PS2_template

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.

This template is provided as reference for instructions. Files with specific naming conventions are requested to be turned in at the end of this problem set. You can use this template to gather notes while completing this assignment. Be sure to upload all relevant materials by the deadline and **double check** to be sure that your offline repository is up-to-date with your online repository. Answers to questions should be included in your final, high-level, report as a `pdf`. This pdf should be generated using Rmarkdown and submitted to Canvas as well as GitHub. Be sure to keep a well-organized, detailed lab notebook!

Dataset:

Each of you will be working with 2 RNA-seq files from two different electric fish studies (PRJNA1005245 and PRJNA1005244). The methods for the PRJNA1005244 dataset are published and the methods for the PRJNA1005245 dataset are written in the third chapter of a thesis. For all steps below, process the two libraries separately. SRR assignments are here: `/projects/bgmp/shared/Bi623/PS2/QAA_data_Assignments.txt`. If you have time, consider claiming and processing additional RNA-seq raw sequencing files via this google doc. Although this is not extra credit, it will make our downstream RNA-seq analysis more interesting and your classmates will appreciate your efforts.

You are responsible for downloading this data from NCBI SRA, dumping into FASTQ files, and zipping those files (check ICA1 for a refresher). We are processing this data for use in a future assignment, so please keep your files well organized. Finally, rename the files to the convention Species_sample_tissue_age/size_sample#_readnumber.fastq.gz.

Reminder: This template file IS not your final product; however, it gives you a space to record all of the necessary information for your final report.

```
## Download your data

mamba activate sra-toolkit_bgmp
```

```
prefetch SRR25630305

prefetch SRR25630397

fasterq-dump --split-files ./SRR25630305.sra

fasterq-dump --split-files ./SRR25630397.sra

gzip SRR25630305_1.fastq

gzip SRR25630305_2.fastq

gzip SRR25630397_1.fastq

gzip SRR25630397_2.fastq
```

Part 1 – Read quality score distributions

1. Create a new conda environment called `QAA` and install `FastQC`, `cutadapt`, and `Trimmomatic`. Google around if you need a refresher on how to create conda environments. Recommend doing this in an interactive session, not the login node! Record details of how you created this environment in your lab notebook! Make sure you check your installation with:
    - `fastqc --version` (should be 0.12.1)

[Record details on how you made the conda environment]

```
mamba create -n QAA

mamba activate QAA

mamba install fastqc

mamba install cutadapt

mamba install trimmomatic
```

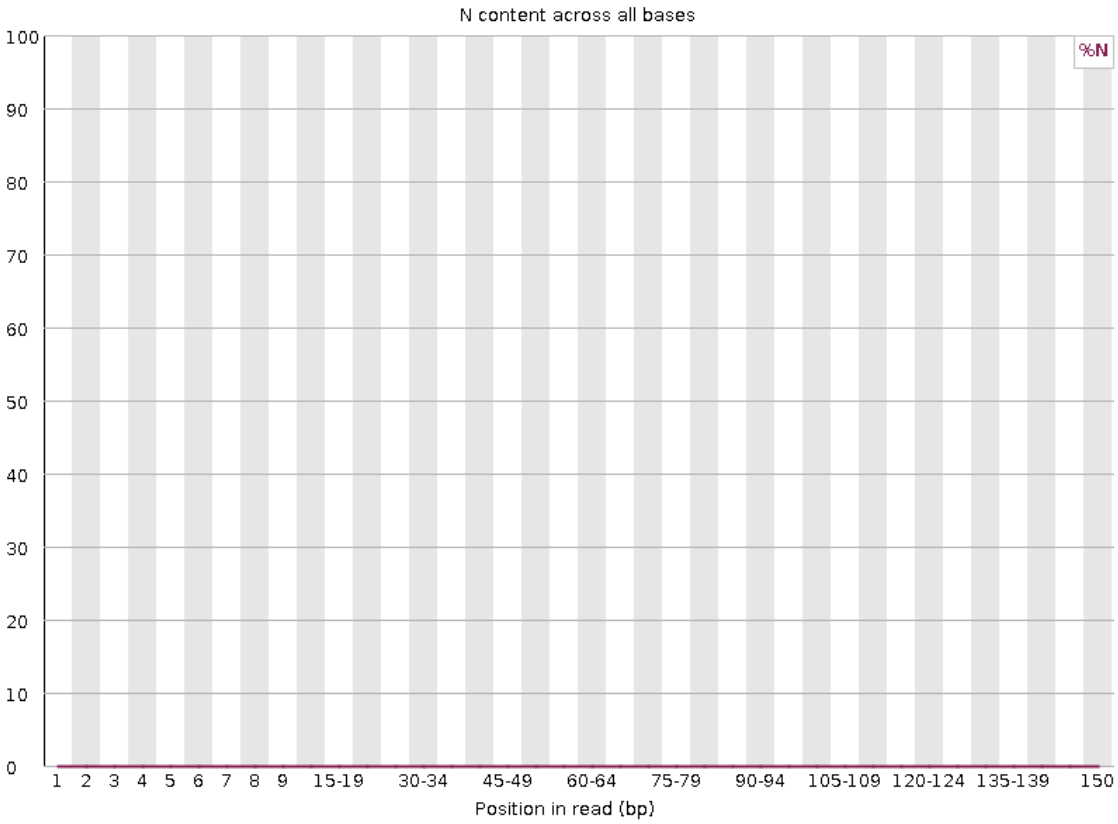
2. Using `FastQC` via the command line on Talapas, produce plots of the per-base quality score distributions for R1 and R2 reads. Also, produce plots of the per-base N content, and comment on whether or not they are consistent with the quality score plots.

[Include FastQC commands, plots of per-base N content, comments on consistency with quality score plots]

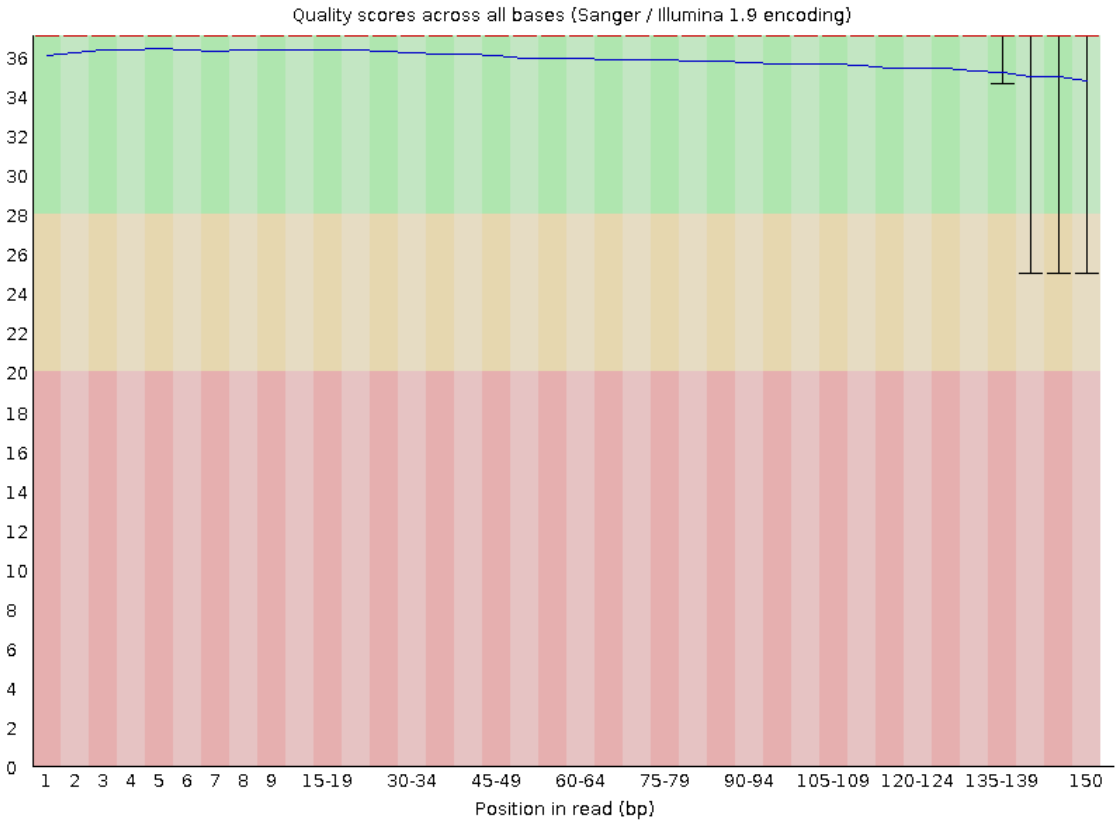
```
/usr/bin/time -v fastqc SRR25630305/*fastq.gz -o /projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630305

/usr/bin/time -v fastqc SRR25630397/*fastq.gz -o /projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630397
```

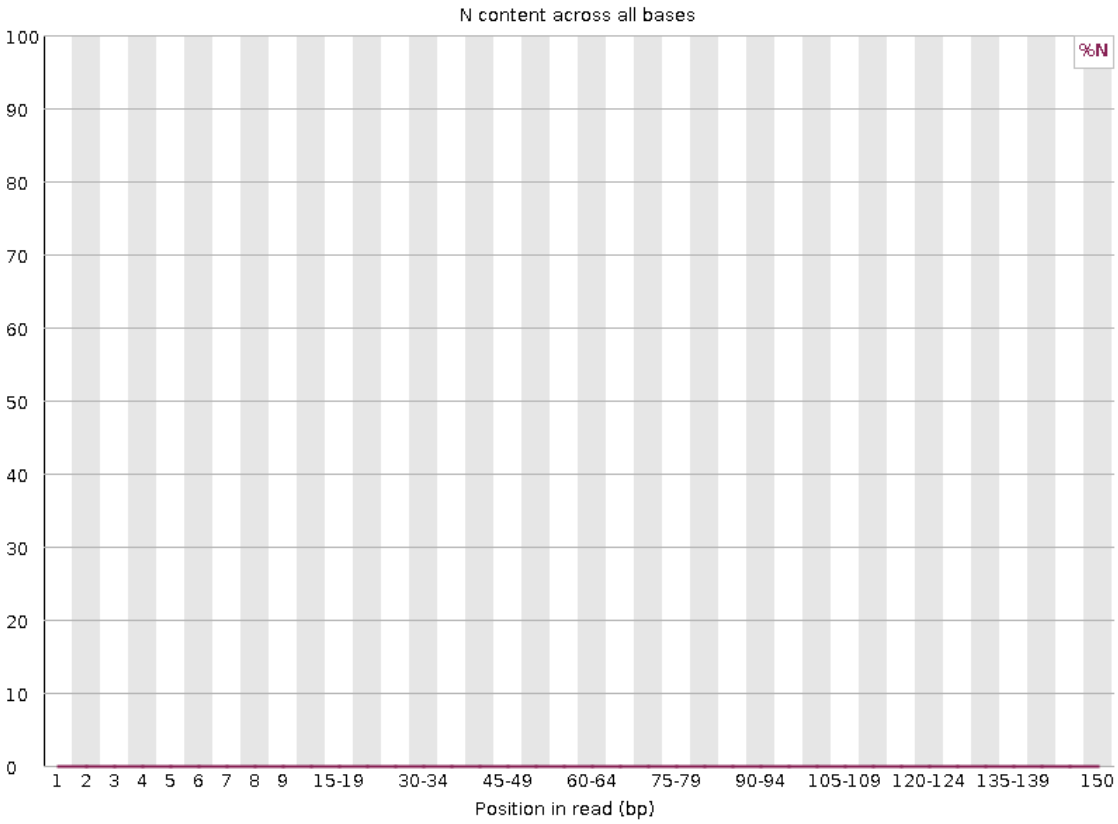
```
knitr::include_graphics("QAA_figs/Cco_com125_EO_6cm_1_per_base_n_content.png")
```



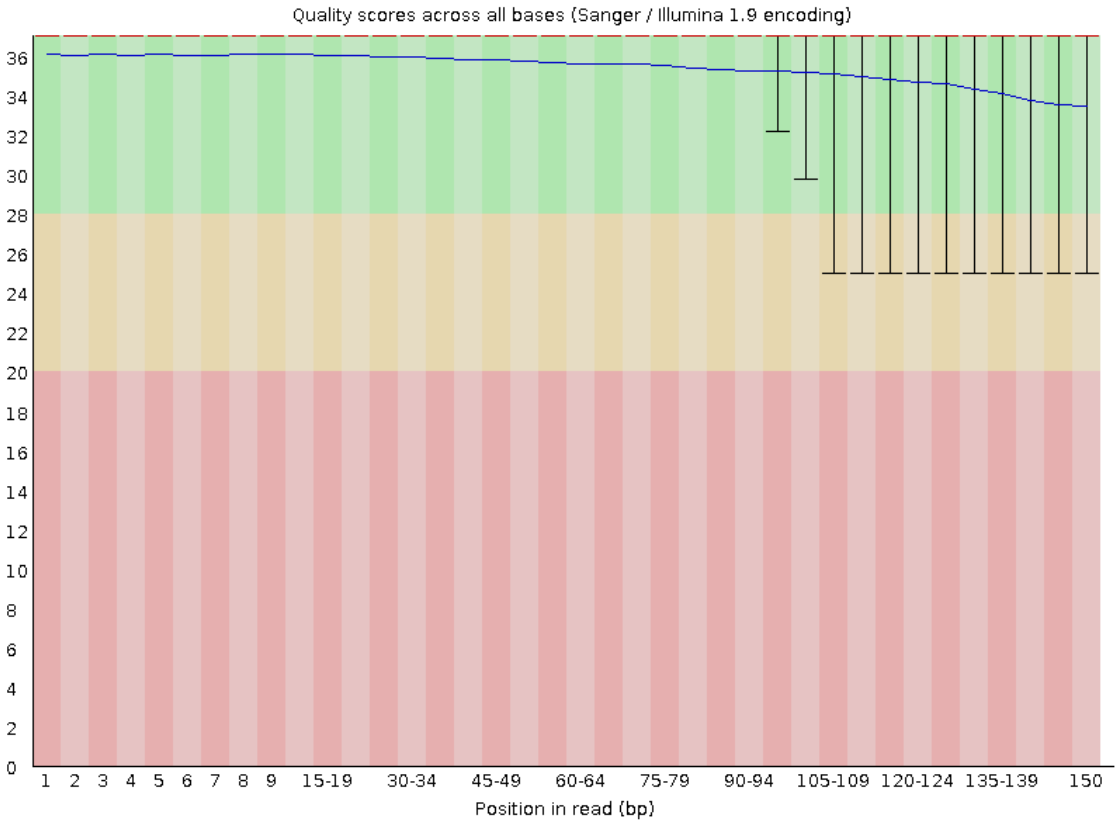

```
knitr::include_graphics("QAA_figs/Cco_com125_EO_6cm_1_per_base_quality.png")
```



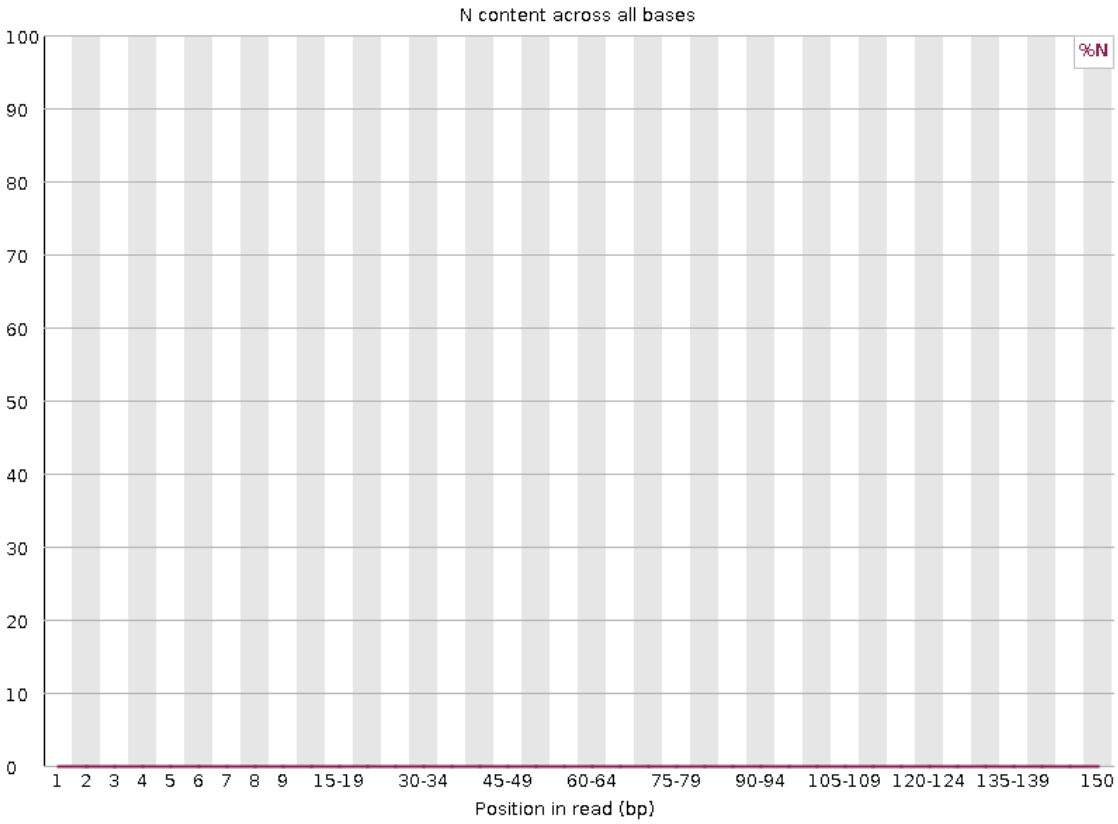

```
knitr::include_graphics("QAA_figs/Cco_com125_EO_6cm_2_per_base_n_content.png")
```



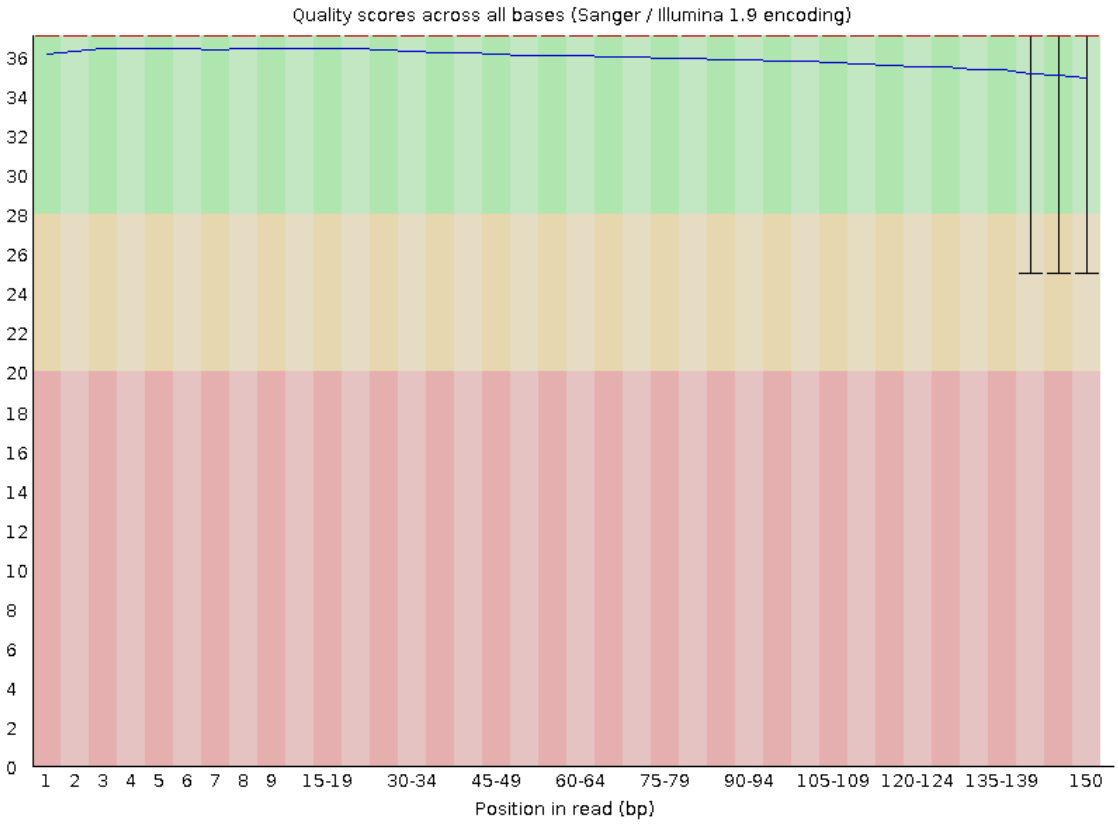

```
knitr::include_graphics("QAA_figs/Cco_com125_EO_6cm_2_per_base_quality.png")
```



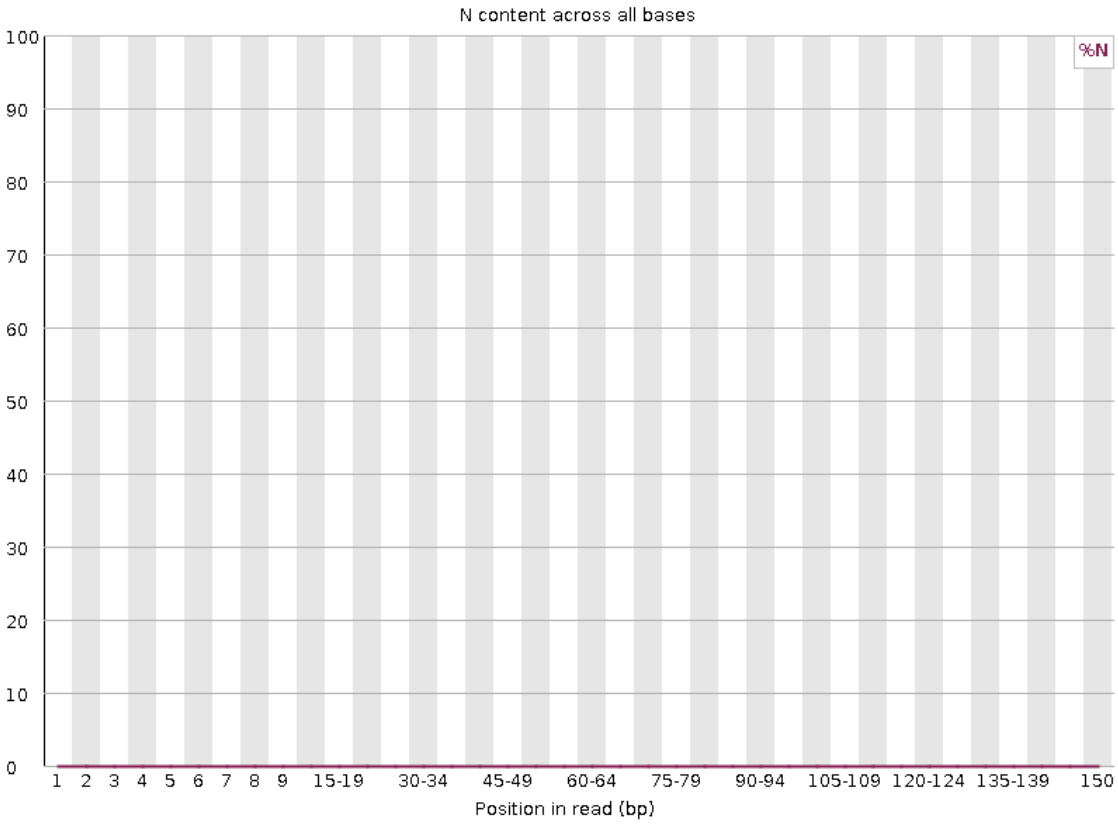

```
knitr::include_graphics("QAA_figs/Crh_rhy107_EO_adult_1_per_base_n_content.png")
```



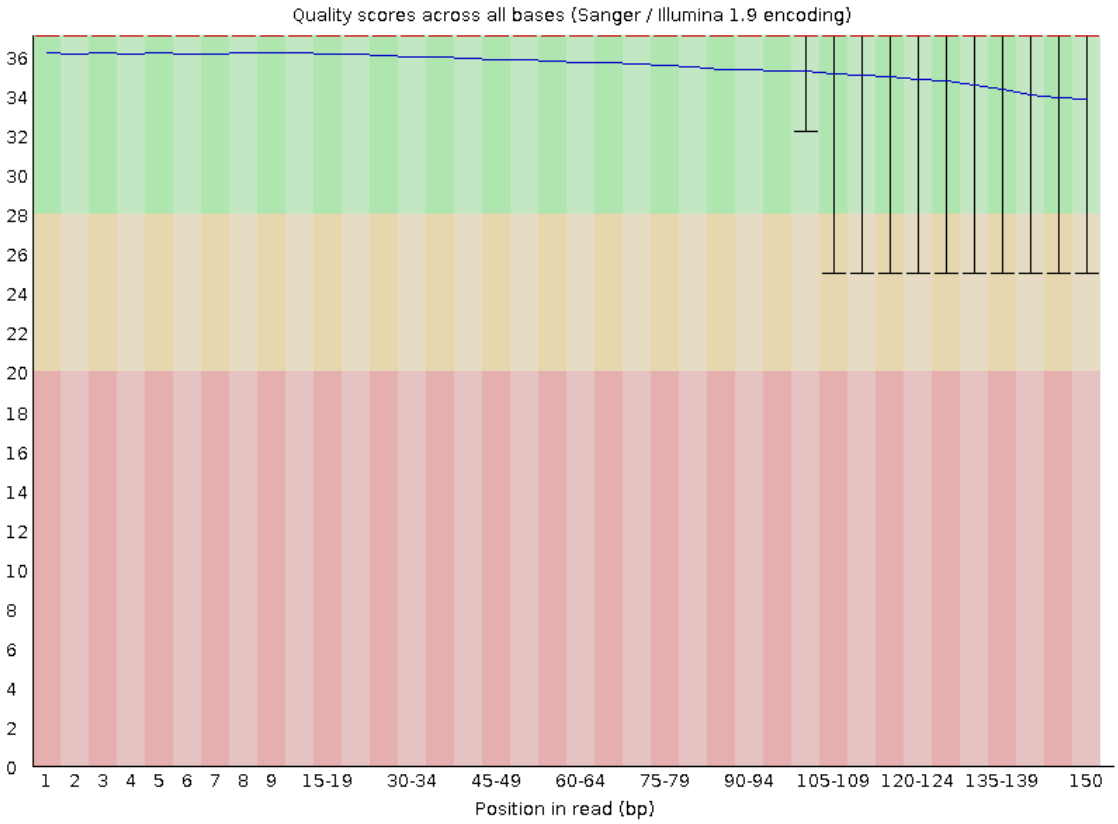

```
knitr::include_graphics("QAA_figs/Crh_rhy107_EO_adult_1_per_base_quality.png")
```




```
knitr::include_graphics("QAA_figs/Crh_rhy107_EO_adult_2_per_base_n_content.png")
```



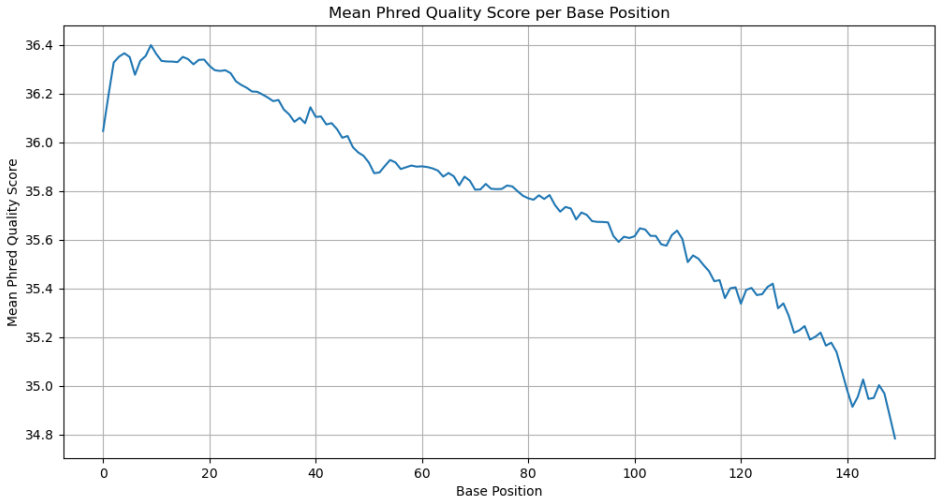

```
knitr::include_graphics("QAA_figs/Crh_rhy107_EO_adult_2_per_base_quality.png")
```



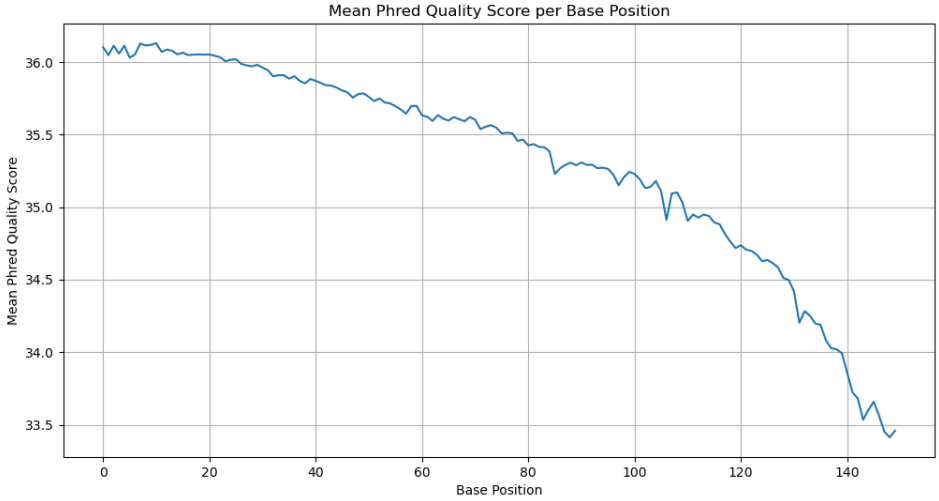
The near-zero N content is consistent with the high base-quality profiles. Bases are confidently called (not "N") across the reads, including at the ends where quality dips slightly but remains high.

3. Run your quality score plotting script from your Demultiplexing assignment in Bi622. (Make sure you're using the "running sum" strategy!!) 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?

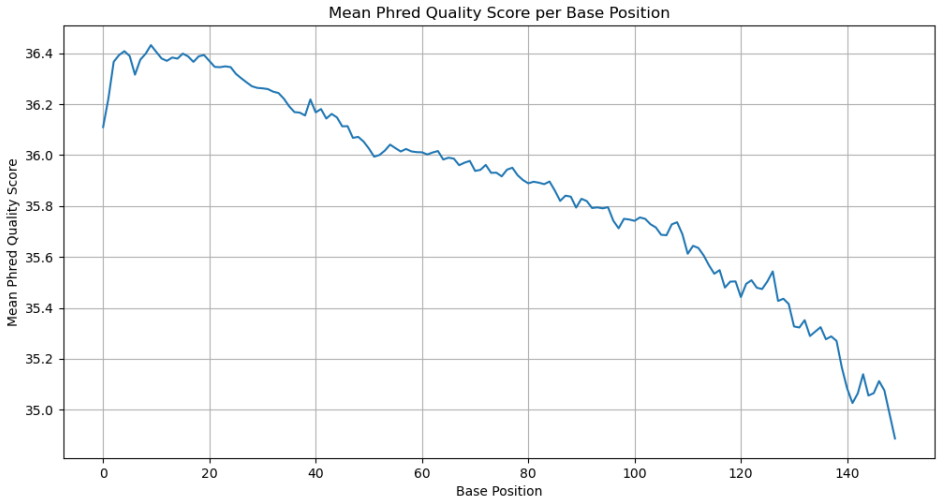
```
knitr::include_graphics("QAA_figs/R1_Cco_com125_EO_6cm_mean_quality_scores.png")
```



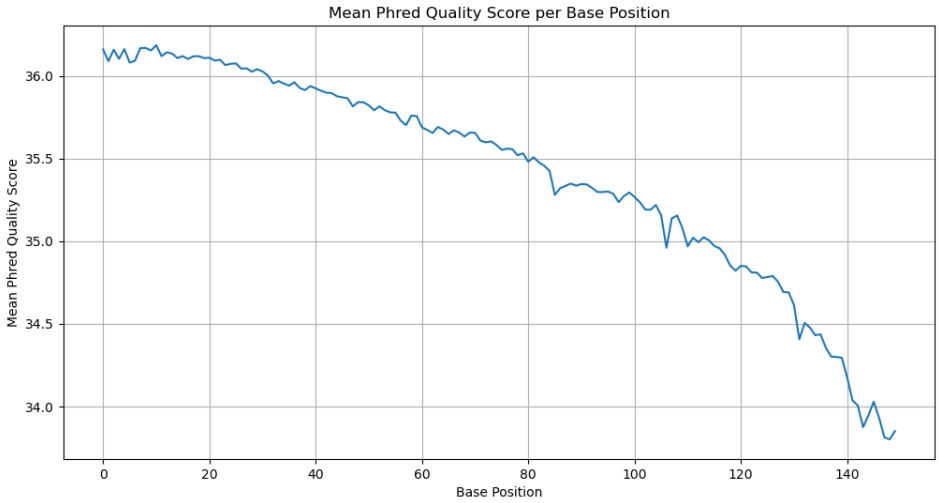

```
knitr::include_graphics("QAA_figs/R2_Cco_com125_EO_6cm_mean_quality_scores.png")
```




```
knitr::include_graphics("QAA_figs/R1_Crh_rhy107_EO_adult_mean_quality_scores.png")
```




```
knitr::include_graphics("QAA_figs/R2_Crh_rhy107_EO_adult_mean_quality_scores.png")
```



The FastQC quality plots and my plots are similar. They all have very high scores at the beginning of that reads (~Q36) with a gradual decline toward the end of the reads, and R2 tails off slightly more than R1. Per-base N content is ~0% in FastQC, which matches the high quality seen in both sets of plots. Small visual differences are expected because FastQC shows median with IQR (box-and-whiskers) while mine plot the mean, FastQC bins later cycles (e.g., 105–109) which smooths the right side. Overall, the results are consistent; any offsets are methodological, not biological. FastQC ran a bit faster because it's an optimized, compiled Java program, whereas my Python script is single-threaded and spends more time in gzip decompression and Python loops. FastQC typically uses more CPU and RAM but finishes sooner; the Python script uses less CPU/RAM but takes longer.

4. 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.

[Include comments on data quality and recommendation on whether this can be used for further analysis]

Both libraries meet the bar for downstream RNA-seq. Quality scores are high across most bases with only a small drop at the end. The N content is ~0% across positions, the Adapter Content panel doesn't show a worrying rise, there aren't big unknown "overrepresented" sequences, duplication levels are typical for RNA-seq (representing highly expressed transcripts), the GC content looks close to what we expect, and the read lengths are consistent. Overall, the data look clean and are good to use for downstream analysis.

Part 2 – Adaptor trimming comparison

5. If you haven't already in your QAA environment, install `Cutadapt` and `Trimmomatic`. Check your installations with:
   - `cutadapt --version` (should be 5.0)
   - `trimmomatic -version` (should be 0.39)

```
cutadapt --version = 5.1

trimmomatic -version = 0.40
```

6. Using `Cutadapt`, properly trim adapter sequences from your assigned files. Be sure to read how to use `Cutadapt`. Use default settings. What proportion of reads (both R1 and R2) were trimmed?

   Try to determine what the adapters are on your own. If you cannot (or if you do, and want to confirm), click here to see the actual adapter sequences used.

   R1: `AGATCGGAAGAGCACACGTCTGAACTCCAGTCA`

   R2: `AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT`

   - *Sanity check*: Use your Unix skills to search for the adapter sequences in your datasets and confirm the expected sequence orientations. Report the commands you used, the reasoning behind them, and how you confirmed the adapter sequences.

[Include commands and report out the proportion of reads trimmed]

```
cutadapt -a AGATCGGAAGAGCACACGTCTGAACTCCAGTCA -A AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT -o Cco_com125_EO_6cm_

cutadapt -a AGATCGGAAGAGCACACGTCTGAACTCCAGTCA -A AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT -o Crh_rhy107_EO_adul
```

For Cco_com125_EO_6cm: Read 1 - 18.2% reads trimmed Read 2 - 18.5% reads trimmed

For Crh_rhy107_EO_adult: Read 1 - 12.5% reads trimmed Read 2 - 13.0% reads trimmed

7. Use `Trimmomatic` to quality trim your reads. Specify the following, **in this order**:
   - LEADING: quality of 3
   - TRAILING: quality of 3
   - SLIDING WINDOW: window size of 5 and required quality of 15
   - MINLENGTH: 35 bases

   Be sure to output compressed files and clear out all intermediate files.

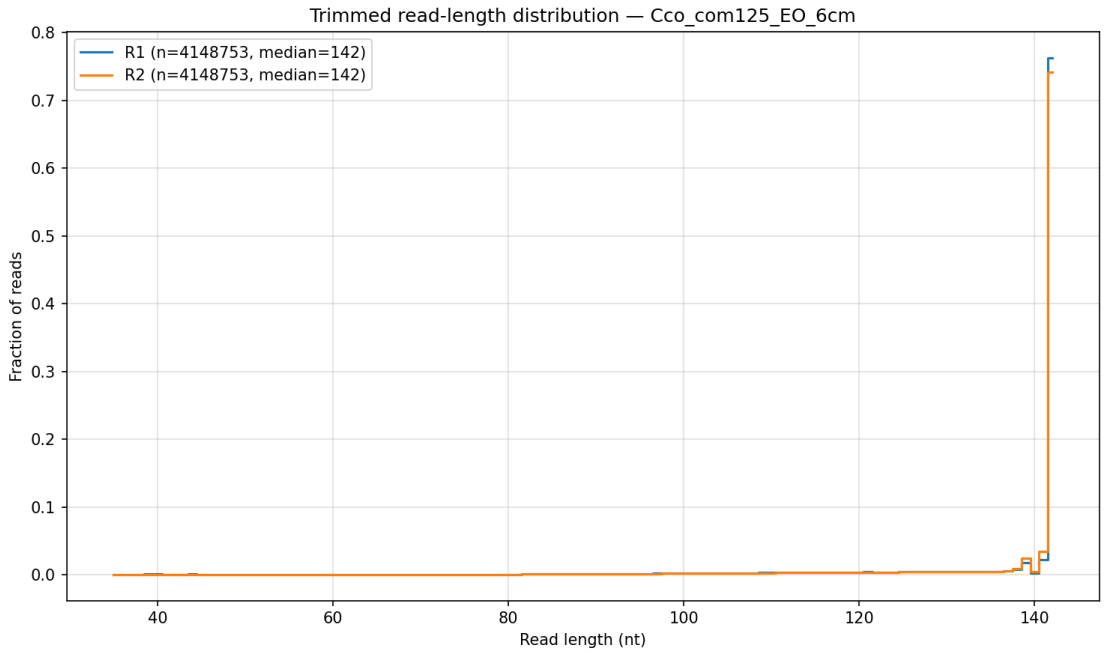
```
/usr/bin/time -v trimmomatic PE \
-threads 4 \
/projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630305/SRR25630305_1.trim.fastq.gz \
/projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630305/SRR25630305_2.trim.fastq.gz \
/projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630305/SRR25630305_1.trim.paired.fastq.gz \
/projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630305/SRR25630305_1.trim.unpaired.fastq.gz \
/projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630305/SRR25630305_2.trim.paired.fastq.gz \
/projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630305/SRR25630305_2.trim.unpaired.fastq.gz \
HEADCROP:8 \
LEADING:3 \
TRAILING:3 \
SLIDINGWINDOW:5:15 \
MINLEN:35

/usr/bin/time -v trimmomatic PE \
-threads 4 \
/projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630397/SRR25630397_1.trim.fastq.gz \
/projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630397/SRR25630397_2.trim.fastq.gz \
/projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630397/SRR25630397_1.trim.paired.fastq.gz \
/projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630397/SRR25630397_1.trim.unpaired.fastq.gz \
/projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630397/SRR25630397_2.trim.paired.fastq.gz \
/projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630397/SRR25630397_2.trim.unpaired.fastq.gz \
HEADCROP:8 \
LEADING:3 \
TRAILING:3 \
SLIDINGWINDOW:5:15 \
MINLEN:35
```

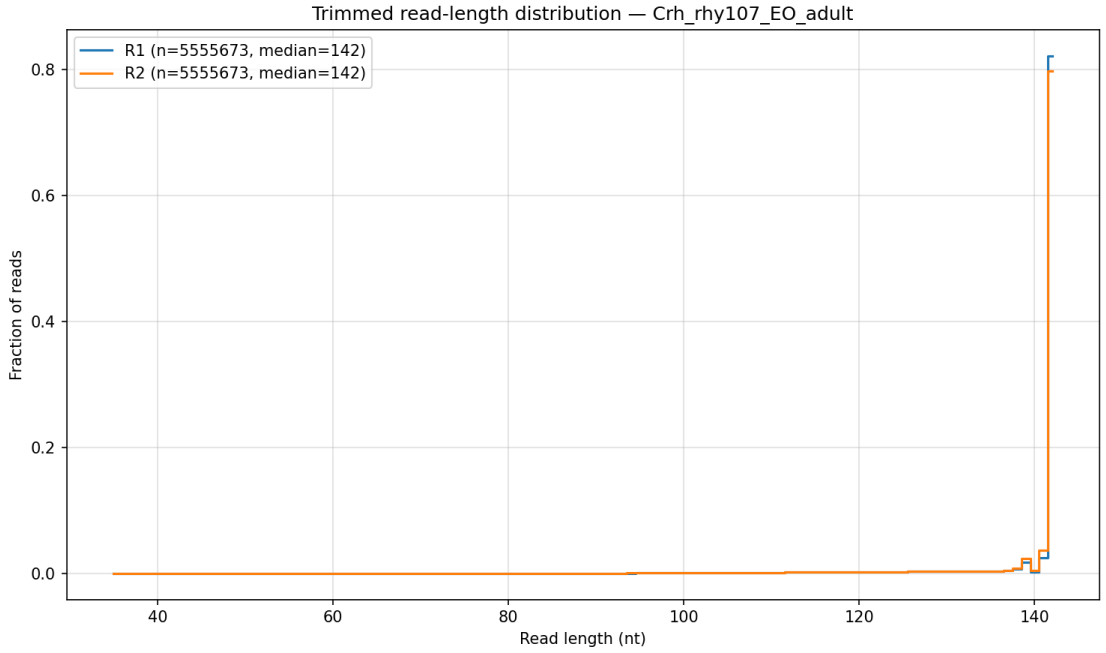
8. Plot the trimmed read length distributions for both paired R1 and paired R2 reads (on the same plot - yes, you will have to use Python or R to plot this. See ICA4 from Bi621). You can produce 2 different plots for your 2 different RNA-seq samples. There are a number of ways you could possibly do this. One useful thing your plot should show, for example, is whether R1s are trimmed more extensively than R2s, or vice versa. Comment on whether you expect R1s and R2s to be adapter-trimmed at different rates and why.

[Include your plot and comment on R1/R2 adapter trimming]

```
knitr::include_graphics("QAA_figs/Cco_com125_EO_6cm_trimmed_lengths.png")
```




```
knitr::include_graphics("QAA_figs/Crh_rhy107_EO_adult_trimmed_lengths.png")
```



R1 and R2 look basically the same. Both end up around 142 nt, so most trimming came from the fixed HEADCROP:8, not from cutting off adapters. In general, R1 and R2 should have about the same amount of adapter trimming because adapters show up when the DNA/RNA fragment is shorter than the read, which affects both reads equally. Any tiny differences are usually just because R2's end is a bit lower quality, not because it has more adapters.

9. Bonus - Run `FastQC` on your trimmed data. Comment on differences you observe between the trimmed and untrimmed data. Include any figures needed to support your conclusions.

[Include command, comments on differences, and plot/s]

Part 3 – Alignment and strand-specificity

10. Install additional software for alignment and counting of RNA-seq reads. In your QAA environment, use conda to install:
    - Star
    - Picard
    - Samtools
    - NumPy
    - Matplotlib
    - HTSeq

[Record details on how you installed these packages]

```
mamba install star

mamba install picard=2.18

mamba install samtools

mamba install numpy

mamba install matplotlib

mamba install htseq #numpy was downgraded to version 1.26.4
```

11. Download the publicly available *Campylomormyrus compressirostris* genome fasta and gff file from Dryad and generate an alignment database from it. If the download fails, the files are available `/projects/bgmp/shared/Bi623/PS2/campylomormyrus.fasta`, `/projects/bgmp/shared/Bi623/PS2/campylomormyr` Align the reads to your *C. compressirostris* database using a splice-aware aligner. Use the settings specified in PS8 from Bi621.

    [!IMPORTANT] You will need to use gene models to perform splice-aware alignment, see PS8 from Bi621. You may need to convert the gff file into a gtf file for this to work successfully.

[Record details on how you downloaded the genome, prepared the dataset for alignment, and commands for generating the alignment database and aligning reads]

```
##I was unable to download the files from Dryad due to an error (ERROR 403: Forbidden.)
cp /projects/bgmp/shared/Bi623/PS2/campylomormyrus.fasta .

cp /projects/bgmp/shared/Bi623/PS2/campylomormyrus.gff .

mamba install -c bioconda gffread -y

gffread -E campylomormyrus.gff -T -o- > campylomormyrus.gtf
```

```
STAR --runThreadN 8 \
    --runMode genomeGenerate \
    --genomeDir /projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630305/Campylomormyrus.Dryad.STAR_2.7.11b
    --genomeFastaFiles ./campylomormyrus.fasta \
    --sjdbGTFfile ./campylomormyrus.gtf

STAR --runThreadN 8 --runMode alignReads \
    --outFilterMultimapNmax 3 \
    --outSAMunmapped Within KeepPairs \
    --alignIntronMax 1000000 --alignMatesGapMax 1000000 \
    --readFilesCommand zcat \
    --readFilesIn /projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630305/Cco_com125_EO_6cm_1.trim.paired.
    --genomeDir /projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630305/Campylomormyrus.Dryad.STAR_2.7.11b
    --outFileNamePrefix Cco_alignments

/usr/bin/time -v STAR --runThreadN 8 \
    --runMode genomeGenerate \
    --genomeDir /projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630397/Campylomormyrus.Dryad.STAR_2.7.11b
    --genomeFastaFiles ./campylomormyrus.fasta \
    --sjdbGTFfile ./campylomormyrus.gtf

/usr/bin/time -v STAR --runThreadN 8 --runMode alignReads \
    --outFilterMultimapNmax 3 \
    --outSAMunmapped Within KeepPairs \
    --alignIntronMax 1000000 --alignMatesGapMax 1000000 \
    --readFilesCommand zcat \
    --readFilesIn /projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630397/Crh_rhy107_EO_adult_1.trim.paire
    --genomeDir /projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630397/Campylomormyrus.Dryad.STAR_2.7.11b
    --outFileNamePrefix Crh_alignments
```

12. Remove PCR duplicates using Picard MarkDuplicates. You may need to sort your reads with `samtools` before running Picard.

- Use the following for running picard: picard MarkDuplicates INPUT=[FILE] OUTPUT=[FILE] METRICS_FILE=[FILENAME].metrics REMOVE_DUPLICATES=TRUE VALIDATION_STRINGENCY=LENIENT

```
samtools sort -O SAM -o Cco_alignmentsAligned.out.sorted.sam Cco_alignmentsAligned.out.sam

samtools sort -O SAM -o Crh_alignmentsAligned.out.sorted.sam Crh_alignmentsAligned.out.sam

picard MarkDuplicates \
  INPUT=/projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630305/Cco_com125_EO_6cm.out.sorted.sam \
  OUTPUT=/projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630305/Cco_com125_EO_6cm.out.sorted.rmdup.sam \
  METRICS_FILE=/projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630305/Cco_com125_EO_6cm.metrics \
  REMOVE_DUPLICATES=TRUE \
  VALIDATION_STRINGENCY=LENIENT

picard MarkDuplicates \
  INPUT=/projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630397/Crh_rhy107_EO_adult.out.sorted.sam \
  OUTPUT=/projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630397/Crh_rhy107_EO_adult.out.sorted.rmdup.sam
  METRICS_FILE=/projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630397/Crh_rhy107_EO_adult.metrics \
  REMOVE_DUPLICATES=TRUE \
  VALIDATION_STRINGENCY=LENIENT
```

13. Using your script from PS8 in Bi621, report the number of mapped and unmapped reads from each of your 2 SAM files post deduplication with picard. Make sure that your script is looking at the bitwise flag to determine if reads are primary or secondary mapping (update/fix your script if necessary).

For Cco_com125_EO_6cm.out.sorted.rmdup.sam: Number of unique mapped reads: 1903088 Number of unique unmapped reads: 958755

For Crh_rhy107_EO_adult.out.sorted.rmdup.sam: Number of unique mapped reads: 2855485 Number of unique unmapped reads: 913124

14. Count deduplicated reads that map to features using `htseq-count`. You should run htseq-count twice: once with `--stranded=yes` and again with `--stranded=reverse`. Use default parameters otherwise. You may need to use the `-i` parameter for this run.

```
/usr/bin/time -v htseq-count -s yes -i gene_id\
  /projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630305/Cco_com125_EO_6cm.out.sorted.rmdup.sam \
  /projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630305/campylomormyrus.gtf \
  > /projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630305/Cco_com125_EO_6cm.htseq.stranded_yes.txt

/usr/bin/time -v htseq-count -s reverse -i gene_id\
  /projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630305/Cco_com125_EO_6cm.out.sorted.rmdup.sam \
  /projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630305/campylomormyrus.gtf \
  > /projects/bgmp/ica/bioinfo/Bi623/PS2/QAA/SRR25630305/Cco_com125_EO_6cm.htseq.stranded_reverse.txt
```

15. Demonstrate convincingly whether or not the data are from "strand-specific" RNA-Seq libraries **and** which `stranded=` parameter should you use for counting your reads for a future differential gene expression analyses. Include any commands/scripts used. Briefly describe your evidence, using quantitative statements (e.g. "I propose that these data are/are not strand-specific, because X% of the reads are y, as opposed to z."). This kit was used during library preparation. This paper may provide helpful information.

    [!TIP] Recall ICA4 from Bi621.

[Describe whether your reads are "string-specific", why you think they are, any evidence, and which stranded parameter is appropriate and why]

calc	assigned	total	percent
1	93,309	4,508,188	2.07%
2	1,942,036	4,508,188	43.08%
3	153,957	6,360,672	2.42%
4	3,344,574	6,360,672	52.58%

These data are strand-specific (reverse-stranded). With htseq-count, –stranded=reverse assigned 43.08% of reads to genes vs 2.07% with –stranded=yes for the Cco_com125_EO_6cm dataset. With htseq-count, –stranded=reverse assigned 52.58% of reads to genes vs 2.42% with –stranded=yes for the Crh_rhy107_EO_adult dataset. Given the kit's dUTP chemistry, this matches expectations. We will use htseq-count –stranded=reverse for downstream differential expression.

16. BONUS - Turn your commands from part 1 and 2 into a script with a loop going through your two SRA files

Bonus (optional!)

Review the publication from PRJNA1005244 or the third chapter of the thesis for the PRJNA1005245 dataset. See if this information leads to any additional insight of your analysis.

[Add insights to the dataset]

Upload your:

- ☐ lab notebook
- ☐ Talapas batch script/code
- ☐ FastQC plots
- ☐ counts files generated from htseq-count (in a folder would be nice; **only include the counts files that would be used in a future differential RNA-seq analysis: use the format Species_sample_tissue_age/size_sample#*readnumber_htseqcounts*[revORyes]stranded.txt**)
- ☐ pdf report (see below; turn into both Github AND Canvas)
- ☐ and any additional plots, code, or code output

to GitHub.

Pdf report details

You should create a pdf file (using Rmarkdown) with a high-level report including:

- ☐ all requested plots
- ☐ answers to questions
- ☐ mapped/unmapped read counts from PS8 script (in a nicely formatted table)
- ☐ It should be named `QAA_report.pdf`
- ☐ Include at the top level of your repo
- ☐ ALSO, submit it to Canvas.

> [!TIP] You may need to install LaTeX to knit your rmarkdown into a pdf file. Run `tinytex::install_tinytex()` to install it on R.

The three parts of the assignment should be clearly labeled. Be sure to title and write a descriptive figure caption for each image/graph/table you present.

> [!TIP] Think about figure captions you've read and discussed in Journal Club. Find some good examples to model your captions on.