A Fun Introductory Command Line Exercise:

Next Generation Sequencing (NGS) Quality Analysis with Emoji 💻 📈 😻

*Unfortunately, this lesson version only works on computers running Mac OSX or Linux 😱

See *Technical requirements* for alternate lesson versions able to run on any machine 😀

**this activity was adapted from code and slides developed by Andrew Lonsdale (@LonsBio) at Melbourne University. Here's a link to a Lightning Talk that Andrew gave in 2017 about FASTQE.*

Goals: Use basic command line coding to:

- Introduce students to writing basic command line scripts
- Analyze & assess the quality of FASTQ formatted NGS data
- Trim/filter low quality reads in FASTQ files

The 1st step of any **Next Generation Sequencing (NGS)** analysis pipeline is checking the quality of the raw sequencing reads in each **FASTQ (.fastq) formatted file**. If the sequence quality is poor, then your resulting downstream analysis will be inaccurate and misleading. **FastQC** is a popular software used to provide an overview of basic quality metrics for NGS data. In this lesson, you will use an even more universal form of communication to analyze FASTQ files, THE EMOJI 😻😻😻.

Technical requirements/limitations:

- Windows does not support the use of emoticons 😿, **so this lesson version only works on computers running Mac OSX or Linux.** You can work as a group on the classroom Macs or use your own Mac computer for this lesson.
- An **alternative lesson version** able to run on any machine is available using the CyVerse Jupyter Notebook platform: https://bit.ly/fastqe-CyVerse 😁
- If using your own Mac computer, you need to install **Anaconda** on your machine (see pre class assignment https://bit.ly/2RxKApp; ~20-30 min to install). Anaconda is a Python-based data processing & scientific computing platform with built-in third-party libraries.
- Lastly, the FASTQE program is limited to short read NGS data of 500bp or less

Like the popular FastQC software, **FASTQE** can be used to analyze the quality of FASTQ file data whether it's from a genome sequencing project, an RNA-seq project, a ChIP-seq project, etc. Here's a brief background on the in class metagenomics project that Dr. Enke's BIO 481 Genomics class is collecting data for. Garter snakes excrete sexually dimorphic pheromones to attract a mate. The hypothesis of their experiment is that male and female garter snakes host unique microbial communities in their mouths, cloacae and musk glands that contribute to sexually dimorphic bioengineering of these pheromone molecules. **Figure 1** provides an overview of their 16S metagenomics analysis pipeline. For this lesson though, all you need are the FASTQ files. Feel free to substitute your own favorite FASTQ files for this activity if you like.

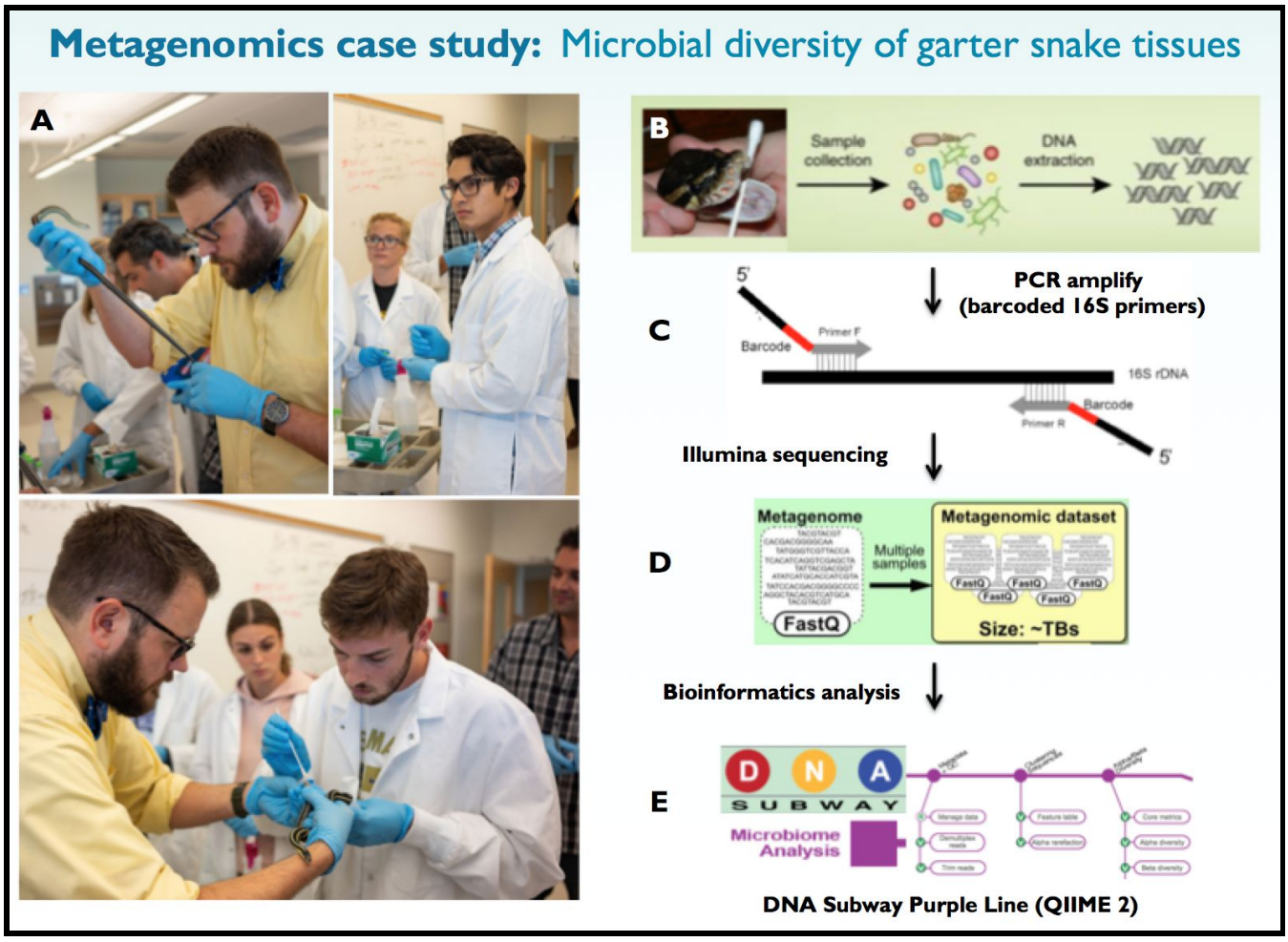


Figure 1. Overview of the in class metagenomics project. Using a saline swabbing technique, microbial samples were collected from garter snake tissues in class **(A)**. Swabs were placed in sterile tubes to release collected microbes & DNA was extracted for downstream analysis **(B)**. Barcoded primers were used to PCR amplify the microbial 16S ribosomal DNA repeat genes for each sample followed by Illumina sequencing of PCR amplicons **(C-D)**. The DNA Subway Purple Line web-based software can be used to analyze FASTQ data files generated from Illumina sequencing to reveal the

microbial population from our swabs **(E)**. Garter snakes were provided by Dr. Rocky Parker in the JMU Department of Biology (**A**; yellow shirt).

As previously discussed, **FASTQE** is a program that analyzes FASTQ files & reads out an emoji output as an indicator of the sequence's quality in the file. So a high quality read may look like this 😀, while this symbol 💩 indicates...well you get the idea.

In-class Assignment: Working in your lab groups, take turns operating the command line to analyze NGS FASTQ file data using **FASTQE** and another program called **FASTP**. All of the instructions & explanations are listed below. Create a new MS Word or GoogleDoc file and provide feedback wherever you see red text. If you get stuck, ask for help! Turn in this document at the end of the activity for your group's graded assignment. Make sure to rotate turns typing commands.

Student #1 starts on command line

- Open up your computer's **Terminal** (ie the **command line;** search Terminal in Mac finder)

When using the 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 using the ***change directory*** command (hit enter)
    - Then type: `cd Desktop` (hit enter)
- Here's what that looks like in Terminal

```
[Rays-Mac:~ rayenke$ #This is the command used to navigate to your desktop
[Rays-Mac:~ rayenke$ cd Desktop
Rays-Mac:Desktop rayenke$
```

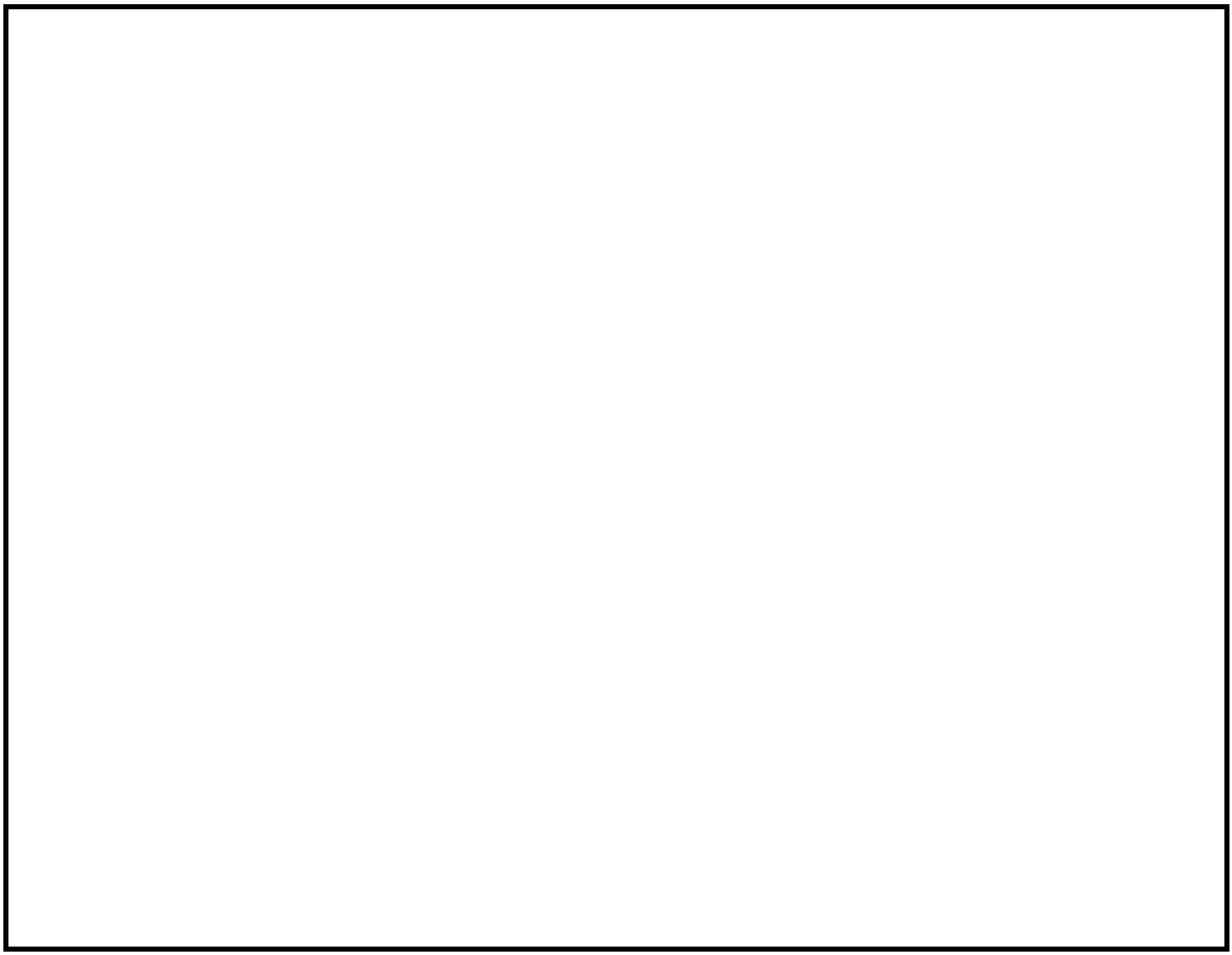
    - The # character is used to indicate comments. A line starting with # line doesn't do anything, in this case it's just telling you what the next line of code does which is navigate to your computer's desktop
    - We will stick with this convention throughout the activity
    - Comments can occur anywhere in a line, anything after the # will be ignored
- #check to see if you're still on your desktop using the ***print working directory*** command

    - Type: `pwd`
    - **Question 1**: If you've printed a path that doesn't make sense (i.e. the directory you navigated to is the incorrect directory) how would you go back to the previous directory? (hint, it includes the *change directory* command)

Today we will analyze 3 FASTQ files from a snake metagenomics project in Dr. Enke's BIO 481 Genomics class. FASTQ files are very large text files so it is common to store them as compressed zip files which need to be unzipped prior to analysis.

- Download the compressed fastq files here: https://bit.ly/3fastq (this is 1 file named *fastq.zip* containing 3 individual .fastq files from our experiment)
    - These files can be alternatively downloaded from the CourseSource journal article associated with this lesson
    - Find the *fastq.zip* file in Downloads folder and drag to desktop
- #unzip files using the ***unzip*** command
- #unzip followed by a file name will unzip and individual file
- #or unzip followed by ***.zip** will unzip all files ending with .zip
    - Type: `unzip *.zip`
    - **Question 2**: What's the purpose of using a compressed .zip file?
- #check to see if your files are unzipped using the ***list*** command
    - Type: `ls`
    - **Action 1:** Screenshot your terminal after using the `ls` command into your assignment. Make sure the 3 unzipped fastq files are present in this directory when you used the `ls` command
- #Install the FASTQE software using the ***pip install*** command
    - Type: `pip install fastqe`
- Note: if you don't have Anaconda installed at this point you will probably get an error message here
- #now you're ready to run FASTQE; you can use individual .fastq file names or type *.fastq to run all .fastq files in your directory
    - Type: `fastqe *.fastq`
    - Remember that fastq files are large so this command will take ~30 seconds/file to run
    - **Question 3**: What are the advantages and disadvantages to using the command `fastqe *.fastq` rather than `fastqe sample.fastq`?

- **Action 2:** Screenshot your Emoji output for your 3 fastq files & paste it into your assignment. It should look something like this:



Switch to student #2 typing on command line

Notice that 1 of your files (Male5-oral1) seems to have lower quality than the others based on the Emoji readout. Let's look more closely to see how bad the data is.

- #Open the FASTQE help page to view the "optional arguments", these are all of the options and setting for the program
- Type: `fastqe --help`
    - **Question 4:** Which optional argument will show the version# of FASTQE?
- #Add the scale command to the fastqe command to view the Phred score associated with each emoji in your output. Try this just for the Male5-oral1 file.
- Type: `fastqe Male5-oral1.fastq --scale`

- **Question 5:** A Phred score of ≤20 is considered a poor quality base call. How many poor quality base calls are at the 3' end of this read?

Switch to student #3 typing on command line

Let's use another program called **Fastp** to get a more conventional readout of .fastq file data. **Fastp** is similar to the **FastQC** program we previously discussed, however it also has a trimming tool to cut out or **filter** low quality sequences out of our file.

- #Type the command `pwd` to determine if the current directory is sufficient for your new directory
- #Create a new folder on the desktop called *fastp* using the ***make directory*** command
- Type: `mkdir fastp`
- Drag your 3 unzipped fastq files from the desktop into this folder
- #navigate to the fastp folder
  - Type: `cd fastp`
- #Install fastp
  - Type: `conda install -c bioconda fastp`
  - If your terminal holds on `a ñtype Y to proceedò` type Y, enter

Fastp Usage Notes:

- `fastp` is the name of software that will check the quality of the fastq file
- `-i [input.fastq]` -i option specifies the input file for fastp (required)
- `-o [ouput.fastq]` -o option specifies the output file for fastp (required)
- `--html [ouput.html]` --html option specifies the name of the HTML report for fastp (optional)
- `--json [ouput.json]` --json option specifies he name of the JSON report for fastp (optional)
- #run fastp on the lower quality Male5-oral1.fastq file
  - Make sure to name your .json and .html output files "Male5-oral1"
  - Type: `fastp -i Male5-oral1.fastq -o out.Male5-oral1.fastq --html` [name of file]`.html --json` [name of file]`.json`
- 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.Male5-oral1.fastq)

- Open your .html QC report & collect some before and after filtering data
  - **Question 6:** From the “Summary” data, how many reads are in this FASTQ file before and after filtering?
  - **Action 3:** From the “Before Filtering” & “After Filtering” data, screenshot the before and after Quality plots into your assignment
    - This is the per base quality plot with quality (or Phred score) on the y and base position on the x axis
  - **Question 7:** How do the before and after plots compare?
  - 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.Male5-oral1.fastq`
- **Action 4:** screenshot your Emoji output for your pre & post trimmed fastq file into your assignment
  - **Question 8:** How do your before & after filtering emoji outputs look?
  - **Question 9:** Which tool (fastqe or fastp) did you find easier to use?
  - **Question 10:** Which tool (fastqe or fastp) do you think is a more reliable research grade tool?

Switch back to student #2 typing on command line

- Move all your fastq.zip file and fastp folder from your desktop to the trash to reset the computer for the next class.
- #navigate back to the desktop
- Type: `cd`
- Type: `cd Desktop`
- #list your desktop directory
- Type: `ls`
  - **Action 5:** Screenshot your desktop directory into your assignment. Make sure the fastq.zip file & fastp folder have been removed from the desktop

To sum up, you just analyzed Illumina NGS .fastq data quality using Emoji output. You then filtered out low quality sequences & output before & after QC plots. **You did all of that on the command line, congrats!** 👋 🎉 👊

*Save and electronically submit your assignment to your instructor

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

#download the 3 compressed fastq files from the web and move to your desktop:

https://bit.ly/3fastq

#navigate to your desktop directory

```
cd Desktop
```

#Unzip fastq files using the unzip command

```
unzip fastq.zip or unzip *.zip
```

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

```
ls
```

#Install the fastqe software

```
pip install fastqe
```

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

```
fastqe *.fastq
```

#Open the FASTQE help page

```
fastqe --help
```

#Add emoji scale to the fastqe output for the Male5-oral1 file

```
fastqe Male5-oral1.fastq --scale
```

#Create a new desktop folder called *fastp*

```
mkdir fastp
```

#Drag your 3 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 Male5-oral1.fastq file

```
fastp -i Male5-oral1.fastq -o out.Male5-oral1.fastq --html
Male5-oral1.html --json Male5-oral1.json
```

#use the fastp output .fastq file to rerun fastqe

```
fastqe out.Male5-oral1.fastq
```

Examples of fastp pre & post filtering metrics (Male5-oral1 file):

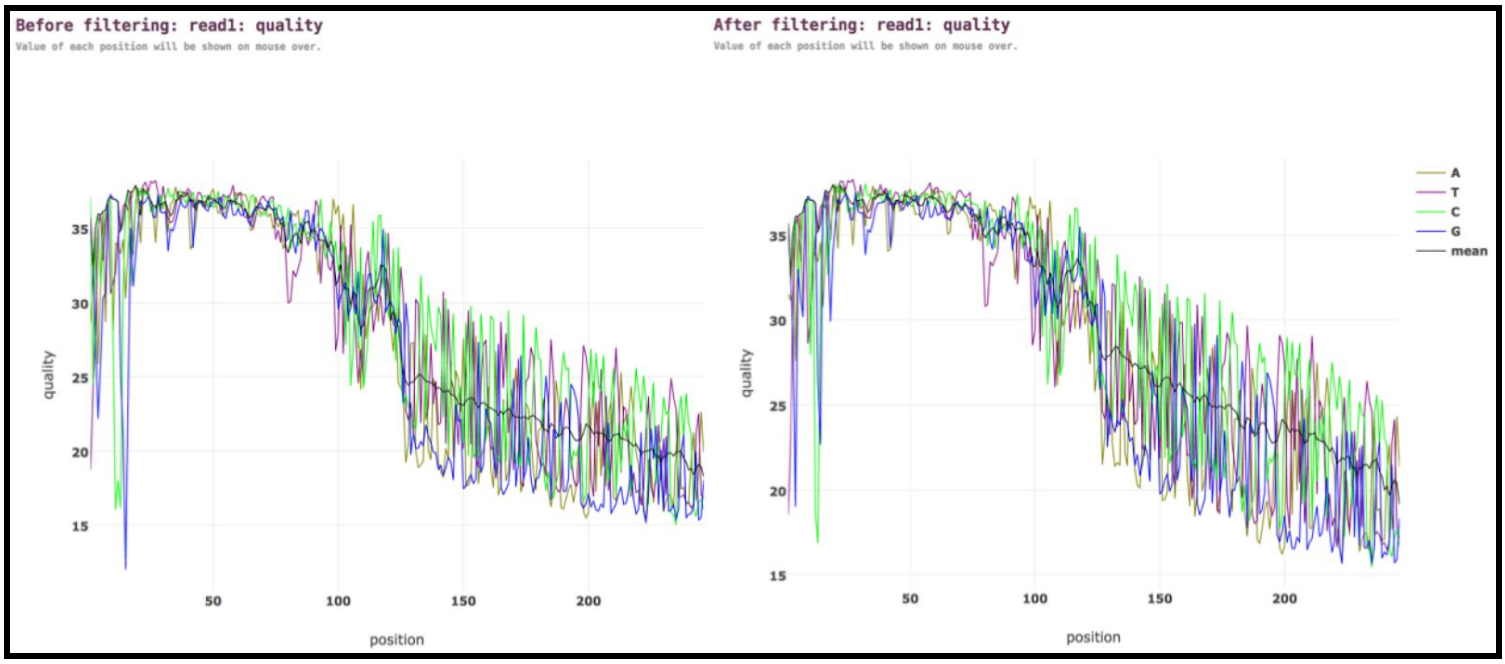
Before filtering

total reads:	21.345000 K
total bases:	5.250870 M
Q20 bases:	3.514259 M (66.927176%)
Q30 bases:	3.074805 M (58.558010%)
GC content:	50.902384%

After filtering

total reads:	15.539000 K
total bases:	3.822594 M
Q20 bases:	2.788131 M (72.938193%)
Q30 bases:	2.496501 M (65.309081%)
GC content:	50.981977%

Examples of fastp pre & post filtering quality plots:



Answer key for instructors:

Question 1: If you've printed a path that doesn't make sense (i.e. the directory you navigated to is the incorrect directory) how would you go back to the previous directory? (hint, it includes the *change directory* command)

Answer 1: Type: `cd`
Then type: `cd Desktop`

Question 2: What's the purpose of using a compressed file?

Answer 2: FASTQ files are large files; compressing them together into 1 .zip file makes them smaller and more convenient to download, but they often need to be unzipped in order to analyze them

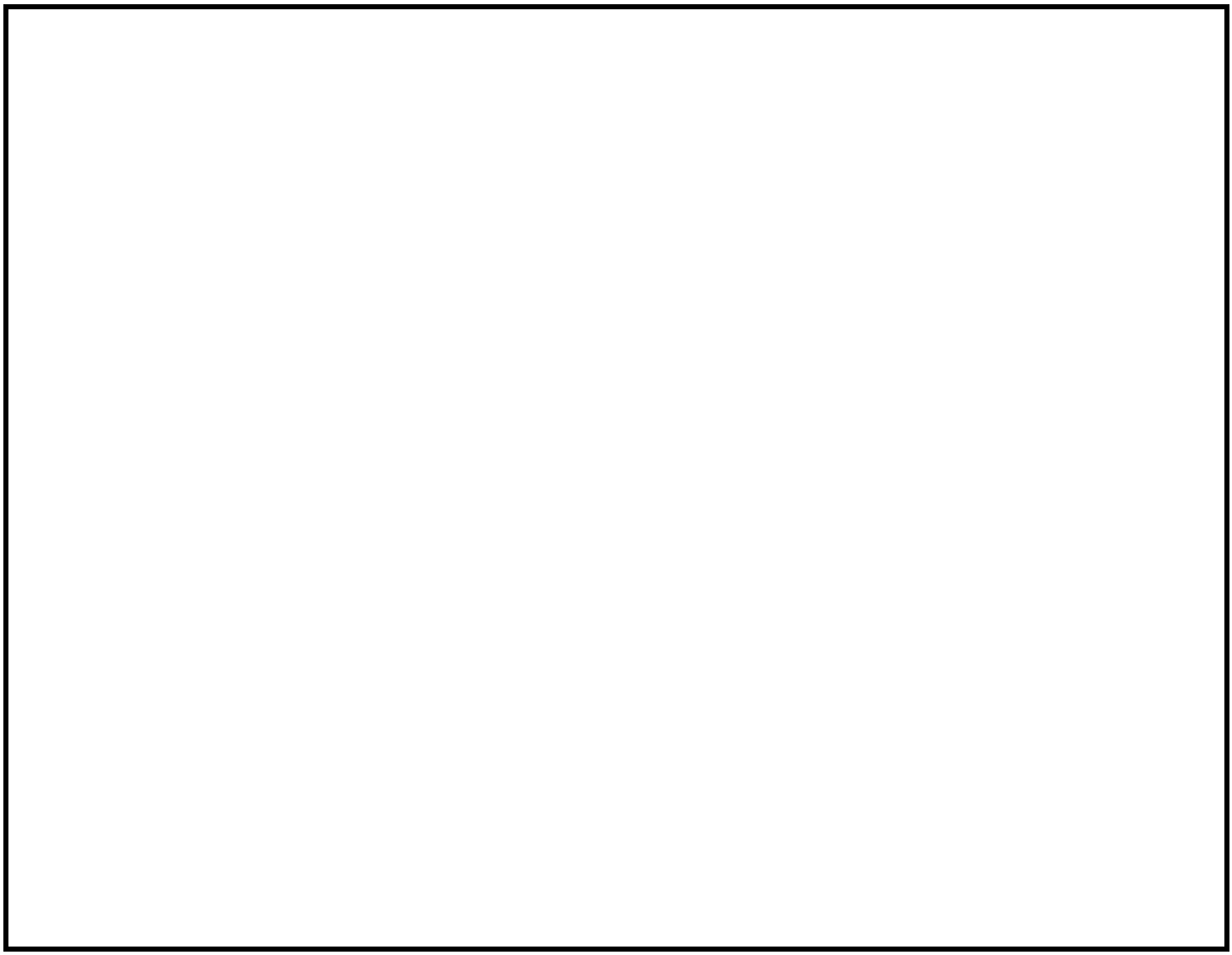
Action 1: Screenshot your terminal after using the `ls` command into your assignment. Make sure the 3 unzipped fastq files are present in this directory when you used the `ls` command

this will look different depending on the computer; the key is that the compressed fastq file and the 3 unzipped fastq files are present

Question 3: What are the advantages and disadvantages to using the command `fastqe *.fastq` rather than `fastqe ñsampleò.fastq`?

Answer 3: Rather than typing in each file name you want to analyze you can use the *.fastq command to analyze any file in your directory that has a .fastq extension.

Action 2: Screenshot your Emoji output for your 3 fastq files & paste it into your assignment. It should look something like this:



Question 4: Which optional argument will show the version# of FASTQE?
Answer 4: --version

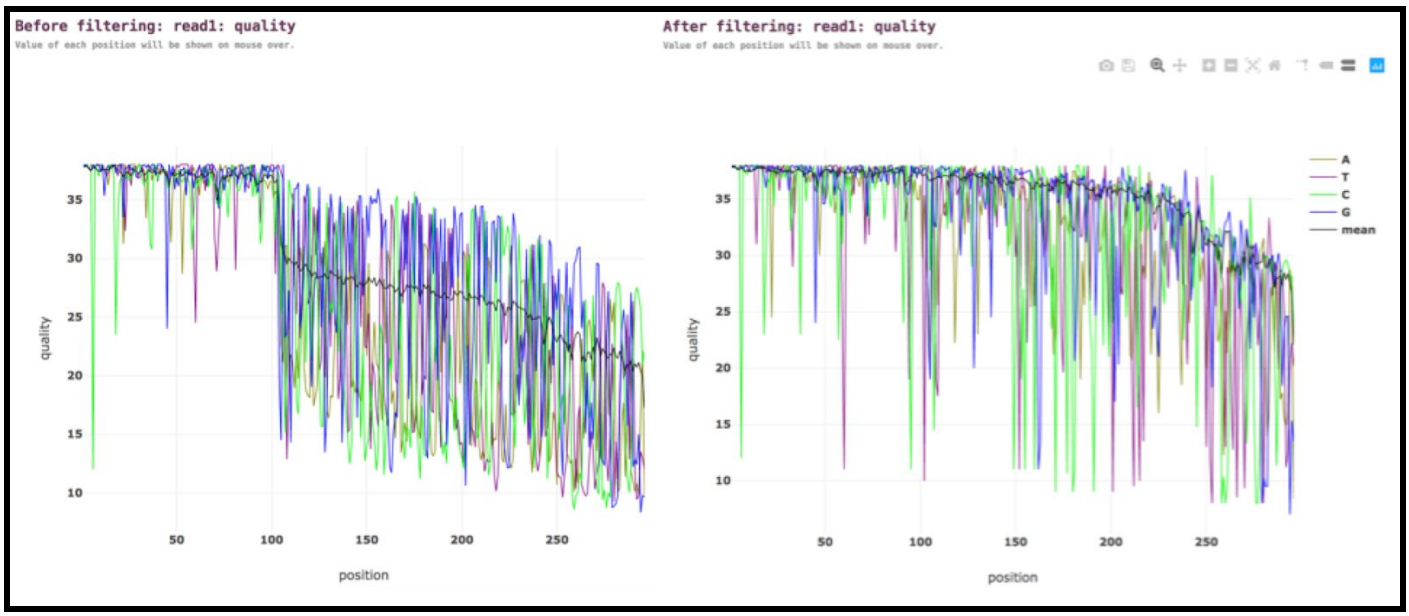
Question 5: A Phred score of ≤20 is considered a poor quality base call. How many poor quality base calls are at the 3' end of this read?

Answer 5: 30 base calls

Question 6: From the “Summary” data, how many reads are in this FASTQ file before and after filtering?

Answer 6: 21.3K reads before filtering; 15.5K reads after filtering (this is from the FASTP table data)

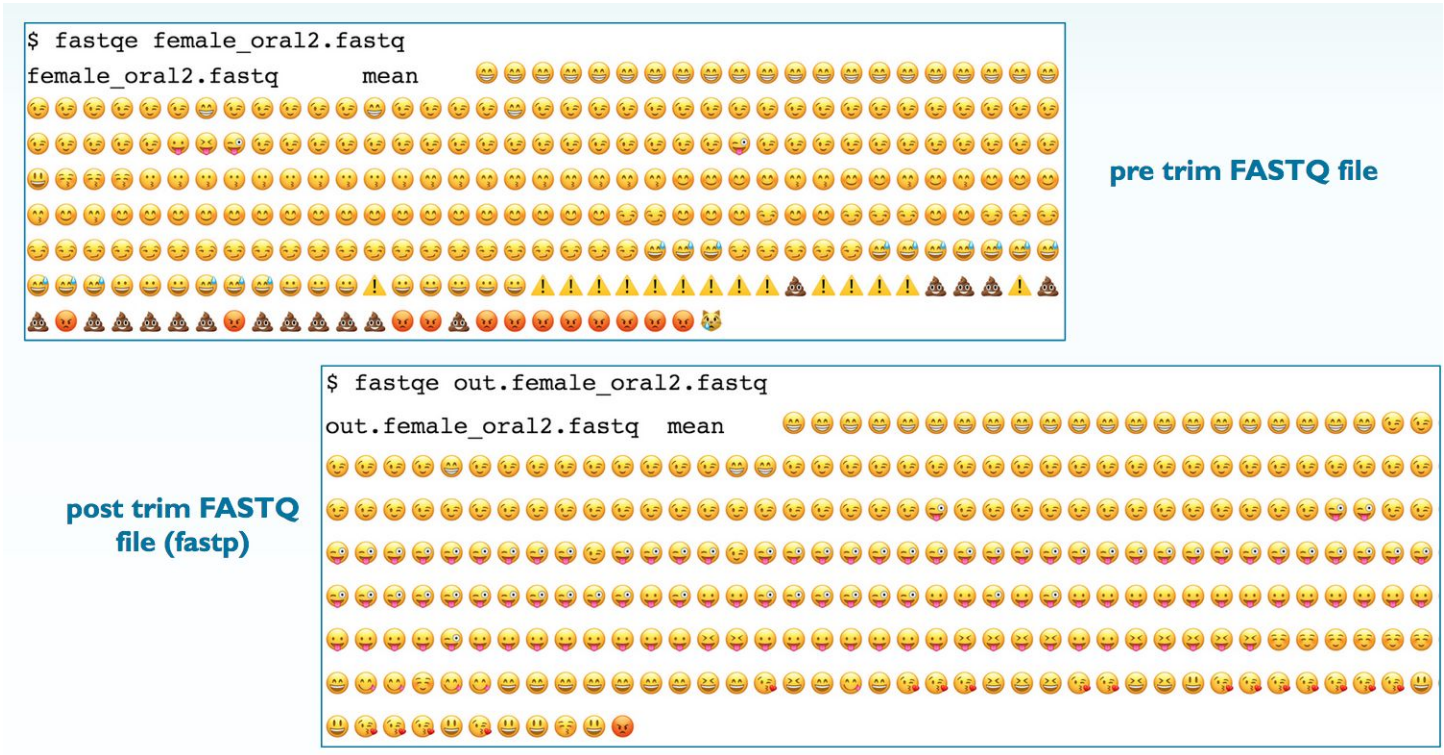
Action 3: From the “Before Filtering” & “After Filtering” data, screenshot the before and after Quality plots into your assignment. They should look like this:



Question 7: How do the before and after plots compare?

Answer 7: By filtering out the low quality sequences in the FASTQ file, the average read quality in the 3’ end of the read is dramatically improved.

Action 4: screenshot your Emoji output for your pre & post trimmed fastq file into your assignment (they should look something like this)



Question 8: How do your before & after filtering emoji outputs look?

Answer 8: The post trimmed file has less negative emojis (poop, !, frowny face, etc)

Question 9: Which tool (fastqe or fastp) did you find easier to use?

Answer 9: This is a subjective question, but generally students find FASTQE easier to use.

Question 10: Which tool (fastqe or fastp) do you think is a more reliable research grade tool?

Answer 10: FASTP is an actual research-grade tool. FASTQE is just for fun!

Action 5: Screenshot your desktop directory into your assignment. Make sure the fastq.zip file & fastp folder have been removed from the desktop

The primary objective of this task is to make sure students delete all of the files for this activity off of the lab computers prior to the next class using them. Please remind them to do this to avoid confusion for subsequent sections.