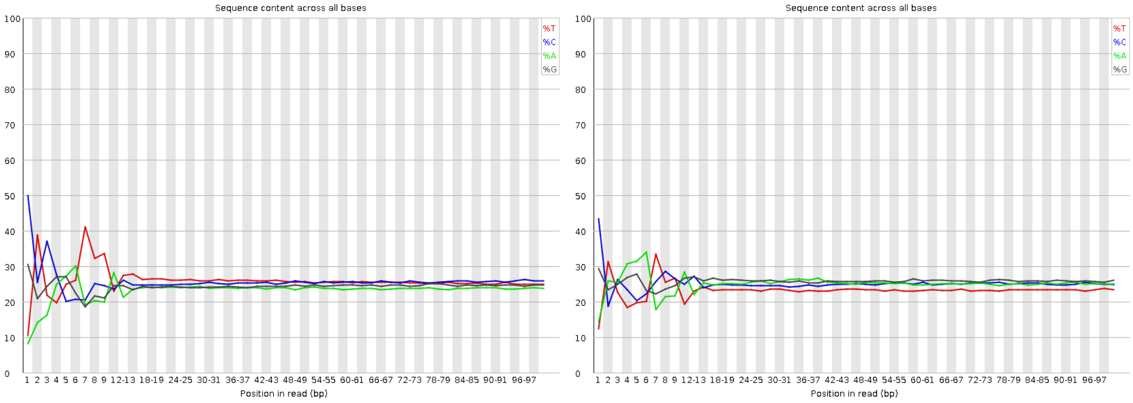


Figure 15: 23-4A Per base sequence content, Read 1 (left) and read 2 (right) per base sequence content. I have the same hypothesis about these reads: The fact that they are not parallel at the beginning of the sequences tells me that there is somthing non-random about the start. Perhaps these all have a similar sequence such as regulatory spots in a 5'UTR.

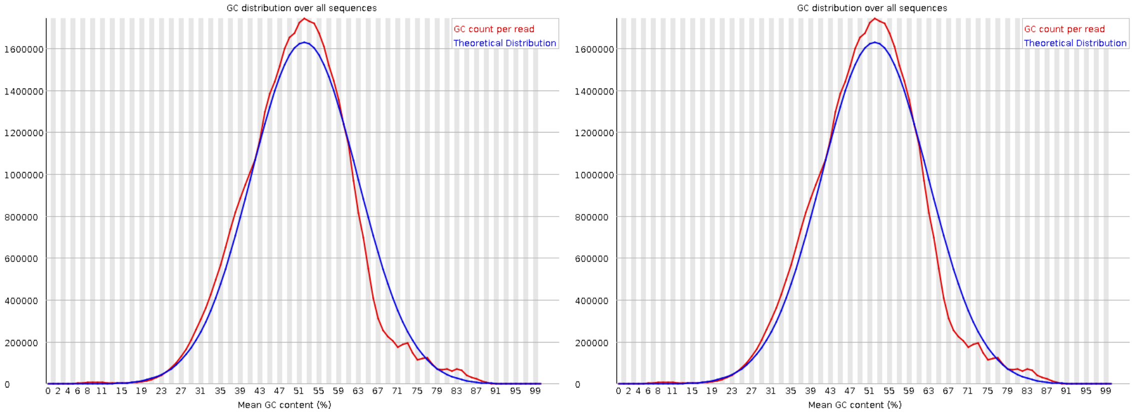


Figure 16: 23-4A Per sequence GC content, Read 1 (left) and read 2 (right) per sequence GC content. As noted above, the GC content in these reads conforms to the expected norm.

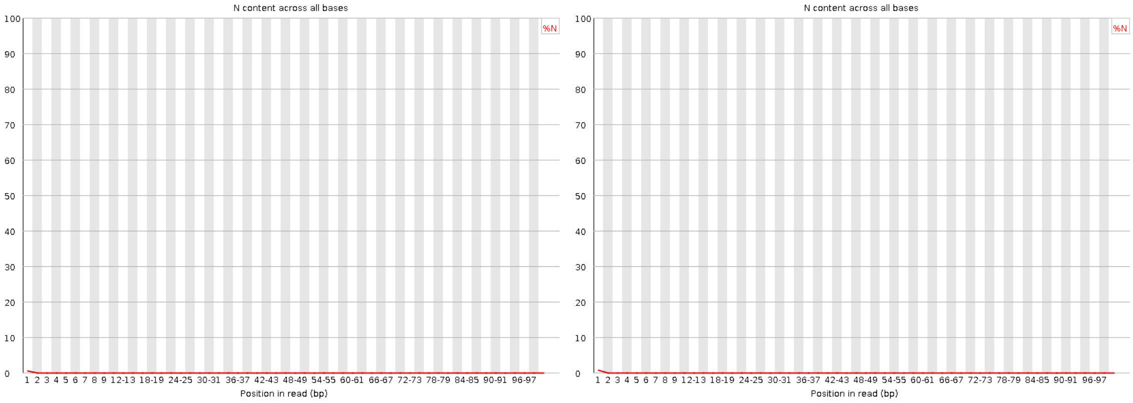


Figure 17: 23-4A Per base N content, Read 1 (left) and read 2 (right) per base N content. These reads have almost no Ns resulting from unknown base calls. This gives further confidence that the reads can be used for further analysis.

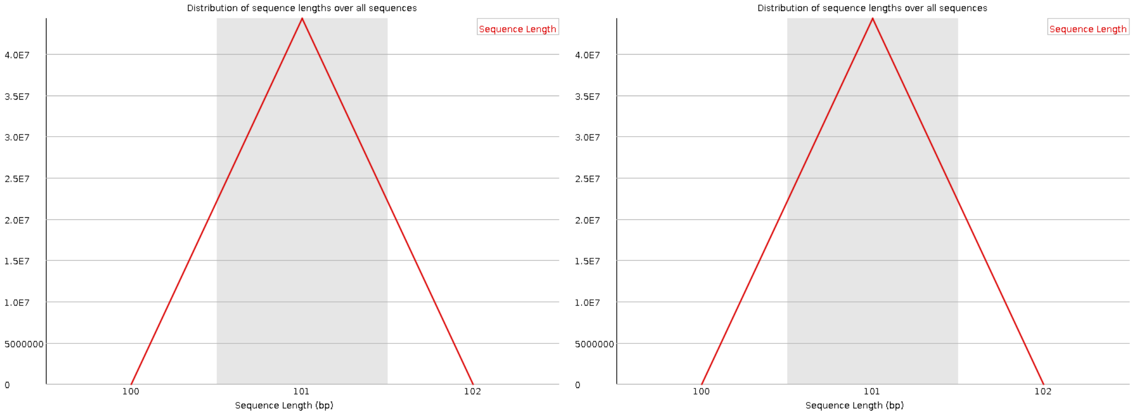


Figure 18: 23-4A Sequence length distribution, Read 1 (left) and read 2 (right) sequence length distribution.These reads all start out 101 bp.

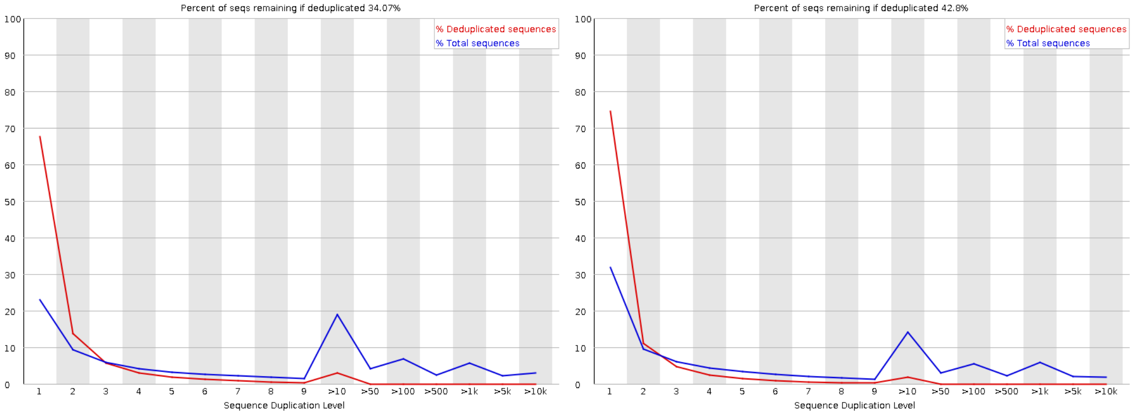


Figure 19: 23-4A Sequence duplication levels, Read 1 (left) and read 2 (right) sequence duplication levels. Again, like the previous two libraries, there is not really any sequence duplication in these reads.

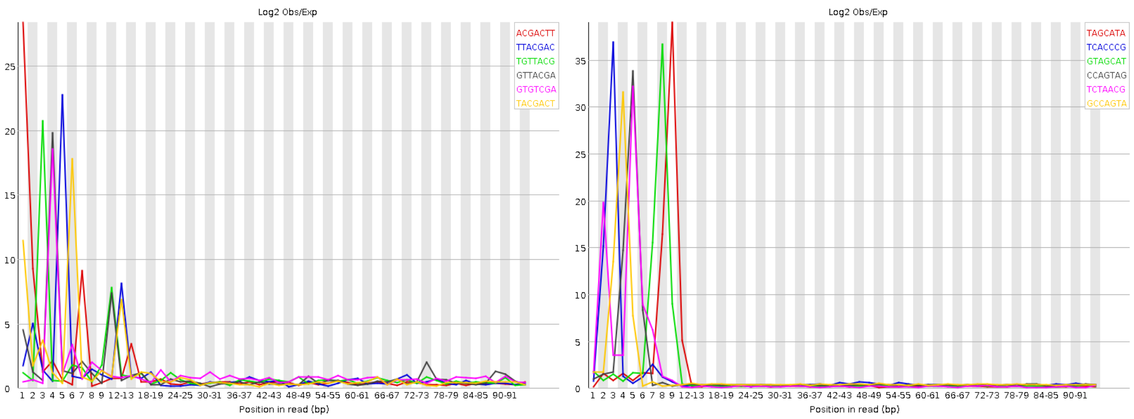


Figure 20: 23-4A Kmer content, Read 1 (left) and read 2 (right) Kmer content. There are few spikes in 5-mer enrichment.

Part 1C: Read quality score distributions – Original quality score plot via Python script

These figures show less information than the FastQC charts because I did not compute quantiles or display range in any way. There's no sense that the quality scores in read 2 suffered to the extent that they did. In addition, these each took about as long to construct as the entire FastQC report took for each library sample. FastQC has proven its value.

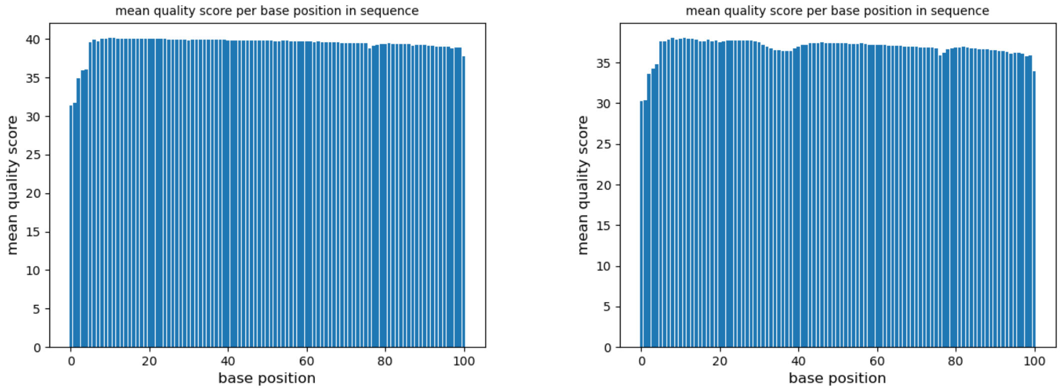


Figure 21: Read 1 (left) and read 2 (right) from library 1-2A. Mean quality score by base position.

It's difficult to decipher just how much better the scores are for read 2 of this library compared to the read 2 scores on 1-2A above. Again, the detail available through FastQC is significant. Caveat: I wrote this script quite early in my coding career. It's entirely possible I could improve upon its performance in the coming years.

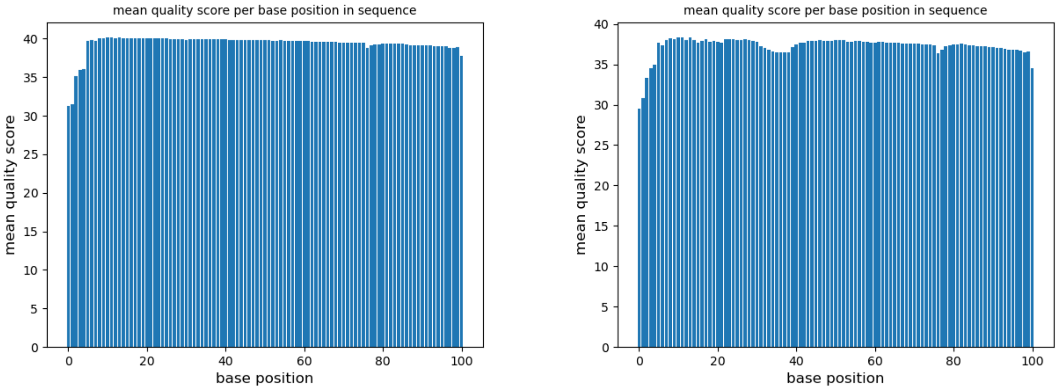


Figure 22: Read 1 (left) and read 2 (right) from library 23-4A. Mean quality score by base position.

Part 2A: Adapter Trimming Comparison

I used cutadapt version 4.4 to trim adapter sequences

Post cutadapt data from 1_2A:

Total read pairs processed: 44,303,262 Read 1 with adapter: 1,359,563 (3.1%) Read 2 with adapter: 1,657,134 (3.7%) Pairs written (passing filters): 44,303,262 (100.0%)

Total basepairs processed: 8,949,258,924 bp Read 1: 4,474,629,462 bp Read 2: 4,474,629,462 bp Total written (filtered): 8,925,098,135 bp (99.7%) Read 1: 4,463,208,431 bp Read 2: 4,461,889,704 bp

=== First read: Adapter 1 ===

Sequence: AGATCGGAAGAGCACACGTCTGAACTCCAGTCA; Type: regular 3'; Length: 33; Trimmed: 1359563 times

Minimum overlap: 3 No. of allowed errors: 1-9 bp: 0; 10-19 bp: 1; 20-29 bp: 2; 30-33 bp: 3

Bases preceding removed adapters: A: 23.4% C: 29.6% G: 30.7% T: 15.0% none/other: 1.3% === Second read: Adapter 2 ===

Sequence: AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT; Type: regular 3'; Length: 33; Trimmed: 1657134 times

Minimum overlap: 3 No. of allowed errors: 1-9 bp: 0; 10-19 bp: 1; 20-29 bp: 2; 30-33 bp: 3

Bases preceding removed adapters: A: 26.3% C: 28.5% G: 33.0% T: 11.1% none/other: 1.1%

I cannot find my record of the second set of reads. I remember the proportion of processed reads being approximately the same (about 3%), but I don't have the data to back it up.

Part 2B: Trimmed read length distributions

I wrote an R script to display R1 and R2 reads.

I expect R2 to be more aggressively trimmed than R1 because trimmomatic uses a sliding window technique that will call a read unpaired as soon as it hits a minimum threshold for sequential base pairs. This prevents the effect of a single sequencer error causing misalignment in typically low coverage RNA seq data. The effect, however, is to create a lot of trimmed, unpaired reads.

We set the minimum length at 35, thus that's the shortest we will see.

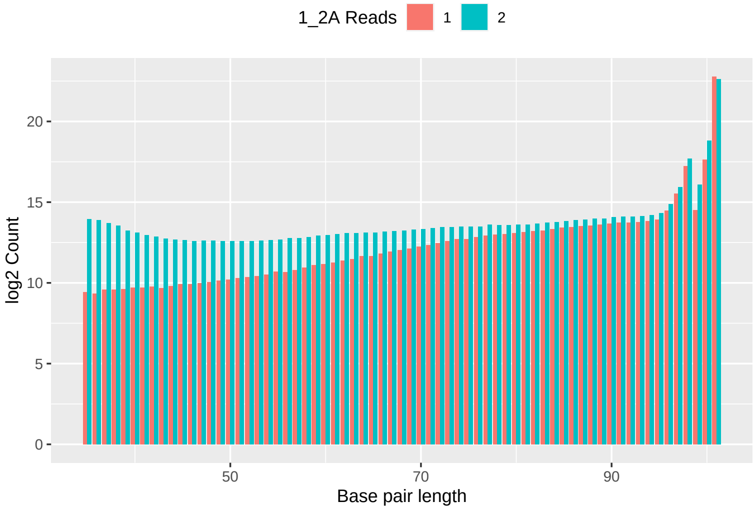


Figure 23: Reads 1 and 2 log2 transformed and plotted on the same axes. Note read 2 count increasing as the sequences decrease.

```
plt23 + theme(legend.position="top") +
  labs(x = "Base pair length", y = "log2 Count", fill = "23_4A Reads")
```

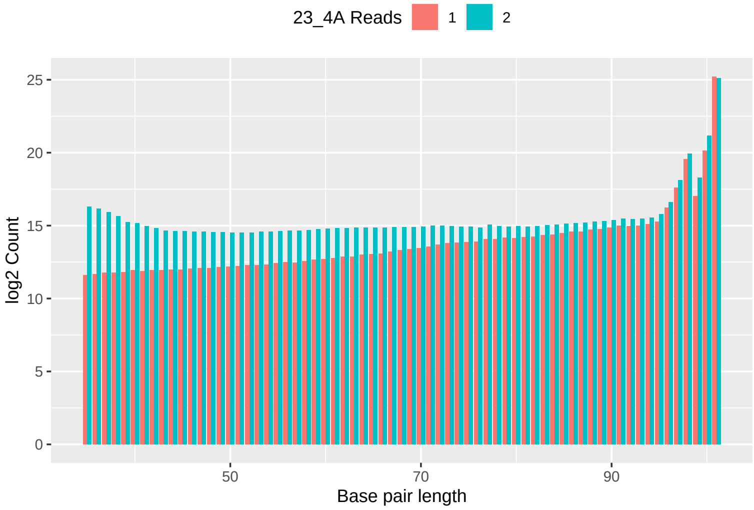


Figure 24: Reads 1 and 2 log2 transformed and plotted on the same axes. Once again, note read 2 count increasing as the sequences decrease.

In both cases, the quality of the reads was decreasing after the long period of time on the sequencer. On a sequencer like the Illumina Novaseq, the molecules had undergone half a day of biology. Read quality degradation is a=known phenomenon with this technology, but our library sequences still have data that tell us a story and we can conitnue to use it for further analysis. The quality scores are still good enough on the remaining reads.

Part 3: Alignment and Strand specificity

I used a simple samparser Python script to count mapped and unmapped reads. (It looks at bitwise flags to determine primary or secondary mapping)

I got the following:

I ran samparser.py version 1.1 on ikes_1_2A_Aligned.out.sam 15627427 reads were mapped 305259 reads were unmapped That's 98% mapped

I ran samparser.py version 1.1 on ikes_24_4A_Aligned.out.sam 79472970 reads were mapped 4640182 reads were unmapped that's 94.2% mapped

This tells me that the reads are mouse (the identity of the alignment genome) but does not tell me strandedness. I used HTSeq-count for that

We need to know if the library is stranded or not. To label a library as stranded means that the directionality of the original sequence is preserved. The sequences will be either sense or anti sense. This means that when aligned, they should all be mapped if the library is stranded. If the library prep was not made in such a way as to preserve strandedness (i.e. the first or second copied strand was NOT destroyed) then the sequences will be roughly 50/50 (sense:anti-sense) and should map in HTSeq at about that ratio no matter which strand of the feature is examined by HTSeq-count. Because the script found an overwhelming proportion of the reads mapped to the reference genome when aligned using STAR. These must be stranded libraries.

I ran an awk command to calculate percent mapped. Here are the results:

```
$ awk '$1 ~ "ENS"{s+=$2}{n+=$2}END{print s*100/n}' 24_4A_htseqout_AlignedRevstr.tsv
78.3448
$ awk '$1 ~ "ENS"{s+=$2}{n+=$2}END{print s*100/n}' 24_4A_htseqout_AlignedStr.tsv
3.68428
$ awk '$1 ~ "ENS"{s+=$2}{n+=$2}END{print s*100/n}' 1_2A_htseqout_AlignedStr.tsv
3.84614
$ awk '$1 ~ "ENS"{s+=$2}{n+=$2}END{print s*100/n}' 1_2A_htseqout_AlignedRevstr.tsv
86.3226
```

These percentages tell me clearly that the libraries were stranded. Because the reverse strand mapped the most, this means the first copied strand was preserved during library prep; the second copy was destroyed.