

Sequence Quality Control

James Thornton Jr

Abstract

This protocol will introduce a workflow for quality control and pre-processing of metagenomic sequence reads using FastQC for visualization and FastX Toolkit for editing the fastq files.

Citation: James Thornton Jr Sequence Quality Control. **protocols.io**

dx.doi.org/10.17504/protocols.io.j2icqce

Published: 25 Sep 2017

Guidelines

[FastX Toolkit Documentation](#)

Protocol

Step 1.

Login to the HPC.

```
cmd COMMAND  
$ ssh hpc  
$ ocelote
```

Step 2.

Make sure you have downloaded your FASTQ files for your project. This is detailed in the following protocol:

<https://www.protocols.io/view/getting-started-on-your-project-jsacnae>

Step 3.

Move into the directory containing your FASTQ files.

```
cmd COMMAND  
$ cd /rsgprps/bh_class/username/fastq  
Replace username with YOUR username.
```

Step 4.

If your FASTQ files are still compressed (.gz extension), make sure to uncompress them.

cmd **COMMAND**

```
$ gunzip *.gz
```

Remember, you must be in the directory containing your fastq files for this to work.

■ **ANNOTATIONS**

Bonnie Hurwitz 28 Sep 2017

what is the expected result when I gunzip?

it should return with no errors

Then you can ls-l and see that none of the files have .gz as an ext.

Step 5.

Load FastQC:

cmd **COMMAND**

```
$ module load fastqc
```

This command can be executed anywhere on the hpc to load the fastqc tool

📌 **NOTES**

James Thornton Jr 19 Sep 2016

The FastQC tools will provide visualization for the quality of the sequence reads for each of your samples.

Step 6.

Make a directory for FastQC output.

cmd **COMMAND**

```
$ mkdir /rsgrps/bh_class/username/quality_control
```

Make sure to replace username with the name of YOUR directory containing YOUR work.

Step 7.

Run FastQC on all of your fastq files and store the output in the directory you made in the previous step.

cmd **COMMAND**

```
$ fastqc *.fastq -o /rsgrps/bh_class/username/quality_control
```

Make sure to move into YOUR directory containing YOUR fastq files. The -o command will put the output from fastqc into the directory specified.

📈 **EXPECTED RESULTS**

Started analysis of SRR1647144.fastq

Approx 5% complete for SRR1647144.fastq
Approx 10% complete for SRR1647144.fastq
Approx 15% complete for SRR1647144.fastq
Approx 20% complete for SRR1647144.fastq
Approx 25% complete for SRR1647144.fastq
Approx 30% complete for SRR1647144.fastq
Approx 35% complete for SRR1647144.fastq
Approx 40% complete for SRR1647144.fastq
Approx 45% complete for SRR1647144.fastq
Approx 50% complete for SRR1647144.fastq
Approx 55% complete for SRR1647144.fastq
Approx 60% complete for SRR1647144.fastq
Approx 65% complete for SRR1647144.fastq
Approx 70% complete for SRR1647144.fastq
Approx 75% complete for SRR1647144.fastq
Approx 80% complete for SRR1647144.fastq
Approx 85% complete for SRR1647144.fastq
Approx 90% complete for SRR1647144.fastq
Approx 95% complete for SRR1647144.fastq
Analysis complete for SRR1647144.fastq

Step 8.

Move into the quality_control directory that now contains the output of FastQC. Delete the .zip folders that were created (we only need the .html files).

cmd **COMMAND**

```
$ cd /rsgrps/bh_class/username/quality_control  
$ rm -rf ./*.zip
```

CAUTION: rm -rf is a no going back command. Files removed are gone FOREVER. Make sure you are in the directory containing the FastQC files and execute as described above.

Step 9.

In order to view the html summary files you must "secure copy (scp)" to your local machine. Open a new terminal (don't log into hpc). Determine where you want to store the files on your local machine and move into that directory.

NOTES

James Thornton Jr 19 Sep 2016

Windows users using Cygwin, your file will be stored in C:/cygwin64/home/USER. Just open a new terminal window and proceed to next step.

Step 10.

Execute the following command to scp the html files to your local machine:

cmd **COMMAND**

```
$ scp jamesthornton@login.hpc.arizona.edu:/rsgrps/bh_class/jetjr/quality_control/*.html .
```

Replace jamesthornton with your own NETID. Also notice there is a period after *.html which indicates to put the files in the current directory. All files containing the .html extension will be copied to the current directory on the local machine.

🔗 NOTES

James Thornton Jr 19 Sep 2016

Keep in mind that anytime FastQC is ran again and a new .html summary file is generated you must scp to your local machine in order to view it.

Emily Wall 27 Sep 2016

At this step it asks me for the two-factor login again? Should I be getting this? It doesn't seem to have saved into my C:/cygwin/home/Emily folder on my computer either? (I have a PC)

■ ANNOTATIONS

James Thornton Jr 25 Sep 2017

See step 12 for an alternative method to obtain the .html files.

Step 11.

Alternative method to obtain .html files. (Skip if scp was successful for you).

If you are having trouble using scp, you can "push" the .html files to your abe487 github repository. This will require you to copy them into your abe487 directory (used for computational homework). Then do a git add, git commit, git push.

cmd COMMAND

```
$ pwd
/rsgrps/bh_class/username/quality_control
$ mkdir ~/abe487/fastqc
$ cp *.html ~/abe487/fastqc
$ cd !$
$ git add *.html
$ git commit -m 'adding qc reports'
$ git push
```

Make sure you are in the quality_control directory that contains your .html files. Once complete, you can go to your github repository to download the report files for viewing.

Step 12.

Now you can view the FastQC results from the .html files.

cmd COMMAND

```
$ open ./*.html
```

Will open all .html files in the current directory into your default browser. Remember this is on a local terminal (not connected to hpc). Double clicking on the files will work as well.

🔗 NOTES

James Thornton Jr 19 Sep 2016

Again, Windows users using Cygwin will have to go to C:/cygwin64/home/USER directory on your windows machine to find the file. In the windows 10 interface open file explorer, click "This PC", double click on Windows (C:), then cygwin64, then home, then USER. Your files should be there.

Step 13.

Determine the quality control steps and parameters needed to improve the quality of the reads by looking at the html summary for each sample. Keep in mind that each sample (file) will likely have different steps and parameters.

■ ANNOTATIONS

Bonnie Hurwitz 20 Sep 2016

See lecture notes for ideas on what to look for

Bonnie Hurwitz 25 Sep 2017

We will review steps in class:

1. Clipping
2. Quality filtering
3. Read length filtering
4. Duplicate sequence removal

Step 14.

Load FastX Toolkit:

cmd COMMAND

```
$ module load fastx
```

The FastX toolkit will allow us to take quality control steps on our sequences.

Step 15.

Summary of tools available in the FastX toolkit can be viewed by the link given below.

Command line usage for these tools is here: http://hannonlab.cshl.edu/fastx_toolkit/commandline.html

🔗 LINK:

http://hannonlab.cshl.edu/fastx_toolkit/

📄 NOTES

Bonnie Hurwitz 20 Sep 2016

We are showing you a limited view into this total software package. You can look here for more options.

Step 16.

The next few steps will introduce some FastX tools and how they can be used one at a time on a file. It's possible to use multiple FastX tools at the same time and this is demonstrated in step 17.

■ ANNOTATIONS

James Thornton Jr 03 Oct 2017

Woops, step 21 actually demonstrates multiple command option.

Step 17.

The fastx_trimmer can be used if you see a decrease in quality at a specific base position:

cmd COMMAND

```
$ fastx_trimmer -f 10 -l 200 -i [Infile] -o [outfile]
```

where -f refers to the first base position and -l refers to the last NOTE: this is just an example of how to use the trimmer. The actual parameters to run depends on your samples.

⊕ NOTES

James Thornton Jr 19 Sep 2016

It is possible that only some or even none of your samples will be trimmed. Look at the FastQC output to determine this.

Bonnie Hurwitz 20 Sep 2016

base pair quality usually decreases with length for most NGS technologies

Bonnie Hurwitz 20 Sep 2016

fastqc can tell you where the drop in quality occurs for the sequences in a given file, reference the fastqc results to decide on the base pair position to trim from.

Step 18.

The fastq_quality_filter can be used to filter out reads that fail to reach a specific quality score:

cmd COMMAND

```
$ fastq_quality_filter -q 20 -p 80 -i [infile] -o [outfile]
```

-q refers to the minimum quality score to keep and -p is the minimum percent bases that must have -q quality

⊕ NOTES

James Thornton Jr 19 Sep 2016

It's a good idea to run the quality filter on all of your samples, even if the reads appear to have good quality already. The parameters used in the command on this step will work as a filter on your samples.

Bonnie Hurwitz 20 Sep 2016

Some reads are just bad, and have poor quality through out. We want to remove these reads. If these reads remain in the dataset, you will have issues down the line with assembly. Garbage in = garbage out

Step 19.

The fastx_clipper can remove reads below a certain minimum length. Remove reads that are less than 70 base pairs long by executing the following command:

cmd **COMMAND**

```
$ fastx_clipper -l 70 -i [infile] -o [outfile]
```

-l sets the minimum length of reads. Fastx_clipper will remove any reads < 70

NOTES

Bonnie Hurwitz 20 Sep 2016

After you trim your reads, some may be super short. These reads are usually not long enough to contribute to down stream analyses such as assembly and taxonomic or functional annotation (we will go through these analysis steps later in the semester).

Step 20.

Finally, the fastx_collapser will collapse identical sequences into a single one. The collapser should always be last in the workflow because the output will be in Fasta format instead of Fastq.

cmd **COMMAND**

```
$ fastx_collapser -i [infile] -o [outfile]
```

output will be in fasta format.

NOTES

Bonnie Hurwitz 20 Sep 2016

Here we are removing duplicates produced by the sequencing technology. These data can bias your final results, so they need to be removed.

Step 21.

You can also pipe together multiple commands:

cmd **COMMAND**

```
cat SRR1647145_trimmed.fq | fastx_trimmer -f 12 -l 175 | fastq_quality_filter -q 20 -p 80 | fastx_clipper -l 60 | fastx_collapser > SRR1647145.fasta
```

fastx_collapser should be last as the output will be in fasta format.

NOTES

Bonnie Hurwitz 20 Sep 2016

Make files are also a great way to do this. See Ken's gitbook and your homework assignment for how to do this.

Bonnie Hurwitz 28 Sep 2016

You can also do this on the fly via a for loop. Note that I put the names of my files in a file called

"list"

```
% for file in `cat list`; do
```

```
> cat $file.fq | fastx_trimmer -f 12 -l 300 | fastq_quality_filter -q 20 -p 80 | fastx_clipper -l 60 |  
fastx_collapser > $file.fasta
```

```
> done
```