**Intro to Command Line Coding Genomics Analysis**

**Rachael St. Jacques, Max Maza & Dr. Ray Enke**

**Bio 481 Genomics James Madison University**

****

*\*this activity was adapted from code and slides developed by Andrew Lonsdale (*[*@LonsBio*](https://twitter.com/lonsbio?lang=en)*) at Melbourne University.* [*Here’s a link*](https://youtu.be/WywQ6a3uQ5I?t=33m18s) *to a Lightning Talk that Andrew gave in 2017 about FASTQE*

You’ve previously used FastQC for FASTQ file analysis in CyVerse Discovery Environment. Today you will use an even more universal form of communication to analyze these files, THE EMOJI 😻😻😻.

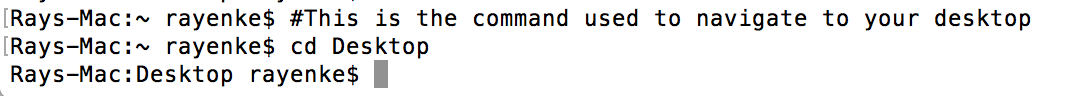
You need to have **Anaconda** installed on your computer (see [pre class assignment](https://docs.google.com/document/d/1Y73xfIWorNVjv4vCFoQP3EhDHgzv_dmtwq33nCjWEUY/edit?usp=sharing) https://bit.ly/2RxKApp). If you haven’t done this yet, please go ahead and do so.

As previously discussed,FASTQE is a program that will analyze a FASTQ file and read out an emoji output as an indicator of each sequence’s quality in the file. So a high quality read may look like this 😃, while this symbol 💩...well you get the idea.

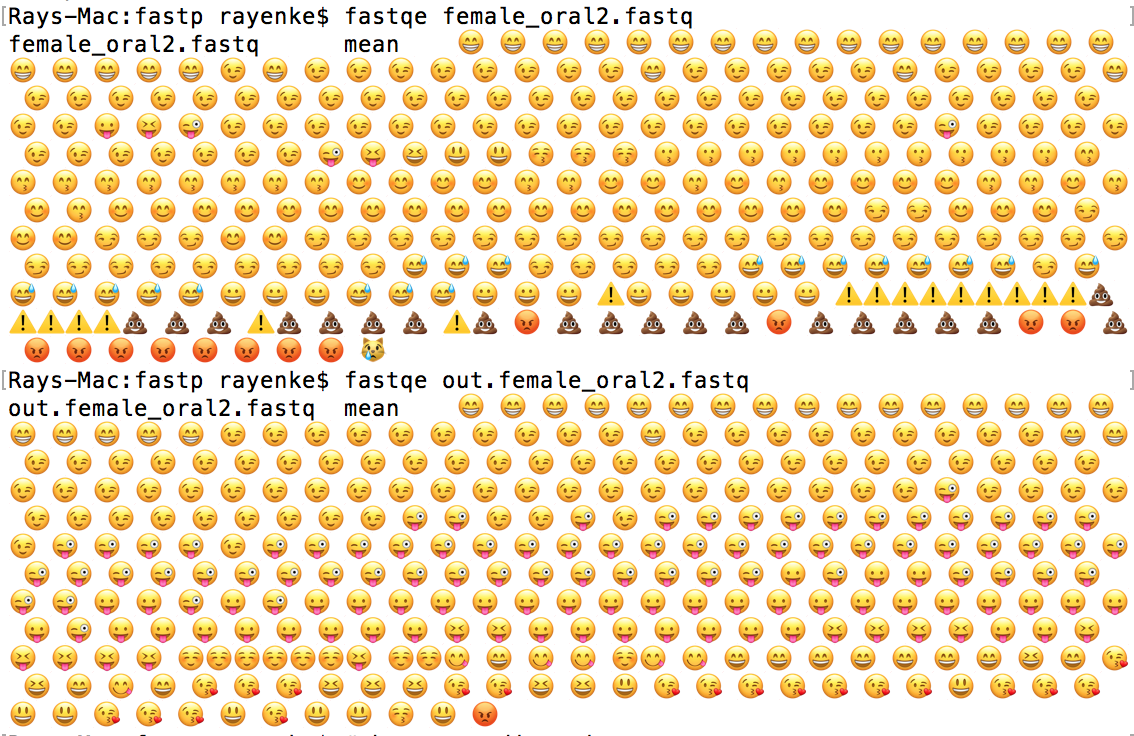
* Open up your computer’s **Terminal** or **command line**

When using command line code, lines that start with the # sign are text descriptors or instructions not lines of code. For example:

* #This is the command used to navigate to your desktop (hit enter)
  + Then type: *cd Desktop* (hit enter)
* Here’s what that looks like in Terminal

**

* + The # line doesn’t do anything, it’s just telling you what the next line of code does which is navigate to your computer’s desktop
  + We will stick with this convention throughout the activity
* #check to see if you are still on your desktop
  + Type: *pwd*
* Download the 2 compressed fastq files from Canvas and put them on your desktop (they have .gz extension and need to be unzipped; also available here for download <https://bit.ly/2FbODRS>)
* #Unzip both files using the **gunzip** command
* #gunzip followed by a fastq file name will unzip and individual file
* #or gunzip followed by **\*.gz** will unzip all files ending with .gz
  + Type: *gunzip* *\*.fastq.gz*
* #check to see if your files are unzipped (.fastq extension)
  + Type: *ls*
* #Install the fastqe software
  + Type: *pip* *install fastqe*
* Note: if you don’t have Anaconda installed at this point you will probably get an error message here
* #you can use individual .fastq file names or \*.fastq for all files
  + Type: *fastqe \*.fastq*
* take a screenshot of your Emoji output for each file & paste it into an MS ppt file
  + It should look something like this:



Notice that 1 of your files (femal\_oral2) seems to have lower quality than the other based on the Emoji readout. Let’s use another program called **Fastp** to get a more conventional readout of the .fastq file data. **Fastp** is similar to the **FastQC** program we previously used, however, it also has a trimming tool to cut out or **filtering** the low quality sequences in our file.

* #Create a new folder on the desktop called fastp using the mkdir command
* Type: *mkdir fastp*
* Drag your 2 fastq files into this folder
* #navigate to the fastp folder
  + Type: *cd fastp*
* #Install fastp
  + Type: *conda install -c bioconda fastp*
* #run fastp on the lower quality female\_oral2.fastq file
  + Type: *fastp -i female\_oral2.fastq -o out.female\_oral2.fastq*
* You should now have 3 new files in your fastp folder

1. A .html file (this is your QC report)
2. A .json file (ignore this for now)
3. A trimmed fastq file (out.female\_*oral*2.fastq)

* Open your html QC report & collect some before and after filtering data
  + How many reads are in this FASTQ file before and after filtering?
  + Copy/paste before and after Quality plots into your ppt file
  + How do they compare?
    - This is the per base quality plot with quality (or Phred score) on the y and base position on the x axis
* Lastly, let’s rerun fastqe to get an emoji output on our trimmed file
* #use the fastp output .fastq file to rerun fastqe
* Type: fastqe out.female\_*oral*2.fastq
  + How does your before and after filtering emoji outputs look for this file?

**To sum up**, you just looked at Illumina FASTQ data quality using Emoji output. You then filtered the low quality sequences in your FASTQ file and output before and after QC plots. Following filtering, you created a new Emoji output of your filter FASTQ file.

**You did all of that on the command line, congrats!**

Here’s the script only version of the above exercise:

**#download the 2 fastq.gz files from Canvas onto your desktop**

**#navigate to your desktop directory**

*cd Desktop*

**#Unzip both fastq files using the gunzip command**

*gunzip* *\*.fastq.gz*

**#check to see if your files are unzipped (.fastq extension)**

*ls*

**#Install the fastqe software**

*pip install fastqe*

**#run fastqe on all .fastq files on the desktop**

*fastqe \*.fastq*

**#Create a new desktop folder called fastp using the mkdir command**

*mkdir fastp*

**#Drag your 2 fastq files into this folder**

**#navigate to the fastp folder**

*cd fastp*

**#Install fastp**

*conda install -c bioconda fastp*

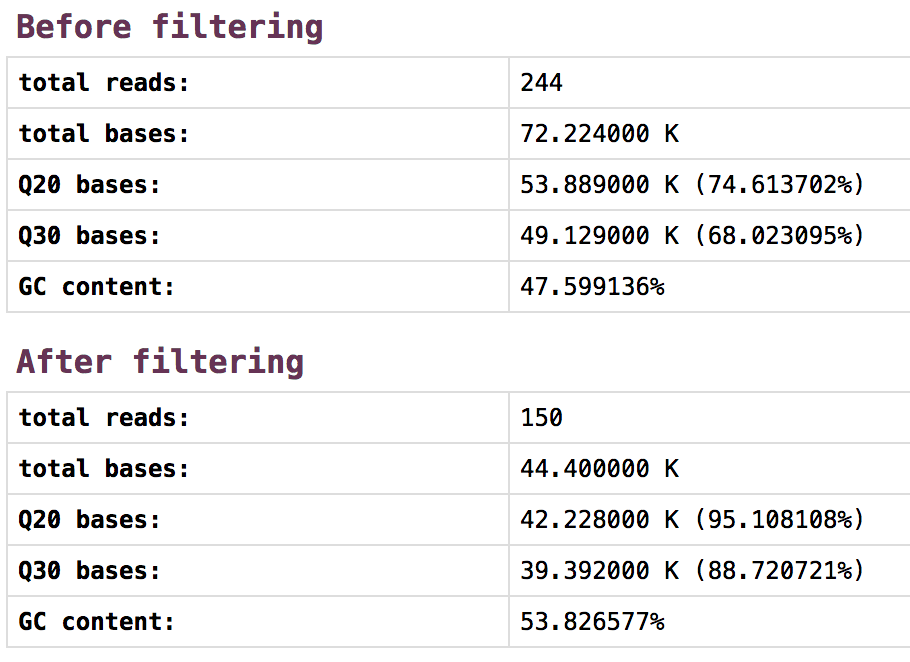
**#run fastp on the lower quality female\_oral2.fastq file**

*fastp -i female\_oral2.fastq -o out.female\_oral2.fastq*

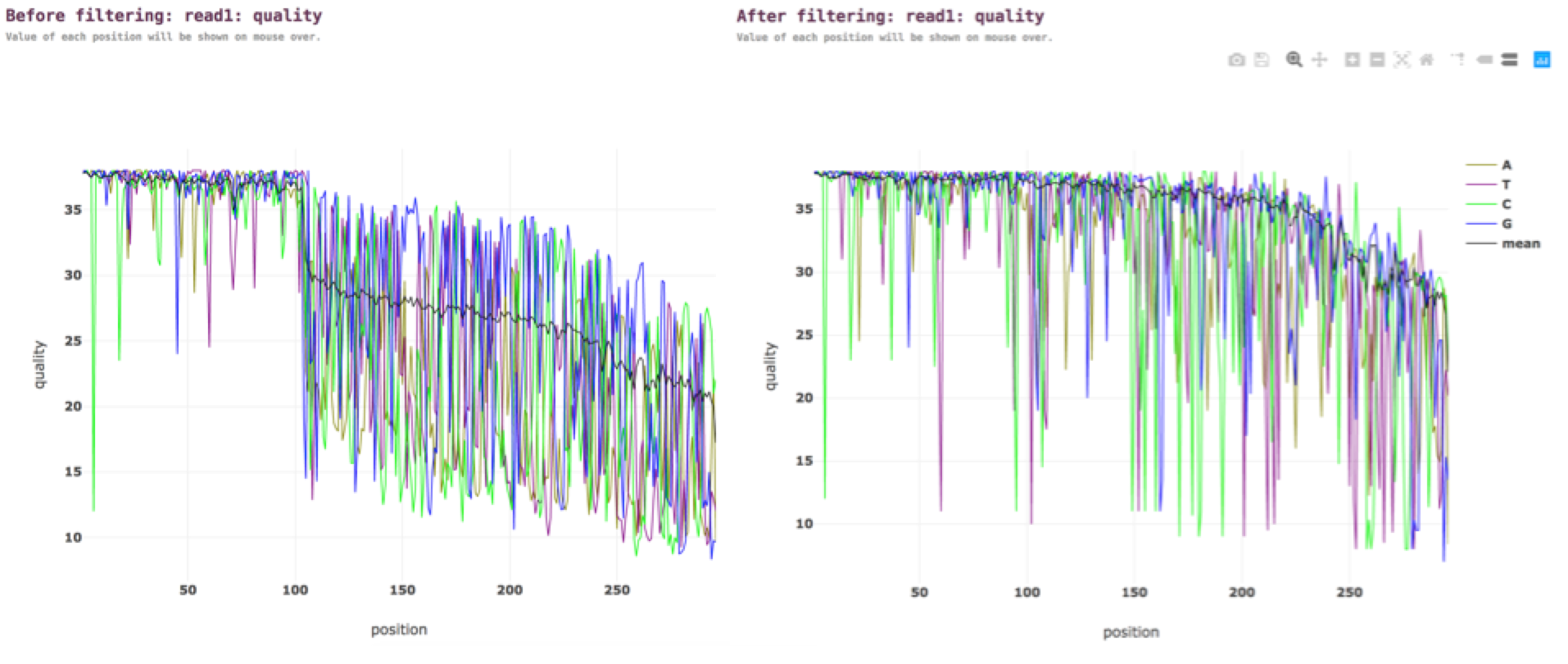
**#use the fastp output .fastq file to rerun fastqe**

*fastqe out.female\_oral2.fastq*

Examples of fastp pre & post filtering metrics:

**

Examples of fastp pre & post filtering quality plots:

**