Task 1: Get fastq files

1. Create a new directory in your directory ~/**sread2019/** directory named **quiz/**. Remember that the " ~ " indicates your home directory. You may also put these files into your scratch home directory **(/scratch/Users/<your_id>/)** which is good practice for this kind of job. Within the **quiz/** directory make the following subdirectories:

    a. **fastq/**

    b. **trimmed/**

    c. **mapping/**

    d. **bam/**

    e. **scripts/**

    f. **e_and_o/** **NOTE:** all sbatch error and output should go to this directory

    g. **answers/**

2. Copy the script **template.slurm** from **/scratch/Workshop/SR2019/5_assessment/quiz/sbatch/** to ~/**sread2019/quiz/scripts/** with the name **rsync.slurm**.

   Edit the script to create a job which will rsync files **day5_01.fastq.gz** and **day5_02.fastq.gz** from **/scratch/Workshop/SR2019/5_assessment/quiz/fastqs/** to ~/**sread2019/quiz/fastq/**. Review how you can copy multiple files within a directory.

Task 2: Quick evaluation of fastq file

3. Determine how many reads are in fastq file. You can use the wc command to determine line counts. Review fastq format to determine from the line counts how many reads are in your files.

    a. Open a file inside **answers/** called **answers.txt** and type:

       "There are _____ reads in the file day5_01.fastq.gz and _____ reads in the file day5_02.fastq.gz"

    b. Based on the wc command output, calculate the number of reads and fill in the blank.

Task 3: FastQC

4. Create a new slurm file for running FastQC:

    a. Open your file **rsync.slurm** in vim. Press [esc] and type **:w fastqc.slurm**. You have now written a new sbatch file in the same directory that will have all of the same contents as **rsync.slurm**. This is a convenient way to create new scripts that still have your sbatch options at the top.

b. Exit out of your current sbatch file and open **fastqc.slurm**.

c. Make any necessary edits to this script to run FastqQC on the fastq file and direct the output to ~**sread2019/quiz/fastq/**.

d. Open your file **answers.txt** from **answers/** directory. On new lines, type:

"The read length for the file day5_01.fastq.gz is **_____**. In this data set, there are _NO/YES_ adapters and (if yes) they are **_____** adapters.

The read length for the file day5_02.fastq.gz is **_____**. In this data set, there are _NO/YES_ adapters and (if yes) they are _____ adapters."

Task 4: Trimming your dataset and QC via FastQC

5. Using the same steps described in question 4, create a new sbatch script named **trim.slurm** and edit the file to run the program Trimmomatic on both fastq files.

   a. Go to the line below where your script says "*Load necessary modules for the job*" and make sure your cursor is on that line. Press [esc] followed by **dG**. You have now deleted everything below your selected line. **HINT:** If you made a mistake, press [esc] again and then **u** to undo your last edit

   b. Now, make any necessary edits to this script to run Trimmomatic on the fastq files and direct the output to ~**sread2019/quiz/fastq/**.

6. For Trimmomatic, the output should go to ~**/sread2019/quiz/trimmed** . Save the file names as **day5_01.trimmed.fastq** and **day5_02.trimmed.fastq.**

   NOTE: Remember that Java is really greedy, so be sure to include the flag to limit the number of memory for this job.

   Note about SE vs PE sequencing: These are single-end reads using TruSeq2 PE adapters. You may run a single-end sequencing reaction with paired-end adapters. If the sequencing quality is low, I could then resequence these same samples using a paired-end sequencing reaction. This is more expensive, so many experiments will use single-end.

7. Using your **fastqc.slurm** script, edit it to again run FastQC on the trimmed files. The output should go to the **/trimmed** directory with the trimmed fastq files.

8. Open your file **answers.txt** from **answers/** directory. On new lines, type:

"For the dataset day05_01:

There are **_____** reads in the fastq file after trimming.

The minimum read length is **_____** and the maximum read length is **_____**.

My assessment of the quality of this dataset prior to trimming was _____ (*select one parameter)* based on the FastQC report. After trimming, the quality of the dataset was _____ (*did the trimming improve the dataset? Select one parameter to discuss*).

For the dataset day5_02:

There are _____ reads in the fastq file after trimming.

The minimum read length is _____ and the maximum read length is _____.

My assessment of the quality of this dataset prior to trimming was _____ (*select one parameter)* based on the FastQC report. After trimming, the quality of the dataset was _____ (*did the trimming improve the dataset? Select one parameter to discuss*)."

9. Compare the two fastq files before and after trimming based on the FastQC reports. Based on the reports, which file do you think needs further assessment before proceeding to mapping. In the **answers.txt** file from **answers/** directory, name at least one parameter that lead you to your answer.

Task 5: Mapping to genome.

In question 9, you were asked it you would proceed with mapping based on the FastQC reports. Let's map both of the dataset to human hg38 reference genome and see if your intuition was correct.

10. Using the same steps described in question 4, create a new sbatch script named **mapping.slurm** and edit the file to run the program Hisat2 on both fastq files. Map the reads to human hg38 reference genome.
11. Make any necessary edits to this script to run Hisat2 on the trimmed fastq files and direct the output to ~**sread2019/quiz/mapping/**. The output should be sam files.
12. Open your file **answers.txt** from **answers/** directory. On new lines, type:

"For the dataset day05_01, _____% of reads mapped to hg38 reference genome.

For the dataset day05_02, _____% of reads mapped to hg38 reference genome."

Task 6: Create a README.txt file for documentation

13. Create a new file in ~**/sread2019/quiz/** named **README.txt**. For each of the two fastq files, fill in the blanks in the following paragraph in **README.txt.**

day5_01.fastq.gz

There are _____ reads in the original file and they are ______ nt long.

I ran a quality report and there are _______ reads with adapters.

I removed adapters by _______.

After removal of the adapters there are ______ reads and they range in length between ______ and _______.

I trimmed ______ amount of reads from my original fastq file.

______% of the trimmed reads mapped to hg38.

day5_02.fastq.gz

There are _____ reads in the original file and they are ______ nt long.

I rand a quality report and there are _______ reads with adapters.

I removed adapters by _______.

After removal of the adapters there are ______ reads and they range in length between ______ and _______.

I trimmed ______ amount of reads from my original fastq file.

______% of the trimmed reads mapped to hg38.

The results for fastq pre-trimming can be found in __________ (directory). The results for fastq post-trimming can be found in _____________ (directory). The trimmed and mapped fastq can be found in _____________ (directory).

Task 7: View the files in IGV (Optional)

After mapping fastq dataset to the reference genome, the result is a large SAM file. It is usually best to convert these SAM files to BAM files for memory, and in addition most downstream application uses a BAM file. Lastly, it is important to do an initial assessment of your mapped reads visually with IGV.

14. Using the same steps described in question 4, create a new sbatch script named **samstool.slurm** and edit the file to run the program Samtools on the trimmed and mapped fastq files.
15. Make any necessary edits to this script to run Samtools on the trimmed fastq files and direct the output to ~**sread2019/quiz/bam/**. Your script to convert SAM to BAM format. In addition, sort your BAM files then index the files.
16. View your sorted BAM files in IGV.

Resource links:

1. FastQC guide: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/
2. QC fail. Useful for looking through different FastQC plots of sequencing libraries that failed: https://sequencing.qcfail.com/software/fastqc/
3. Trimmomatic guide: http://www.usadellab.org/cms/?page=trimmomatic

4. BBDuk guide: https://jgi.doe.gov/data-and-tools/bbtools/bb-tools-user-guide/bbduk-guide/
5. Mapping via HISAT2 guide: https://ccb.jhu.edu/software/hisat2/manual.shtml
6. Illumina igenome. Provides packages for genome for model organism: https://support.illumina.com/sequencing/sequencing_software/igenome.html
7. Samtools guide: https://www.htslib.org/doc/samtools.html