

# RNA-seq Quality Assessment Assignment

Gera

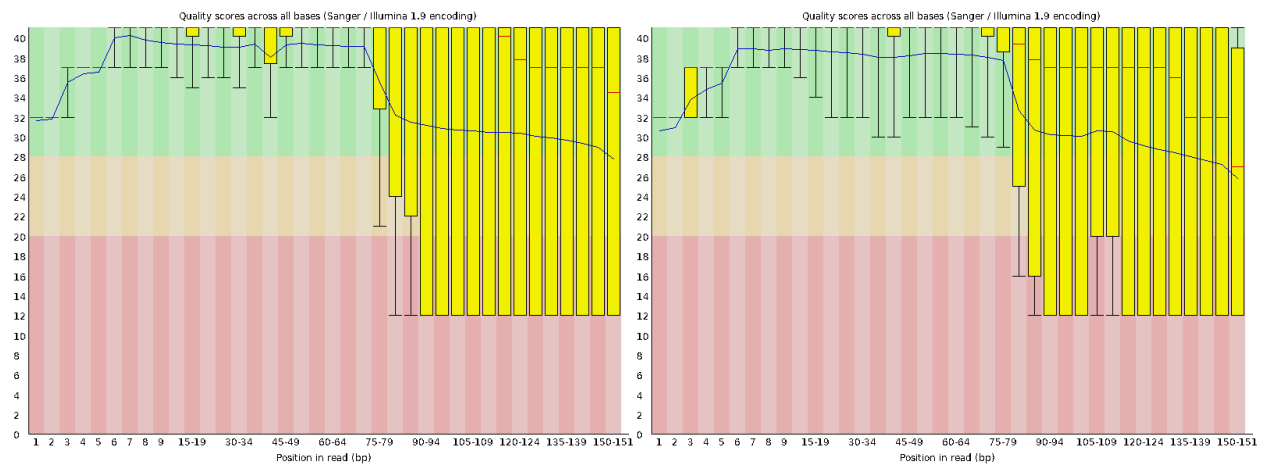
10/12/2019

## Bi 624 (Fall 2019) – Problem Set 1

1. Using FastQC on Talapas, produce plots of quality score distributions for forward and reverse reads. Also, produce plots of the per-base N content

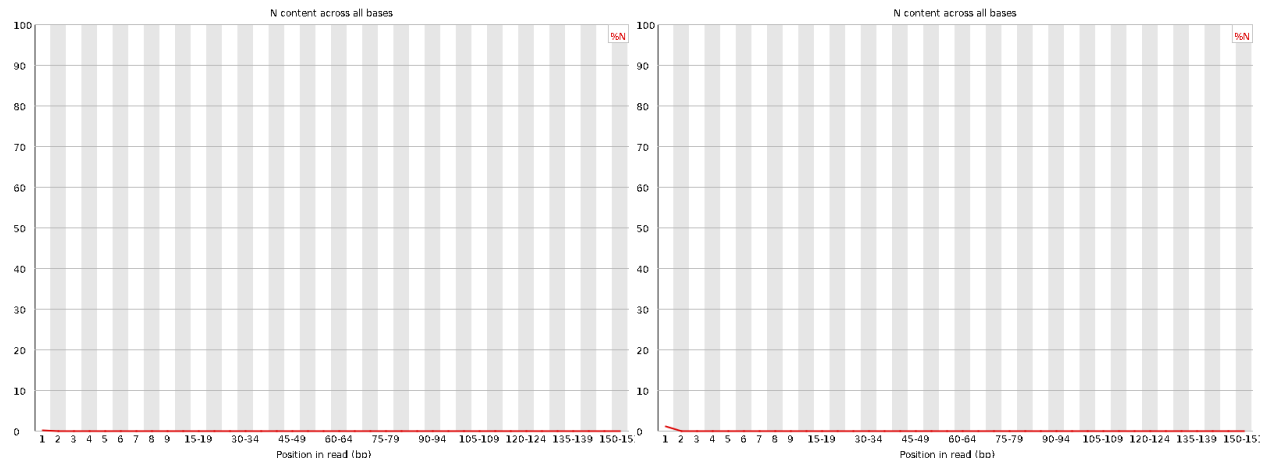
Figure 1: RAL306W-F3\_S79\_L006 fastqc

### a. per base quality



Fastqc Per base sequence quality distributions output png files. Left is RAL306W-F3\_S79\_L006\_R1\_001. Right is RAL306-F3\_S79\_L006\_R2\_001

### b. Per base n content



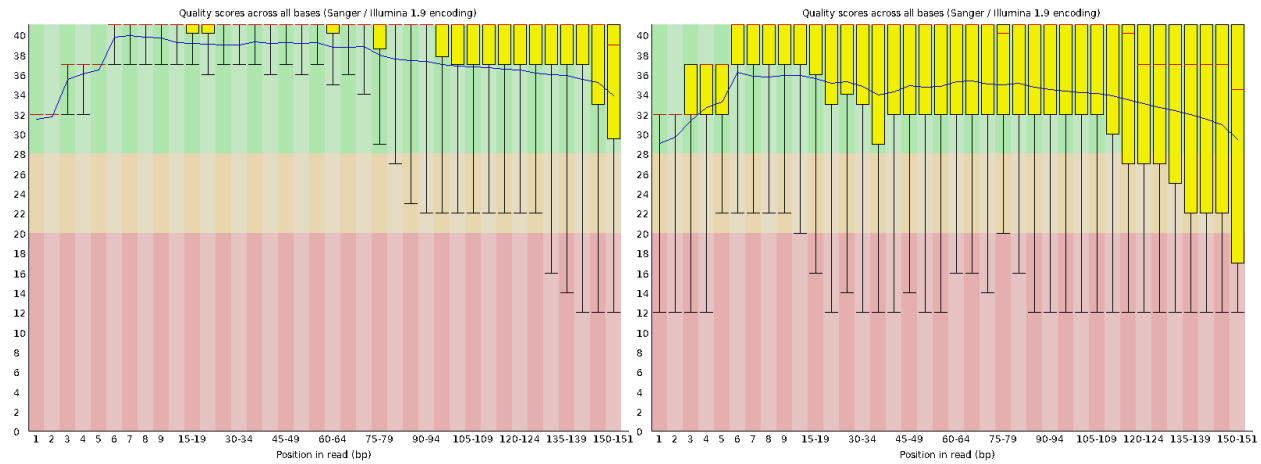
Fastqc Per base n content output png files. Left is RAL306W-F3\_S79\_L006\_R1\_001. Right is RAL306-F3\_S79\_L006\_R2\_001

and comment on whether or not they are consistent with the quality score plots.

The N cotent across the bases was consistent in which there was no N bases. The base quality scores were good, specifically 1 to 75-79 in RAL306W-F3\_S79\_L006 reads.

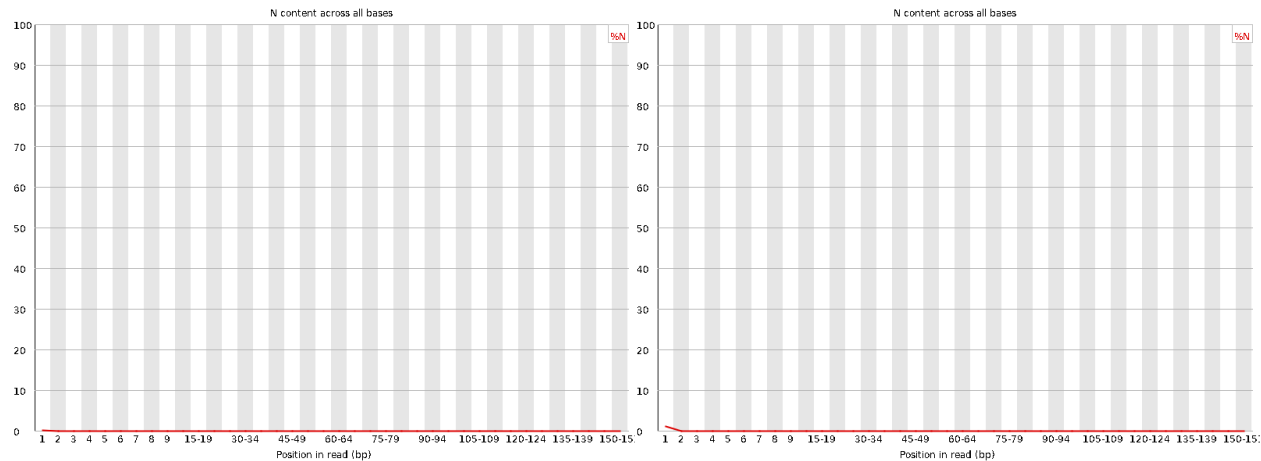
**Figure 2: RAL306W\_plus\_M3\_S78\_L006 fastqc**

**a. per base quality**



*Fastqc Per base sequence quality distributions output png files. Left is RAL306W\_plus\_M3\_S78\_L006\_R1\_001. Right is RAL306W\_plus\_M3\_S78\_L006\_R2\_001*

**b. Per base n content**



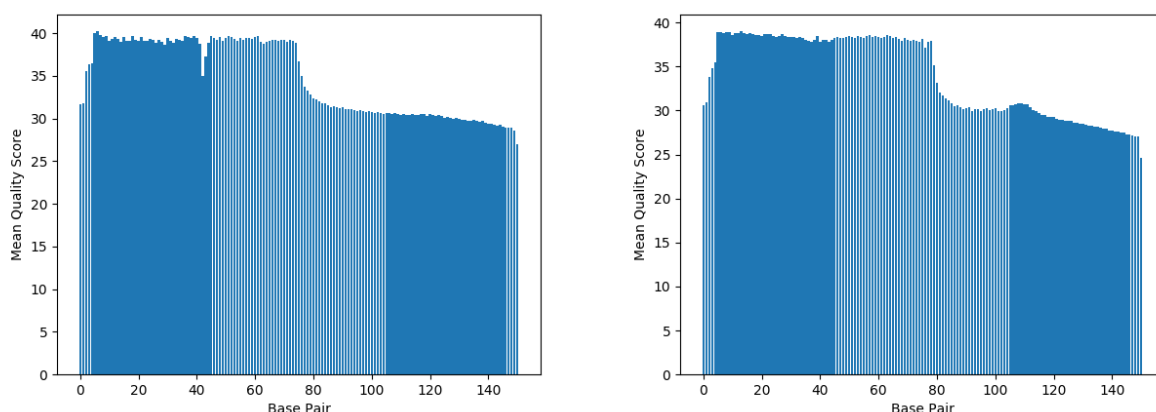
*Fastqc Per base n content output png files. Left is RAL306W\_plus\_M3\_S78\_L006\_R1\_001. Right is RAL306W\_plus\_M3\_S78\_L006\_R2\_001*

**and comment on whether or not they are consistent with the quality score plots.**

The N cotent across the bases was consistent in which there was no N bases. The mean curves look the same. RAL306W\_plus\_M3\_S78\_L006\_R1\_001 base quality scores were good from 1 to 75-79 but R2 was not as good. There was no N cotent across the bases in both reads.

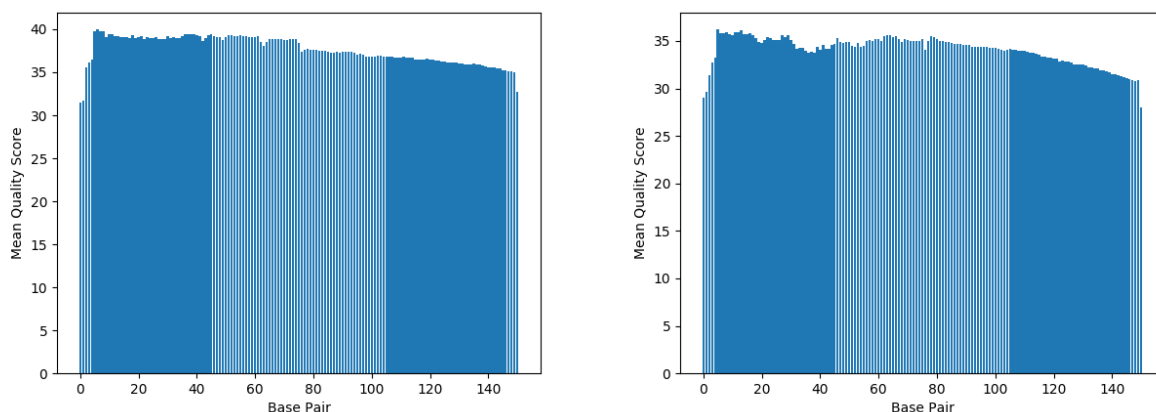
**Run your quality score plotting script from your Demultiplexing assignment from Bi 622**

**Figure 3: RAL306W-F3\_S79\_L006 python script**



*Python script for quality score along each of the basepair distributions. Left is RAL306W-F3\_S79\_L006\_R1\_001. Right is RAL306W-F3\_S79\_L006\_R2\_001*

**Figure 4: RAL306W\_plus\_M3\_S78\_L006 python script**



*Python script for quality score along each of the basepair distributions. Left is RAL306W\_plus\_M3\_S78\_L006\_R1\_001. Right is RAL306W\_plus\_M3\_S78\_L006\_R2\_001*

### Describe how the FastQC quality score distribution plots compare to your own

The mean score curve in the plots between fastqc and Demultiplexing code looks about the same. The run time was different fastqc was 3-5 fold faster than the demultiplexing code. This can be due to fastqc using bins of base pairs instead of individual base pairs. Also, multithreading can make fastqc more efficient. In addition, fastqc uses the language Java which is faster than python. Java is a compiled language where it runs faster compared to python as interpreted language.

Table 1

##	fast_qc (h:mm:ss)
## RAL306W-F3_S79_L006_R1_001	"0:14.15"
## RAL306W-F3_S79_L006_R2_001	"0:13.13"
## RAL306W_plus_M3_S78_L006_R1_001	"4:21.43"
## RAL306W_plus_M3_S78_L006_R2_001	"4:37.23"
##	demultiplexing_code (h:mm:ss)
## RAL306W-F3_S79_L006_R1_001	"0:48.04"
## RAL306W-F3_S79_L006_R2_001	"0:50.98"
## RAL306W_plus_M3_S78_L006_R1_001	"21:09.52"

## RAL306W\_plus\_M3\_S78\_L006\_R2\_001 "20:44.86"

*Table shows the differences in times running fastqc and python script for the from the R1 and R2 of the 2 demultiplexed file pairs.*

### **Comment on the overall data quality of your two libraries.**

The overall quality data of RAL306W-F3\_S79\_L006 reads were good. The mean scores curve for 1 to 75-79 where in the green then started to dip into the abnormal. Fastqc summary states that they are good. For RAL306W\_plus\_M3\_S78\_L006 R1, the mean score curve was good, good reads overall but started slightly deeping 1 to 75-79. Where for RAL306W\_plus\_M3\_S78\_L006 R2 it was not as good as R1 but the mean score curve did stay in the green started to fall 30-34.

## **Part 2 – Adaptor trimming comparison**

### **Look into the adaptor trimming options for cutadapt and Trimmomatic (all on Talapas), and briefly describe the differences.**

Trimmomatic can read single and paired ends. Removes adapters, leading or trailing low quality bases and N bases. Can cut adapter and other illumina-specific sequences, can cut bases off the start or the end of the read by a threshold quality. Has a sliding window trimming. Can drop read by a specific length. Has the convenience that it decides what adapter to use but can cut out overrepresented areas which can cut more than your adapter

Cutadapt can read single and paired ends. Removes adapter primers, sequences, poly-A tails and other unwanted sequences. Can modify and filter reads in various ways. IUPAC wildcard characters are allowed in adapter sequences. In cutadapt, you have to specify the adapters. It throws out reads that are less than the adapters size.

For the this assignment I used cutadapt because it was easier to get it running compared to trimmomatic.

### **What proportion of reads (both forward and reverse) was trimmed?**

According cutadapt for RAL306W-F3\_S79\_L006 R1 and R2 pairs 767,924 of 1,153,925, resulting in 66.5% of the reads were trimmed.

According cutadapt for RAL306W\_plus\_M3\_S78\_L006 R1 and R2 pairs 29,169,013 of 30,943,940 resulting in 94% of the reads were trimmed.

**Sanity check: Use your Unix skills to search for the adapter sequences in your datasets and confirm the expected sequence orientations. You may want to refer to Graded Assignment 4 from Bi623. Report the commands you used**

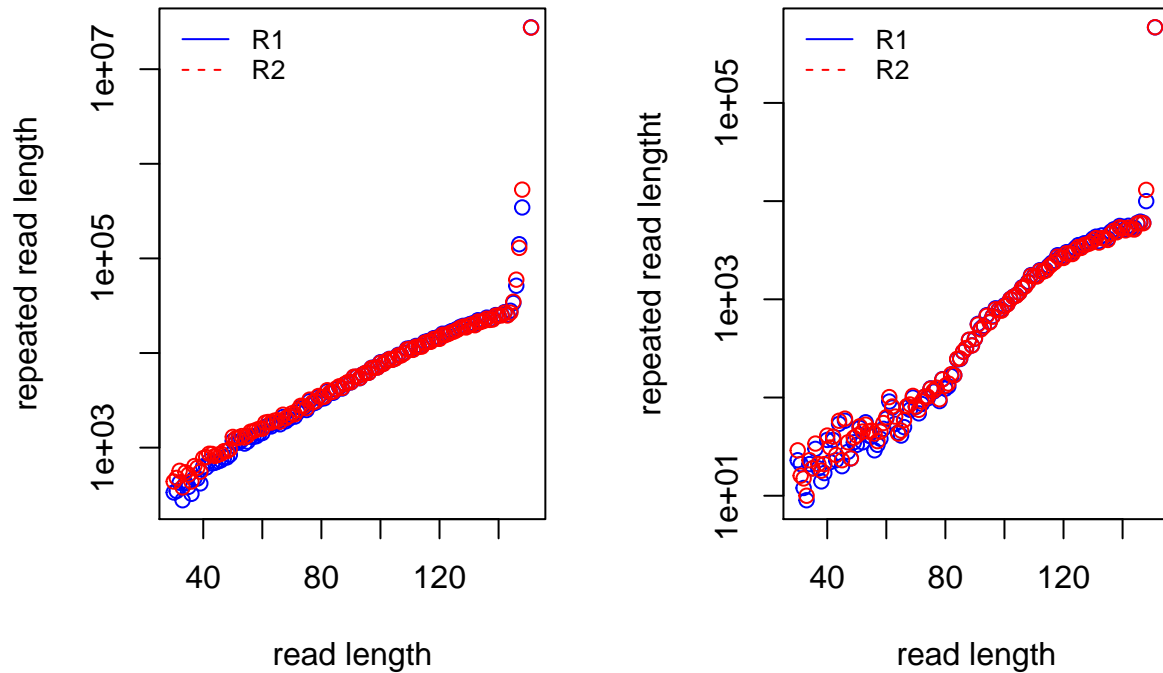
**the reasoning behind them, and how you confirmed the adapter sequences.**

The reason behind this was to confirm by greping and word count line of the Illumina adapter sequences, from the R1 and R2 of the 2 demultiplexed file pairs, are in the correct orientation pre-trimmed and absent post trimmed. The Illumina's TruSeq LT and TruSeq HT-based kits Read 1 and Read 2 adapter sequences were used. The confirmation came from having a large value in the grep word count line using Read 1 adapter on RAL306W-F3\_S79\_L006\_R1\_001 and doing the same for Read 2 adapter on RAL306W-F3\_S79\_L006\_R2\_001. Which did result into outputting large values. For coriousity, the same commands were used using Read 1 adapter on RAL306W-F3\_S79\_L006\_R2\_001 and Read 2 adapter on RAL306W-F3\_S79\_L006\_R1\_001 to confirm there was 0 which was the result. This same approach was used for RAL306W\_plus\_M3\_S78\_L006 R1 and R2 files. Furthermore, using postrimmed files, these same commands were run and resulted in 0 which is what we expected.

<https://support.illumina.com/bulletins/2016/12/what-sequences-do-i-use-for-adapter-trimming.html>

**Plot the trimmed read length distributions for both forward and reverse reads (on the same plot). You can produce 2 different plots for your 2 different RNA-seq samples**

**Figure 5 a: Trimmed read length distribution for RAL306W\_plus\_M3\_S78\_L006** **b: Trimmed read length distribution for RAL306W-F3\_S79\_L006**



Trimmed read length distributions for R1 and R2 of the 2 demultiplexed file pairs. Figure 5 a shows at the beginning read length R1 having less repeated read lengths compared to R2 but then overlapping until last few points. Figure 5 b shows at the beginning reads scattered but some overlap and more overlap as the reads length increases until the last few points.

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

We should definitely see expect R1s and R2s to be adapter-trimmed at different rates.

### Part 3 – rRNA reads and strand-specificity

**Find publicly available *Drosophila melanogaster* rRNA sequences and generate an alignment database (e.g. STAR) from them.**

Downloaded the *Drosophila\_melanogaster*.BDGP6.22.ncrna.fa.gz file from ensembl.org. and did a gunzip.

**Align the reads to your fly rRNA database (e.g. STAR) and report the proportion of reads that likely came from rRNAs.**

**and report the proportion of reads that likely came from rRNAs**

From the alignment result and looking into the aligned\_RAL306W-F3\_S79\_L006\_rRNA Log.final.out file, rRNA uniquely mapped reads were 1.13%. As for, the aligned\_RAL306W\_plus\_M3\_S78\_L006\_rRNA Log.final.out file, rRNA uniquely mapped reads were 0%.

These numbers state that 2 trimmed demultiplexed file pair reads of rRNA proportions were small to nothing. This is hard to believe. There is most likely an incorrect error in running STAR that needs to be explored and corrected. Due to time, I went forward with these alignments to explore further questions.

**Demonstrate convincingly whether or not the data are from “strand-specific” RNA-Seq libraries. There are a few possible strategies to address this problem, but you need only implement one**

From the alignment output result sam file, we can use the bit wise flag 64, which is “First in pair” and 16, which is “Read reverse strand” to determine the strand-specific. From running this, if we see an even distribution in forward and reverse, this would mean no strandness. If we see a lopsided distribution then it would mean strandness.

Python script to counts up the number of reads that are properly forward and reverse from a sam file

Running the python script using bash, the output for running the aligned\_RAL306W-F3\_S79\_L006\_rRNA sam file was forward: 0, reverse: 758684, unmapped: 19747. As for running aligned\_RAL306W\_plus\_M3\_S78\_L006\_rRNA sam file was forward: 0, reverse: 29169000, unmapped: 52. From these results, the data are from a strand specific RNA-Seq libraries.