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]

    2. Exactly 5 characters

    ANS: .{5}

    3. Any letter followed by a number

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

    4. 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

    5. The first 3 bases in a DNA sequence

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

    6. The last 3 bases in a DNA sequence

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

    7. Two numbers followed by 2 lower case letters

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

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

    ANS: It matches any number with three decimal points

Regular expression command exercises

2. 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

    2. Use **grep** to get all genes on chr22

    ANS: grep "chr22" knownGene.txt (displays all the information about the genes)

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

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

3. 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;}'

2. Change the chr position of every other line to cow

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

3. Delete the lines that have cow in them

ANS: sed ‘/cow/d’ knownGene.txt

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

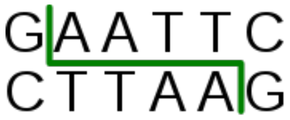
ANS: grep "chr22" knownGene.txt | sed -n '{p;p;}' > result.txt

 sed -i ‘s/chr/cow/g;n’ result.txt

 sed -i ‘/cow/d’ result.txt

Biologically-inspired problem

4. 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:

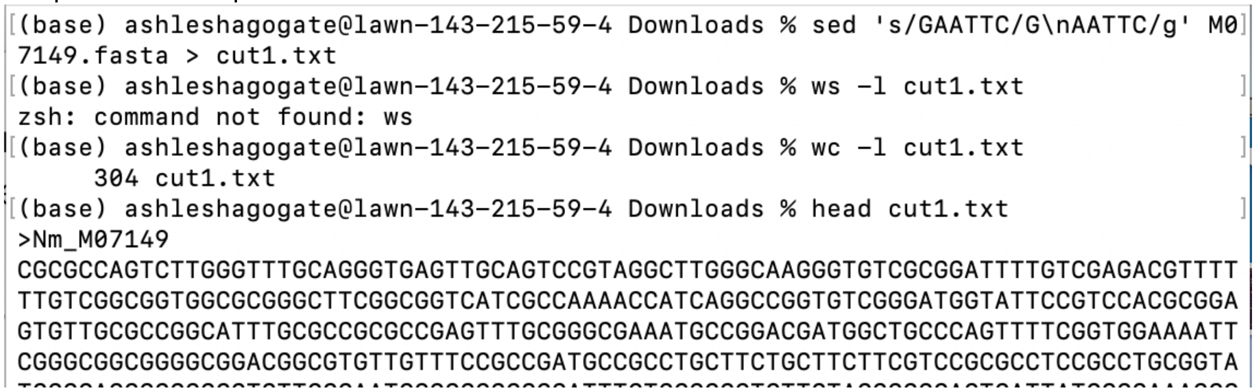


Cut the genome into pieces using this restriction enzyme (sed)! Store the fragmented genome in a new file. How many pieces did you get? (Don’t count this manually – use a command like wc).

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.



```
(base) ashleshagogate@lawn-143-215-59-4 Downloads % sed 's/GAATTC/G\nAATTC/g' M0
7149.fasta > cut1.txt
(base) ashleshagogate@lawn-143-215-59-4 Downloads % ws -l cut1.txt
zsh: command not found: ws
(base) ashleshagogate@lawn-143-215-59-4 Downloads % wc -l cut1.txt
     304 cut1.txt
(base) ashleshagogate@lawn-143-215-59-4 Downloads % head cut1.txt
>Nm_M07149
CGCGCCAGTCTTGGGTTTGCAGGGTGAGTTGCAGTCCGTAGGCTTGGGCAAGGGTGTCGCGGATTTTGTCGAGACGTTTT
TTGTCGGCGGTGGCGCGGGCTTCGGCGGTCATCGCCAAAACCATCAGGCCGGTGTCGGGATGGTATTCCGTCCACGCGGA
GTGTTGCGCCGGCATTTGCGCCCGCGCCGAGTTTGCGGGCGAAATGCCGGACGATGGCTGCCCAGTTTTCGGTGGAAAATT
CGGGCGGCGGGGCGGACGGCGTGTTGTTTCCGCCGATGCCGCCTGCTTCTGCTTCTTCGTCCGCGCCTCCGCCTGCGGTA
```

2. 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

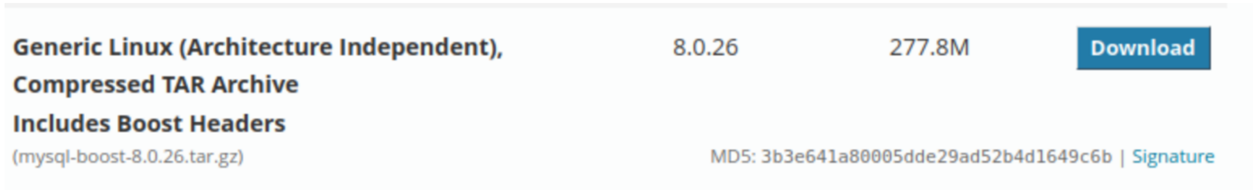
3. You underestimated the strength of this enzyme – it can also vary its length. The updated list of patterns has the following letters being optional: third (A or T), fourth (T) and last (C or G). Update the pattern to get the new number of pieces. How many did you get this time?

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

5. 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.
   1. Download the latest source code for MySQL (http://dev.mysql.com/downloads/mysql/), not the precompiled binaries.

   ANS:



Generic Linux (Architecture Independent), Compressed TAR Archive Includes Boost Headers 8.0.26 277.8M Download
(mysql-boost-8.0.26.tar.gz) MD5: 3b3e641a80005dde29ad52b4d1649c6b | Signature

   2. 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.

   3. 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`

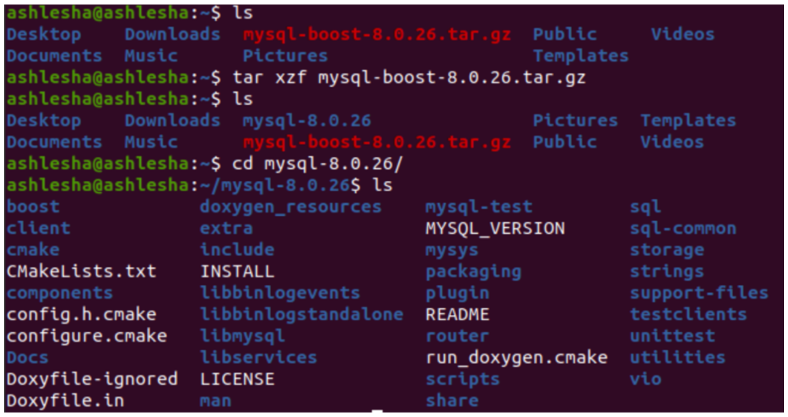
sudo apt-get install cmake libblkid-dev e2fslibs libboost-all-dev libaudit-dev

sudo apt-get install cmake libcurses*

sudo apt-get install build-essential

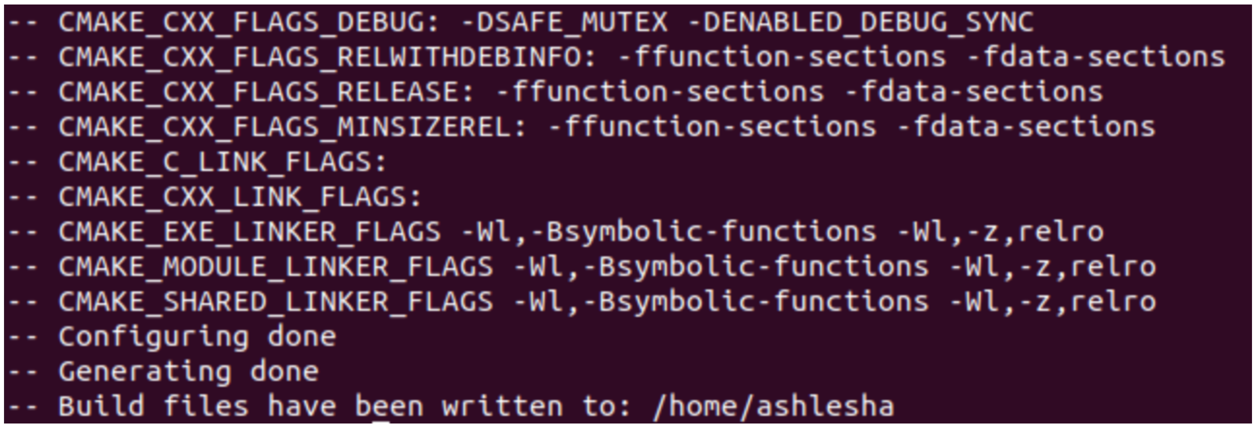
sudo apt-get install libssl-dev

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



```
ashlesha@ashlesha:~$ ls
Desktop    Downloads  mysql-boost-8.0.26.tar.gz  Public     Videos
Documents  Music      Pictures                   Templates
ashlesha@ashlesha:~$ tar xzf mysql-boost-8.0.26.tar.gz
ashlesha@ashlesha:~$ ls
Desktop    Downloads  mysql-8.0.26               Pictures  Templates
Documents  Music      mysql-boost-8.0.26.tar.gz  Public    Videos
ashlesha@ashlesha:~$ cd mysql-8.0.26/
ashlesha@ashlesha:~/mysql-8.0.26$ ls
boost             doxygen_resources     mysql-test         sql
client            extra                 MYSQL_VERSION      sql-common
cmake             include               mysys              storage
CMakeLists.txt    INSTALL               packaging          strings
components        libbinlogevents       plugin             support-files
config.h.cmake    libbinlogstandalone   README             testclients
configure.cmake   libmysql              router             unittest
Docs              libservices           run_doxygen.cmake  utilities
Doxyfile-ignored  LICENSE               scripts            vio
Doxyfile.in       man                   share
```

sudo cmake -DDOWNLOAD_BOOST=1 -DWITH_BOOST=/home/ -DFORCE_INSOURCE_BUILD=1



```
-- CMAKE_CXX_FLAGS_DEBUG: -DSAFE_MUTEX -DENABLED_DEBUG_SYNC
-- CMAKE_CXX_FLAGS_RELWITHDEBINFO: -ffunction-sections -fdata-sections
-- CMAKE_CXX_FLAGS_RELEASE: -ffunction-sections -fdata-sections
-- CMAKE_CXX_FLAGS_MINSIZEREL: -ffunction-sections -fdata-sections
-- CMAKE_C_LINK_FLAGS:
-- CMAKE_CXX_LINK_FLAGS:
-- CMAKE_EXE_LINKER_FLAGS -Wl,-Bsymbolic-functions -Wl,-z,relro
-- CMAKE_MODULE_LINKER_FLAGS -Wl,-Bsymbolic-functions -Wl,-z,relro
-- CMAKE_SHARED_LINKER_FLAGS -Wl,-Bsymbolic-functions -Wl,-z,relro
-- Configuring done
-- Generating done
-- Build files have been written to: /home/ashlesha
```

4. Build the MySQL executables with make

ANS: make

5. Try to install them with make install

ANS: make install

6. That should have failed. Why?

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

7. 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

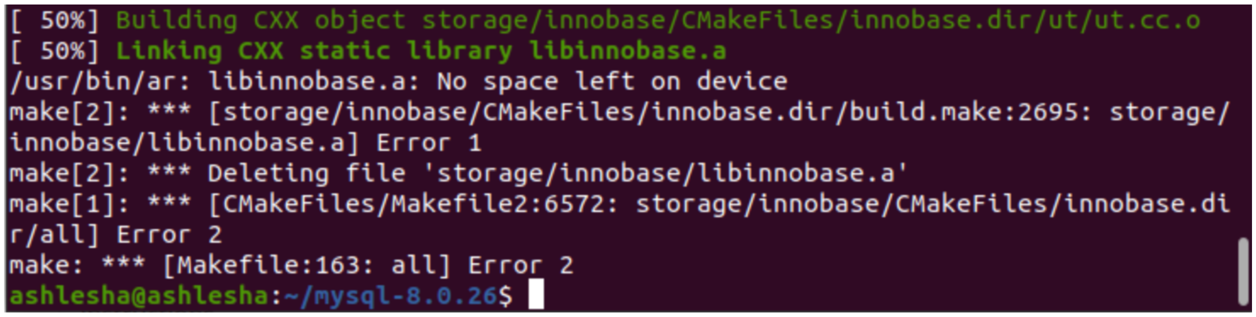
sudo make install

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



```
[ 50%] Building CXX object storage/innobase/CMakeFiles/innobase.dir/ut/ut.cc.o
[ 50%] Linking CXX static library libinnobase.a
/usr/bin/ar: libinnobase.a: No space left on device
make[2]: *** [storage/innobase/CMakeFiles/innobase.dir/build.make:2695: storage/
innobase/libinnobase.a] Error 1
make[2]: *** Deleting file 'storage/innobase/libinnobase.a'
make[1]: *** [CMakeFiles/Makefile2:6572: storage/innobase/CMakeFiles/innobase.di
r/all] Error 2
make: *** [Makefile:163: all] Error 2
ashlesha@ashlesha:~/mysql-8.0.26$
```

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