Genomes mini-project

Genomes: | *Caenorrhabditis elegans* | *Borrelia burgdorferi* | *Ambystoma* |

Create a repository on GitHub for your Genome mini-project

In a README.md file, give basic information on the genome, including:

- Size of genome
- Structure (such as, linear or circular, how many chromosomes, how many plasmids)
- Estimated number of protein-coding genes
- Where to access the genome sequence and at what coverage it was sequenced
- Any other interesting facts about the genome that turn up

On your own, search for sites from which you can download the genome sequence of your organism and make a note of them (paste the url in your lab notebook, for example).

Download the genome sequence to your Discovery Cluster "scratch" account (`/scratch/<username>`) using good practices for how you store the files (in other words, set up a set up folders and subfolders that have a logical structure).

Analysis: part one

To analyze your organism's genome sequence, you'll use two programs that are available as module on Discovery: **seqtk** and **emboss**.

Remember that you need to load the modules into your workspace (scratch) to use them.

```
module load emboss

module load seqtk
```

When you download your organism's genome sequence, multiple fasta files are usually included in one file. You can use the emboss command 'infoseq' to see what a file contains:

```
infoseq <filename>
```

One item on your deliverables list is to generate the reverse complement of a genomic sequence from your organism. First you'll need to extract just one fasta sequence from the whole batch. You can use a seqtk command to extract a sequence. First you need to put the sequence identifier (usually the accession number) into a file. You can do this quickly using echo.

```
echo '<identifier>'  >  <make-up-a-filename>
```

For example, it might look like this:

```
echo 'NC_008524.2' > list.txt
```

This will create a file called `list.txt` that contains just one line: NC_008524.2.

Then I use the command that tells seqtk to extract that sequence from the larger file and store it in a new file by itself:

```
seqtk subseq <file-with-many-sequences> list.txt >
NC_008524.2.txt
```

Now we can generate the reverse complement of this sequence, putting it into a new file:

```
revseq NC_008524.2.txt NC_008524.2.rev
```

Remember to keep using tab-complete, and check after commands that the new file is added to your directory by using `ls`.