Assignment 2 - fun with scripting tools

RWH

January 25, 2019

Learning to use awk, grep and sed

There is a protein database file (4HKD.pdb) in Dr. Harrison's main directory. These files consist of many individual records (lines) each of which starts with a keyword that identifies it. The files are somewhat complicated. Your task is to use Unix tools to simplify looking at these files. *I*t is a good idea to write script files out as demonstrated in class, rather than trying to compose them on the command line. - you can turn the scripts in as part of the answer.

You should copy the file to your own area and write programs to solve the following problems.

1. Records other than "ATOM","CONNECT", "HETATM", "TER" and "END" are considered header records which describe the metadata about the molecule. Use grep to generate the header. Please give the grep command(s) and the header you found.

2. The records that have "HETATM" and "MSE" should be "ATOM " (the two spaces after ATOM are important) and "MET" respectively. (This reflects an experimental technique used to solve the structure - but results in a syntactical inconsistency that can cause problems). Please use sed and/or awk to fix this by replacing HETATM with ATOM and MSE with MET. Please give the commands you used and show the corrected lines.

3. Use awk to find the maximum and minimum x,y,z values for the ATOMs

```
ATOM     93  OG  SER A  12      20.901  10.643  45.146  1.00 34.66           O
ATOM     94  N   MET A  13      22.086  11.751  41.731  1.00 22.99           N
```

   The 7th through 9th fields are the x,y,z positions.

4. find the mean values for x,y,z for the HETATM records (same fields as ATOM records)

5. The standard name for a water molecule is HOH. Unfortunately it needs to be called WAT to be used by some (slightly braindead) computational chemistry program. Make the changes automatically with sed. What command did you use?

6. produce a list of atoms sorted by their b-factor (11th position in an ATOM record). How did you do it?

Lab Work

It was a dark and stormy night, well not really, but that small army of barred owls calling "no soup for you" has kept you awake all night. A distinctly large and hairy man nabs you on the way to Starbucks, shoves a moldy shoe in your hand, and whisks you to a castle in remote Scotland to help with something called 'muggle studies.' (You can see where this is going.) Due to an overly complicated monetary system based on prime numbers, the owners of the castle are nearly bankrupt. In a last desperate attempt to avoid foreclosure they turn to you to do the accounts. Fortunately, due to the 'magic of the ELF' the linux partition on your laptop still runs. Less fortunately, 'for security reasons' due to an unfortunate incident in the recent past Python does not. So you'll have to use bash and awk to achieve your magic. Since your ability to make it back to GSU for the next 3320 class depends on them staying solvent this is a matter of the first importance. (besides the beer is weak and tastes of butter.)

You are presented with a file of figures of the form 1/3/3 or -10/3/2 which correspond to galleons, sickles and knuts. There are 23 sickles in a galleon and 17 knuts in a sickle.

1. Write an AWK program to convert a string of the form +- n/m/o into an absolute number of knuts. 1/0/0 should go to 23*17 or 391. It may or may not be helpful to use SED to remove the '/'

2. Write another AWK program to generate a string of the form +- n/m/o from and absolute number of knuts. (this is the inverse of the last one.)

3. Use these programs to sum a list of numbers. (This will probably be a bash script and use another awk program to perform the sum.)

4. (time permitting) Find the ELF.