# Programming for Bioinformatics | BIOL7200

## Week 3 Exercise

#### General regex

1. Create regular expressions for the following; this is a theoretical exercise, but you're welcome to try out the regex using grep:
   1. Only a number that is a multiple of 5

ANS: **\d\*[05]**

* 1. Exactly 5 characters

ANS: **.{5}**

* 1. Any letter followed by a number

ANS: **[a-z|A-Z]\d+**

* 1. The first 3 columns of a BED file (Google UCSC BED format to find out the specifications of the standard BED format)

ANS: **chr.+\t\d+|NA\t\d+|NA**

* 1. The first 3 bases in a DNA sequence

ANS: ^**[A,T,G,C]{3}**

* 1. The last 3 bases in a DNA sequence

ANS: **[A,T,G,C]{3}$**

* 1. Two numbers followed by 2 lower case letters

ANS: **[0-9]{2}[a-z]{2}**

* 1. What does this regular expression match? \d\*\.\d{3}

ANS: It matches any number with three decimal points

#### Regular expression command exercises

1. Searching a file with grep
   1. Extract the knownGene.txt.gz from the files you downloaded from Canvas. Google the command if you don’t know how to extract it.

ANS: **gzip -d knownGene.txt.gz**

* 1. Use grep to get all genes on chr22

ANS: **grep “chr22” knownGene.txt** (displays all the information about the genes)

* 1. Use grep to get all and only those genes that occur on chr1

ANS: **grep chr1[^0-9] knownGene.txt**

1. Editing data streams with sed
   1. Take the results from **2.2** and duplicate each line

ANS: **grep "chr22" knownGene.txt | sed -n '{p;p;}'**

* 1. Change the chr position of every other line to cow

ANS: sed ‘s/chr/cow/g;n’ knownGene.txt

* 1. Delete the lines that have cow in them

ANS: **sed ‘/cow/d’ knownGene.txt**

* 1. Repeat **1**-**3**, but this time do it “in-place”. Read the man page to figure out what this means.

ANS: **grep "chr22" knownGene.txt | sed -n '{p;p;}' > result.txt**

sed -i ‘s/chr/cow/g;n’ result.txt

**sed -i ‘/cow/d’ result.txt**

#### Biologically-inspired problem

1. An *in silico* restriction enzyme digestion.  
   In a parallel universe, restriction enzymes are called sed, and cut microbial genomes on specific patterns. One such enzyme has magically found its way to your computer. Download the M07149.fasta from Canvas; we've got some cutting to do!
   1. The restriction enzyme works on the pattern **GAATTC** and cuts right after the G like this:



Cut the genome into pieces using this restriction enzyme (sed)! Store the fragmented genome in a new file. How many pieces did you get? (Don’t count this manually – use a command like wc).

ANS: **sed ‘s/GAATTC/G\nAATTC/g’ M07149.fasta > cut1.txt**

**wc -l cut1.txt**

**Number of pieces- 303. The above command returns 304 but the first line is not the part of the sequence.Text

Description automatically generated**

* 1. Upon further investigation, you found that the restriction enzyme is a little flexible. It can actually cut after the first base in the following patterns:

GAATTC, GAATTG,

GATTTC, GATTTG,

CAATTC, CAATTG,

CATTTC, CATTTG

Update your pattern to cut the genome accordingly. How many pieces did you get this time?

**ANS: sed 's/[G,C]A[A,T]TT[C,G]/[G,C]\nA[A,T]TT[C,G]/g' M07149.fasta > cut2.txt**

**Number of lines 6780**

* 1. You underestimated the strength of this enzyme – it can also vary its length. The updated list of patterns has the following letters being optional: third (A or T), fourth (T) and last (C or G). Update the pattern to get the new number of pieces. How many did you get this time?

ANS: **sed 's/[G,C]A[A,T]\*T\*T[C,G]\*/[G,C]\nA[A,T]\*T\*T[C,G]\*/g' M07149.fasta > cut3.txt**

**Number of lines- 82767**

#### Harder installation problem

1. Continuing our installation discussion from last week, this week we will install MySQL without using root. MySQL is a relational database management system. If that doesn’t mean anything to you right now that’s okay, but databases are extremely useful in bioinformatics. I recommend relational databases (taught in CS 4400) for everybody. MySQL is also a good example for typical compilation/installation.
2. Download the latest source code for MySQL (<http://dev.mysql.com/downloads/mysql/>), not the precompiled binaries.

ANS: Graphical user interface, text

Description automatically generated

1. Next step requires cmake. What is cmake?

ANS: **cmake is an open source software which is used to package and install softwares using a compiler-independent method. It generates another system’s build files.**

1. Unpack the source and run cmake . in the directory you just created. If you don’t have cmake in your system, get it using apt-get. Don’t attempt cmake install without root, it’s a harder install.

ANS: **tar -xzf mysql-8.0.26.tar.gz**

**sudo apt-get install cmake**

**sudo apt-get install cmake libblkid-dev e2fslibs libboost-all-dev libaudit-dev**

**sudo apt-get install cmake libcurses\***

**sudo apt-get install build-essential**

**sudo apt-get install libssl-dev**

**The tar line unpacks it into a folder named mysql-8.0.26. Now navigate to that folder using cd**

Text

Description automatically generated

**sudo cmake -DDOWNLOAD\_BOOST=1 -DWITH\_BOOST=/home/ -DFORCE\_INSOURCE\_BUILD=1**

**Text

Description automatically generated**

1. Build the MySQL executables with make

**ANS: make**

1. Try to install them with make install

ANS: make install

1. That should have failed. Why?

**ANS: It didn’t work because it said permission denied. We are not giving make sudo privileges.**

1. How would you get around this with sudo? How would you get around this with cmake? (Hint: you have to tell cmake where YOUR bin directory is. Run cmake --help )

**ANS: sudo make**

**sudo make install**

**To get around this with cmake, use**

**cmake -D CMAKE\_INSTALL\_PREFIX:PATH=/home/ashlesha/**

**This installs it in the path specified after the prefix rather than the /usr/bin/. You can then run the program by specifying the path or you can add the path to your environment variable to run it by using the name.**

**export PATH=/home/ashlesha/:$PATH**

**Text

Description automatically generated**

**Since there was no space left on my device, I repeated the steps on another system.**