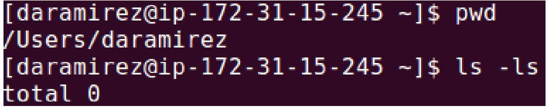
Worksheet 4.1 - Checking fastq file sequencing quality using fastQC

Authors: Mary Allen & Daniel Ramirez

FastQC webpage: https://www.bioinformatics.babraham.ac.uk/projects/fastqc/

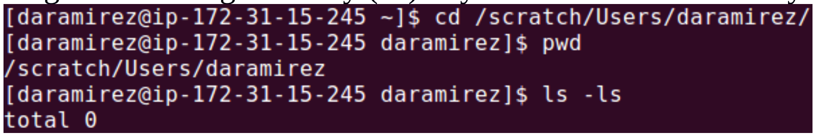
Username: Screenshots show 'daramirez', though you will see your own username!

1. Using an appropriate terminal, log on to the cluster where you will use **fastQC**:
   a. Use ***pwd*** to make sure you know where you are.



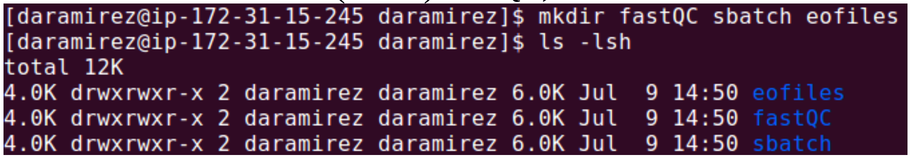
```
[daramirez@ip-172-31-15-245 ~]$ pwd
/Users/daramirez
[daramirez@ip-172-31-15-245 ~]$ ls -ls
total 0
```

   b. Change the working directory (***cd***) to your own scratch directory.



```
[daramirez@ip-172-31-15-245 ~]$ cd /scratch/Users/daramirez/
[daramirez@ip-172-31-15-245 daramirez]$ pwd
/scratch/Users/daramirez
[daramirez@ip-172-31-15-245 daramirez]$ ls -ls
total 0
```

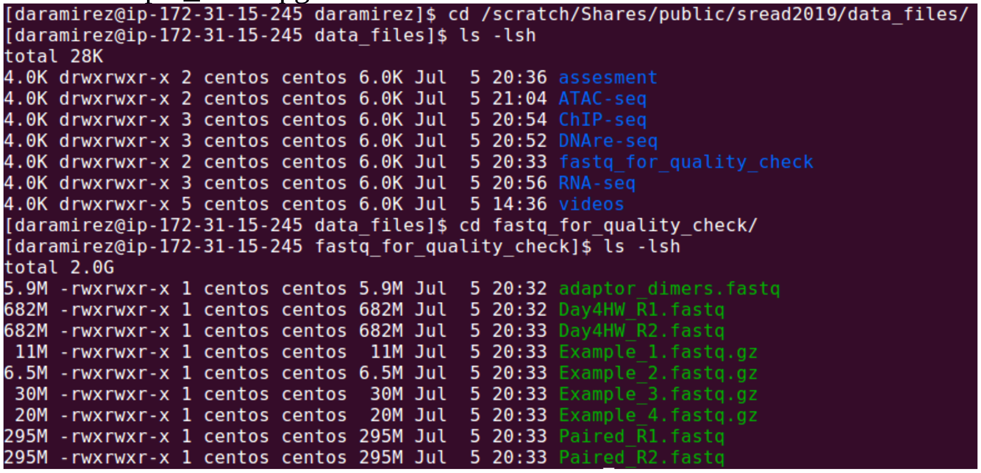
2. Make 3 new directories/folders (***mkdir***): fastQC, sbatch and eofiles.



```
[daramirez@ip-172-31-15-245 daramirez]$ mkdir fastQC sbatch eofiles
[daramirez@ip-172-31-15-245 daramirez]$ ls -lsh
total 12K
4.0K drwxrwxr-x 2 daramirez daramirez 6.0K Jul  9 14:50 eofiles
4.0K drwxrwxr-x 2 daramirez daramirez 6.0K Jul  9 14:50 fastQC
4.0K drwxrwxr-x 2 daramirez daramirez 6.0K Jul  9 14:50 sbatch
```

   These are the directories that will contain the results from fastQC, the error and output files generated by your batch scripts jobs, and the batch scripts themselves.

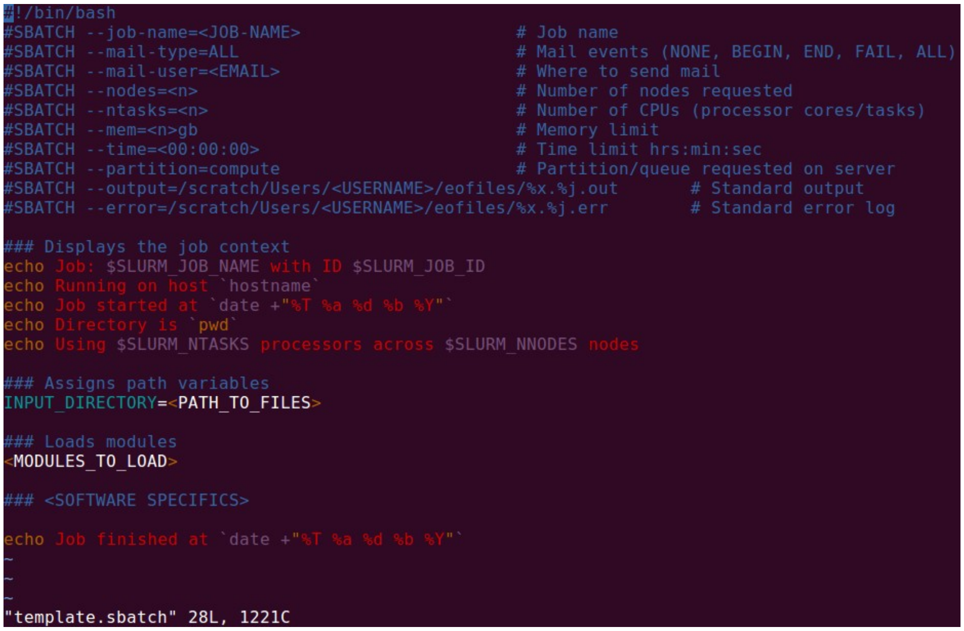
3. Check the fastq data files in the following public directory using ***cd*** and ***ls***: /scratch/Workshop/SR2019/4_qc/fastq . There are many fastq files in that directory. Some of them are zipped (.gz), some are not. Pick one. In the following examples here I picked "Example_1.fastq.gz".



```
[daramirez@ip-172-31-15-245 daramirez]$ cd /scratch/Shares/public/sread2019/data_files/
[daramirez@ip-172-31-15-245 data_files]$ ls -lsh
total 28K
4.0K drwxrwxr-x 2 centos centos 6.0K Jul  5 20:36 assesment
4.0K drwxrwxr-x 2 centos centos 6.0K Jul  5 21:04 ATAC-seq
4.0K drwxrwxr-x 3 centos centos 6.0K Jul  5 20:54 ChIP-seq
4.0K drwxrwxr-x 3 centos centos 6.0K Jul  5 20:52 DNAre-seq
4.0K drwxrwxr-x 2 centos centos 6.0K Jul  5 20:33 fastq_for_quality_check
4.0K drwxrwxr-x 3 centos centos 6.0K Jul  5 20:56 RNA-seq
4.0K drwxrwxr-x 5 centos centos 6.0K Jul  5 14:36 videos
[daramirez@ip-172-31-15-245 data_files]$ cd fastq_for_quality_check/
[daramirez@ip-172-31-15-245 fastq_for_quality_check]$ ls -lsh
total 2.0G
5.9M -rwxrwxr-x 1 centos centos 5.9M Jul  5 20:32 adaptor_dimers.fastq
682M -rwxrwxr-x 1 centos centos 682M Jul  5 20:32 Day4HW_R1.fastq
682M -rwxrwxr-x 1 centos centos 682M Jul  5 20:33 Day4HW_R2.fastq
 11M -rwxrwxr-x 1 centos centos  11M Jul  5 20:33 Example_1.fastq.gz
6.5M -rwxrwxr-x 1 centos centos 6.5M Jul  5 20:33 Example_2.fastq.gz
 30M -rwxrwxr-x 1 centos centos  30M Jul  5 20:33 Example_3.fastq.gz
 20M -rwxrwxr-x 1 centos centos  20M Jul  5 20:33 Example_4.fastq.gz
295M -rwxrwxr-x 1 centos centos 295M Jul  5 20:33 Paired_R1.fastq
295M -rwxrwxr-x 1 centos centos 295M Jul  5 20:33 Paired_R2.fastq
```

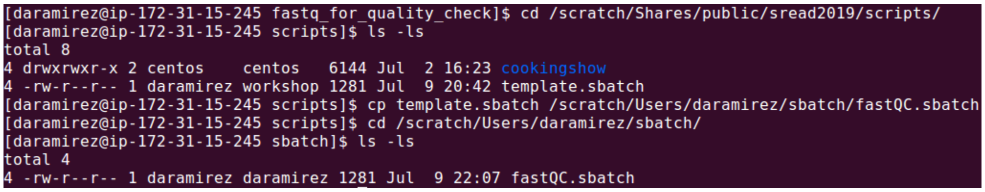
4. Find and explore the contents (e.g. ***vim <file>***) of the script batch template “template.sbatch” in the directory: /scratch/Workshop/SR2019/4_qc/sbatch

    You cannot edit, only look. The top of the file has information for the queue. The middle section contains job specific documentation. We will change this file so that it can be used for fastQC. This is your template. When you are done looking use ***:q!*** then press enter to exit the file.



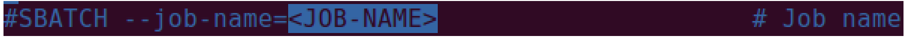
```
#!/bin/bash
#SBATCH --job-name=<JOB-NAME>                     # Job name
#SBATCH --mail-type=ALL                           # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=<EMAIL>                       # Where to send mail
#SBATCH --nodes=<n>                               # Number of nodes requested
#SBATCH --ntasks=<n>                              # Number of CPUs (processor cores/tasks)
#SBATCH --mem=<n>gb                               # Memory limit
#SBATCH --time=<00:00:00>                         # Time limit hrs:min:sec
#SBATCH --partition=compute                       # Partition/queue requested on server
#SBATCH --output=/scratch/Users/<USERNAME>/eofiles/%x.%j.out      # Standard output
#SBATCH --error=/scratch/Users/<USERNAME>/eofiles/%x.%j.err       # Standard error log

### Displays the job context
echo Job: $SLURM_JOB_NAME with ID $SLURM_JOB_ID
echo Running on host `hostname`
echo Job started at `date +"%T %a %d %b %Y"`
echo Directory is `pwd`
echo Using $SLURM_NTASKS processors across $SLURM_NNODES nodes

### Assigns path variables
INPUT_DIRECTORY=<PATH_TO_FILES>

### Loads modules
<MODULES_TO_LOAD>

### <SOFTWARE SPECIFICS>

echo Job finished at `date +"%T %a %d %b %Y"`
~
~
~
"template.sbatch" 28L, 1221C
```

5. Copy the script batch “template.sbatch” that you just looked at to your previously created sbatch directory “/scratch/Users/<username>/sbatch/” using the new name “fastQC.sbatch” (***cp <input> <output>***). Check that copying worked by moving to the sbatch directory and listing its contents (hint: ***cd*** & ***ls***).



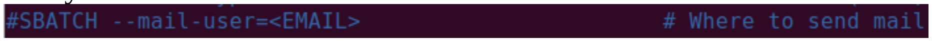
```
[daramirez@ip-172-31-15-245 fastq_for_quality_check]$ cd /scratch/Shares/public/sread2019/scripts/
[daramirez@ip-172-31-15-245 scripts]$ ls -ls
total 8
4 drwxrwxr-x 2 centos    centos   6144 Jul  2 16:23 cookingshow
4 -rw-r--r-- 1 daramirez workshop 1281 Jul  9 20:42 template.sbatch
[daramirez@ip-172-31-15-245 scripts]$ cp template.sbatch /scratch/Users/daramirez/sbatch/fastQC.sbatch
[daramirez@ip-172-31-15-245 scripts]$ cd /scratch/Users/daramirez/sbatch/
[daramirez@ip-172-31-15-245 sbatch]$ ls -ls
total 4
4 -rw-r--r-- 1 daramirez daramirez 1281 Jul  9 22:07 fastQC.sbatch
```

6. Complete the new “fastQC.sbatch” file with the right content to run fastQC. (hint: transition to insert mode by pressing ***i*** if using vim.)
    a. Change the name of the script batch from <JOB-NAME> to something more useful, such as “fastQC”.



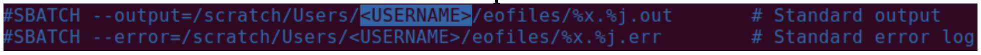
```
#SBATCH --job-name=<JOB-NAME>                     # Job name
```

    b. Replace <EMAIL> with your own email address to which you want to receive any notifications.



```
#SBATCH --mail-user=<EMAIL>                       # Where to send mail
```

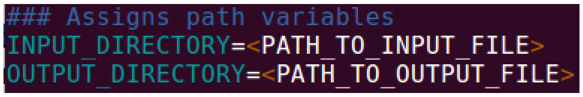
c. Replace <USERNAME> with your own username to complete the path directory to where to store the error and output files.



```
#SBATCH --output=/scratch/Users/<USERNAME>/eofiles/%x.%j.out       # Standard output
#SBATCH --error=/scratch/Users/<USERNAME>/eofiles/%x.%j.err        # Standard error log
```

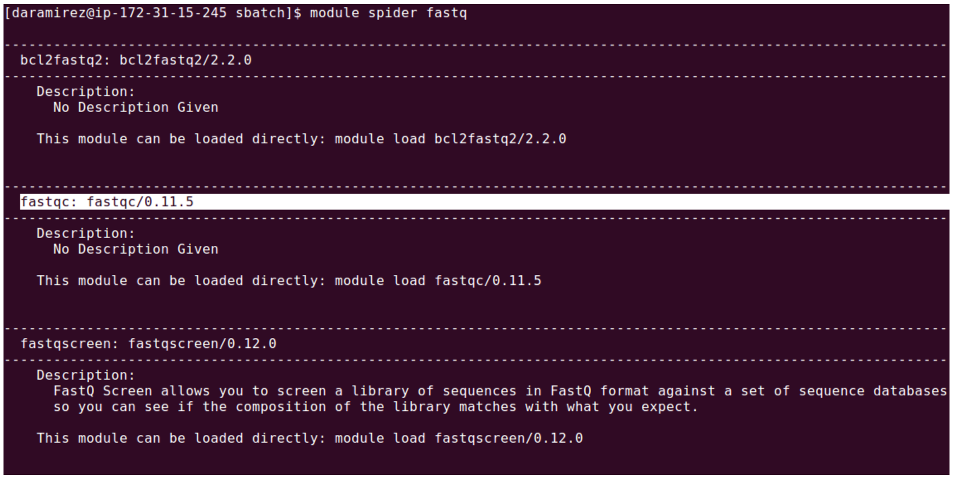
d. Complete the following fields: nnodes, ntasks, mem and time. FastQC cannot use multiple processors per input file. So 1 node, 1 task or processor, 10gb for memory and 1 hour for wall time should be enough.

e. Specify first the path of the fastq file that you selected earlier as the value of the variable “INPUT_DIRECTORY”, and second the path that leads to the directories you created earlier in your scratch directory as the value of the variable “OUTPUT_DIRECTORY”. For example, I decided to use the file “Example_1.fastq.gz”, so I will type this file’s complete path directory “/scratch/Workshop/SR2019/4_qc/fastq” to the INPUT_DIRECTORY variable, and I will type “/scratch/Users/daramirez/fastQC/”
to the OUTPUT_DIRECTORY variable.



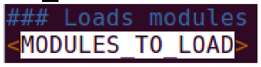
```
### Assigns path variables
INPUT_DIRECTORY=<PATH_TO_INPUT_FILE>
OUTPUT_DIRECTORY=<PATH_TO_OUTPUT_FILE>
```

f. Assign the required modules necessary to run this fastQC job. To do this, exit vim by saving all changes (press ***ESC*** and ***:wq!***). To look for the correct fastQC module, list all available modules on the computer cluster that contain the word “fastq” in them. Type the following command ***module spider <string>*** and look for the one for fastQC.



```
[daramirez@ip-172-31-15-245 sbatch]$ module spider fastq

----------------------------------------------------------------------------------------------------
  bcl2fastq2: bcl2fastq2/2.2.0
----------------------------------------------------------------------------------------------------
    Description:
      No Description Given

    This module can be loaded directly: module load bcl2fastq2/2.2.0


----------------------------------------------------------------------------------------------------
  fastqc: fastqc/0.11.5
----------------------------------------------------------------------------------------------------
    Description:
      No Description Given

    This module can be loaded directly: module load fastqc/0.11.5


----------------------------------------------------------------------------------------------------
  fastqscreen: fastqscreen/0.12.0
----------------------------------------------------------------------------------------------------
    Description:
      FastQ Screen allows you to screen a library of sequences in FastQ format against a set of sequence databases
      so you can see if the composition of the library matches with what you expect.

    This module can be loaded directly: module load fastqscreen/0.12.0
```

Copy “module load fastqc/0.11.5”. Open again the file “fastQC.sbatch” using vim and replace “MODULES_TO_LOAD” with what you just copied.



```
### Loads modules
<MODULES_TO_LOAD>
```

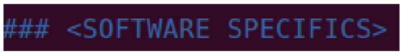
g. The last edit you need to do is the actual text that runs fastQC!

The syntax to use fastQC is as follows:

fastqc --format <format> --threads <n> --outdir <output_file> <input_file>

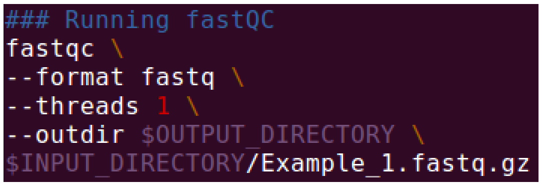
Where <format> is the format of the input file “fastq”, <threads> is 1 (processors or CPUs), <output_file> is the path and name that you want to specify to where to store the results, and <input_file> is the path and name of the fastq file you want to run. We can take advantage of the variables that we created INPUT_DIRECTORY and OUTPUT_DIRECTORY. Though this may seem silly, creating variables makes longer pieces of script much more readable when you re-utilize a given path many times.

So we can go from having in the template:



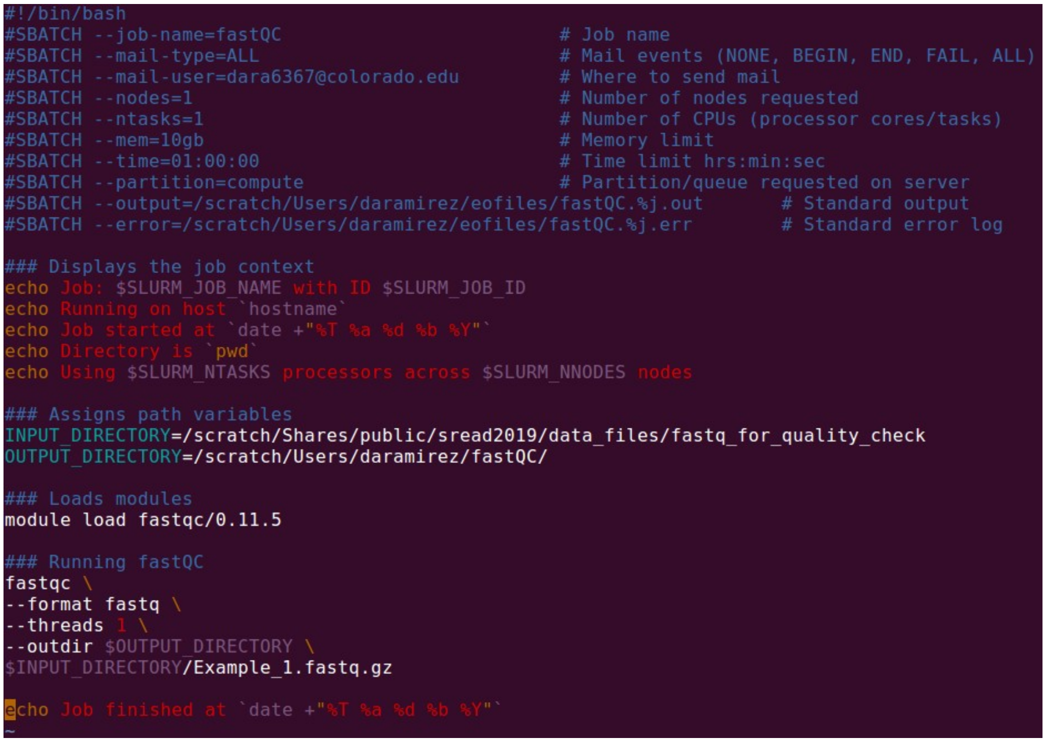
```
### <SOFTWARE SPECIFICS>
```

To having a complete fastQC command:



```
### Running fastQC
fastqc \
--format fastq \
--threads 1 \
--outdir $OUTPUT_DIRECTORY \
$INPUT_DIRECTORY/Example_1.fastq.gz
```

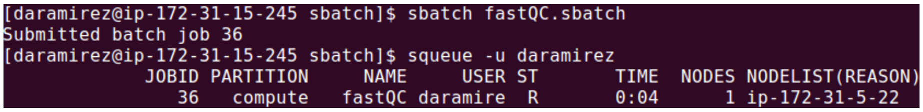
The \ at the end of every line is used to break up what would be a long and confusing single line command into pieces corresponding to every part of the command, just for clarity purposes.



```
#!/bin/bash
#SBATCH --job-name=fastQC                          # Job name
#SBATCH --mail-type=ALL                            # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=dara6367@colorado.edu          # Where to send mail
#SBATCH --nodes=1                                  # Number of nodes requested
#SBATCH --ntasks=1                                 # Number of CPUs (processor cores/tasks)
#SBATCH --mem=10gb                                 # Memory limit
#SBATCH --time=01:00:00                            # Time limit hrs:min:sec
#SBATCH --partition=compute                        # Partition/queue requested on server
#SBATCH --output=/scratch/Users/daramirez/eofiles/fastQC.%j.out        # Standard output
#SBATCH --error=/scratch/Users/daramirez/eofiles/fastQC.%j.err         # Standard error log

### Displays the job context
echo Job: $SLURM_JOB_NAME with ID $SLURM_JOB_ID
echo Running on host `hostname`
echo Job started at `date +"%T %a %d %b %Y"`
echo Directory is `pwd`
echo Using $SLURM_NTASKS processors across $SLURM_NNODES nodes

### Assigns path variables
INPUT_DIRECTORY=/scratch/Shares/public/sread2019/data_files/fastq_for_quality_check
OUTPUT_DIRECTORY=/scratch/Users/daramirez/fastQC/

### Loads modules
module load fastqc/0.11.5

### Running fastQC
fastqc \
--format fastq \
--threads 1 \
--outdir $OUTPUT_DIRECTORY \
$INPUT_DIRECTORY/Example_1.fastq.gz

echo Job finished at `date +"%T %a %d %b %Y"`
~
```

7. Congratulations! You have written your first batch script. You just need to submit the script to the job manager SLURM for it to begin processing.

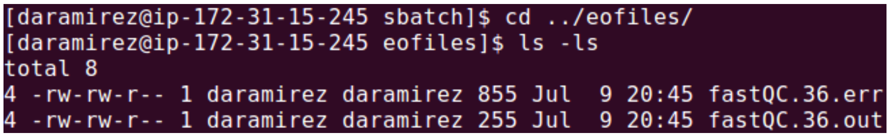
    a. Save all changes to “fastQC.sbatch” and exit vim. In the terminal, located in the directory where “fastQC.sbatch” lives, type ***sbatch <sbatch file>***. The job manager will give you a job number. Once submitted, you can check on the status of jobs by typing ***squeue -u username***.



```
[daramirez@ip-172-31-15-245 sbatch]$ sbatch fastQC.sbatch
Submitted batch job 36
[daramirez@ip-172-31-15-245 sbatch]$ squeue -u daramirez
             JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
                36   compute   fastQC daramire  R       0:04      1 ip-172-31-5-22
```

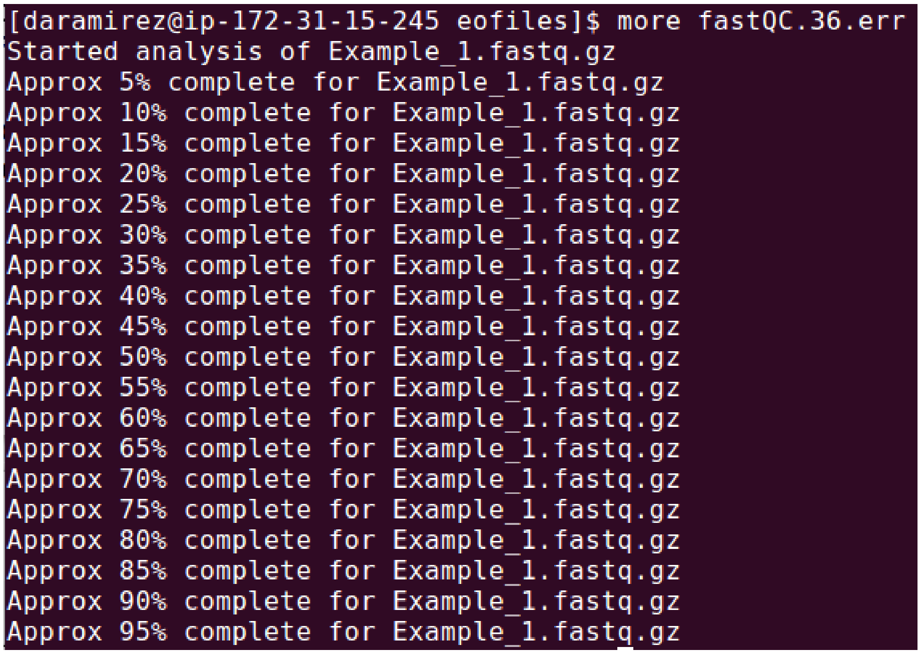
8. Move to the “eofiles” directory. If you cannot remember where you told SLURM to put the error and output files, go back and check the “fastQC.sbatch” file.

    a. There should be two files in that directory.
        One named <job_name>.<job_number>.out
        and one named <job_name>.<job_number>.err

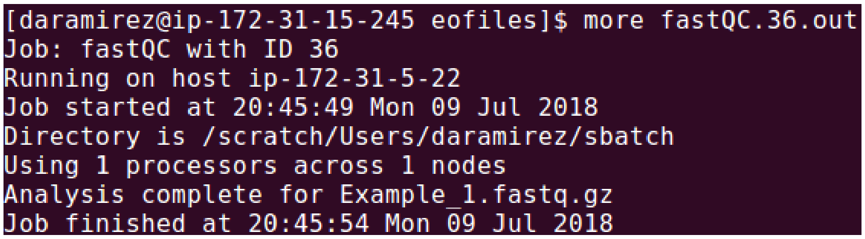


```
[daramirez@ip-172-31-15-245 sbatch]$ cd ../eofiles/
[daramirez@ip-172-31-15-245 eofiles]$ ls -ls
total 8
4 -rw-rw-r-- 1 daramirez daramirez 855 Jul  9 20:45 fastQC.36.err
4 -rw-rw-r-- 1 daramirez daramirez 255 Jul  9 20:45 fastQC.36.out
```

    b. Look at both of those files. Use vim or ***less*** or ***more*** or ***head*** or ***tail***.
        If your files look like this, then your fastQC job completed successfully.

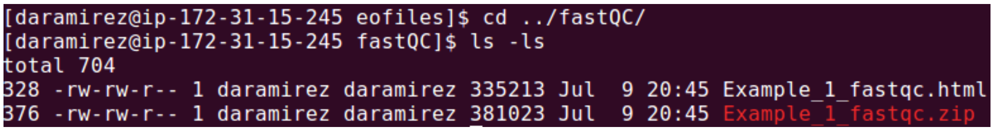


```
[daramirez@ip-172-31-15-245 eofiles]$ more fastQC.36.err
Started analysis of Example_1.fastq.gz
Approx 5% complete for Example_1.fastq.gz
Approx 10% complete for Example_1.fastq.gz
Approx 15% complete for Example_1.fastq.gz
Approx 20% complete for Example_1.fastq.gz
Approx 25% complete for Example_1.fastq.gz
Approx 30% complete for Example_1.fastq.gz
Approx 35% complete for Example_1.fastq.gz
Approx 40% complete for Example_1.fastq.gz
Approx 45% complete for Example_1.fastq.gz
Approx 50% complete for Example_1.fastq.gz
Approx 55% complete for Example_1.fastq.gz
Approx 60% complete for Example_1.fastq.gz
Approx 65% complete for Example_1.fastq.gz
Approx 70% complete for Example_1.fastq.gz
Approx 75% complete for Example_1.fastq.gz
Approx 80% complete for Example_1.fastq.gz
Approx 85% complete for Example_1.fastq.gz
Approx 90% complete for Example_1.fastq.gz
Approx 95% complete for Example_1.fastq.gz
```



```
[daramirez@ip-172-31-15-245 eofiles]$ more fastQC.36.out
Job: fastQC with ID 36
Running on host ip-172-31-5-22
Job started at 20:45:49 Mon 09 Jul 2018
Directory is /scratch/Users/daramirez/sbatch
Using 1 processors across 1 nodes
Analysis complete for Example_1.fastq.gz
Job finished at 20:45:54 Mon 09 Jul 2018
```

9. Next, move and look at the other output files stored in the “fastQC” folder that you created earlier. The two files have .zip and .html extensions.

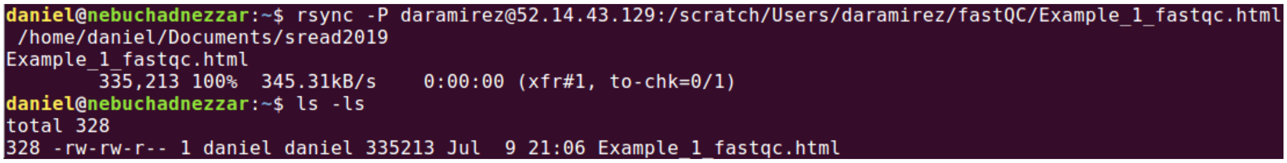


```
[daramirez@ip-172-31-15-245 eofiles]$ cd ../fastQC/
[daramirez@ip-172-31-15-245 fastQC]$ ls -ls
total 704
328 -rw-rw-r-- 1 daramirez daramirez 335213 Jul  9 20:45 Example_1_fastqc.html
376 -rw-rw-r-- 1 daramirez daramirez 381023 Jul  9 20:45 Example_1_fastqc.zip
```

10. Transfer the .html file to your own computer so that you can open it using a web browser. You can use ***rsync***, ***scp*** or other command to do so. If you are on windows you will use another method.

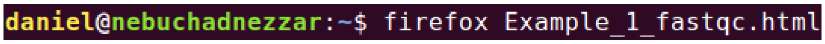
    a. Open a new terminal. Do not log into the computer cluster. This terminal window is on your computer. You can tell because it does not say “[username@ip-172-31-15-245]$” at the beginning of every line, but says my computers username and name.

    b. Make a new directory to put your html file in. Then use ***rsync*** (or your command of preference) to move the html file from the cluster to your home machine.



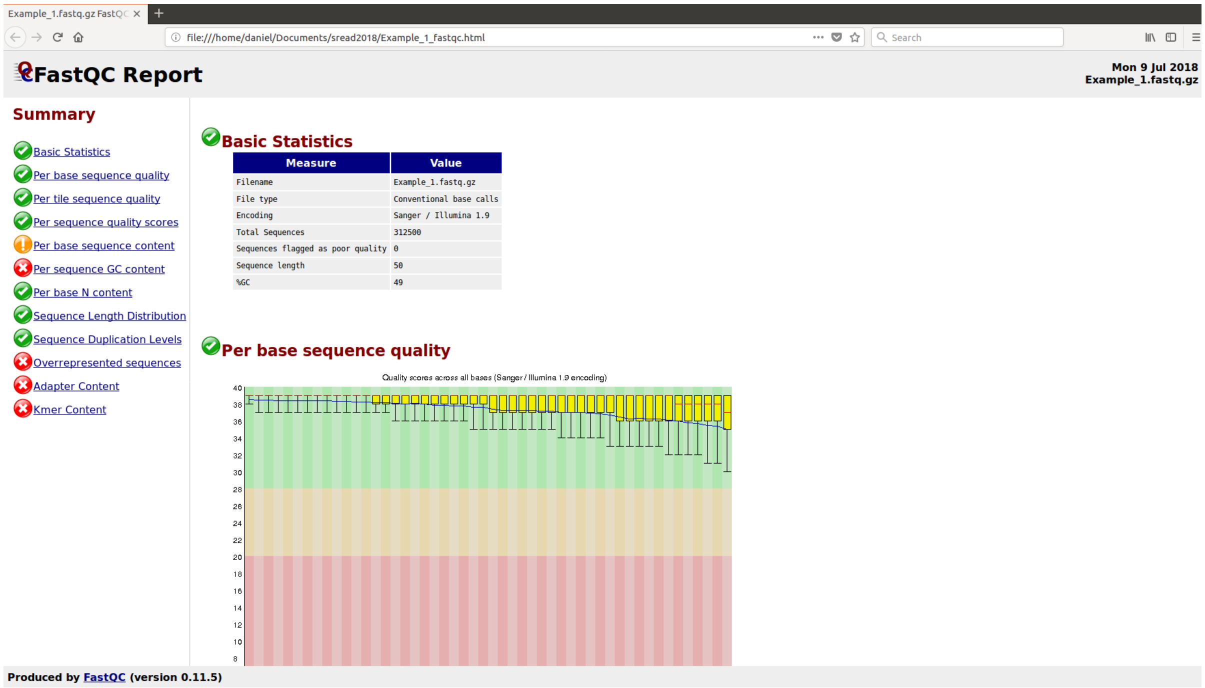
```
daniel@nebuchadnezzar:~$ rsync -P daramirez@52.14.43.129:/scratch/Users/daramirez/fastQC/Example_1_fastqc.html
 /home/daniel/Documents/sread2019
Example_1_fastqc.html
        335,213 100%  345.31kB/s    0:00:00 (xfr#1, to-chk=0/1)
daniel@nebuchadnezzar:~$ ls -ls
total 328
328 -rw-rw-r-- 1 daniel daniel 335213 Jul  9 21:06 Example_1_fastqc.html
```

11. Open the html file you just downloaded.



```
daniel@nebuchadnezzar:~$ firefox Example_1_fastqc.html
```

12. The html report will look like this. You can navigate it just like a website.



Example_1.fastq.gz FastQC
file:///home/daniel/Documents/sread2018/Example_1_fastqc.html
Search

FastQC Report

Mon 9 Jul 2018
Example_1.fastq.gz

Summary

- ✓ Basic Statistics
- ✓ Per base sequence quality
- ✓ Per tile sequence quality
- ✓ Per sequence quality scores
- ! Per base sequence content
- ✗ Per sequence GC content
- ✓ Per base N content
- ✓ Sequence Length Distribution
- ✓ Sequence Duplication Levels
- ✗ Overrepresented sequences
- ✗ Adapter Content
- ✗ Kmer Content

✓ **Basic Statistics**

Measure	Value
Filename	Example_1.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	312500
Sequences flagged as poor quality	0
Sequence length	50
%GC	49

✓ **Per base sequence quality**

Quality scores across all bases (Sanger / Illumina 1.9 encoding)
40
38
36
34
32
30
28
26
24
22
20
18
16
14
12
10
8

Produced by FastQC (version 0.11.5)