4 Sep 2025

I had forgotten that Hope suggested we start on this Wednesday at the latest... anyways, it's Thursday and about time to get some work done here

My SRR assignments are `SRR25630409` and `SRR25630385`.

SRA toolkit

sbatched these – get_SRR25630409.sh and get_SRR25630385.sh: For SRR25630409: prefetch

```
Percent of CPU this job got: 27%
Elapsed (wall clock) time (h:mm:ss or m:ss): 2:04.79
Maximum resident set size (kbytes): 56696
```

fasterq-dump

```
Percent of CPU this job got: 98%
Elapsed (wall clock) time (h:mm:ss or m:ss): 4:39.26
Maximum resident set size (kbytes): 593292
```

For SRR25630385: prefetch

```
Percent of CPU this job got: 27%
Elapsed (wall clock) time (h:mm:ss or m:ss): 1:36.12
Maximum resident set size (kbytes): 56444
```

fasterq-dump

```
Percent of CPU this job got: 98%
Elapsed (wall clock) time (h:mm:ss or m:ss): 3:35.10
Maximum resident set size (kbytes): 592372
```

fastqc

Was getting signal 9 killing it (out of memory) with 64 GB of ram with maximum memory allowed (10000MB) .. ran again with 128 GB of ram and the maximum amount permitted by –memory (10000 MB) and no problem.

```
/usr/bin/time -v fastqc --svg -t 8 --memory 10000 SRR25630385_1.fastq
```

Weirdly, however, the maximum memory used was only ~5.7 GB.

```
Percent of CPU this job got: 196%
Elapsed (wall clock) time (h:mm:ss or m:ss): 3:03.77
Maximum resident set size (kbytes): 5784548
```

forgot to gzip first but don't want to rerun that^

gzipping

don't want to wait around soo

```
/usr/bin/time -v gzip -1 *.fastq
```

```
    Percent of CPU this job got: 98%
    Elapsed (wall clock) time (h:mm:ss or m:ss): 11:12.58
    Maximum resident set size (kbytes): 2092
```

plotting with python

per base q scores For the same two fastq files (no longer run in parallel) with `plot_qscores.py` sbatched with `plot_qscores.sh`:

```
    Percent of CPU this job got: 99%
    Elapsed (wall clock) time (h:mm:ss or m:ss): 19:29.89
    Maximum resident set size (kbytes): 71692
```

Unsurprisingly, this took a lot less memory, and a lot more time than the fastqc analysis.

per base n Wrote `plot_per_base_n.py` and `plot_per_base_n.sh` to sbatch it based on `plot_qscores.py`

Realized this wasn't actually asked for in the assignment instructions but it hurts too much to delete the files so they are staying as a testament to what (relatively little) time it took to write them.

Putting it in one script/Part 2

6 Sept 2025 From now on, throwing everything into one sbatch script. Commenting out things as we goso we aren't rerunning more code, but this will make it easier to run it all together in one go later.

Part 3

7 Sep 2025 ### STAR installed necessary packages:

```
mamba install star
mamba install picard
mamba install bioconda::samtools
mamba install numpy
mamba install matplotlib
mamba install htseq
```

also installed this for purposes of GFF to GTF conversion:

```
mamba install bioconda::agat
```

all commands still located in QAA.sh

star database creation:

```
    Percent of CPU this job got: 388%
    Elapsed (wall clock) time (h:mm:ss or m:ss): 7:17.76
    Maximum resident set size (kbytes): 24136384
```

got this warning:

```
!!!!! WARNING: --genomeSAindexNbases 14 is too large for the genome size=862592683, which may cause seg
```

will rebuild if we are getting seg-faults during mapping, but keeping as is for now.

star aligning usage (threw a bunch of cores at it since I'm tired and want to finish ASAP):

```
    Percent of CPU this job got: 1921%
    Elapsed (wall clock) time (h:mm:ss or m:ss): 2:19.56
    Maximum resident set size (kbytes): 13793504
```

Picard

Now onto removing PCR duplicates.. in order to do this (with picard), we have to sort the SAM files output from STAR. To do that, first we have to convert them to BAM files.

Converting:

```
    Percent of CPU this job got: 99%
    Elapsed (wall clock) time (h:mm:ss or m:ss): 4:54.27
    Maximum resident set size (kbytes): 7648
```

Sorting:

```
    Percent of CPU this job got: 754%
    Elapsed (wall clock) time (h:mm:ss or m:ss): 0:55.36
    Maximum resident set size (kbytes): 20866120
```

errored out with Picard:

```
WARNING 2025-09-08 05:04:43 AbstractOpticalDuplicateFinderCommandLineProgram    A field field parsed ou
[Mon Sep 08 05:04:43 PDT 2025] picard.sam.markduplicates.MarkDuplicates done. Elapsed time: 0.00 minute
Runtime.totalMemory()=536870912
To get help, see http://broadinstitute.github.io/picard/index.html#GettingHelp
Exception in thread "main" java.lang.NullPointerException: Cannot invoke "htsjdk.samtools.SAMReadGroupR
    at picard.sam.markduplicates.MarkDuplicates.buildSortedReadEndLists(MarkDuplicates.java:558)
    at picard.sam.markduplicates.MarkDuplicates.doWork(MarkDuplicates.java:270)
    at picard.cmdline.CommandLineProgram.instanceMain(CommandLineProgram.java:281)
    at picard.cmdline.PicardCommandLine.instanceMain(PicardCommandLine.java:105)
    at picard.cmdline.PicardCommandLine.main(PicardCommandLine.java:115)
```

need to add readgroups (again with samtools), added to pipeline in QAA.sh

got this warning/error but it finished running so we're going to continue:

```
Exception in thread "main" htsjdk.samtools.SAMFormatException: SAM validation error: ERROR::INVALID_FLA
```

```
    Percent of CPU this job got: 129%
    Elapsed (wall clock) time (h:mm:ss or m:ss): 8:02.06
    Maximum resident set size (kbytes): 2291628
```

Needed to make it into a sam file again for the sake of reading the file in python..

```
    Percent of CPU this job got: 96%
    Elapsed (wall clock) time (h:mm:ss or m:ss): 0:37.62
    Maximum resident set size (kbytes): 7472
```

sam file is nearly 20 GB

Counting mapped unmapped for SRR25630385 (`count_mapped_SRR25630385_38069906.out`):

```
    Percent of CPU this job got: 99%
    Elapsed (wall clock) time (h:mm:ss or m:ss): 1:02.74
    Maximum resident set size (kbytes): 12904

# mapped: 22910539
# unmapped: 17341635
```

and for SRR25630409 (`count_mapped_SRR25630409_38069907.out`):

```
# mapped: 33240914
# unmapped: 17431172
```

htseq-count

For forward: `htseq-count_SRR25630409_38070692.err`

```
    Percent of CPU this job got: 99%
    Elapsed (wall clock) time (h:mm:ss or m:ss): 20:53.67
    Maximum resident set size (kbytes): 238280
```

`htseq-count_SRR25630385_38070689.err`

```
    Percent of CPU this job got: 99%
    Elapsed (wall clock) time (h:mm:ss or m:ss): 15:12.21
    Maximum resident set size (kbytes): 238252
```

Reverse: `htseq-count_rev_SRR25630409_38070692.err`

```
    Percent of CPU this job got: 99%
    Elapsed (wall clock) time (h:mm:ss or m:ss): 20:57.13
    Maximum resident set size (kbytes): 238380
```

`htseq-count_rev_SRR25630385_38070689.err`

```
    Percent of CPU this job got: 99%
    Elapsed (wall clock) time (h:mm:ss or m:ss): 15:26.62
    Maximum resident set size (kbytes): 238084
```

All commands run successfully.. scripts are QAA and QAA2. Didn't run them in a loop so it would be parallelized and faster. Can write a separate script in the future to call the script passing through what the SRR should be based off a given list files or something similar.

9/10/2025 Realized I was missing some of the plots.. added the calls for them to the QAA except for the python plotting. That still needs argparse added so leaving that as a for later thing and hardcoding that for now since it isn't really crucial to the analysis, given fastqc gives the same plots.

Wrapping up the Rmd report, `QAA_report.Rmd`.

Forgot to add justification for forward/reverse setting for htseq-count. Did code in the report markdown and added that.

Submitting now (finally).