# Bi623 RNA-Seq QAA

Shayal Pratap

2024-09-08

**RNA_Seq QAA**

## Part 1 – Read quality score distributions

The goal of this assignment was to perform initial quality control on two RNA-seq samples (**15_3C_mbnl_S11_L008 & 24_4A_control_S18_L008**) prepared by the 2017 BGMP cohort (details for the experimental setup can be found here. The provided fastq were demultiplexed prior to quality assessment. For the purpose of this report, the samples will be referred to as **S11** and **S18**.

Table 1. Summary metrics resulting from inital file exploration

| File Name | Total Num. Reads | Phred encoding | File Size (M) | Read Length |
|---|---|---|---|---|
| 15_3C_mbnl_S11_L008_R1_001.fastq.gz | 7,806,403 | phred+33 | 407 | 101 |
| 15_3C_mbnl_S11_L008_R2_001.fastq.gz | 7,806,403 | phred+33 | 465 | 101 |
| 24_4A_control_S18_L008_R1_001.fastq.gz | 10,515,874 | phred+33 | 578 | 101 |
| 24_4A_control_S18_L008_R2_001.fastq.gz | 10,515,874 | phred+33 | 595 | 101 |

We're using FastQC for initial quality checking of raw sequencing data. This program takes in a file (.sam/.bam or .fastq) and outputs an HTML file reporting the resuts from its analyses. Some of the plots that are reported are distributions of the per-base average quality score (Fig. 1-4), the per-base N content (Fig. 5), and per tile plots (Fig. 6).

A B

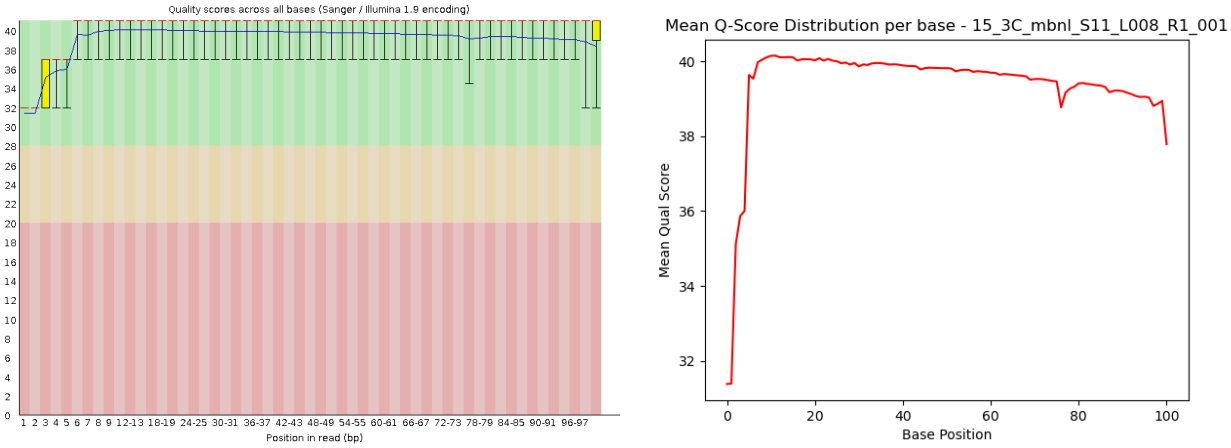


Fig. 1 Comparison of S11 R1 Per Base average quality score plot distrubtion generated by (A) FastQC and (B) Python script from Bi622 (Demultiplex Assignment).

A B

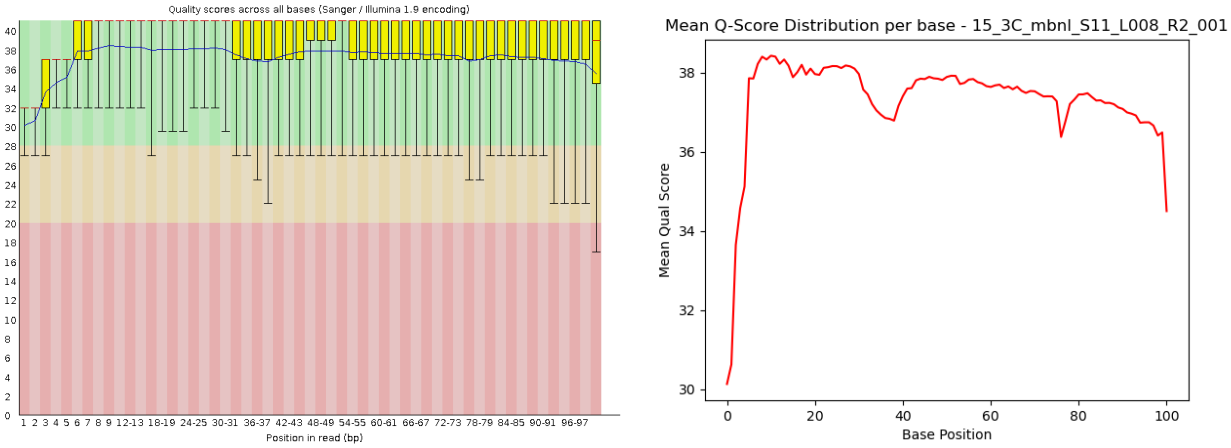


Fig. 2 Comparison of S11 R2 Per Base average quality score plot distrubtion generated by (A) FastQC and (B) Python script from Bi622 (Demultiplex Assignment).

A B

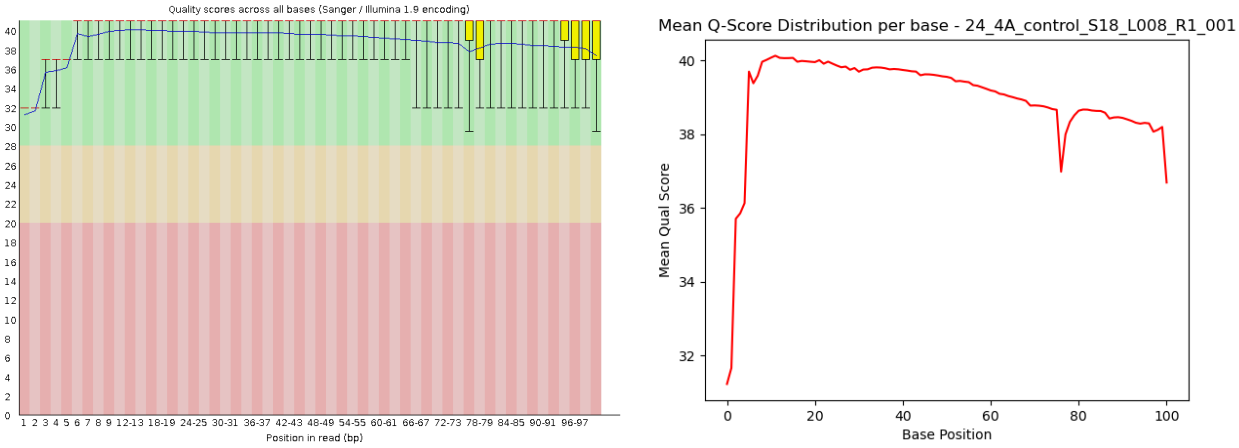


Fig. 3 Comparison of S18 R1 Per Base average quality score plot distrubtion generated by (A) FastQC and (B) Python script from Bi622 (Demultiplex Assignment).

A B

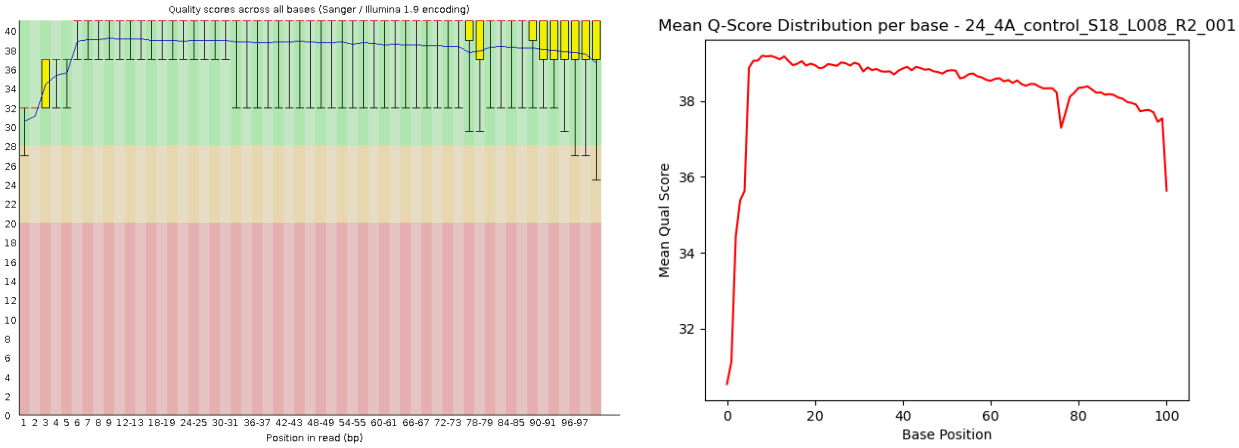


Fig. 4 Comparison of S18 R2 Per Base average quality score plot distrubtion generated by (A) FastQC and (B) Python script from Bi622 (Demultiplex Assignment).

While both plots illustrate a similar overall trend in average quality score per base, the ones generated by FastQC provide more information. By breaking up the data into quartiles, the FastQC plot makes it easy on the operator/interpreter to quickly gauge where the data falls which could be used to estabilish a q-score cutoff. The y-axis for both plots have the same range and scale however for the x-axis, the FastQC plots clustered their base positions in bins which could lead to a loss of data. Similarly, it's important to note that

the Python script from the Bi622 Demultiplex assignment was written with a week whereas FastQC was developed by seasoned experts and is a maintained program.

In regard to runtime, the FastQC was able to output multiple plots and metrics for four fastq files within 2.63 minutes. The plots generated from the Bi622 code used two scripts - one to generate a .tsv file reporting the base position and average qual score at that position and a second to plot the .tsv file. This process wasn't submitted as a batch script so runtime and CPU storage wasn't recorded but with the touch-points required, it definitely was not fasther than the FastQC pipeline.

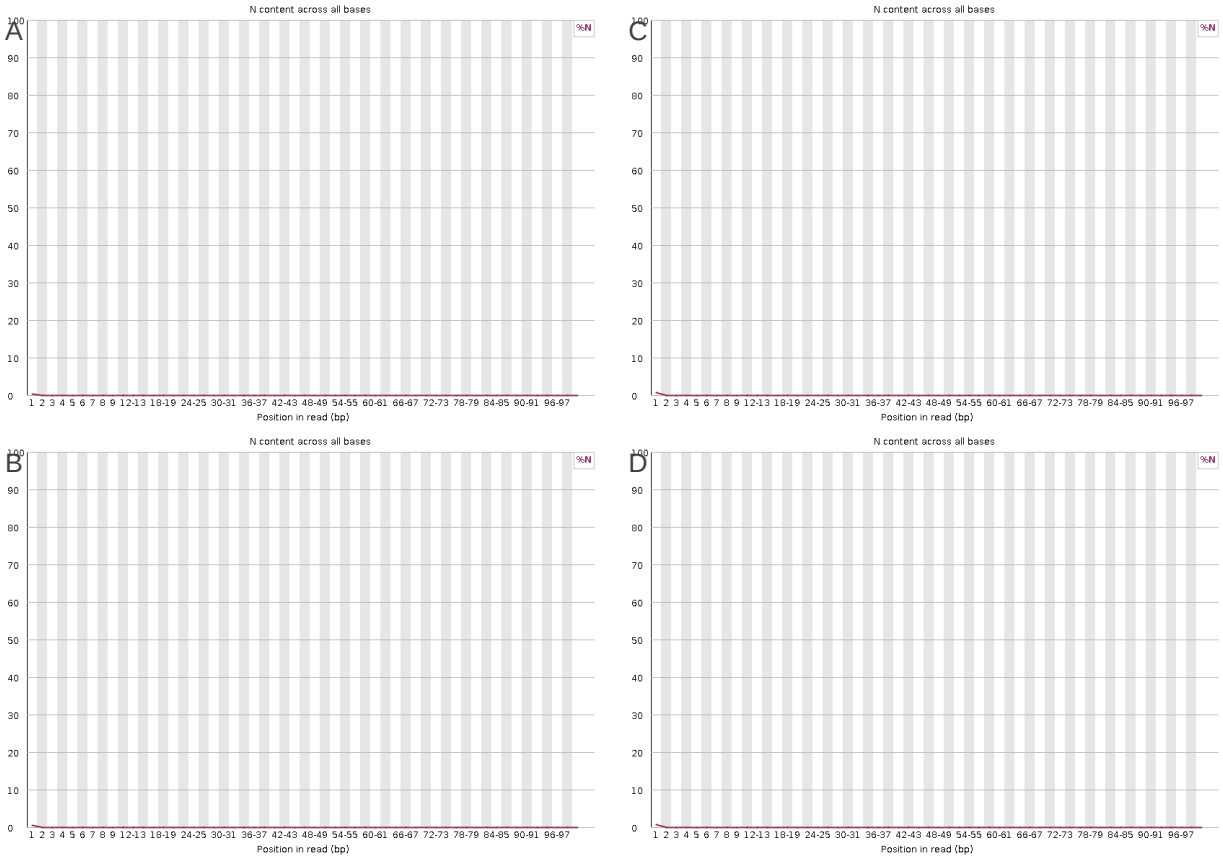


Fig. 5 Per Base N content plot for (A) S11 R1, (B) S11 R2, (C) S18 R1, (D) S18 R2

A base value of N is substituted by the sequencer if it's not able to make a call with enough confidence. The Per Base N content plots (Fig. 5) the percentage of base calls at each position where an N was called. Based on the mean quality score distribution (Plot A from Fig. 1-4), where most average qscores are above a value of 32, the N content plots are consistent with those results.

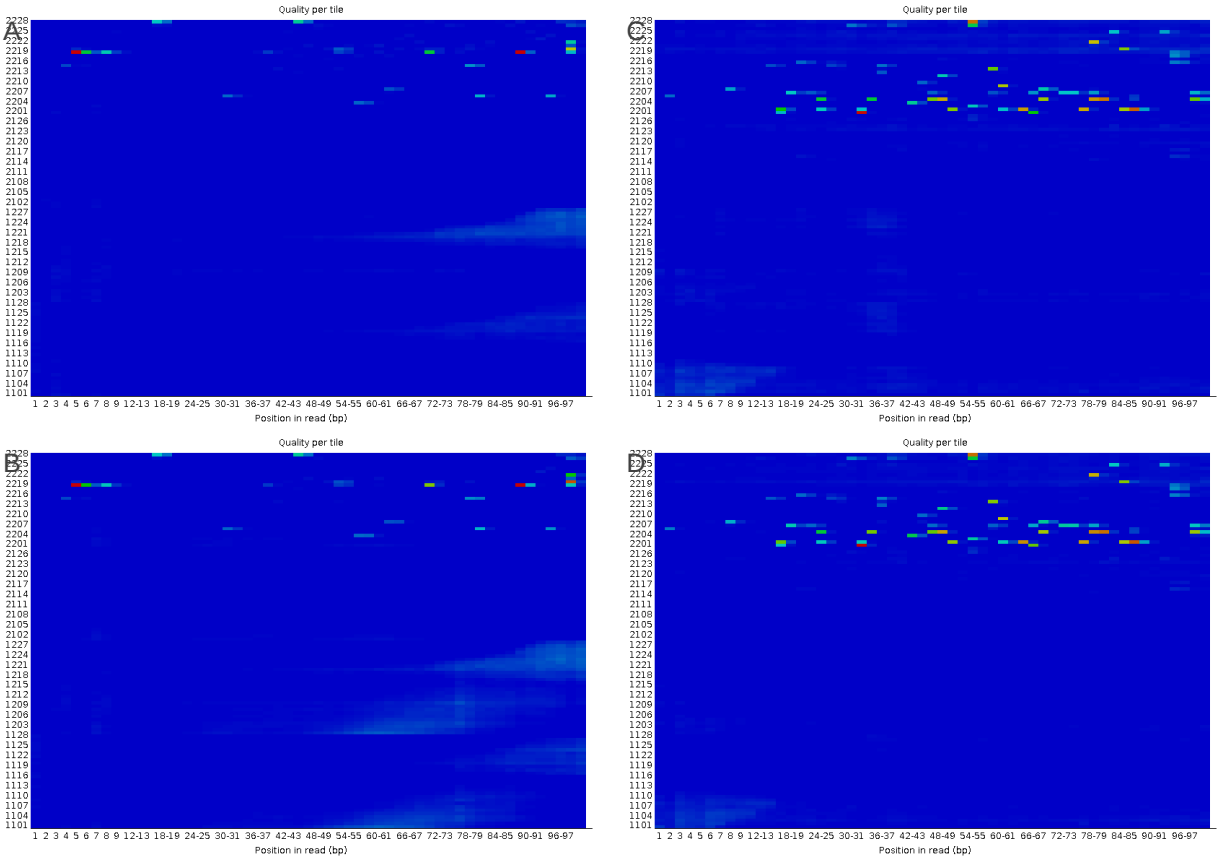


Fig. 6 Per-tile sequence quality distribution for (A) S11 R1, (B) S11 R2, (C) S18 R1, (D) S18 R2

Table 2. Summary of FastQC output per file

| | S11_R1 | S11_R2 | S18_R1 | S18_R2 |
|---|---|---|---|---|
| Base Statistics | Pass | Pass | Pass | Pass |
| Per base sequence quality | Pass | Pass | Pass | Pass |
| Per tile sequence quality | **Fail** | Warning | **Fail** | **Fail** |
| Per sequence quality scores | Pass | Pass | Pass | Pass |
| Per base sequence content | **Fail** | Warning | Warning | Warning |
| Per sequence GC content | Pass | Warning | Pass | Pass |
| Per base N content | Pass | Pass | Pass | Pass |
| Sequence Length Distribution | Pass | Pass | Pass | Pass |
| Sequence Duplication Levels | Warning | Warning | Warning | Warning |
| Overrpresented sequences | Pass | Pass | Pass | Pass |
| Adapter Content | Pass | Pass | Pass | Pass |

Both libraries have sequences lengths of 101 bp, which is the expected size of standard Illumina library, and all average quality scores per position are above 36. Similarly, the per base N content is low and the per-tile quality distribution (Fig. 6) is mostly blue indicating high-quality. Looking at the FastQC summary (Table 2), a majority of the metrics resulted in "Pass." Overall, the data is of high enough quality to use for further analysis.

# Part 2 – Adaptor trimming comparison

Per the metadata, the libraries were prepped with the KAPA Stranded mRNA-Seq Kit which uses Illumina TruSeq adapters:

**Adapter Sequences**
**Read 1**: AGATCGGAAGAGCACACGTCTGAACTCCAGTCA
**Read 2**: AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT

Cutadapt and Trimmomatic were used to trim the adapter sequences

Table 3. Proportion of reads that were trimmed - CutAdapt Output

| S11 | Num. Reads | % total reads |
|---|---|---|
| Total read pairs processed | 7,806,403 | N/A |
| Read 1 with adapter | 417,810 | 5.4% |
| Read 2 with adapter | 477,359 | 6.1& |

| S18 | Num. Reads | % total reads |
|---|---|---|
| Total read pairs processed | 10,515,874 | N/A |
| Read 1 with adapter | 335,742 | 3.2% |
| Read 2 with adapter | 417,709 | 4.0& |

**Visual confirmation of expected sequence orientations** The orientations were confirmed using the following commandline script:

```
zcat <filepath to fastq.gz> | grep --color=always "<adapter sequence>"|head
```

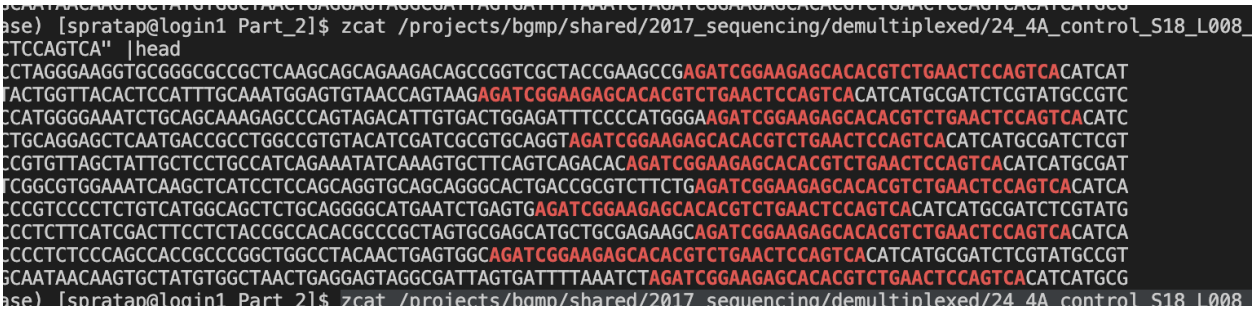


Figure 1: Adapter Orientation

Fig. 7 Bash command line output using S18 R1 fastq.gz and grepping for the Read 1 Adapter Sequence.

Figure 7 shows the adapter sequences are found closer to the 3' end of the read.

A B

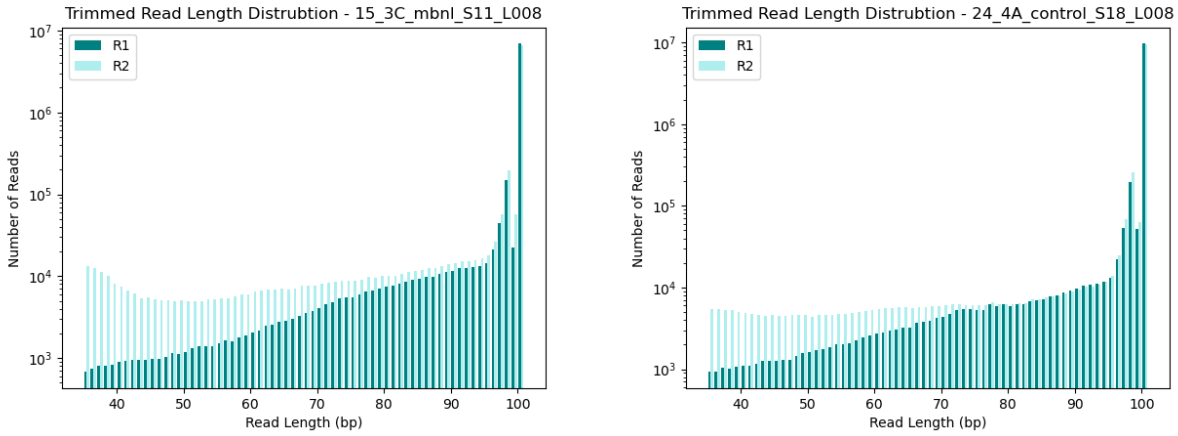


Fig. 8 Read length distribution of **trimmed** (A) S11 Reads 1 and 2 and (B) S18 Reads 1 and 2. Plots the number of reads (log base 10) against the read length.

Figure 8 shows that R2 for both samples was trimmed more extensively than R1 which is also corroborated by Table 3 where the % of reads trimmed with the Read 2 adapter is higher than the Read 1 adapter. This is expected because R2 is the last sequence to be read on the sequencer. At this point in the sequencing run, the samples and reagents have been on the sequencer for a long time. If the concentration of reagents starts to fluctuate at the end of the run or the DNA starts to degrade, it could impact sequencing quality. For example. less Mg++ leads to decreased polymerase activity which could cause the polymerase to displace and skip regions, which would truncated the insert and include more adapter sequence into the read.

## Part 3 – Alignment and strand-specificity

The Mus musculus (house mouse) genome was obtained from Ensemble (release 112) and the trimmed reads from S11 aqnd S18 were aligned it using STAR.

Table 4. Number of mapped and unmapped reads from SAM files (output from STAR aligner)

| | S11 | S18 |
|---|---|---|
| Num. Mapped Reads | 14,436,372 | 19,780,624 |
| Num. Unmapped Reads | 400,402 | 710,240 |

Table 5. Proportion of reads mapped to the Mus musculus genome

| Sample | Stranded = “ ” | Total Mapped Reads | Total Reads | % Mapped |
|---|---|---|---|---|
| S11 | yes | 267,375 | 7,806,403 | 3.42% |
| S11 | reverse | 6,164,597 | 7,806,403 | 79.0& |
| S18 | yes | 323,588 | 10,515,874 | 3.08% |

| Sample | Stranded = " " | Total Mapped Reads | Total Reads | % Mapped |
|---|---|---|---|---|
| S18 | reverse | 8,376,547 | 10,515,874 | 82.7% |

Strandedness refers to whether or not the directionality of the library molecule was retained. During library prep, when the second cDNA strand is being synthesized, a stranded prep will incorporated dideoxy nucleotides (dUTPs) that will later be targeted by an excision enzyme (like Endo VIII or USER). This ensures that the insert is the same sequence as the template/non-coding strand found in the genome. In an non-stranded library prep, the information regarding directionality is not retained. Given that ~80% of the reads mapped back to the genome when the stranded condition is set to "reverse" (Table 5), this suggests that the data is strand-specific.