Bioinformatics Pipeline

1. Finding your sample

1. When you first log onto Ron, you will be in your home directory. If you are ever lost, you can return to your home directory by typing: **cd**
2. The directory containing your data is located in your home directory. It should start with: **Sample_**. If you do not know what the directory is called, type: **ls**
3. Use the **cd** command to change into the directory containing your sample data. For instance, if the directory containing your sample data is called **Sample_FP999**, you would do:

```
cd Sample_FP1
```

4. Confirm that you are in the correct directory by typing: **pwd**

2. Running trim_script.sh

1. Type **ls**. You should see trim_script.sh in your sample directory. You should also see your reads (they end with fastq.gz).
2. If your sample had a lot of data, the reads are probably split up between four or more files. To make sure you all the forward and reverse reads merged together, enter the two following commands in order (yes, those are asterisks):

```
cat *R1* > forward.fastq.gz
cat *R2* > reverse.fastq.gz
```

3. Type **ls**. Check to make sure that **forward.fastq.gz** and **reverse.fastq.gz** exists in your directory now.
4. It's time to use trim_script.sh! Type in the following command:

```
trim_script.sh forward.fastq.gz reverse.fastq.gz
```

5. Wait until the trimming is finished. Afterwards, type **ls**. You should see paired_forward.fastq.gz and paired_reverse.fastq.gz. You will be using these later.

3. Running Spades

1. You should be in your sample data directory, where the trimmed reads are located.
2. Spades can take a while to run, so we use the **nohup** command to make sure it runs even if we disconnect from the server.
3. You can run the assembler using this command *(this might take a while)*:

```
nohup spades.py -o assembly -1 paired_forward.fastq.gz -2 paired_reverse.fastq.gz
```

4. It should say "appending to nohup.out". You might need to hit enter.
5. Wait until this finishes (you will get your command prompt back)
6. Type **cd assembly** to change into the assembly directory.
7. Type **ls**. You should see the **contigs.fasta** file.
8. Type **less contigs.fasta** to view the fasta file. You can exit by typing **q**.

4. Running Quast

1. Make sure you are in your assembly directory (see last step).
2. You can run QUAST on your assembled contigs by doing the following:

```
quast.py -o quast_report contigs.fasta
```

3. Wait for QUAST to finish. Type **ls**. You should see the **quast_report** directory.
4. Go into this directory by typing **cd quast_report**
5. Type **ls**. You should see **report.txt**, which contains quality metrics.
6. Type **less report.txt** to see the contents of this file. Make note of the N50 result found in this file.
7. Change back into your assembly directory by typing: **cd ..**

5. Running Prokka

1. You should be back in your assembly directory. If you type **ls**, you should see the **contigs.fasta** file.
2. You can run prokka by doing the following (assuming you're analyzing bacteria):

```
prokka -o prokka_report contigs.fasta
```

3. Wait for prokka to finish. Type **ls**. You should see the **prokka_report** directory.
4. Change into the directory by typing **cd prokka_report**

6. Analyzing Prokka Data

1. You should be in your prokka_report directory. Type **ls** to see its contents.
2. You should see files beginning with PROKKA, each with a diffferent extension.
   - **faa:** this file contains annotated proteins found in your sample.
   - **ffn:** this file contains annotated CDS found in your sample.
   - **gff:** this file contains annotations that indicate where genes/RNA/tRNA are found.
   - **txt:** this file contains summary statistics about what was found in your sample. For instance, it lists the total number of CDS that were found.