FastQC

A beginner's guide to using FastQC on BisonNet,
Bucknell's High Performance Research Computing Network

What is FastQC? What does it do?

FastQC is a tool used by genomic researchers to provide additional quality control checks on raw sequence data from high throughput sequencing pipelines. This provides an HTML based permanent report that contains a summary of potential problem areas in the form of graphs and tables.

Why is this necessary?

Although most sequencers generate their own quality control as a part of their analysis pipeline, it's usually focused on identifying problems generated by the sequencer itself. A primary goal of FastQC is to spot problems that originate in either the sequencer machinery or the raw sequence material. By providing a quick summary report, researchers can quickly identify any potential problems or biases in the data before continuing with further analysis.

What does the report contain?

The report starts with a table of basic information about the file such as its name, type, encoding machinery, total sequences, poor quality sequences, sequence length, and %GC content. Then it goes on to show graphical representations of the following categories:

1. Per base sequence quality
2. Per tile sequence quality
3. Per sequence quality scores
4. Per base sequence content
5. Per base N content
6. Sequence Length Distribution
7. Sequence Duplication Levels
8. Overrepresented sequences
9. Adapter Content

How do I know if my data is good data?

Fortunately for us, the Babraham Institute of Babraham Bioinformatics has provided example reports of both *good* Illumina data and *bad* Illumina data. They even have an online video tutorial walking through a FastQC file linked here.

What file formats does it support?

FastQ, Cassava FastQ files, Colorspace FastQ, GZip compressed FastQ, SAM, BAM, SAM/BAM Mapped only

What version is available on BisonNet?

0.11.7

However, the most up to date version available, 0.11.9, was released on 08-01-19.

What module does it need?

qc

How do you run it?

FastQC can be run in one of two modes. For an immediate analysis of a small number of FastQ files, it can be run as a stand-alone interactive application, or it can be run in a non-interactive mode more suitable for integrating into a larger analysis pipeline.

And because this page is primarily to explain how to utilize Bucknell's high-performance computing cluster, BisonNet, for our very large sequencing datasets, I will be providing a step-by-step demonstration of both processes. Please note that in this example script, I've highlighted the steps that require user input in blue and have indicated comments to explain each step in bold.

Option 1: Immediate analysis of a few FASTQ files

Step 1: Logging onto the main computer of BisonNet

The first purple line is an executable command that tells the computer to log into the "head node" of the boss (main) computer using a secure shell protocol. This main computer is responsible for scheduling different tasks for the other computers, also known as "compute nodes". The text that is highlighted in orange indicates where you should put your respective Bucknell email ID.

```
michelle@DELLXPS-9370:~$ ssh mtp006@bisonnet-hpc.bucknell.edu
mtp006@bisonnet-hpc.bucknell.edu's password: <ENTER PASSWORD>

Last login: Wed Nov 18 14:20:15 2020 from wifi178-065.bucknell.edu
Welcome to Bright release         9.0

                                  Based on CentOS Linux 7
                                         ID: #0000022

Use the following commands to adjust your environment:

'module avail'            - show available modules
'module add <module>'     - adds a module to your environment for this session
'module initadd <module>' - configure module to be loaded at every login

-------------------------------------------------------------------------------
```

Step 2: Log into an interactive session

In order to keep the head node in working order and prevent the system from crashing, it's best practice to log into one of the compute nodes directly via an interactive session when performing intense calculations on a large file.

```
[mtp006@bisonnet-hpc ~]$ srun -n 1 -p short --pty /bin/bash
[mtp006@hpc-4 ~]$
```

You can read more about each term used in the aforementioned command in the "Interactive Jobs" section of BisonNet's Job Submission Management webpage.

Step 3: Navigate to a folder (directory) that contains the file(s) you will be working with

In this example, I have a folder called “Practical_2” that contains a fastq (.fq) file named “Ppyra3_antenna_5mil_1.fq”

```
[mtp006@hpc-4 ~]$ cd Practical_2
```

Step 4: Make a subdirectory called “FASTQC_OUTPUT” within your current directory

```
[mtp006@hpc-4 Practical_2]$ mkdir FASTQC_OUTPUT
```

Step 5: Run FastQC on the desired file(s)

After loading the proper module from BisonNet, the command to run fastqc follows a general format of : *fastqc -o <name_of_output_directory> <name_of_input_file>*

In this example, the -o flag is used to indicate that the next group of characters following the space represent the name of the output directory (FASTQC_OUTPUT), immediately followed by the name/location of the input file (Ppyra3_antenna_5mil_1.fq).

```
[mtp006@hpc-4 Practical_2]$ module load qc

[mtp006@hpc-4 Practical_2]$ fastqc -o FASTQC_OUTPUT/ Ppyra3_antenna_5mil_1.fq

Started analysis of Ppyra3_antenna_5mil_1.fq
Approx 5% complete for Ppyra3_antenna_5mil_1.fq
Approx 10% complete for Ppyra3_antenna_5mil_1.fq
Approx 15% complete for Ppyra3_antenna_5mil_1.fq
Approx 20% complete for Ppyra3_antenna_5mil_1.fq
Approx 25% complete for Ppyra3_antenna_5mil_1.fq
Approx 30% complete for Ppyra3_antenna_5mil_1.fq
Approx 35% complete for Ppyra3_antenna_5mil_1.fq
Approx 40% complete for Ppyra3_antenna_5mil_1.fq
Approx 45% complete for Ppyra3_antenna_5mil_1.fq
Approx 50% complete for Ppyra3_antenna_5mil_1.fq
Approx 55% complete for Ppyra3_antenna_5mil_1.fq
Approx 60% complete for Ppyra3_antenna_5mil_1.fq
Approx 65% complete for Ppyra3_antenna_5mil_1.fq
Approx 70% complete for Ppyra3_antenna_5mil_1.fq
Approx 75% complete for Ppyra3_antenna_5mil_1.fq
Approx 80% complete for Ppyra3_antenna_5mil_1.fq
Approx 85% complete for Ppyra3_antenna_5mil_1.fq
Approx 90% complete for Ppyra3_antenna_5mil_1.fq
Approx 95% complete for Ppyra3_antenna_5mil_1.fq
Approx 100% complete for Ppyra3_antenna_5mil_1.fq
Analysis complete for Ppyra3_antenna_5mil_1.fq
```

After loading the module once, you can repeat the same general fastqc command for several other files if needed.

Step 5: After the analysis is complete, import the HTML file FastQC produced on the interactive session to your local computer for viewing.

Unfortunately, you are unable to view the HTML file from BisonNet directly, so you will need to somehow copy it over to your local computer. Here are two ways to securely copy the data to your desired location:

1. *On a BisonNet terminal, you can "push" the file to your local computer*

   In pink, is the absolute path reference to the file you would like to push. Ideally, you are already in the directory that contains this file. The asterisk (*) is a wildcard indicator that will tell the computer to secure copy any existing HTML files from your BisonNet directory to your local computer. Remember to put a single space between the pink and yellow sections!

   The yellow highlighted section of text should be interchanged with your respective local user and computer names. If you're confused and/or don't know how to identify that, it is what is displayed in the very first line when you open a terminal session on your local computer.

   And in the green, what follows the semicolon is the absolute path reference destination of where you would like the file to be saved on your local computer. There is no space between the yellow and green!

```
[mtp006@hpc-4 ~]$ scp ./*.html michelle@DELLXPS-9370:~/Practical_2/fastqc_output/
```

   OR

2. *On your local computer, you can "pull" the file from BisonNet*

   Similar to choice one, you should already be in your destination folder.

   Highlighted in yellow is the area to be replaced with your respective Bucknell email ID. In green is the absolute path reference location of the HTML file being pulled. And the pink period (.) can be interchanged with an absolute path reference to the desired destination folder on your local computer. This example just has a period because my current directory is my desired destination folder.

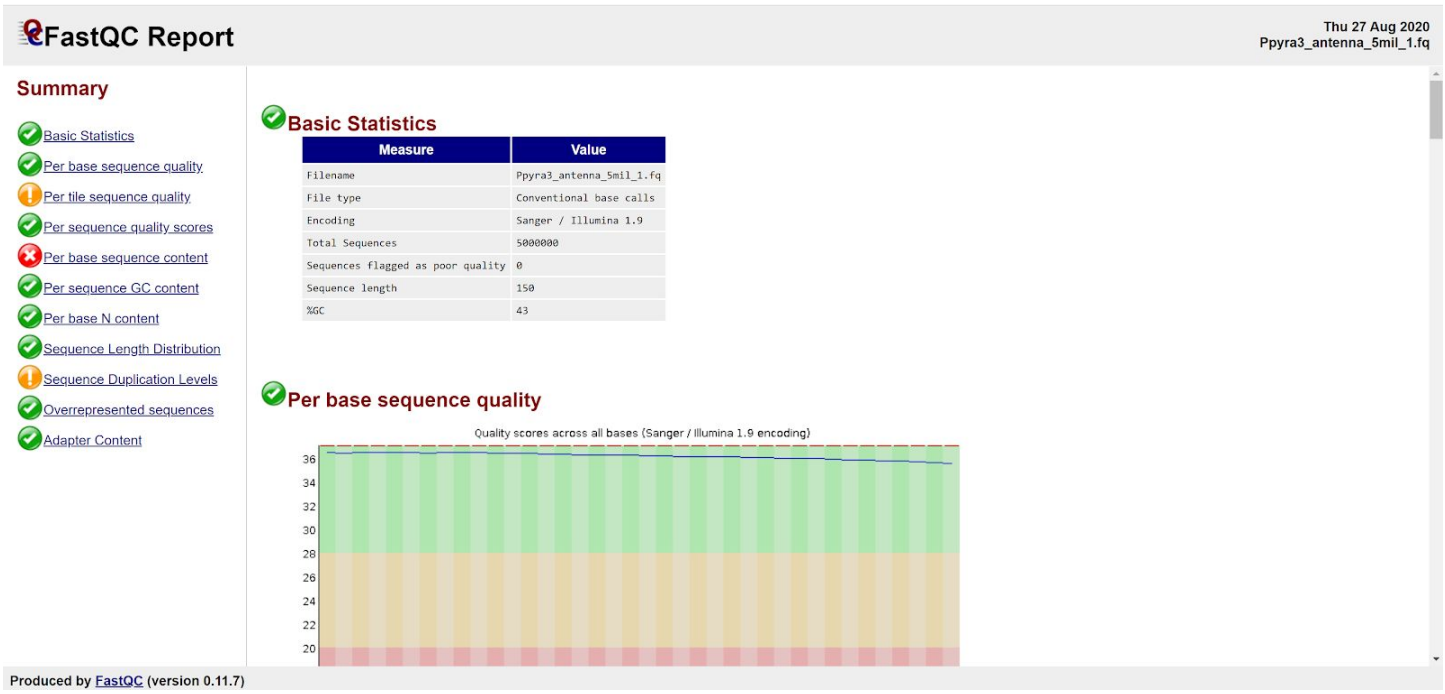
```
michelle@DELLXPS-9370:~/BIOL325_EvolGen/Practical_2$ scp -r
mtp006@bisonnet-hpc.bucknell.edu:~Practical_2/fastqc_output/*.html .
```

Step 6: Open the file and read away!

On iOS, you can simply navigate to the HTML file through the iOS file manager and open up the FastQC report. However, on Windows operating systems it can be a little trickier to find the directory you actually chose. So as a tip for Windows users, after navigating to the directory containing the FastQC HTML file on the terminal, use this useful command to bring up a more intuitive (and likely familiar!) file explorer graphical interface: **explorer.exe .**

But make no mistake - there *is* a space between the last e and the standalone period. Again, the period symbol represents a file path reference that indicates that you are already in the desired directory with the files you want to view. Once you've done this, you can simply open the HTML with

whatever program you choose, such as Google Chrome, and will likely be brought to a page that looks like this:



Option 2: Non-interactive session on BisonNet for large quantities of FastQ files

If you don't have the time to actively stay logged on an interactive session and/or keep your computer running, but still have files you would like to create a FastQC for, luckily there is another way! You can create a script, which combines a set of commands into one executable file and send it to BisonNet's head node queue. From there the script will be assigned a compute node that will automatically execute the script commands and run the programs for you.

Step 1: Log onto BisonNet - but NOT in an interactive session

```
michelle@DELLXPS-9370:~$ ssh mtp006@bisonnet-hpc.bucknell.edu
mtp006@bisonnet-hpc.bucknell.edu's password: <ENTER PASSWORD>

Last login: Wed Nov 18 14:20:15 2020 from wifi178-065.bucknell.edu
Welcome to Bright release         9.0

                                  Based on CentOS Linux 7
                                  ID: #0000022

Use the following commands to adjust your environment:

'module avail'            - show available modules
'module add <module>'     - adds a module to your environment for this session
'module initadd <module>' - configure module to be loaded at every login
```

```
--------------------------------------------------------------------------
```

Step 2: Move into your desired directory and make an output directory

```
[mtp006@hpc-4 ~]$ cd Practical_2
[mtp006@hpc-4 Practical_2]$ mkdir FASTQC_OUTPUT
```

Step 3: Create a script and save it

```
[mtp006@hpc-4 Practical_2]$ nano runFastQC.sh
```

^ This command will open the script editor, where you should input the following text:

```
#!/bin/bash

#SBATCH -p medium  # partition (queue)
#SBATCH -N 1 # (leave at 1 unless using multi-node specific code)
#SBATCH -n 4 # number of cores
#SBATCH --mem-per-cpu=8192 # memory per core
#SBATCH --job-name="mpFASTQC" # job name
#SBATCH -o slurm.%N.%j.stdout.txt # STDOUT
#SBATCH -e slurm.%N.%j.stderr.txt # STDERR
#SBATCH --mail-user=mtp006@bucknell.edu # address to email
#SBATCH --mail-type=ALL # mail events (NONE, BEGIN, END, FAIL, ALL)

#General Usage: sbatch runFastQC.sh

#loads the correct
module load qc

#runs FASTQC on all the files listed by a loop, with file names separated by a single space
for file in Ppyra3_antenna_5mil_1.fq Ppyra3_antenna_5mil_1.fq Ppyra2_antenna_5mil_1.fq
Ppyra2_antenna_5mil_1.fq
do
      fastqc -o FASTQC_OUTPUT/ $file
done

#print 'done!' to the terminal screen
echo "done!"
```

If you would like to understand what the first 11 lines of the script (aka the header) mean, please read more about it on the BisonNet webpage.

For any line following the header (after #SBATCH), if it starts with a pound symbol (#), those are lines are comments used to describe what the commands are actually doing. They will not be executed by the compute node, but are rather there for your clarification and future reference.

The text highlighted in purple should be adjusted to fit the description of your input files and user info.

Step 3: Change the permissions on the file to make it executable

```
[mtp006@hpc-4 Practical_2]$ chmod u+x runFastQC.sh
```

Step 4: Submit the job to the queue

```
[mtp006@hpc-4 Practical_2]$ sbatch runFastQC.sh
Submitted batch job 10646
```

Step 5: Review results

Once the job has finished, you should receive an email with the subject name along the lines of "Slurm Job_id=10646 Name=mpFASTQC" Ended, Run time 04:07:40, COMPLETED, ExitCode 0"

Then you may view your results (FastQC HTML files) by following steps 5 and 6 as outlined above for option 1.