MCB 5472 Assignment #3: Determining Genome Quality

February 5, 2014

In deference to cancelled class, assignment #2 will be due at 1pm Thursday Feb 6/14. Similarly, today's assignment is abbreviated to only require the coding techniques we have covered to this point. Please post additional questions to the website and/or email myself or Peter if you have further questions.

Before the start of the next class (Wed Feb 12/14): Write each script and run it. Send your files to Jonathan as a .zip folder (jonathan.klassen@uconn.edu; please remember to put MCB 5472 somewhere in the subject line).

From last week's Assignment #2 Question #3: [5 marks] Write a script that takes any input gene sequence and outputs its translation. Send Jonathan your input file, output file and script. Recall our discussion on hash data structures, and especially consider its application to converting codons to amino acids. Also recall our discussion of `for` loops and how you can precisely control moving through an array (in codon-like steps of three, for example). Remember that `for` returns array positions, in contrast to `foreach` which returns array values. `for` is therefore more flexible because you can ask for the data in any array location, and even other locations at the same time if you define them relative to the first.

This week's new assignment question: [10 marks] Go to NCBI and download the following genomes and their corresponding protein sequences:

Escherichia coli O104:H4 str. 2009EL-2050 (complete)
Escherichia coli O104:H4 str. 2009EL-2071 (complete)
Escherichia coli O104:H4 str. 01-09591 (draft)
Escherichia coli O104:H4 str. 11-02030 (draft)
Escherichia coli O104:H4 str. 11-02092 (draft)
Escherichia coli O104:H4 str. E112/10 (draft)
Escherichia coli O104:H4 str. LB226692_2.0 (draft)

All of these strains are from the same *E. coli* outbreak in Germany a year ago and are highly related as befits their having the same serotype. The top two are complete genomes and the rest are drafts. All can be obtained through either the NCBI Genome portal or the FTP site.

Note #1: Draft genomes on the NCBI FTP site come as "tarballs", a zipped file format common on UNIX-based operating systems. Each needs to be extracted using this command: "`tar –zxvf [tarball name]`*". Tarballs often have file endings like *.tgz or *.tar.gz.*

Note #2: All of the genomes listed are exclusively contigs or complete chromosomes/plasmids (even if the file is labelled scaffolds!). They can be most easily be worked with when joined into a single file using the `cat` *function in the terminal, e.g., "*`cat [file1] [file2] > [newfile]`*" (logically: join file1 and file2 into a new file newfile) or "*`cat *.fna > all.fna`*" (logically: join all files in this directory ending in .fna into a new file all.fna).*

For each genome, calculate the number of contigs/chromosomes+plasmids, the number of proteins that they encode, and the genome size. You can write this in as many scripts as you want (I did it in two myself). Run your script(s) on each genome and tabulate the results any way you like; send me your script(s) and this table before next Wednesday's class.

Note #3: Some of you already figured this out, but perl has a `length` *function that will return the number of characters in a string. Remember that this will include any sort of white space (i.e., spaces, tabs, new lines). Example: "*`$string_length = length $string`*";* `$string_length` *will contain the number of characters in* `$string`*.*

Note #4: Because you need to analyze multiple input files, input file names should be passed to your scripts using the `@ARGV` *syntax. In reality, there are very few instances where this is not true.*

Note #5: A point I probably should of mentioned last week but neglected (my apologies! But you all seem to have found workarounds anyway): `for`, `foreach` *and* `while` *loops can be controlled using two other commands,* `next` *and* `last`*.* `next` *immediately jumps the code to the next iteration of that loop, whereas* `last` *jumps the code to the end of the loop code block (i.e., closing curly bracket). Examples:*

```
foreach (@inputfile){
      if ($line =~ /^>/){
            next; # jumps to next element in @inputfile
      }
}
```

```
foreach (@inputfile){
      if ($line =~ /^>/){
            last; # jumps out of the loop
      }
} # jumps to here!
```

Incidentally, the genomes that I chose for this exercise also demonstrate some of the issues that we talked about during my lecture on Monday, i.e., that there can be quality issues with some draft genomes. You should be thinking about your results in this context and be prepared for a class discussion of them next Wednesday.