Micro 612.2 – Bacterial Genomics Pre-course Homework

Instructor – Evan Snitkin (esnitkin@umich.edu)
Course assistant – Ali Pirani (apirani@umich.edu)
Course assistant – Shawn Whitefield (swhitefi@umich.edu)

The goal of this homework assignment is to prepare you to get as much as possible out of the workshop. While the focus of the workshop is on analyzing bacterial genomes, you will also be learning about doing computational research using a combination of Unix and R. Unix and R form the foundation for not just microbial genomics research, but bioinformatics research in general, which is why we had you first take the Software Carpentry workshop before this one.

Review you Unix and R
To start with, I strongly recommend reviewing the SWC materials on Unix (https://swcarpentry.github.io/shell-novice/) and R (http://swcarpentry.github.io/r-novice-inflammation/). The Unix material is especially important, as many of the tools we will be using are only available on a Unix platform. If you would like additional practice with Unix, check out command line boot camp: http://rik.smith-unna.com/command_line_bootcamp/?id=9xnbkx6eaof

Getting setup on Flux
Next, we want to get you comfortable using the Flux compute cluster, which is the platform we will be working on during the workshop. First, to use flux you have to request an account (https://arc-ts.umich.edu/fluxform) and activate Duo authentication (http://its.umich.edu/accounts-access/uniqnames-passwords/two-factor-authentication/enroll-in-duo).

Once you have gotten your Flux account activated, try to login. To do this, open a Unix session on your computer (terminal for Macs and Gitbash for Windows) and establish an ssh (secure shell) connection to flux by typing the following command:

ssh username@flux-login.arc-ts.umich.edu

You will then be prompted for your level 1 password, followed by your preferred method for Duo authentication. If you have successfully logged in, you should see some welcome text and usage agreement information.

Getting oriented on Flux
When you log in to a remote system, you are typically placed in your home directory. Use the "pwd" command (present working directory) to see where your home directory is on flux.

For the purposes of the class we created an alternative home directory inside the class folder.

To verify that your home directory exists:

i) Go to "/scratch/micro612w17_fluxod/"

ii) List the directories and verify that a directory exists that corresponds to your username

iii) Go into your user directory

iv) Type "pwd" again to see what the path is to your class home directory

Creating a shortcut to your class home directory

You learned in SWC that there are several ways to enter "cd" commands to get back to your home directory (one is to just type "cd" with no specified directory). Since you will be working in your class home directory a lot, it would be nice to have an easier way to get there than typing "cd /scratch/micro612w17_fluxod/username". To do this we are going to make a shortcut, which is essentially a user-defined keyword that represents some longer Unix command. We are going to put this shortcut into a special file that is located in your home directory named ".bash_profile". What's special about the ".bash_profile" file is that every time you log into Flux, the commands in that file are executed. So, if we define our shortcut in there, it will always be available to use whenever you login ☺. More information here and here

To setup your shortcut do the following:

i) Go to your home directory (type "cd" or "cd /home/username")

ii) Type "ls –a". "ls" lists the contents of your home directory and "-a" lists hidden files (e.g. those that start with a period). You should see among other things, a file called ".bash_profile".

iii) Open your ".bash_profile" file with pico, or another text editor of choice

iv) Add in the following at the end of the file:

alias micro612="cd /scratch/micro612w17_fluxod/username"

v) Save and exit out of your text editor and type "source .bash_profile". This will execute the commands in your ".bash_profile" (*Note – you only have to use source this one time, in the future the commands in .bash_profile will be executed each time you log in).

vi) Verify that your short cut works by typing "micro612" in terminal and then "pwd" to verify that you are in your class home.

Your first command line sequence analysis!!!

Up until now you've probably accessed sequence data from NCBI by going to the website, laboriously clicking around and finally finding and downloading the data you want. There are a lot of reasons that is not ideal:

i) It's frustrating and slow to deal with the web interface

ii) It can be hard to keep track of where the data came from and exactly which version of a sequence you downloaded

iii) Its not conducive to downloading lots of sequence data

To download sequence data in Unix you can use a variety of commands (e.g. sftp, wget, curl). Here, we will use the curl command to download some genome assemblies from the NCBI ftp site:

i) Go to you class home directory (use your micro612 shortcut!)

ii) Create a directory for this assignment ("mkdir pre_hw") and go into the directory ("cd pre_hw")

iii) Now get three genome sequences with the following commands:

```
curl
ftp://ftp.ncbi.nlm.nih.gov/genomes/refseq/bacteria/Acinetobacter_baumannii/latest_assembly_versions/GCF_000018445.1_ASM1844v1/GCF_000018445.1_ASM1844v1_genomic.fna.gz >
Acinetobacter_baumannii.fna.gz

curl
ftp://ftp.ncbi.nlm.nih.gov/genomes/refseq/bacteria/Klebsiella_pneumoniae/latest_assembly_versions/GCF_000220485.1_ASM22048v1/GCF_000220485.1_ASM22048v1_genomic.fna.gz > Klen_pneu.fna.gz

curl
ftp://ftp.ncbi.nlm.nih.gov/genomes/refseq/bacteria/Escherichia_coli/all_assembly_versions/GCF_000194495.1_ASM19449v2/GCF_000194495.1_ASM19449v2_genomic.fna.gz > E_coli.fna.gz
```

iv) Decompress the compressed fasta file using gzip

```
gzip -d Acinetobacter_baumannii.fna.gz
```

```
gzip -d Klen_pneu.fna.gz
gzip -d E_coli.fna.gz
```

v) These files are genome assemblies in fasta format. Fasta files are a common sequence data format that is composed of alternating sequence headers (sequence names and comments) and their corresponding sequences. Of great importance, the sequence header lines must start with ">". These genome assemblies have one header line for each contig in the assembly, and our goal will be to count the number of contigs/sequences. To do this we will string together two Unix commands: "grep" and "wc". "grep" (stands for global regular expression print), is an extremely powerful pattern matching command, which we will use to identify all the lines that start with a ">". "wc" (stand for word count) is a command for counting words, characters and lines in a file. To count the number of contigs in one of your fasta files enter:

```
grep ">" E_coli.fna | wc -l
```

vi) Try this command on other assemblies to see how many contigs they have

Your first sequence analysis program!!!

OK, so now that we have a useful command, wouldn't it be great to turn it into a program that you can easily apply to a large number of genome assemblies? Of course it would! So, now we are going to take our cool contig counting command, and put it in a shell script that applies it to all files in the desired directory.

i) Copy "/scratch/micro612w17_fluxod/shared/fasta_counter.sh" to your current directory (Hint – use the "cp" command)

ii) Open "fasta_counter.sh" in pico and follow instructions for making edits so it will do what we want it to do

iii) Run "bash fasta_counter.sh ./" on the current directory and verify that you get the correct results (*Note - ./ represents the current directory, you could also put /scratch/micro612w17_fluxod/username/pre_hw/)

```
bash fasta_counter.sh ./
```

Plotting genomic coverage in R

Data visualization plays an important role in organizing, analyzing and interpreting large amount of omics data. R is a flexible and powerful tool for manipulating and visualizing these types of data. The following task will help you brush up on how to load and plot data as you visualize some high-throughput sequencing data.

Background:

A common analysis of sequencing data is to map raw sequences back to some reference genome. For example, this will be done when trying to identify variation

in your sequenced genome relative to a reference, or when performing a counting experiment like RNA-seq. When doing a first pass evaluation of your sequencing experiment and mapping analysis, it is always a good idea to visualize the mapping coverage across the genome (e.g. number of times each base was sequenced or "covered" by a read). For instance, when performing whole genome sequencing, you ideally want uniform coverage across the genome, such that you have sufficient depth at each position to confidently distinguish sequencing errors from actual variants.

Here we are going to look at the sequencing coverage for an *E. coli* whole genome sequencing experiment. The input for this task is a comma-separated file, which contains average sequencing coverage information (i.e average number of reads mapped) to each 1000 base pair window in reference genome.

Let's copy Ecoli_coverage_average_bed.csv file from flux shared directory to your desktop using 'scp'. 'scp' stands for secure copy and is used for securely transferring files between remote host/server(flux) and your local computer system. (Both directions)

i) Open a new terminal window in your local system (We don't need to login to flux) and type the below command.

scp username@flux-xfer.arc-ts.umich.edu:/scratch/micro612w17_fluxod/shared/Ecoli_coverage_average_bed.csv ~/Desktop/

**Note: You can use your choice of folder/path to copy the file instead of "~/Desktop/"

ii) Now, fire up R console or studio and import the file (Ecoli_coverage_average_bed.csv) using any type of data import functions in R (read.table, read.csv etc.)

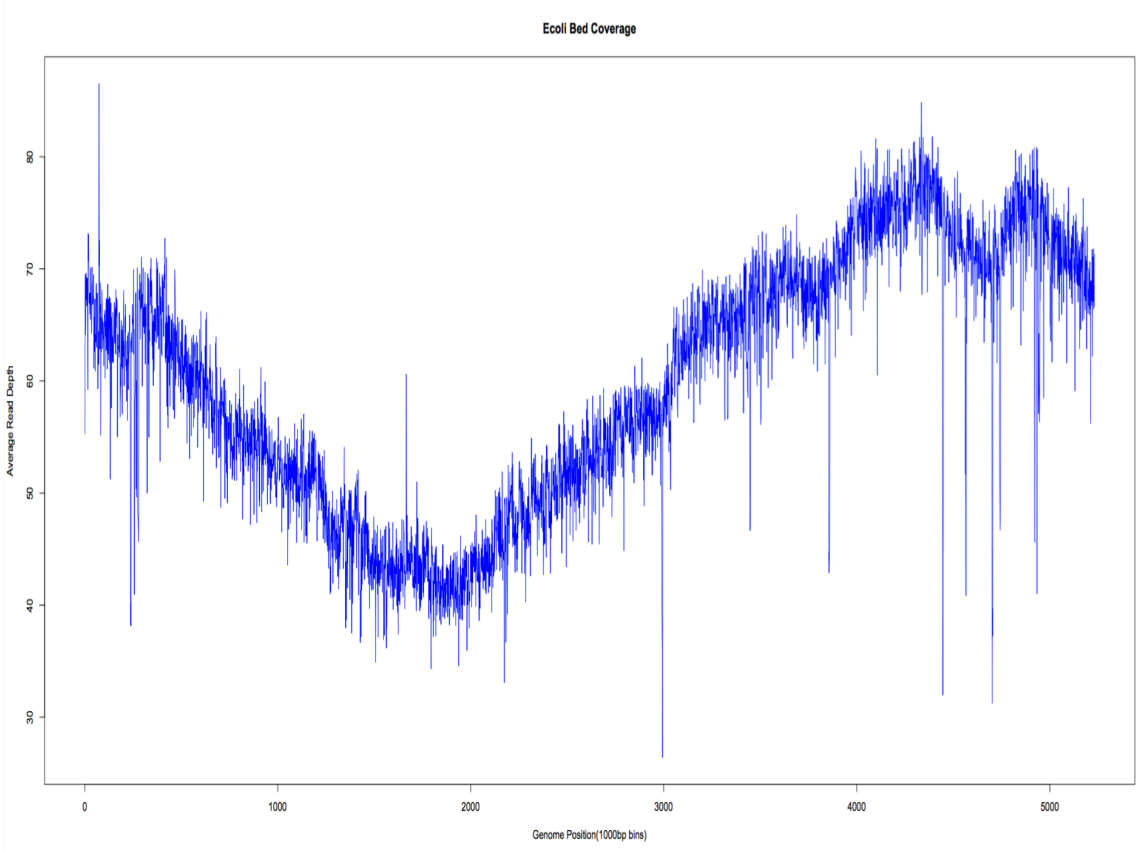
**Hint: The file is comma-separated and contains a header line ("bin,Average_coverage") so use appropriate parameters while importing the file.

iii) Once the data is imported into R object, you can plot the column "average coverage" as a time series plot to assess the coverage of your mapped reads across genome.

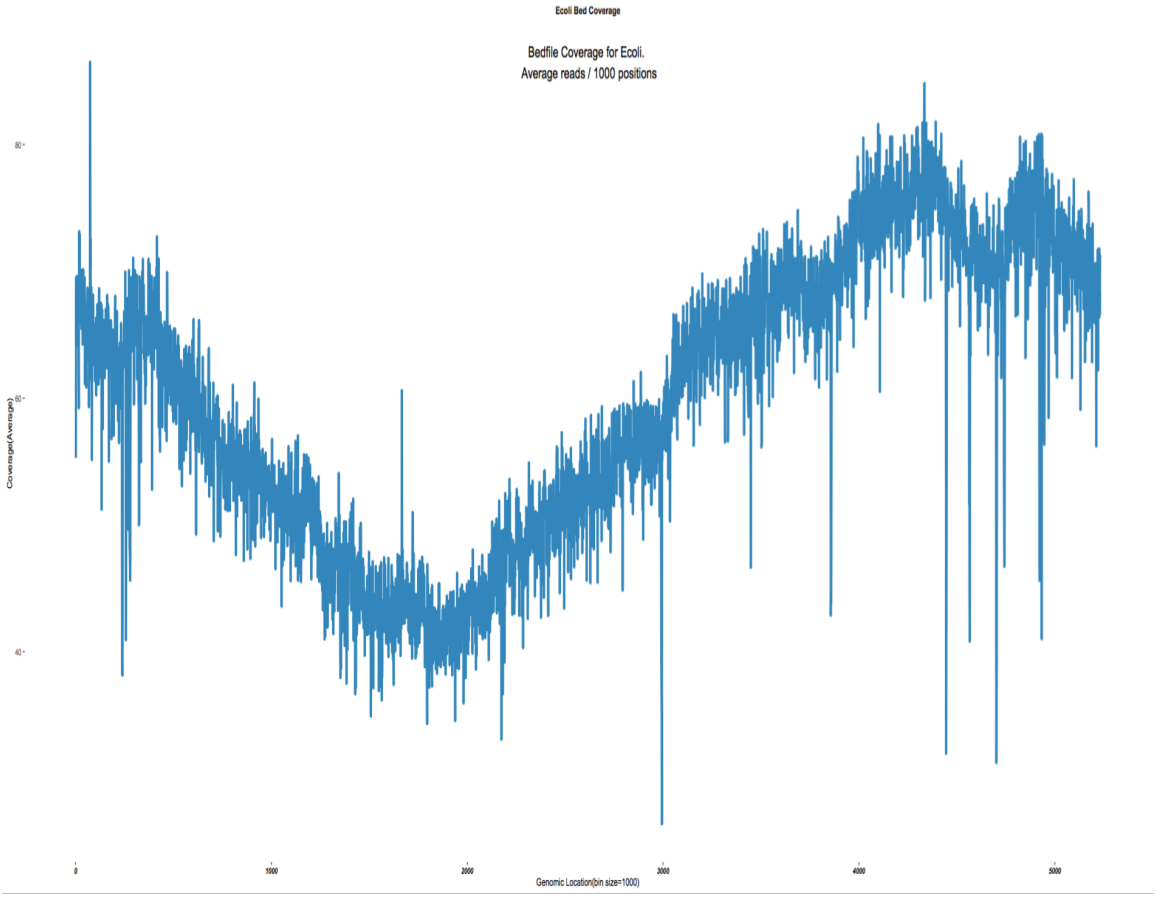
**Note: A time series plot is a graph that you can use to evaluate patterns and behavior in data over time. Here, we can employ the same plot to see the pattern i.e read depth/coverage at each 1000 bases (represented by a column named "bins" where each row represents a bin and the

corresponding second column represents Average number of reads mapped to each bin) using the simplest R function for time series such as "plot.ts" (http://stat.ethz.ch/R-manual/R-devel/library/stats/html/plot.ts.html)

An example plot.ts plot for Ecoli_coverage_average_bed.csv is shown below for your reference. Notice that the coverage does not look even, but rather has a wave pattern. Any ideas of what might be causing this uneven coverage (hint – the highest coverage is at the origin of replication and the lowest coverage is at the terminus).



For advance and more beautiful visualization, ggplot2 can be employed to display the same plot. An example ggplot2 plot for Ecoli_coverage_average_bed.csv is shown below for your reference.



Genomics tools for the workshop

In addition to the work we will be doing on Flux and R, we will use several graphical tools for the purposes of data visualization.

Please download and install the following on your computer.

i) IGV - https://www.broadinstitute.org/software/igv/log-in
ii) Artemis - http://www.sanger.ac.uk/science/tools/artemis
iii) ACT - http://www.sanger.ac.uk/science/tools/artemis-comparison-tool-act
iv) Mauve - http://darlinglab.org/mauve/download.html
v) Seaview - http://doua.prabi.fr/software/seaview

Additional background reading

i) Bacterial genomics review
http://www.nature.com/nrmicro/journal/v13/n12/full/nrmicro3565.html

ii) Review on the use of genomics to study epidemiology
http://www.nature.com/nrg/journal/v13/n9/full/nrg3226.html

http://www.sciencedirect.com/science/article/pii/S1369527414001635

iii) Review on the use of genomics to study bacterial pathogenesis
http://www.sciencedirect.com/science/article/pii/S1931312816301512