RNA-seq Quality Assessment

Ryan Murnane

2025-09-07

This project is assessing the quality of two RNA-seq runs from the Sequence Read Archive. Here SRR25630312 and SRR25630410 are analyzed.

Assessing Data Quality

To first assess data quality, FastQC quality analysis is examined for each fastq read file from SRR25630312 and SRR25630410. The per base quality score plots are then compared here with Python generated plots to check for accuracy.

FastQC analysis

SRR25630312

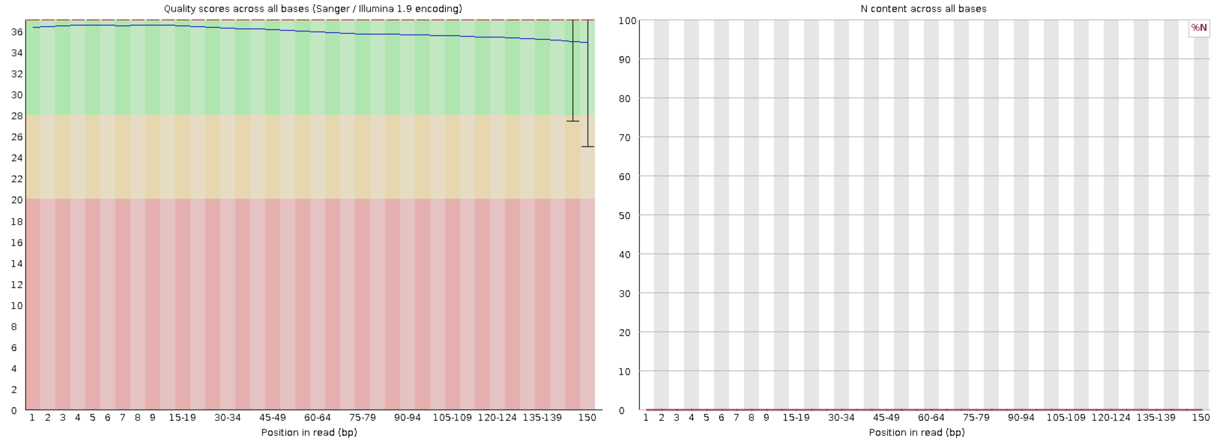

Figure 1: Per Base Quality Score and N content for SRR25630312 R1

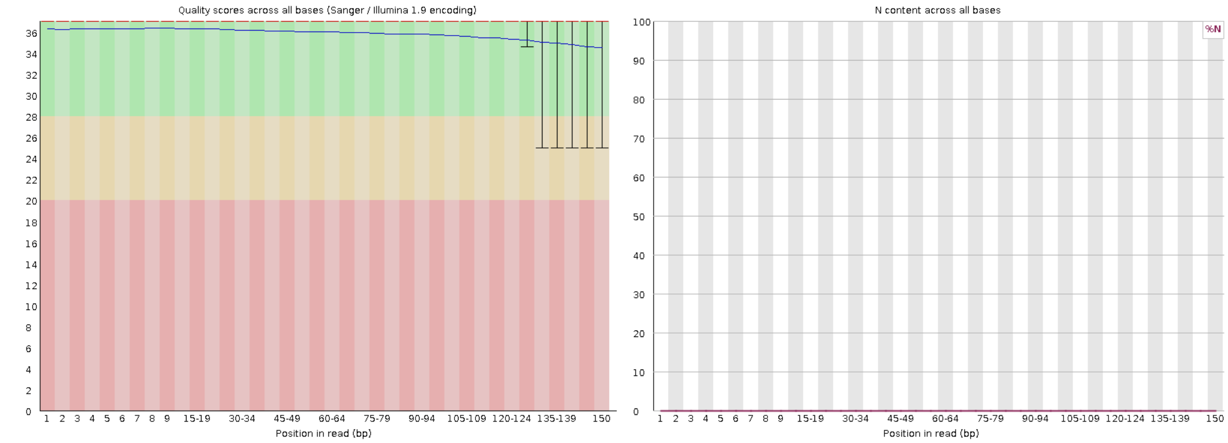

Figure 2: Per Base Quality Score and N content for SRR25630312 R2

SRR25630410

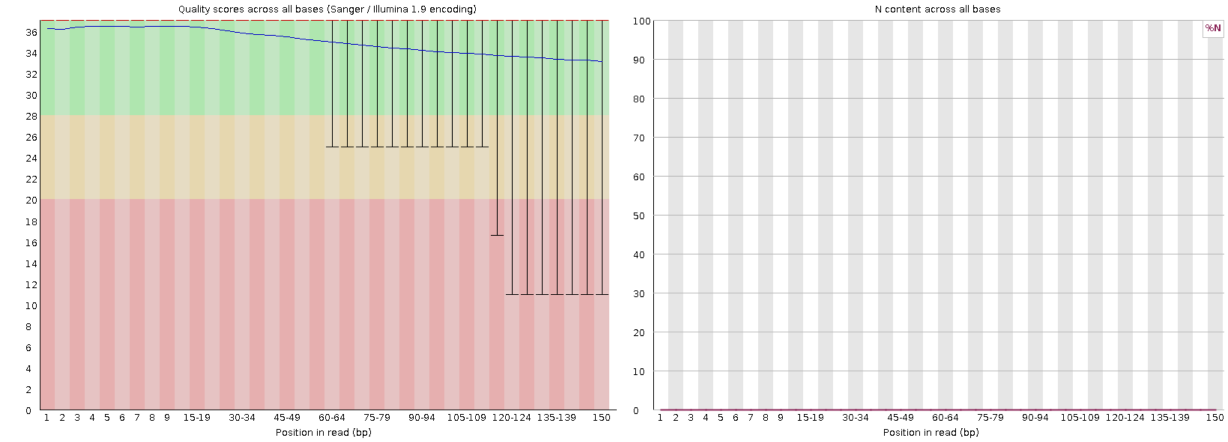

Figure 3: Per Base Quality Score and N content for SRR25630410 R1

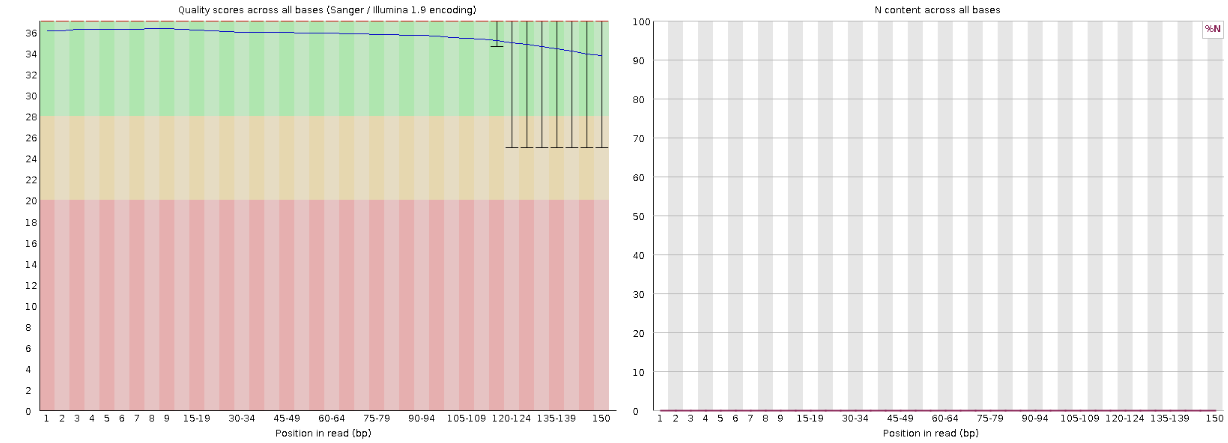

Figure 4: Per Base Quality Score and N content for SRR25630410 R1

Analysis

The quality scores are all high on a per base level. SRR25630410 R1 does seem to have slightly lower quality scores. The per base N scores are all around 0, matching the quality score plots which are all high quality per each base pair.

Python plots

For comparison here are quality score distributions created using a program built in Python.

SRR25630312:

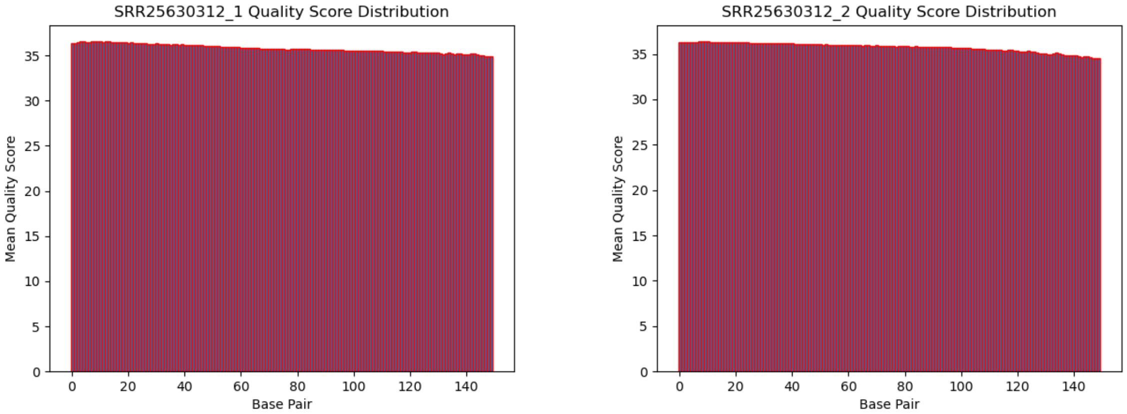

Figure 5: Python Generated Per Base Quality Scores for SRR25630312 R1 and R2

SRR25630410

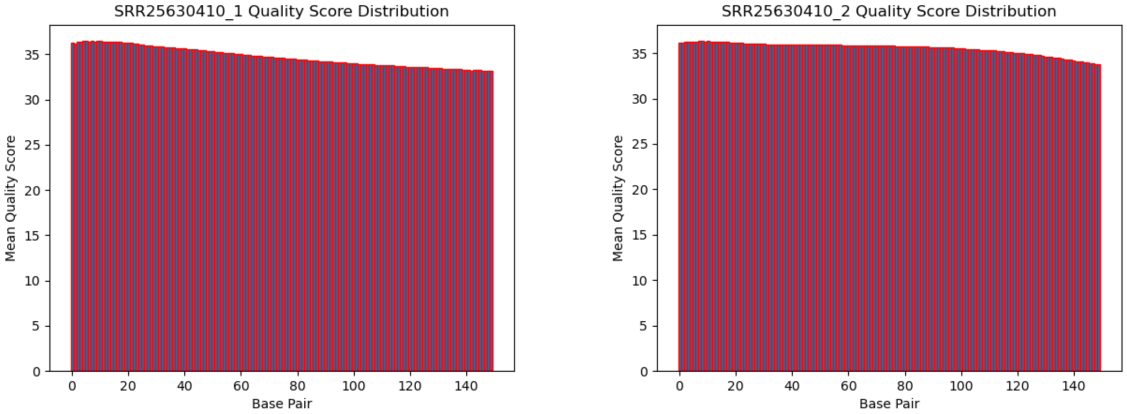

Figure 6: Python Generated Per Base Quality Scores for SRR25630410 R1 and R2

Results

The plots show the same means with the same trends per base pair between the reads for both the fastqc generated and Python generated plots. The fastqc plots also show error bars. The Python generated plots took way longer to generate as they took around 14 minutes each while the fastqc plots took around 4. The fastqc output also includes a multitude of other plots and information so is a lot more efficient in terms of both time and usefulness. The fastqc program is not written in Python and can therefore be faster.

Quality Assessment Takeaways

The data quality across all four files is good and ready for further analysis. The per base quality score distribution for each file is very good with the average quality per read peaking highly at 36 for each file. Accordingly, the per base N content is also nonexistent or very low for each file. The GC content for each file also seems to be normally distributed which is a good sign. All the sequence lengths look to be uniform at 150 bp for each file.

Read Cutting and Trimming Analysis

Here Cutadapt is used to cut adapters out of the reads, while Trimmomatic is used to quality trim them. The results are then analyzed.

Cutadapt

For cutadapt, here are the adapters present in the forward and reverse read fastq files: R1: `AGATCGGAAGAGCACACGTCTGAACTCCAGTCA` R2: `AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT`

Both are found at the 3' ends of their respective reads

Using cutadapt, the adapters get cut out of the reads.

Trimmomatic

Taking the outputs of the cutadapt, Trimmomatic is used to further quality trim the reads taking the various parameters specified.

Here the results are plotted, showing Read Length Distributions for each file.

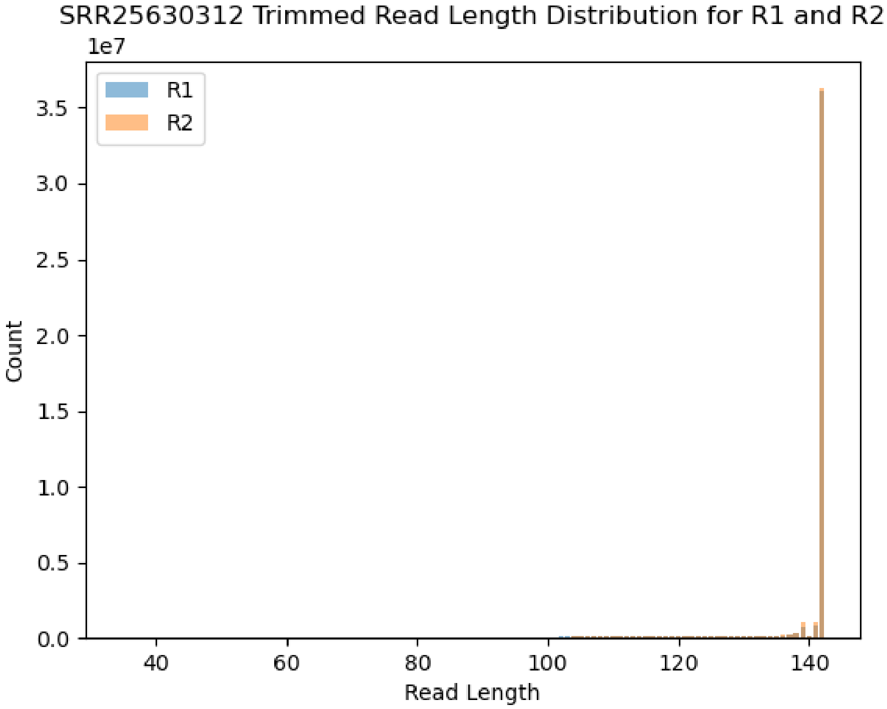

Figure 7: Trimmed Read Length Distributions for SRR25630312

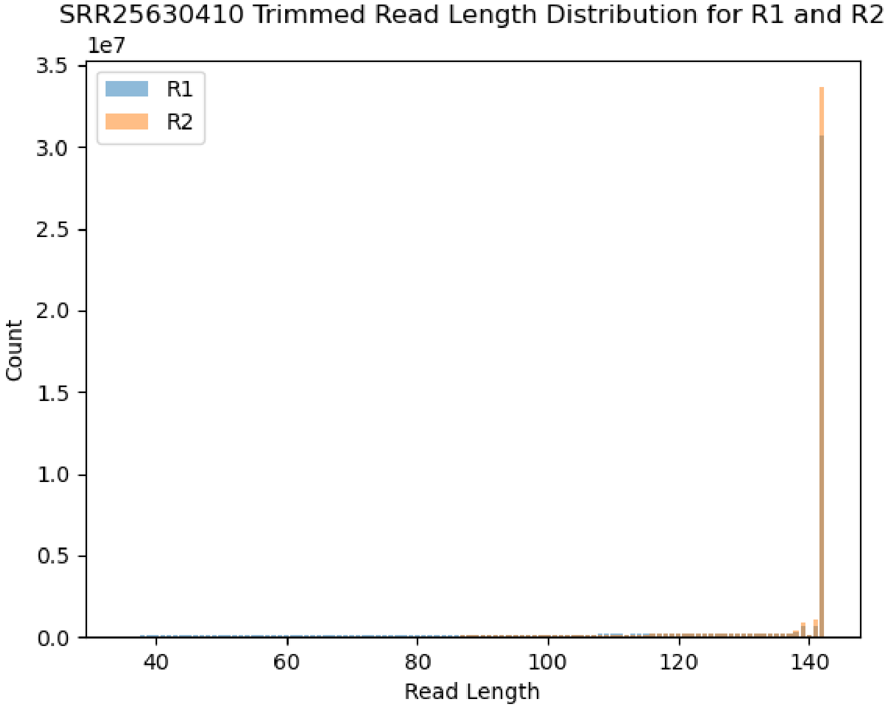

Figure 8: Trimmed Read Length Distributions for SRR25630410

Quality Trimming Results

From the data it seems that R1 is trimmed slightly more than R2 for SRR25630312, with R1 being trimmed significantly more for SRR25630410. In both cases then, R1 is being trimmed at a higher rate. This makes sense when taking the quality score assessment for each read file into account. For SRR25630312 the quality score distributions are very similar between R1 and R2, so they are both quality trimmed somewhat equally. The quality scores per base pair for R1 in SRR25630410 are notably lower than R2 and thus reads are trimmed to a further extent.

Alignment and Assessing Strand Specificity

Here STAR is used to create a *Campylomormyrus compressirostris* alignment database and align the trimmed reads. Picard is used to remove PCR duplicates and the read mapping is analyzed. HTSeq is used to count reads that are mapped to features and determine if the data is strand-specific.

Picard Results

After using Picard to remove duplicates, each SAM file is checked for mapped and unmapped reads.

Table 1: SAM File Read Counts After Picard

File	Mapped	Unmapped
SRR25630312	37341592	16674535
SRR25630410	29216546	29857662

HTSeq Results

Here are the results from the HTSeq runs. Each run is ran with stranded set to yes or reverse to see what is best to use moving forward.

Table 2: Reads Mapped to Features

File	Counts
SRR25630312_stranded	593376
SRR25630312_reverse	10877058
SRR25630410_stranded	483234
SRR25630410_reverse	8607298

Based on these counts, the data used is from reverse strand RNA-Seq libraries. This is because the read counts using reversed are way higher than specifying yes for stranded. The stranded parameter should be set to reversed moving forward.