RNA-seq Quality Assessment

Farris Tedder

2023-09-15

Sequences Assessed in Report

Library	Group	Treatment	Index	Index Sequence
32	4G	Both	B10	AGAGTCCA
27	4C	MBNL	C2	ATCGTGGT

*Sequences are refered to by their library number for the remainder of the report.

Part 1 – Read Quality Score Distributions

FastQC Quality Assessments

Library 27

Figure 1.

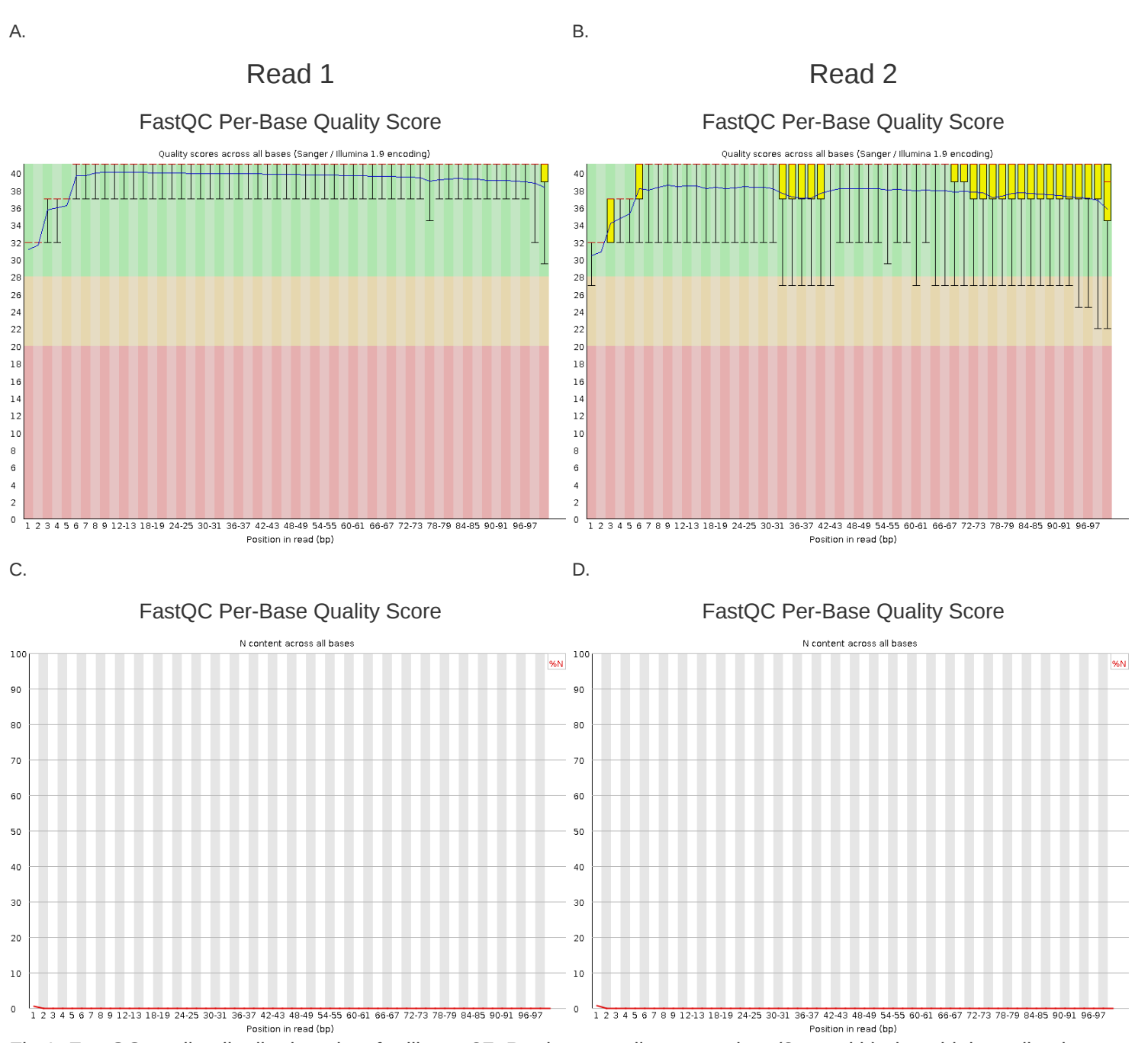

Fig 1: FastQC quality distribution plots for library 27. Per-base quality score plots (2.a and b) show high quality data. Read 2 is lower quality, as predicted due to longer sequence time. N content for both reads is near 0 (1.c and d) with only a small uptick at bp 1.

Library 32

Figure 2.

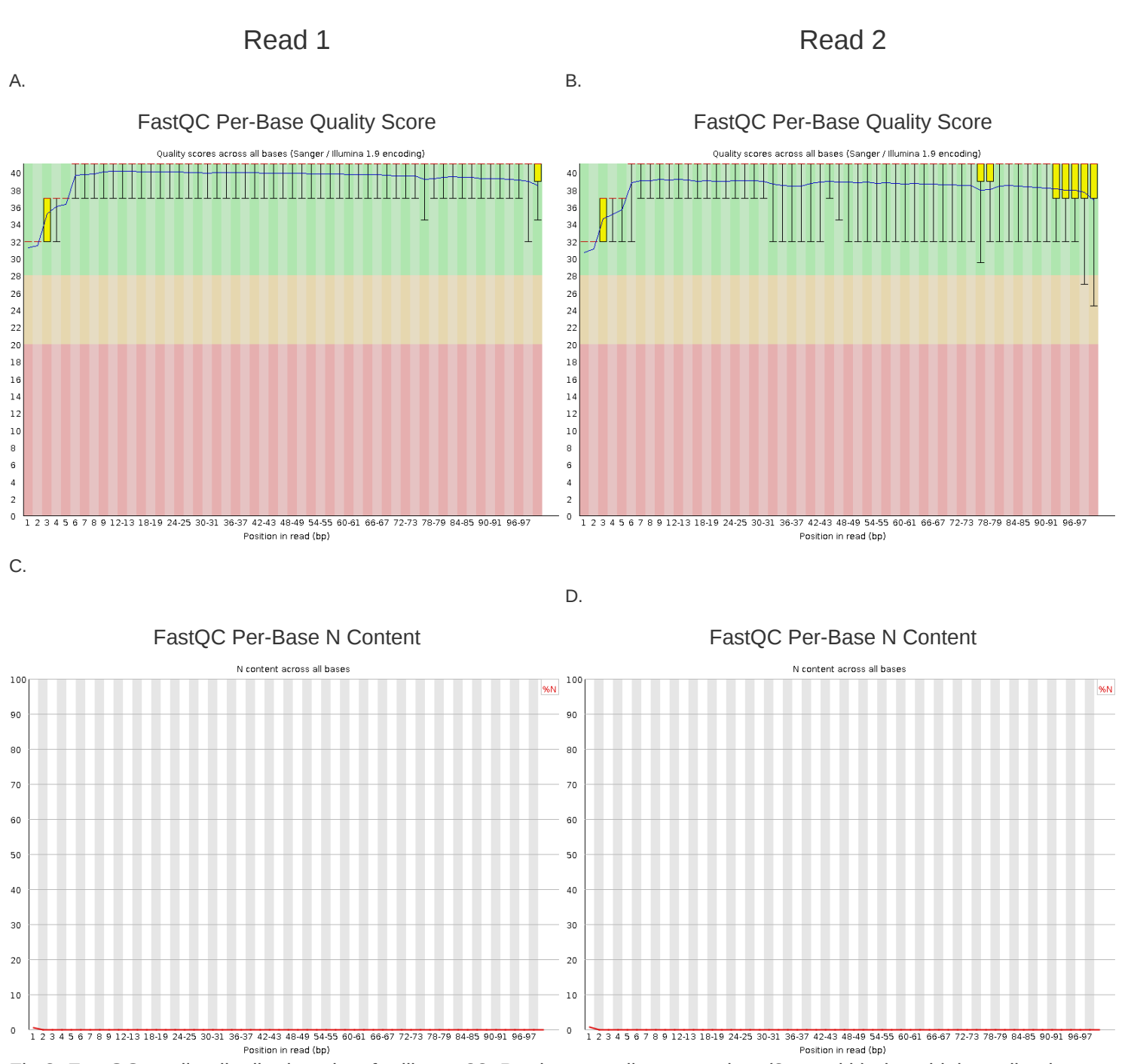

Fig 2: FastQC quality distribution plots for library 32. Per-base quality score plots (2.a and b) show high quality data. Read 2 is lower quality, as predicted due to longer sequence time. N content for both reads is near 0 (2.c and d) with only a small uptick at bp 1.

For all reads quality is high overall with a dip in the first ~7 base pairs and a very small decline in the last few base pairs. N abundance is virtually zero throughout, save the first 1-2 base pairs which account for part of the early base pair quality decline.

My Quality Assesments

Library 27

Figure 3.

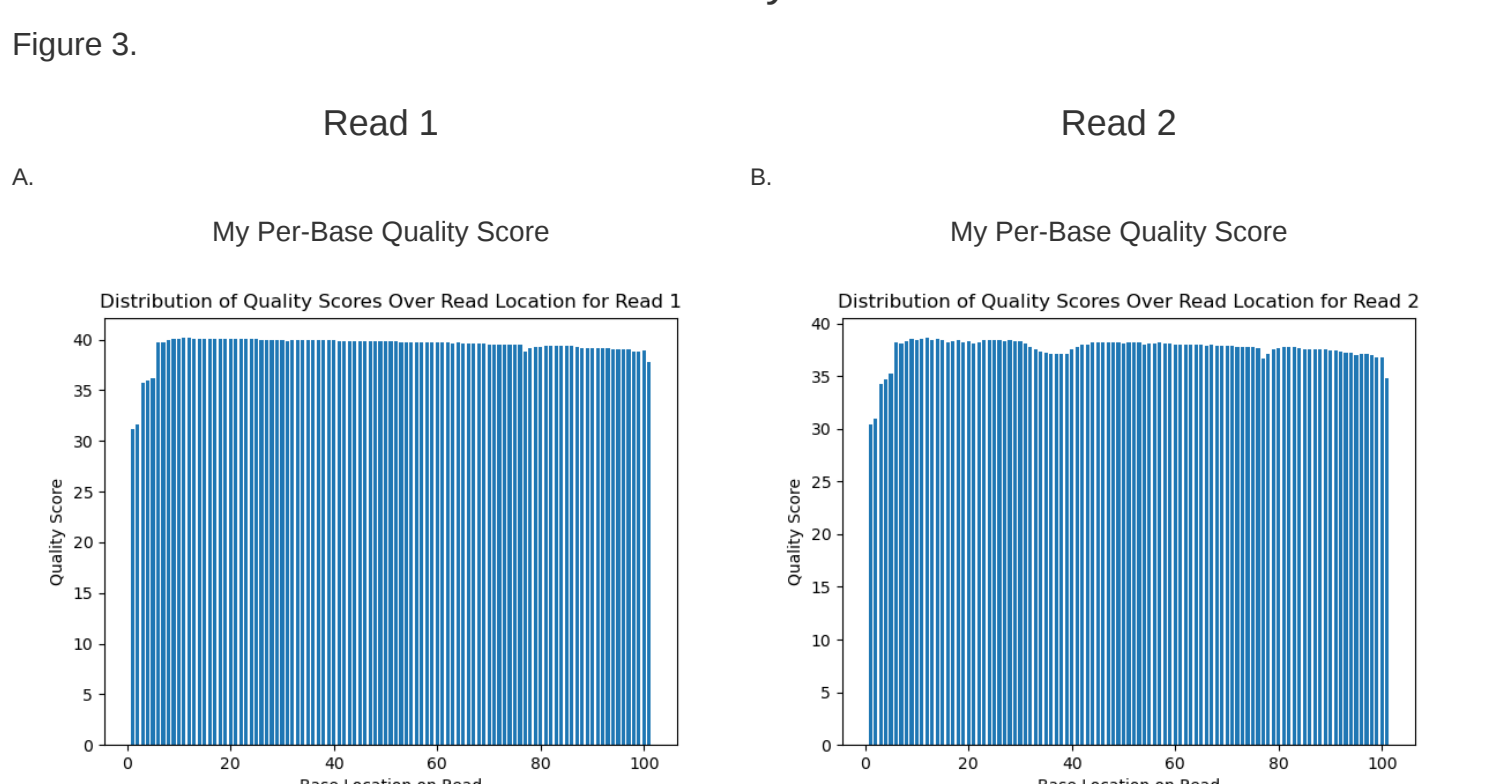

Fig 3: Per-base quality score distabution plots generated for library 27from my python script. The overall trend matches the FastQC plots well.

Library 32

Figure 4.

Fig 4: Per-base quality score distribution plots for library 32 generated from my python script. The overall trend matches the FastQC plots well.

Like the FastQC quality graphs, my graphs are also consistently high with a dip in the early base pairs. Both programs take between 30 seconds to a minute to generate graphs per file, with FastQC taking 30 seconds more overall. As FastQC is generating much more than a per-base quality score, it is a more efficient program than mine despite taking longer in this instance.

Overall, the data quality in both libraries is high enough for analysis. Read 1 especially has little variation among Q scores even towards the end of the sequence. Read 2 in Library 27 has the worst quality with variation barely dipping into the yellow, but it still has medians firmly planted in the high section. This read 2 quality dip is expected so I don't find it a reason to disqualify the data. N content is near 0. The only over-repesented sequence was found in Library 27, and it was the adapter. Trimming removed this repeat and improved overall quality (see below).

Post-Trimming Quality Distribution

Figure 5.

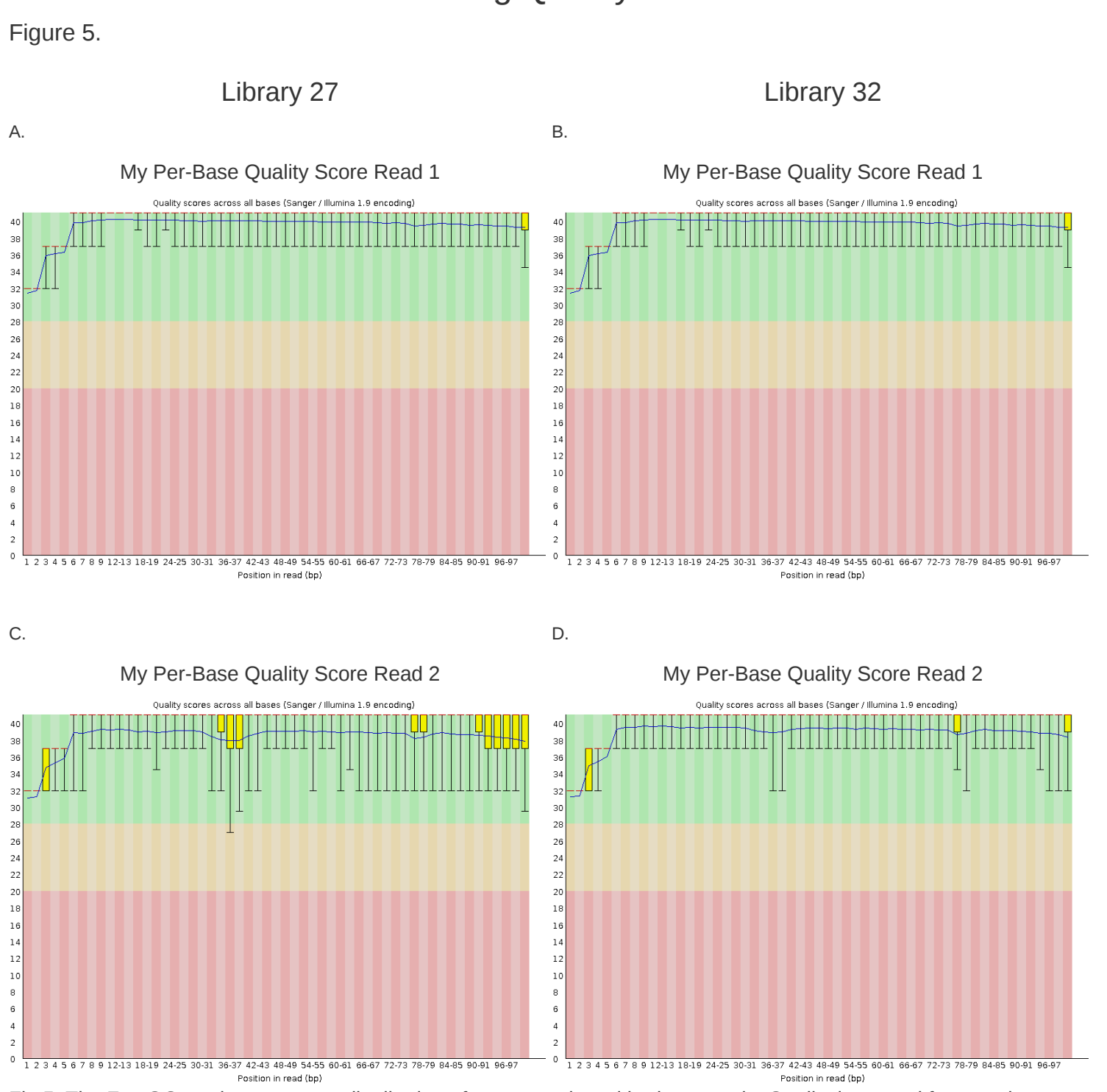

Fig 5: The FastQC per-base q-score distribution after processing with trimmomatic. Quality improved for most bases. The mid range bases (~36-37) for read 2 in library 27 are still touching the lower quality zone (5.c) but otherwise all scores are now fully in the green.

Part 2 – Adaptor trimming comparison

Trimming Results

Figure 6.

Percentages of Reads Lost After Processing

	Library 27		Library 32	
Stage	Read 1	Read 2	Read 1	Read 2
Cut	93.7%	93.4%	93.6%	93.1%
Trim	12.5%	6.5%	12.0%	9.9%

Fig 6: Percentage change after use of cutadapt and trimmomatic. Cutting the data removed around 93% of the sequences in all cases. Read 1 lost around 12% after trimming while read 2 loss less, 6.5% and 9.9% for libraries 27 and 32 respectivly.

Figure 7.

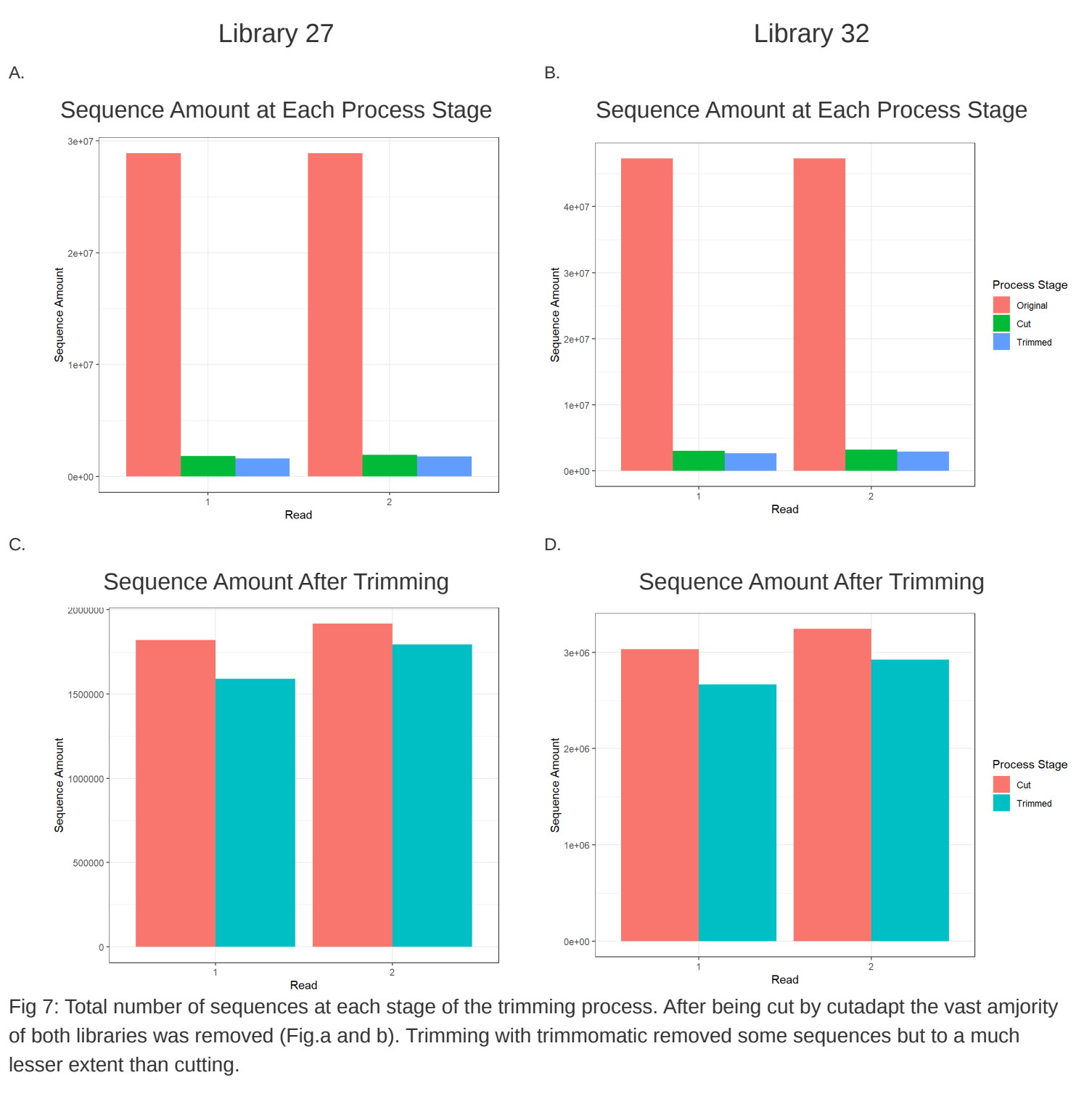

Fig 7: Total number of sequences at each stage of the trimming process. After being cut by cutadapt the vast amjority of both libraries was removed (Fig.a and b). Trimming with trimmomatic removed some sequences but to a much lesser extent than cutting.

Adapters

Adapters Used in Library Prep

Read 1	AGATCGGAAGAGCACACGTCTGAACTCCAGTCA
Read 2	AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT

Adapters can be found at:

https://knowledge.illumina.com/library-preparation/general/library-preparation-general-reference_material-list/000001314.

Adapters were confirmed through bash searches. Example of bash code:

```
zcat <file name> | grep "AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT" | wc -l
```

The listed adapters matched to Read 1 and Read 2 files respecivly. Adapters did not match to the other read in their library. Reverse compliments of the adapters were not found in any file. The adapters were not found in the cut files, as expected.

Read Length Distributions

The following data is log2 transformed

Figure 8.

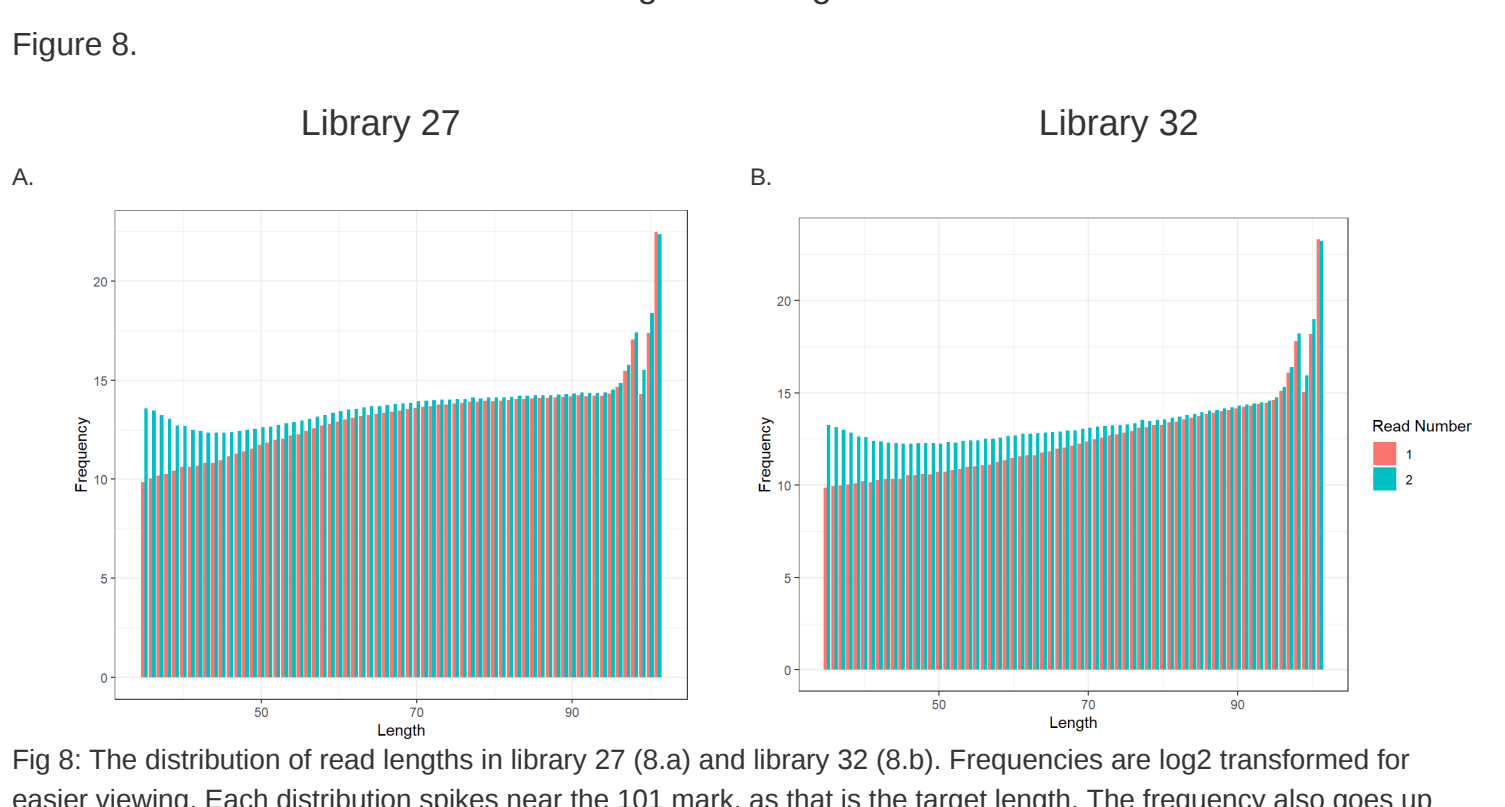

Fig 8: The distribution of read lengths in library 27 (8.a) and library 32 (8.b). Frequencies are log2 transformed for easier viewing. Each distribution spikes near the 101 mark, as that is the target length. The frequency also goes up slighly for read two near the lower lengths. This is likely due to lower data quality, requiring more trimming.

In both libraries read 1 is trimmed more than read 2. Because read 2 is on the sequencer longer it is more prone to accumulating error. Errors in the adapter effect trimming rate, as they could make the adapter unrecognizable. To verify this I checked the files for the last 14bp of their respective adapters. Results as follows:

Read 1 Sequence: CTGAACTCCAGTCA

Read 2 Sequence: TAGGGAAAGAGTGT

Presence of Adapter Fragment Before and After Cutting

Figure 9.

	Library 27		Library 32	
	Read 1	Read 2	Read 1	Read 2
Before	149644	150697	44828	45625
After	133	144	84	97
Percent Change	99.911%	99.904%	99.813%	99.787%

Fig 9: The number of times the test sequence could be found in each file. The amount went down after cutting for all files, but less so percentage wise for read 2 in both libraries.

The difference is not statistically significant, and some of those remaining hits may be part of the genome. However, read 2 has more adapter fragment hits than read 1 in both libraries. With more samples I believe we would see more significance.

Part 3 – Alignment and strand-specificity

Figure 10.

Mapped Reads after STAR

	Library 27	Library 32
Mapped	13320030	22404319
Unmapped	433880	533613

Fig 10: The number of mapped and unmapped reads for each library.

htseq

Counts of Reads Not Matching to Features

Figure 11.

	Library 27		Library 32	
	Count	Percentage of Reads	Count	Percentage of Reads
Stranded	6599565	96%	11024855	96%
Reverse	1189188	19%	1555049	15%

Fig 11: Chart showing sum of all unmatched strands per library. Percentage of reads refers to what percentage of all reads were unmatched. Library 32 is larger, leading to larger raw numbers, but has equivalent percentages to Library 27. The reverse method lead to less unmatched strands than the stranded method for both libraries.

Distribution of Cause for Mismatch

Figure 12.

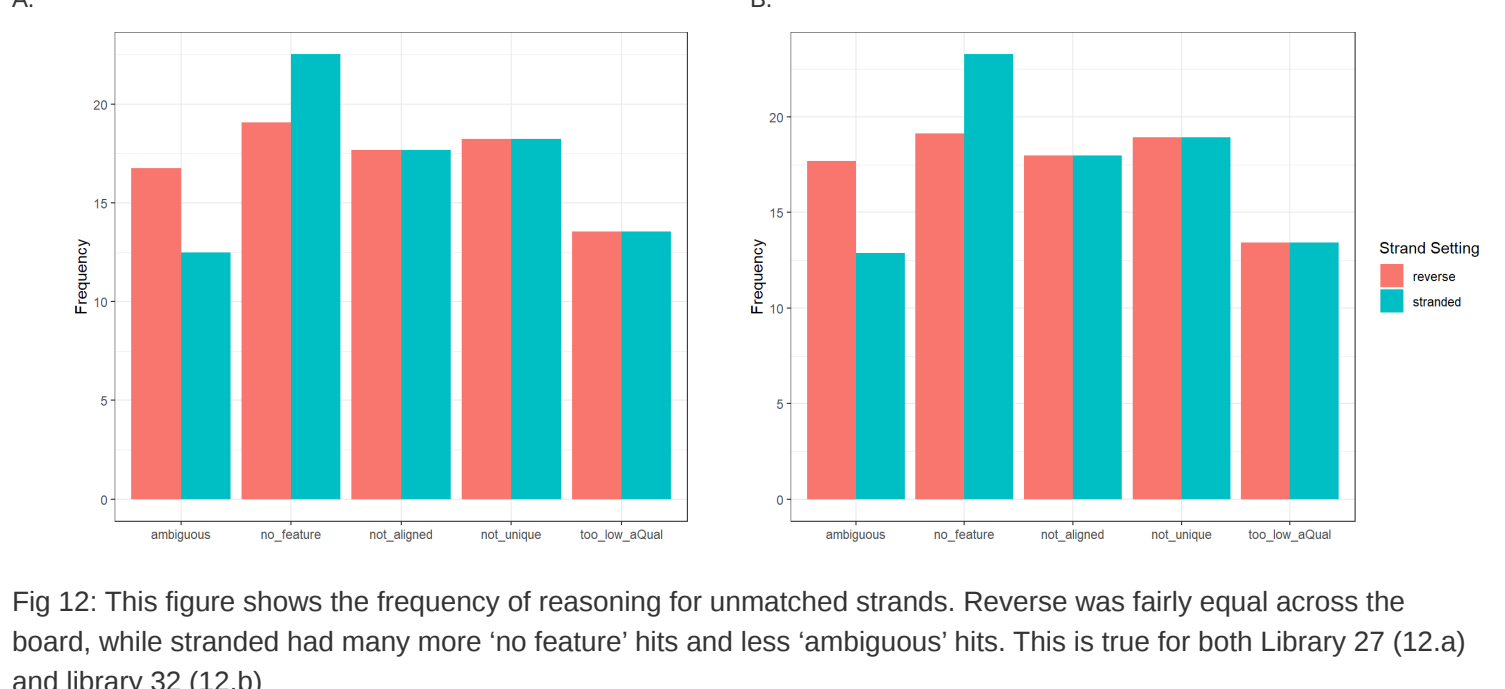

Fig 12: This figure shows the frequency of reasoning for unmatched strands. Reverse was fairly equal across the board, while stranded had many more 'no feature' hits and less 'ambiguous' hits. This is true for both Library 27 (12.a) and library 32 (12.b)

We know these libraries are stranded because they were prepared using KAPA's Stranded mRNA-Seq kit, according to the metadata. We can confirm this by looking at the htseq output. When we set stranded to 'yes' 96% of the data did not match to a feature for both libraries. When stranded was set to reverse only 19% and 15% did not match to a feature. Because 'reverse' output higher quality data it is most likely to match the correct strandedness setting. Reverse can only apply to stranded libraries.