Kayla Hudson
BIOL 3411
14 February 2024, 21 February 2024, 15 March 2024

Deliverable 2C: Working with Illumina sequencing reads.

✅✅ 1. Log onto a computing node before beginning this exercise.

- ✅ Perform the operations below first using the week4 sequence files.
- ✅ Then repeat the operations using the Day data files in the shared folder.

✅✅ 2. You've downloaded and uncompressed a set of sequencing files into their own directory. The original file ended in tar.gz. Now the various files that resulted from uncompressing end in: **.gz**

✅✅ 3. Write a shell script that:

- unzips each file but leaves the original copy intact

```
#!/bin/bash
#SBATCH --partition=short
#SBATCH --job-name=biol3411_unzip_and_mkdir_20240221
#SBATCH --time=24:00:00
#SBATCH --nodes=1
#SBATCH --cpus-per-task=2
#SBATCH --mem=256G
#SBATCH --mail-user=hudson.ka@northeastern.edu
#SBATCH --mail-type=ALL
#SBATCH --output=%j.output
#SBATCH --error=%j.error

/courses/BIOL3411.202430/students/hudson.ka/class_examples/fastq_work_wk6_20240221
for zipped in *.gz; do gunzip -c "$zipped" > "$zipped.fastq"; done
```

✅✅ 4. Now the directory has two sets of files, each with a different extension. Make a new directory and move all the unzipped files to it.

```
[hudson.ka@c2000 fastq_work_wk6_20240214]$ mkdir unzipped_fastq
[hudson.ka@c2000 fastq_work_wk6_20240214]$ mv *.fastq unzipped_fastq/
```

✅✅ 5. But wouldn't it have been more efficient to include this step in the original shell script? Let's try that. First, remove the directory that contains the unzipped files.

```
[hudson.ka@c2000 fastq_work_wk6_20240214]$ rm -ir unzipped_fastq/
ls
```

✅✅ 6. Then go back to the shell script and amend it to include commands to:

- make a new directory

- place the unzipped files there

```
#!/bin/bash
#SBATCH --partition=short
#SBATCH --job-name=biol3411_unzip_and_mkdir_20240221
#SBATCH --time=24:00:00
#SBATCH --nodes=1
#SBATCH --cpus-per-task=2
#SBATCH --mem=256G
#SBATCH --mail-user=hudson.ka@northeastern.edu
#SBATCH --mail-type=ALL
#SBATCH --output=%j.output
#SBATCH --error=%j.error

/courses/BIOL3411.202430/students/hudson.ka/class_examples/fastq_work_wk6_20240221
for zipped in *.gz; do gunzip -c "$zipped" > "$zipped.fastq"; done
mkdir unzipped_fastq
mv *.fastq unzipped_fastq/
```

☑☑ 7. Use the grep command to determine the number of reads in one unzipped file

```
[hudson.ka@login-00 unzipped_fastq]$ grep -i -c "^@" SRR6805880.tiny.fastq.gz.fastq
```

☑ 8. then in all the unzipped files, printing to the screen

```
[hudson.ka@login-00 unzipped_fastq]$ grep -i -c "^@" *.fastq
```

☑☑ 9. then in all the unzipped files, printing to a new file.

```
[hudson.ka@login-00 unzipped_fastq]$ grep -i -c "^@" *.fastq > read_counts_all_seqs.txt
```

☑ 10. Use head to examine the start of each read and see if there is an obvious adapter sequence to trim. Each read starts with the same small section of nucleotides: TGCAG. This is likely an adaptor that has been added to each sequence. This will need to be trimmed.

☑☑ 11. Now you want to generate quality control reports for each of the files, using fastqc/0.11.9. This package requires that you first load the module OpenJDK/19.0.1, and the command operates on zipped files (ending in fastq.gz).
First test out the operation on only one file, using the command line.

☑☑ 12. Now try performing this task on all the files, in two different ways-- using a shell script and using a bash script.

Shell Script:

```
module load OpenJDK/19.0.1
module load fastqc/0.11.9

fastqc *.gz
```

```
mkdir html_qc_reports_sh_output
mv *.html html_qc_reports_sh_output

echo "Script finished. All qc reports generated!"
```

Bash Script:

```
#!/bin/bash
#SBATCH --partition=short
#SBATCH --job-name=biol3411_qc_reports_20240221
#SBATCH --time=24:00:00
#SBATCH --nodes=1
#SBATCH --cpus-per-task=2
#SBATCH --mem=256G
#SBATCH --mail-user=hudson.ka@northeastern.edu
#SBATCH --mail-type=ALL
#SBATCH --output=%j.output
#SBATCH --error=%j.error

/courses/BIOL3411.202430/students/hudson.ka/class_examples/fastq_work_wk6_20240221
module load OpenJDK/19.0.1
module load fastqc/0.11.9
for zipfile in *.gz; do fastqc "$zipfile"; done
mkdir html_qc_reports
mv *.html html_qc_reports
```

☑☑ 13. After generating the new files, make a separate directory and put the .html files into it (you can try adding this step directly tur script from above, if you like).

☑☑ 14. Open one of the .html files (you'll need to do this through the OOD, using a browser to view it).
Completed— note, you have to download the html file from OOD before viewing it

☑ 15. Watch this video on interpreting the report.

All Unit 2 work is due by Mar 10.

Each deliverable is complete when you have:

- answered each question
- saved a terminal session or screenshots demonstrating your performance of the commands (on Discovery or posted on GitHub)
- indicated in a "TOC" (table of contents) file where your work is found (GitHub repo or specific path/name_of_file on the cluster).

Sources referenced in completing this worksheet:

- use gunzip with -c option to decompress file and keep original zipped file: https://superuser.com/questions/45650/how-do-you-gunzip-a-file-and-keep-the-gz-file
- grep: https://www.hostinger.com/tutorials/grep-command-in-linux-useful-examples/
- specify email options in bash script: https://hpcc.umd.edu/hpcc/help/jobs.html#email
- squeue documentation: https://slurm.schedmd.com/squeue.html
    - squeue -u Hudson.ka == squeue –me
- squeue status and reason codes: https://curc.readthedocs.io/en/latest/running-jobs/squeue-status-codes.html
- Loop structure: https://www.cyberciti.biz/faq/linux-unix-shell-unzipping-many-zip-files/