Small Group Exercise #1:

Assessing Quality and QC of FASTQ files using the FASTX-Toolkit

1. Explore the documentation for the FASTX-Toolkit using the documentation at https://hpcdocs.asc.edu.

    i. Login to the portal.
    ii. Select 'Other bioinformatics' from left menu
    iii. Select 'Fastx-Toolkit' and read the documentation

2. As covered in the documentation above, you will need a script to run FASTX-Toolkit. An example script named `FASTX_example.sh` is located at: `/home/shared/biobootcamp/data/example_ASC_queue_scripts`. For practice, survey the contents of this directory using the `ls -alh` (BTW, what do these flags do for the `ls` command?) and absolute path of the directory to preview the names of some other scripts you will be using during the Bootcamp.

3. You will need to make a copy of the above script to your home directory. Let's do that:

    i. `cd` (to take you to the top level of your home directory)
    ii. `mkdir fastx_working_example` (create a directory to work in)
    iii. `cd fastx_working_example` (to change into the directory)
    iv. `cp /home/shared/biobootcamp/data/example_ASC_queue_scripts/FASTX_example.sh .` (to copy the script to your working directory; note the `.` at the end of this command, specifying the current directory)

4. Now use a text editor to open the script (nano is commonly available on most Linux systems and easy to work with):

    i. `nano FASTX_example.sh`

5. Note that comments in the script indicate where you will be adding your commands once you decide what you specifically want to do. Close nano by holding down `Ctrl` and then `x` on your keyboard.

6. Let's copy some raw FASTQ files to our working directory:

    i. `cp /home/shared/biobootcamp/data/Lamellibrachia_luymesi_sequence_reads_for_assembly/Lamellibrachia_luymesi_transcriptomic_sub1M_L001_R1_001.fastq .`

ii. `cp /home/shared/biobootcamp/data/Lamellibrachia_luymesi_sequence_reads_for_assembly/Lamellibrachia_luymesi_transcriptomic_sub1M_L001_R2_001.fastq .`

7. Quantify the number of reads in each file and error check them using `wc -l` and `fastQValidator` (https://github.com/statgen/fastQValidator), respectively, and make note of what they are. For `fastQValidator`, run the command without any flags or a filename to receive a Help message and options on how to run the program and the error check (i.e., see the 'Examples' section in the Help message).

8. To generate boxplot graphs of quality for each of the read sets using the FASTX-Toolkit, you will first need to generate statistics for each read set. This can be done in the following manner:

   i. `module load fastx/0.0.14` (to load the FASTX-Toolkit into your workspace)
   ii. `fastx_quality_stats -h` (to get a listing of available options, also "fast" with tab complete to see all modules)
   iii. the three options that you will need to provide the fastx_quality_stats command with are:
      a. `-i <INFILE_NAME>.fastq`
      b. `-o <OUTFILE_NAME>.stats`
      c. `-Q33` (this tells fastx_quality_stats to use Phred 33 as the quality score range)
   iv. Draft your two fastx_quality_stats commands by writing them out by hand. Do not execute them, just write them down to add to your script shortly. A hint for the first one is below. NOTE: the backslash `\` placed at the end of the first line below means "continue to the next line" when the script is read by the system. Thus, the command below is considered a single line but can be broken up in your script like this in order to increase legibility:

```
fastx_quality_stats \
   -Q33 \
   -i Lamellibrachia_luymesi_transcriptomic_sub1M_L001_R1_001.fastq \
   -o Lamellibrachia_luymesi_transcriptomic_sub1M_L001_R1_001.stats
```

9. We will generate the boxplot graphs of the quality score statistics for each read set next:

   i. `fastq_quality_boxplot_graph.sh -help` (to get a listing of available options)
   ii. the two minimal options that you will need to provide the fastq_quality_boxplot_graph.sh command with are:
      a. `-i <INFILE_NAME>.stats`
      b. `-o <OUTFILE_NAME>.png`
   iii. Draft your two `fastq_quality_boxplot_graph.sh` commands out by hand. Use the example above as a starting point.

10. Once you have outlined your commands, add them to the appropriate section of the `FASTX_example.sh` script using nano (you should have 4 similar, but different, commands, each on a separate line, in your script file when you are done).

11. Save your script and exit out of nano. Review the ASC Queue Tutorial and submit the script to the ASC queue system using the directions at the bottom of the script.

12. Monitor your job using `squeue` and watching for new content (i.e., creation of the two `*.stats` and `*.png` files) in the working directory.

13. Once the job is complete, download the two `*.png` files to your laptop and open them in any image viewer or web browser.

14. As you examine the boxplots, answer the following questions:

    i. What are the x- and y-axes?
    ii. What do the boxplots represent at each interval?
    iii. Do the two graphs look similar or different?
    iv. How do they differ from each other?
    v. `cp /home/shared/biobootcamp/data/Various/qual_score_boxplot_explained_2018.pdf .` (what did you just do in this command (i.e., remember what the "." at the end means)?). Now download this PDF to your laptop like you did for the `*.png` files above for a further explanation and interpretation of quality score boxplots.

15. Let's see how the boxplots change once QC is introduced into the equation. NOTE: for this part of the exercise, pair with another person in your group. Discuss with them 1) potential options to use for (b) below, 2) what options might be most appropriate given the data and 3) independently execute the same commands (i.e., act as each other's "control" in this experiment).

    i. If you logged out of the ASC in the previous step you will need to run `module load fastx/0.0.14` again.
    ii. `fastq_quality_trimmer -help` (to get a listing of available options)
    iii. the five options that you will need to provide the fastq_quality_trimmer command with are:
        a. `-i <INFILE_NAME>.fastq`
        b. `-o <OUTFILE_NAME>.QCed.fastq`
        c. `-Q33 (see #8 above for details)`
        d. `-t` (see `fastq_quality_trimmer -help` for definition and potential values)
        e. `-l` (see `fastq_quality_trimmer -help` for definition and potential values)
    iv. Draft your two fastq_quality_trimmer commands out on paper.

16. Now add these two fastq_quality_trimmer commands to the beginning of the workflow in your FASTX_example.sh script. You can keep your other four previous commands from above BUT edit them by: 1) changing their input flags to use the appropriate `*.QCed.fastq` files and 2) modifying their subsequent output `*.stats` and `*.png` file names (i.e., so you can distinguish them from the ones you originally created in #8 and #9). NOTE: the sequential order of the commands is important, so (a) trim first, (b) produce stats for trimmed data next and (c) graph those stats last. Save your script and exit out of nano. Now submit it to the ASC queue system using the directions at the end of the script like you did in Step #11 above.

17. Monitor your job using `squeue` and watching for new content (i.e., creation of the two new `*.fastq`, `*.stats` and `*.png` files) in the working directory.

18. Once the job is complete, transfer the two new `*.png` files to your own computer and open them in any image viewer or web browser. Also open the two original `*.png` files from #13.

19. As you examine the new boxplots and compare them to the originals (i.e., raw data), answer the following questions:

    i. How have the boxplots changed following QC?
    ii. Do the two graphs from the QC data now look more similar or different to each other?
    iii. Has the amount of data (i.e., number of reads) changed in each file? If so, by how much? (you can use `wc -l` and `fastQValidator` for this quantification).
    iv. How does one "control" for the impact that QC might have on downstream analyses?

Notes:
Everyone has a symlink or shortcut to the shared bootcamp directory called `biobootcamp` in their home folder. As in examples above we can type `biobootcamp` in place of the absolute path `/home/shared/biobootcamp/` e.g., `ls biobootcamp/data/example_ASC_queue_scripts/` for this exercise and all other bootcamp activities this week. If you aren't currently in your home directory we use `~/biobootcamp`, where the tilde always means your home directory.