**Day 4 Worksheet – Trimmomatic**

Author: Jessica Westfall 2021

Edited: Ariel Eraso, Lynn Sanford 2022-23

*Introduction: Now that we have evaluated our sequence library initially to determine if the libraries are worth analyzing, we will do some “cleaning up” by trimming unwanted sequences such as adapter sequences. This step is necessary for improved alignment and mapping to the reference genome downstream. Once trimming is completed, we will reevaluate our trimmed files with FastQC for quality to decide if we will move forward with mapping.*

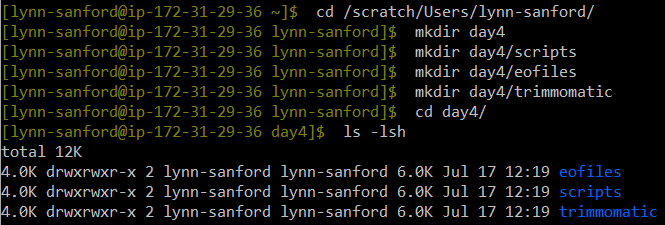
** Note: The directory and username used in the screenshot will be for my working directory and username and will be different than yours.

**Make working directories**

Yesterday, we made working directories for running fastQC. Repeat the same process, but this time we will make a directory for trimmomatic.

1. Use command **pwd** to determine what directory you are in and if necessary, **cd** to the directory that you want to place your new trimmomatic directory in.

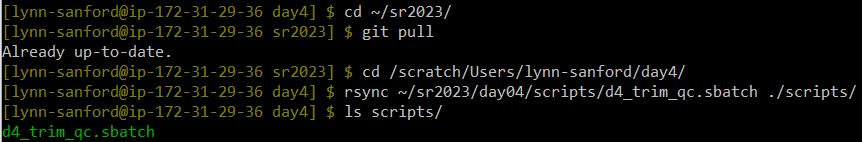
2. Make a few new directories using the **mkdir** command. Use command **ls -lsh** to confirm the folders are present.



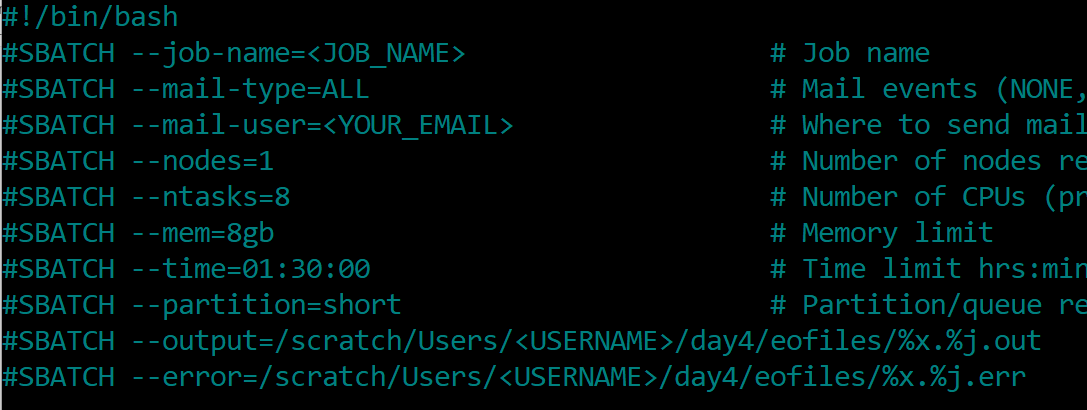
**Trimmomatic**

3. Git pull within the sr2023 repo. Then copy (**rsync**) the **d4\_trim\_qc.sbatch** script from the day04/scripts/ into your script directory. Use **ls -lsh** to confirm the file is present in the directory. You can **ls** with an absolute path as well as relative path.

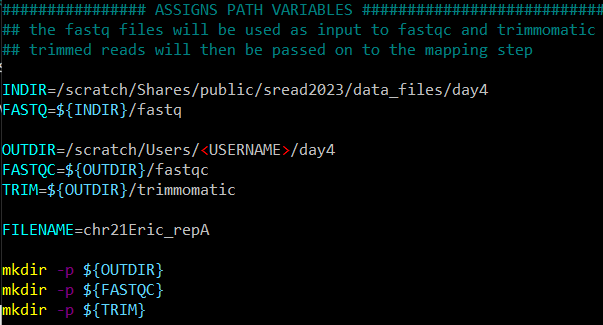
To copy the script, the command syntax is **rsync <input> <output>**

****

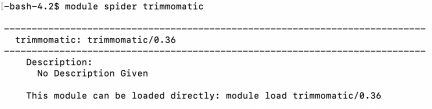
4. Edit the sbatch script by using **vim <sbatch>** to open a text editor on your sbatch script. Type **i** to toggle into edit/insert mode. Similar to the previous exercise you will need to change the job name, user email, and the standard output and error log directories. Change the **–job-name=<JOB\_NAME>** to a name related to the job you will be running, for example ‘trim\_qc’. Additionally you will want to change the **–mail user=<YOUR\_EMAIL>** to your email, as well as the path to your eofiles directory for the standard output (**--output)** and error log (**--error)**. The **%x** will be replace by your **-job name** and the **%j** will be replace by the job id that will be assigned by the job manager when you run your sbatch script.



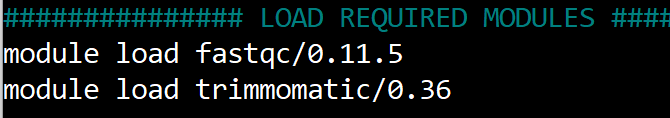
For this script, I will be change my CPU and nodes for trimmomatic which can use multiple processors per input file. I am going to request 1 node, 8 tasks, 8gb of memory and 90 minutes of wall time.

5. Assigning path variables will make your scripts easier to read. In addition, this makes it easier to reference to a given path and utilize it in your scripts. For the **INDIR=**change the path to where the data files directories are located and specifically the fastq data. For the **OUTDIR=**, point to the appropriate output file directories for our fastQC and trimmed fastq files. I also use the command **mkdir -p** just in case for my output directories.

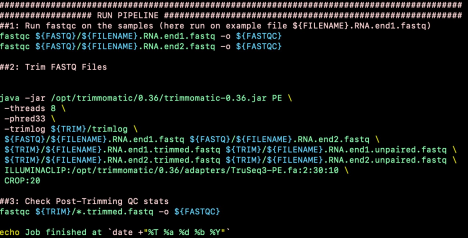
6. Load the require modules for running this pipeline. We will be using fastQC and the trimming program trimmomatic. Similar to fastqc, if you are not sure which version of the program is available on the cluster you can use the command **module spider <string>** to find the available versions.



Now I can add the appropriate versions for the modules I want to load in the pipeline.



7. For the meat of the script, we will be running 3 steps in the pipeline. (1) To run fastQC on the sample, (2) trim the fastQC and (3) reevaluate the quality of the trimmed fastq with fastQC.



In this script we are running paired end reads. Trimmomatic can be used on both single end or paired-end reads. When setting your parameters use the appropriate adapters.

Below are the syntaxes needed to run trimmomatic:

Illuminaclip parameter (see below for quick reference to trimming)

ILLUMINACLIP:<path\_adapters\_fasta>:<seed\_mismatches>:

<palindrome\_clip\_threshold>:<simple\_clip\_threshold> LEADING:<quality>

TRAILING:<quality> SLIDINGWINDOW:<window\_size>:<required\_quality> MINLEN:<length>

For single-end reads

java jar /opt/trimmomatic/0.36/trimmomatic-0.36.jar SE [ -threads <n> ] [ -phred33 | -phred64 ] [ -trimlog <output\_trimlog> ] <input\_file>

<output\_file> ILLUMINACLIP

For pair-end reads

java jar /opt/trimmomatic/0.36/trimmomatic-0.36.jar PE [ -threads <n> ] [ -phred33 | -phred64 ] [ -trimlog <output\_trimlog> ] <input\_file1>

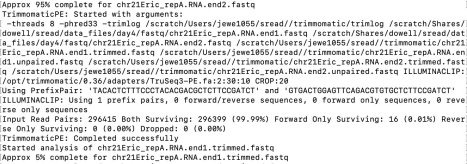
<input\_file2> <output\_fileP1> <output\_fileU1> <output\_fileP2>

<output\_fileU2> ILLUMINACLIP

Recall that the ‘**\**’ at the end is used to break the code up for clarity purpose. We can write this syntax as a single line but it is harder to read. If ‘**\**’ does not change color as you see above (it may not be yellow, but it should change), you may have an extra space after the ‘**\**’. Remove that space or your code will not run properly.

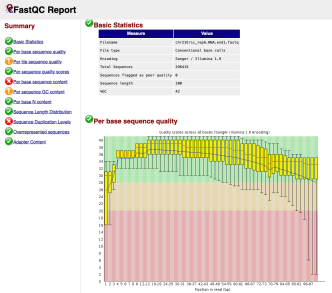
8. Save your sbatch script. Press **esc** to exit out of edit mode, then type **:wq**. This will write/save (w) and quit (q) the script.

9. Let’s run the script. Submit the job to the job manager SLURM using the command **sbatch <sbatch\_file>**. The job manager will assign a job id to your run. 12. This pipeline has more tasks than the previous worksheet, so you will want to check the status of your job using the command **squeue -u <username>** to see if the job is running (R) or completed (C). If there are any errors, often time these are just typos in your scripts, you will want to access your error log to make necessary corrections. I will **ls -lahtr /path/to/eofiles** to get the name of the error log for the job id so that I can view it using **more**, **less**, or **cat**. I use **-tr** with the **ls** command to get order my files based on time so I can quickly find the latest error log.

10. Check the error log to find information about the fastqc and trimming job.

**Pre- and post-trim fastQC**

Pre-trimming



Post-trimming

