QAA

Perris Navarro

2023-09-13

Contents

This assignment focused on using existing tools for quality assessment and adaptor trimming and comparing them to those of my own software. This document speaks to the difference between the two methods, and why existing tools are beneficial for bioinformaticians to know about and use.

Two samples are analyzed in this report: 1_2a_control_S1_L008 and 17_3E_fox_S13_L008. Both samples have two reads, R1 and R2. For the remainder of this document, R1 refers to Read 1 and R2 refers to Read 2.

Part 1 – Read quality score distributions.

FastQC Output: Mean Quality Score Distributions for 1_2A_control_S1_L008.

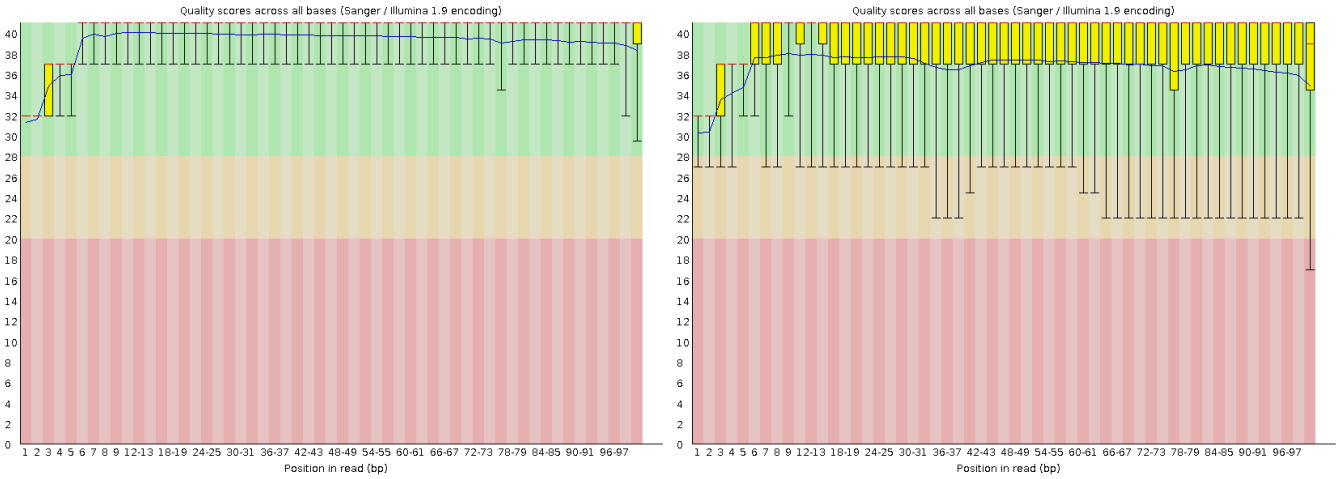

Figure 1: Quality score distributions for 1-2A-control-S1-L008.The line represents the mean quality score for each bp, or position in the read. The error bars represent the distribution of scores found at that particular base. Green indicates an acceptable quality score. As you can see, R2 has a much wider distribution of scores than R1. The left graph is R1, the right graph is R2.

Python script Output: Mean Quality Score Distributions for 1_2A_control_S1_L008.

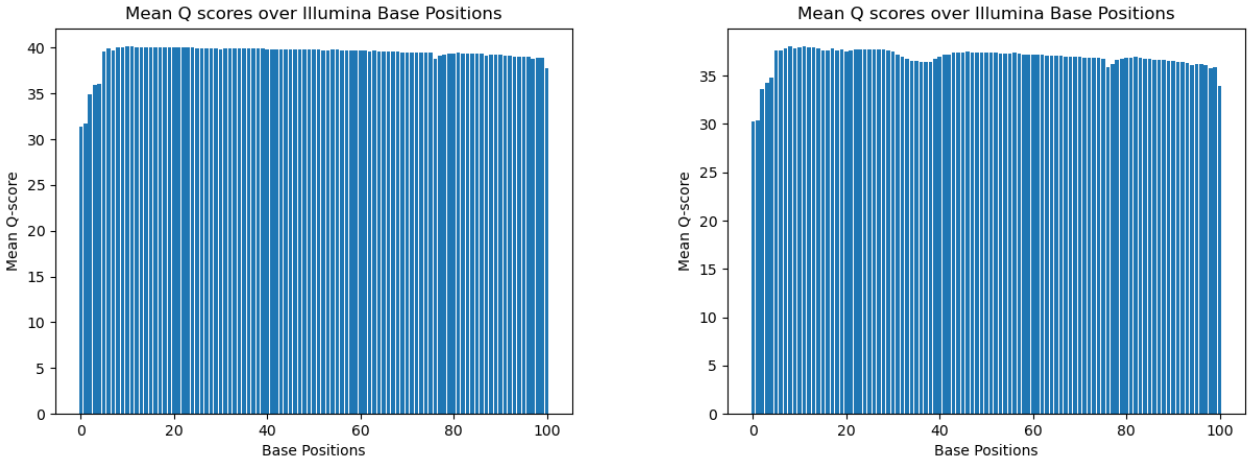

Figure 2: Quality score distributions for 1-2A-control-S1-L008. R2 has lower quality overall, as the scale changes and the mean quality scores are lower. Left is R1, right is R2

FastQC Output: Mean Quality Score Distributions for 17_3E_fox_S13_L008.

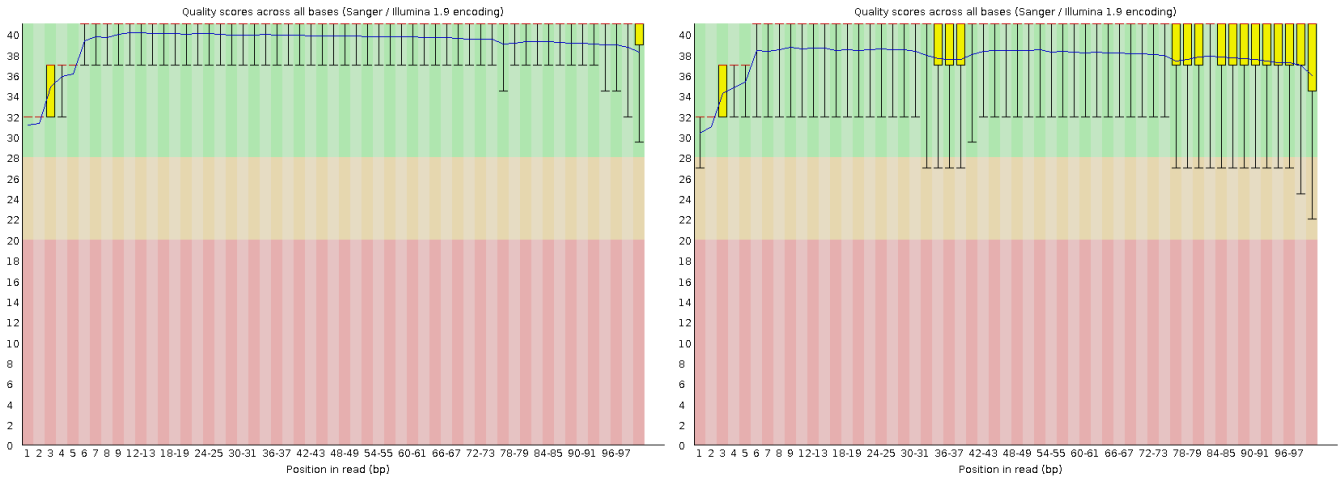

Figure 3: Quality score distributions for 17-3E-fox-S13-L008. The line represents the mean quality score for each bp, or position in the read. The error bars represent the distribution of scores found at that particular base. Green indicates an acceptable quality score. As you can see, R2 has a much wider distribution of scores than R1. The left graph is R1, the right graph is R2.

Python script Output: Mean Quality Score Distributions for 17_3E_fox_S13_L008.

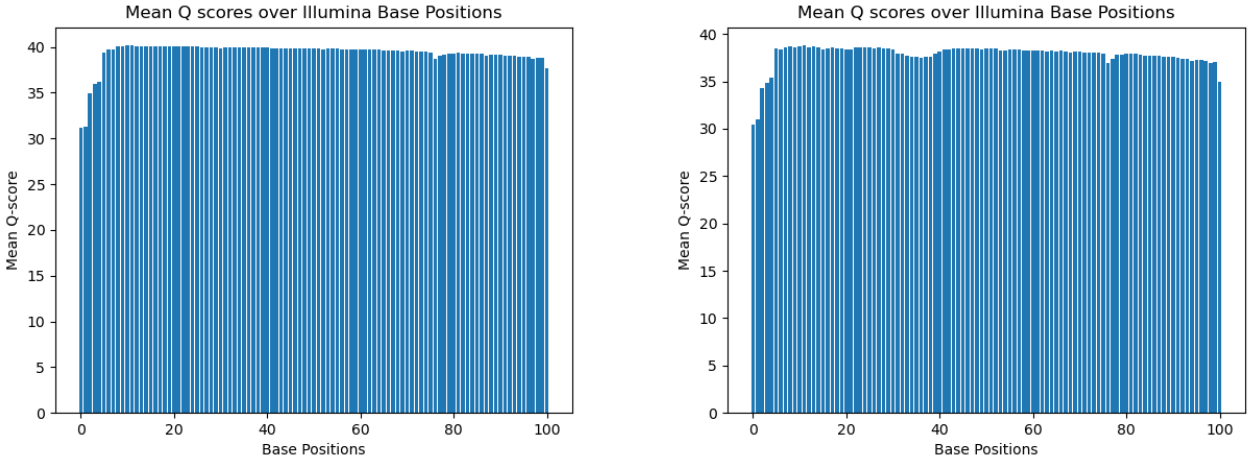

Figure 4: Quality score distributions for 17-3E-fox-S13-L008. R2 has lower quality overall, as the scale changes and the mean quality scores are lower. Left is R1, right is R2

The graphs from the FastQC output and the Python script seem to be very similar. The distributions match very well and follow the same trends.The mean quality score at each base position are at the same values for each graph. FastQC simply has more informative graphs with error bars and regions for good vs. bad quality scores.

The run times for the two were as follows:

FastQC for 1_2A_control_S1_L008_R1: Elapsed (wall clock) time (h:mm:ss or m:ss): 0:48.80
Python for 1_2A_control_S1_L008_R1: Elapsed (wall clock) time (h:mm:ss or m:ss): 3:24.76

FastQC for 1_2A_control_S1_L008_R2: Elapsed (wall clock) time (h:mm:ss or m:ss): 0:49.51
Python for 1_2A_control_S1_L008_R2: Elapsed (wall clock) time (h:mm:ss or m:ss): 3:13.29

FastQC for 17_3E_fox_S13_L008_R1: Elapsed (wall clock) time (h:mm:ss or m:ss): 1:06.44
Python for 17_3E_fox_S13_l008_R1: Elapsed (wall clock) time (h:mm:ss or m:ss): 4:24.84

FastQC for 17_3E_fox_S13_L008_R2: Elapsed (wall clock) time (h:mm:ss or m:ss): 1:06.93
Python for 17_3E_fox_S13_l008_R2: Elapsed (wall clock) time (h:mm:ss or m:ss): 4:28.10

FastQC run times were much faster than my python script. This could be because my python script employs multiple loops. While I cannot confirm with the FastQC code, I can assume it is much more optimized.

FastQC Output: Per-base N content of 1_2A_control_S1_L008

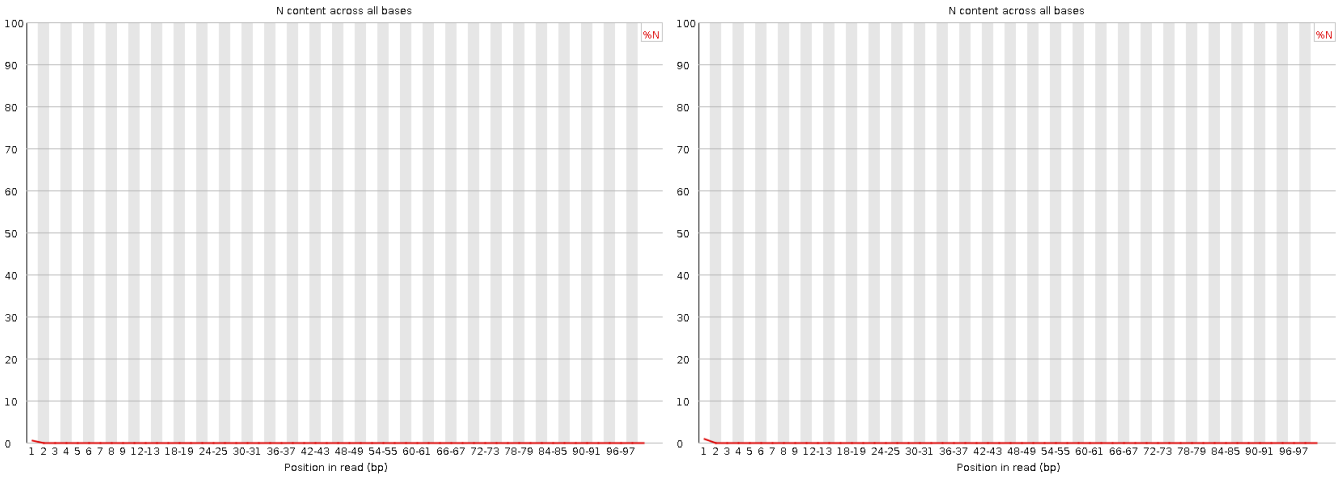

Figure 5: N content distributions for 1-2A-control-S1-L008. The line indicates how many N base calls were made at a specific position. The left graph is R1, the right graph is R2.

FastQC Output: Per-base N content for 17_3E_fox_S13_L008.

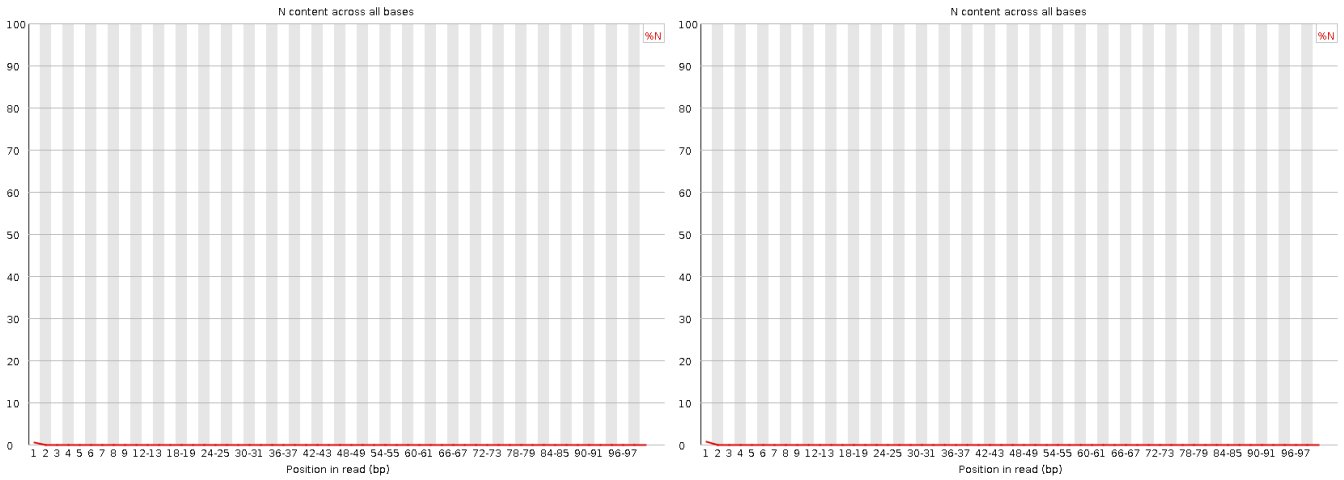

Figure 6: N content distributions for 17-3E-fox-S13-l008. The line indicates how many N base calls were made at a specific position. The left graph is R1, the right graph is R2.

The quality of the data seems to be high. There is a low number of "N" content across the base positions, meaning that most base assignments were not ambiguous.This supports the findings from the mean quality score graphs. The mean quality score was above 30 which is a typical cutoff for Illumina quality.

Part 2: Adaptor Trimming Comparison

Example command to look at adapters

Below is two example command used to estimate the number of adapters found in each file. Refer to lab notebook for the additional commands:

zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/1_2A_control_S1_L008_R1_001.fastq.gz | grep "AGATCGGAAGAGCACACGTCTGAACTCCAGTCA" | wc -l
zcat /projects/bgmp/shared/2017_sequencing/demultiplexed/1_2A_control_S1_L008_R1_001.fastq.gz | grep "AGATCGGAA GAGCGTCGTGTAGGGAAAGAGTGT" | wc -l

Each adapter was found in each file in the correct orientation and absent in the second file where the orientation was incorrect. This confirms that these are the correct adapters.

Cutadapt Output

1_2A_control_S1_L008: Read 1 with adapter: 468,835 (5.5%) Read 2 with adapter: 537,308 (6.3%)

17_3E_fox_S13_L008: Read 1 with adapter: 1,024,588 (8.7%) Read 2 with adapter: 1,104,503 (9.4%)

R2 was adapter trimmed more than R1. This is expected because R2 is typically of lower quality than R1 and cutadapt will also remove primers, poly-A tails and other types of unwanted sequence.

Trimmed read length distributions

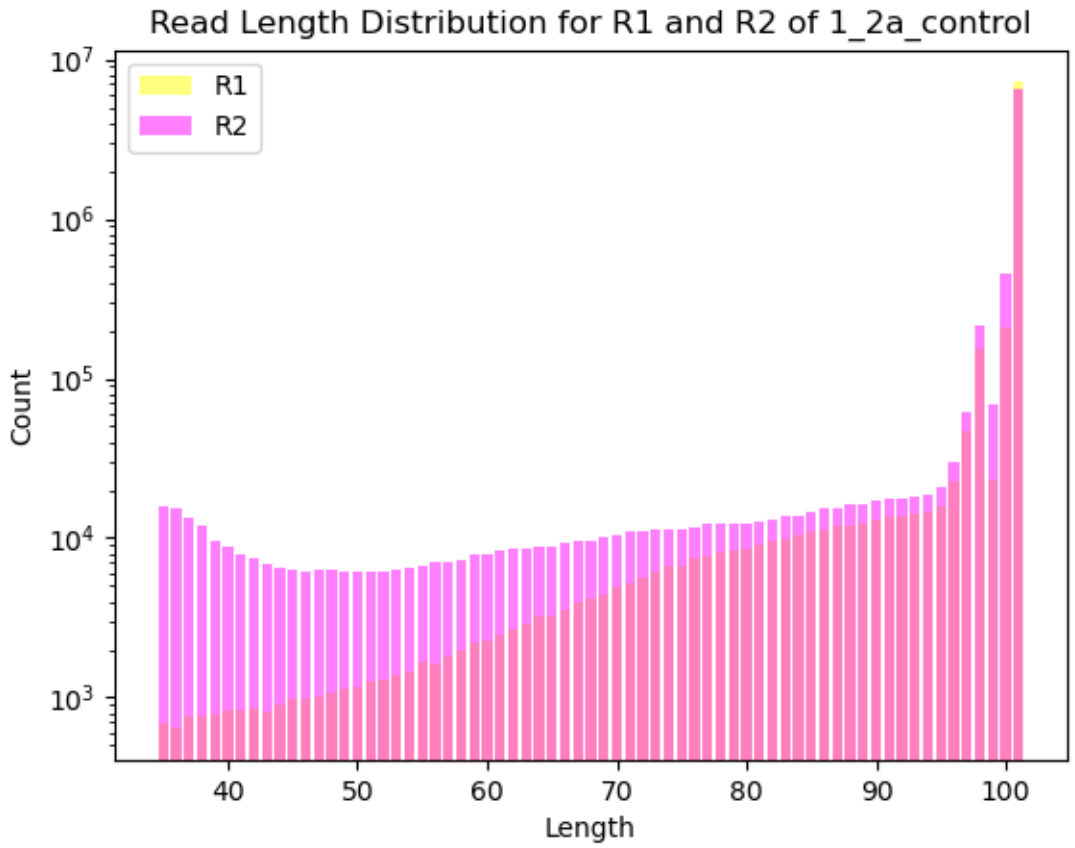

Figure 7: Trimmed read length distribution for 1-2A-control-S1-L008. This shows the difference between R1 and R2, labeled and yellow and pink respectively in the graph. The darker pink color shows where the two overlap. R2 has more shorter reads than R1.

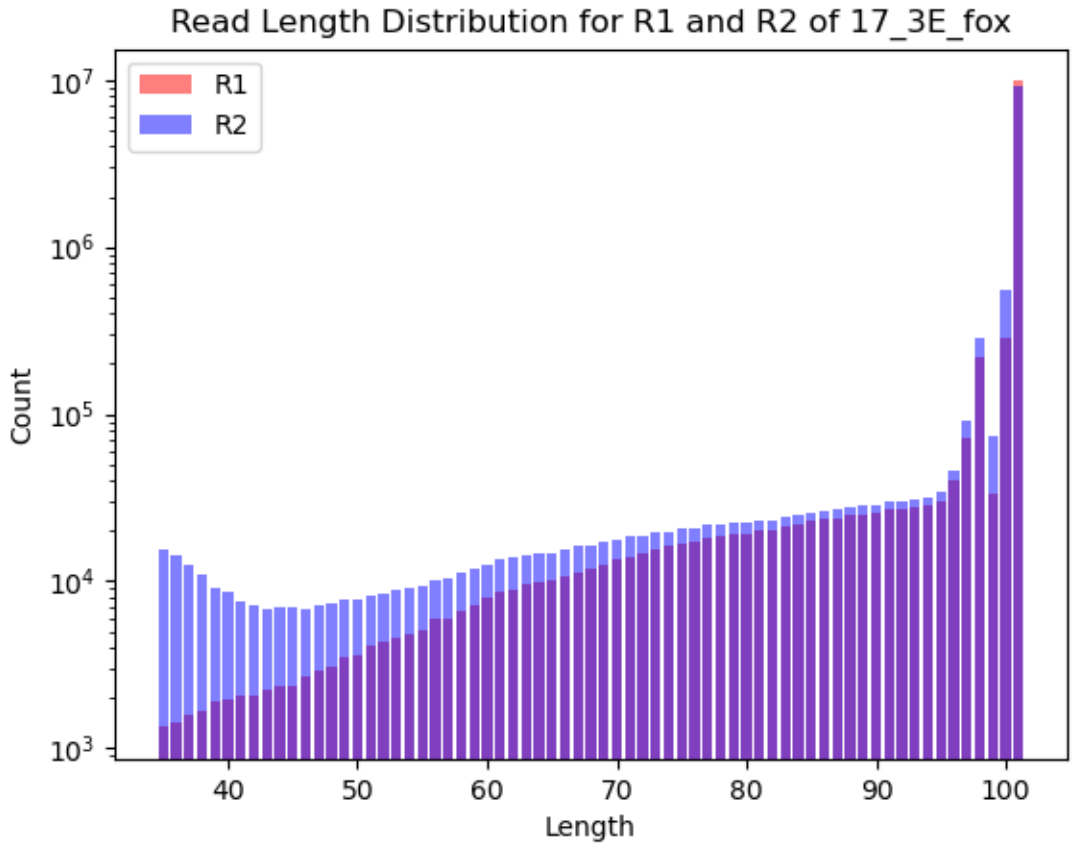

Figure 8: Trimmed read length distribution for 17-3E-fox-S13-L008. This shows the difference between R1 and R2, labeled as red and blue respectively in the plot. The darker color represents where the two overlap.

The plot overlays show that R2 was trimmed more extensively than R1 for both samples. This is expected because R2 is typically lower quality than R1; one reason could be that R2 is typically held on the sequencer longer.

Part 3: Alignment and strand-specificity

Mapped v. unmapped in SAM alignment files using BASH commands

Table 1. Counts of reads that were mapped vs unmapped. Obtained using BASH commands.

Sample	Mapped	Unmapped	Proportion Mapped
1_2a_control_S1_1008	15,627,427	305,259	98.1%
17_3E_fox_S13_1008	21,352,803	948,729	95.8%

Forward vs. reverse mapped reads in htseq-count

Table 2. Counts of forward and reverse reads that mapped to gene features via htseq-count. Obtained using BASH commands.

Sample	R1, mapped	R2, mapped	Total, R1	Total, R2	Proportion of Mapped Reads, R1	Proportion of Mapped Reads, R2
1_2a_control_S1_1008	306,397	6,876,756	7,966,343	7,966,343	3.8%	86.3%
17_3E_fox_S13_1008	436,515	8,952,040	11,240,766	11,240,766	3.9%	79.6%

In order to determine whether the reads were strand-specific, you can look at the proportion of mapped reads for each strand. For unstranded libraries because you don't know where the feature is coming from the proportions should be split evenly. In this case, there is a much higher proportion on the reverse reads.I propose that these data are strand-specific, because >79% of the reads are R2, as opposed to R1, for both samples. You can therefore infer that this is the strand that was not degraded during library preparations and this is a strand specific assay.