QAA Report

2025-04-25

RNA-seq Quality Assessment Assignment - Bi 623 (Summer 2025 Assignment/PS 2)

Overall assignment:

In this assignment, you will process electric organ and/or skeletal muscle RNA-seq reads for a future differential gene expression analysis. We will be completing the differential gene expression analysis in our last bioinformatics assignment of this class. You will learn how to use existing tools for quality assessment and read trimming, compare quality assessments to those created by your own software, and how to align and count reads. Additionally, you will learn how to summarize important information in a high-level report. You should create a cohesive, well written report for your "PI" about what you've learned about/from your data.

Dataset:

Files SRR25630297 spots read : 58,398,524 reads read : 116,797,048 reads written : 116,797,048

SRR25630381
spots read : 6,776,454 reads read : 13,552,908 reads written : 13,552,908

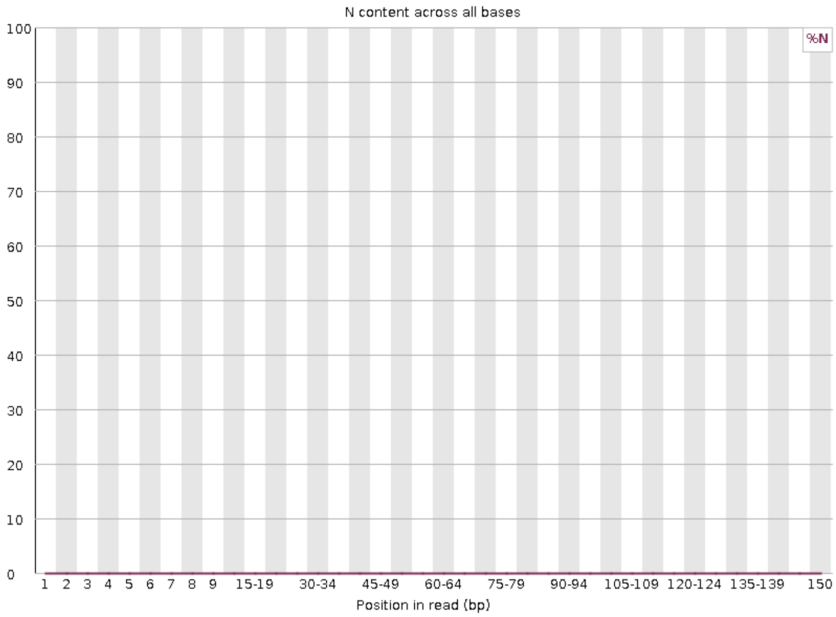
Part 1 – Read quality score distributions

Produce plots of the per-base N content, and comment on whether or not they are consistent with the quality score plots.

```
knitr::include_graphics("SRR25630297_1.fastq.png")
```

SRR25630297_1.fastq

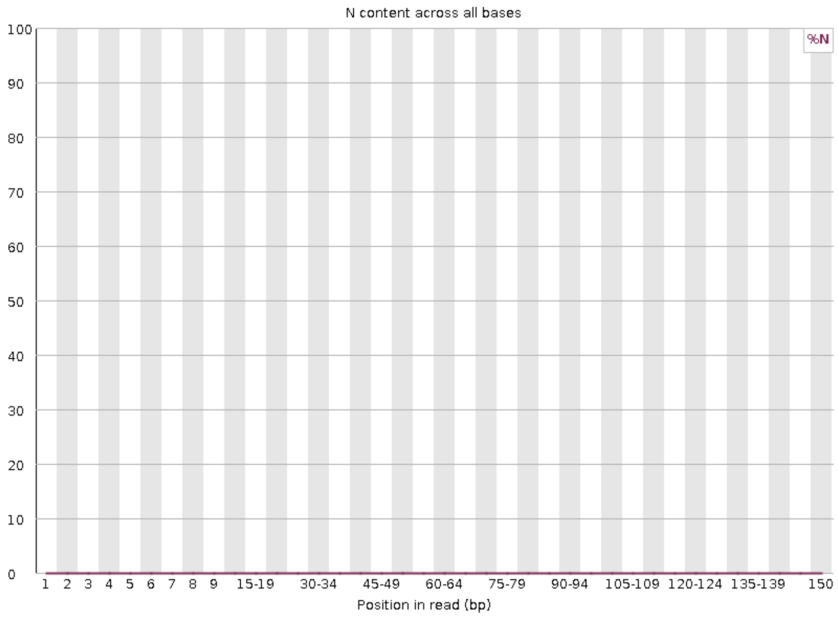
Per base N content



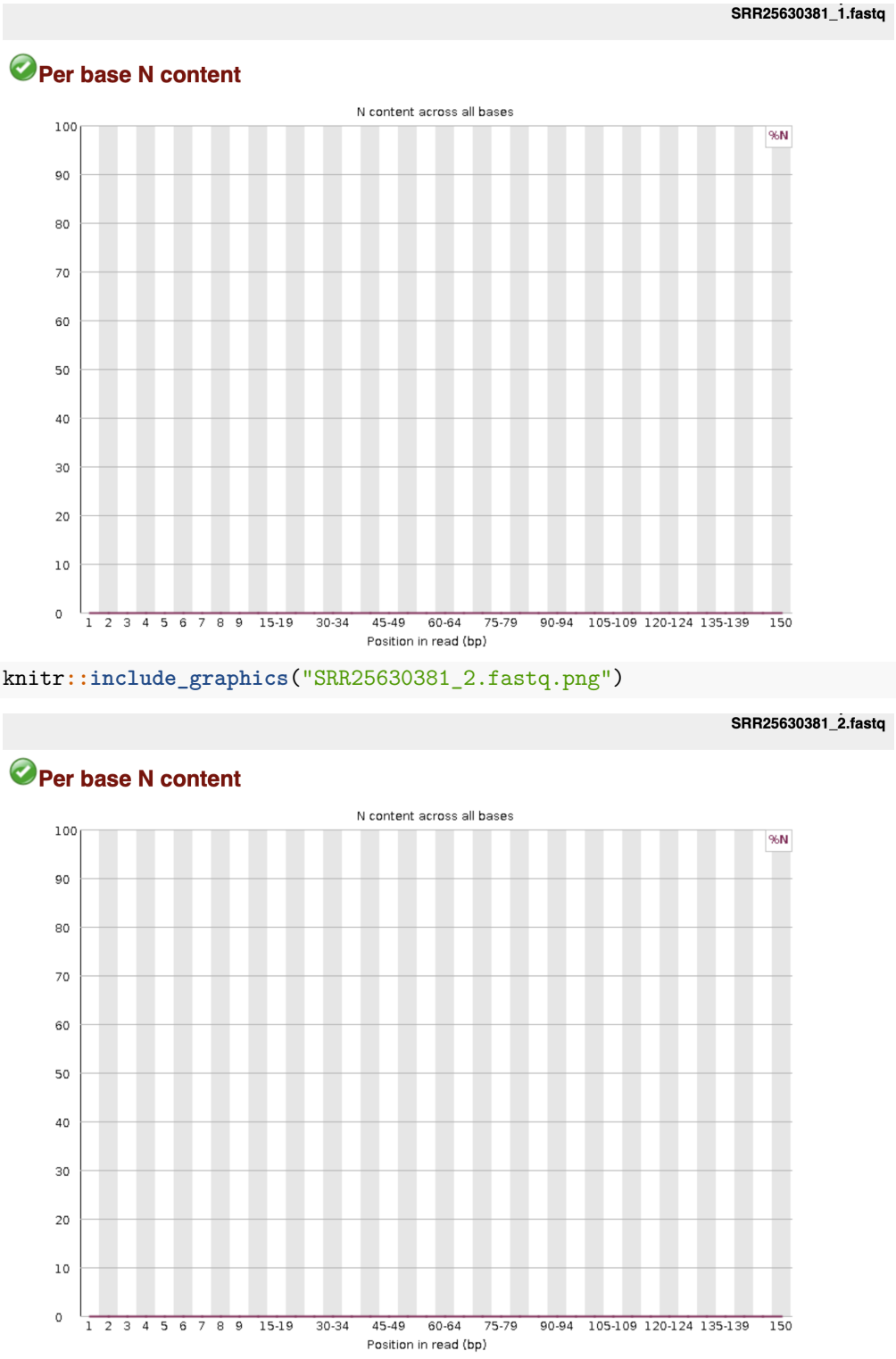

```
knitr::include_graphics("SRR25630297_2.fastq.png")
```

SRR25630297_2.fastq

Per base N content




```
knitr::include_graphics("SRR25630381_1.fastq.png")
```




```
knitr::include_graphics("SRR25630381_2.fastq.png")
```


Figure 1. Per-base quality score distributions from FastQC analysis showing the N content (%) across all bases at each position. An "N" results if the sequencer is not able to make a base call with confidence.

All of the reads from each file do not indicate any N reads at any position. The per sequence quality scores look the same between the 4 fastq files, with the average quality per read being being a Phred score of 36. The per base sequence quality looks similar, or of good quality (green), between SRR25630297_1.fastq.png, SRR25630297_2.fastq.png, SRR25630381_1.fastq.png, but SRR25630381_2.fastq.png but the last third of

the reads are of reasonable quality (orange) and last position in the reads of poor quality (red). Quality can degrade on runs towards the nd of a sequence so its not uncommon to see the quality decrease.

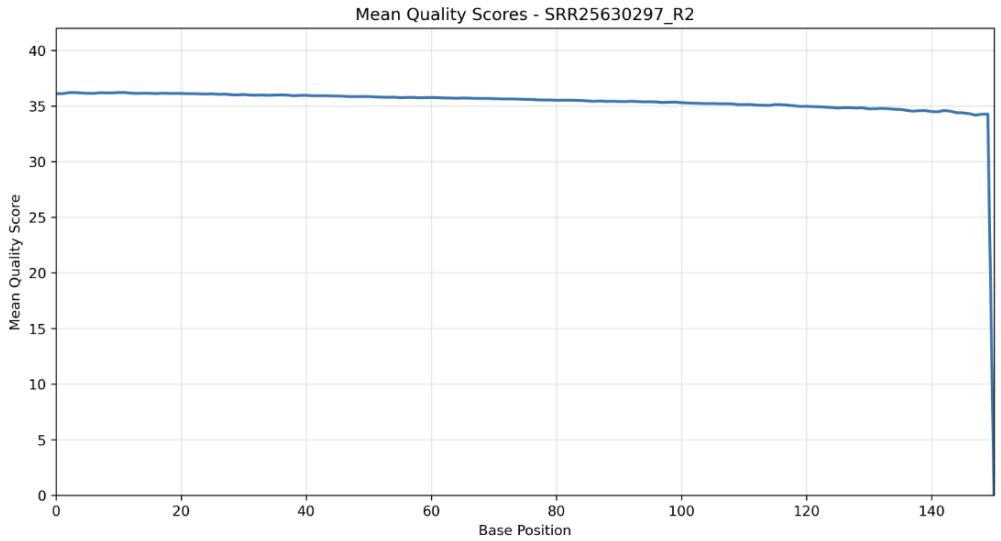
Describe how the `FastQC` quality score distribution plots compare to your own. If different, propose an explanation. Also, does the run time differ? Mem/CPU usage? If so, why?

The mean quality scores look the same between the fastqc and the script from the demultiplexing assignment, but the fastqc ran much quicker than the python demultiplexing script. The larger of the fastq files took 20 min to run with the demultiplexing script while the fastqc took less than 20 min to run all 4 files.

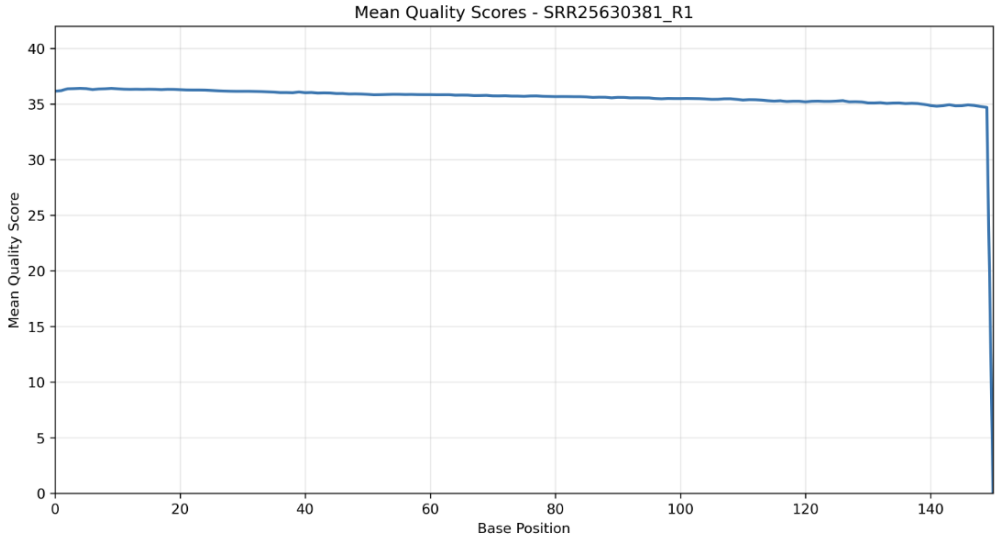
Timed results for SRR25630297_1.fastq Command being timed: "python qscore_dist.py -f SRR25630297_1.fastq -l SRR25630297_R1" User time (seconds): 1237.05 System time (seconds): 9.90 Percent of CPU this job got: 99% Elapsed (wall clock) time (h:mm:ss or m:ss): 20:55.60

```
knitr::include_graphics("SRR25630297_1.fastq_py.png")
```


```
knitr::include_graphics("SRR25630297_2.fastq_py.png")
```




```
knitr::include_graphics("SRR25630381_1.fastq_py.png")
```




```
knitr::include_graphics("SRR25630381_2.fastq_py.png")
```

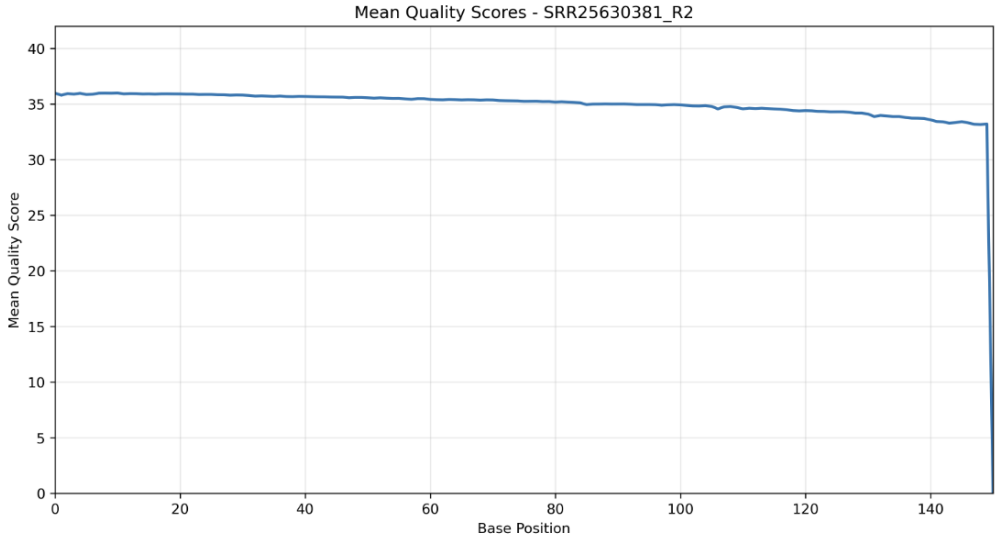


Figure 2: Per base mean quality score from the demultiplexing scripts for SRR SRR25630297 and SRR25630381 reads 1 and 2. Quality scores are in the Phred +33 encoding. The average quality of each read was 36, with a gradual decline around 100 bp and a drop close to the 150 bp. This is reflective of the quality drop exected from Illumina read sequencing.

Comment on the overall data quality of your two libraries. Go beyond per-base qscore distributions. Examine the `FastQC` documentation for guidance on interpreting results and planning next steps. Make and justify a recommendation on whether these data are of high enough quality to use for further analysis.

Looking at the per tile sequence quality for all the plots, which shows if there is a loss of quality associated with a part of the flow cell. If the plot is completely blue, that indicates a good quality plot, if there are warmer colors, that indicates poorer quality. SRR25630297 overall had a "good" looking plot. Read1 had a couple lines of orange, and Read2 was completely blue. SRR25630381 had streaks of warmer colors around the 125 to 150 bp range at 2109, 2160, and 2607. Read2 had more of the yellows as opposed to greens like Read2 indicating poorer quality from the flow cell in Read2. This could be caused by inhibitors from the library, incorrect insert size, or incorrect library quality or quantity. SRR25630381 Read2 shows some adapter content starting from reads 105 to 135, so there are Illumina Universal Adapters in 5-10% of the reads. The first SRR25630297 is good for further analysis. The SRR25630381 needs adaptor trimming, but can be used

for further analysis.

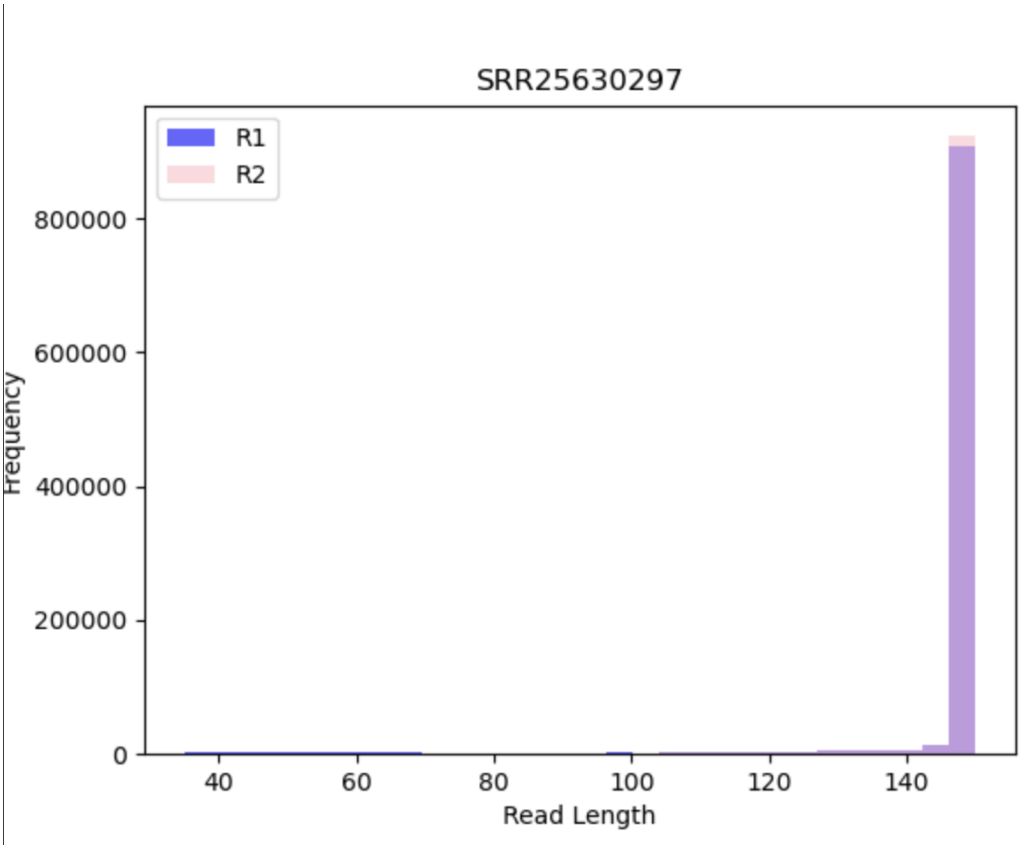
Part 2 – Adaptor trimming comparison

Cut adapt results: SRR25630297: 5.8% R1 and 6.4% R2 SRR25630381: 14.3% for both R1 and R2

Comment on whether you expect R1s and R2s to be adapter-trimmed at different rates and why.

The R1/R2 for both files show minimal differences on the adapter trimming. Read2 shows slightly more, but in general, R2 is expected to have a higher rate of adapter trimming compared to R1. Differences can be attributed to R2 being read after R1 on illumina, allowing the sample to degrade slighly and R2 reads can have lower quality scores towards the 3' ends.

```
knitr::include_graphics("SRR25630297.png")
```




```
knitr::include_graphics("SRR25630381.png")
```

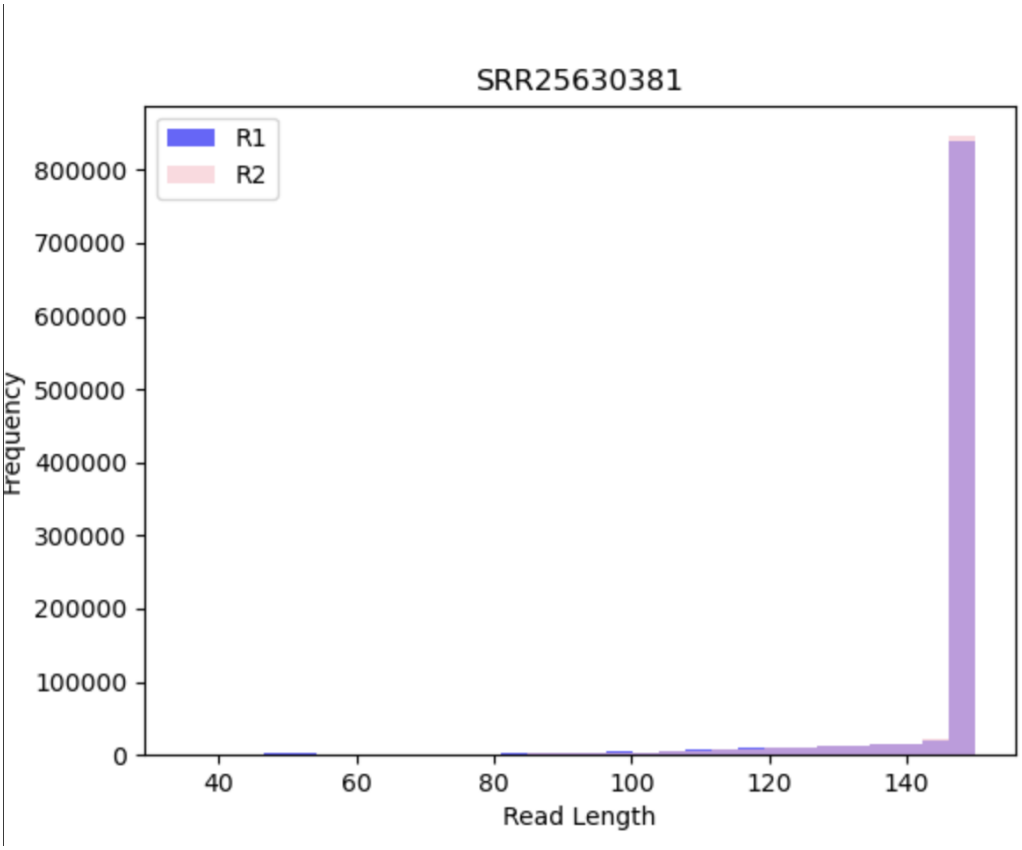


Figure 3: Read length after Trimmomatic processing for Read1 (blue) vs Read2 (pink) for both samples overlayed. Most reads remained at about 150 bp after the trimming, indicating minimal trimming across all the files.

Part 3 – Alignment and strand-specificity

SRR25630381 Mapped reads: 7580940 Unmapped reads: 751006

SRR25630297 Mapped reads: 36643440 Unmapped reads: 7255102

SRR25630297: 83.6% mapped (36,643,440/43,898,542) SRR25630381: 91.0% mapped (7,580,940/8,331,946)

```
sample <- c("SRR25630381", "SRR25630297")
mapped <- c(7580940, 36643440)
unmapped <- c(751006, 7255102)
total <- mapped + unmapped

mapped_unmapped_results <- data.frame(
  Sample = sample,
  Mapped_Reads = mapped,
  Unmapped_Reads = unmapped,
  Total_Reads = total)

knitr::kable(mapped_unmapped_results, format.args = list(big.mark = ","))
```

Sample	Mapped_Reads	Unmapped_Reads	Total_Reads
SRR25630381	7,580,940	751,006	8,331,946
SRR25630297	36,643,440	7,255,102	43,898,542

The reads are strand specific since the reverse strand specified reads are showing there are fewer unassigned reads compared to the forward strand parameter. The no_feature for the “yes” paramerter has 33667781 reads that are not assigned to any gene feautre but the “reverse” parameter shows 12501778 reads that are unassigned. This means the “reverse” is assigning reads to genes successfully.