Shell Programming Lab 1

Due: Thursday 24 January 2019 (just before class starts)
Points: 10

This assignment is designed to reinforce and build on what you learned in chapters 4, 5 and 6 of PCFB. Becoming comfortable working in the shell is an important step towards harnessing the power of the machine[1]. Work alone or in pairs, but each of you will need to *individually produce* your own complete solution for this lab.

Your work-product will be a single Bash script that accomplishes these six tasks:

1. Retrieve three sequence data files from a remote server, the URIs for the files are:
   `https://cs.earlham.edu/~charliep/courses/BioI/first.dat`
   `https://cs.earlham.edu/~charliep/courses/BioI/second.dat`
   `https://cs.earlham.edu/~charliep/courses/BioI/third.dat`

   These are small subsets of the output from sequencing 16S rRNA genes extracted from soil microbes from a local farm field. They are about 1/2,000,000th of the total sequencer output for those samples.

2. Extract all the sequences (lines) from first.dat and second.dat that contain the string 'AACCTTNN'. All of those sequences should end-up together in one file called fourth.dat

3. Append the contents of third.dat to fourth.dat to create fifth.dat

4. Display the number of lines, words and characters in fifth.dat

5. Extract all the lines (sequences) in fifth.dat that do not contain the string 'AACCTTNN'. All those sequences should end-up together in one file called sixth.dat

6. Using one command line, display only the number of lines in fourth.dat and sixth.dat

7. Using one command line, extract sequences in {first, second, third}.dat that contain ATG but do not contain TAG, the results should end-up together in one file called seventh.dat

8. Using one command line, extract sequences in {first, second, third}.dat that contain ATG and contain TAG, the results should end-up together in one file called eighth.dat

9. Using one command line, display only the number of lines in seventh.dat and eighth.dat

All of these tasks can be accomplished with basic shell commands, and while they could also be done with Python the point of this exercise is shell programming so let's stick with Bash for now. When your script is done and tested you should be able to run it with one command (i.e. "`$ ./my-script.sh`") and it should complete all of the nine steps listed above displaying the requisit output as it goes. If you have questions there are CS tutors available in the Lovelace lab on the second floor of the CST, or find Chris or Laurence

Your script should have a comment at the top with your name and the assignment name. Make a PDF file of your Bash script and a screenshot/typescript of a run of your script, and then upload that PDF to the Moodle assignment.

---

[1] If you are curious about why shells are so much more powerful than GUIs see Neal Stephenson's *In the Beginning... Was the Command Line*.