#### 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

grep -E ‘^\d\*[5,0]$’ filename

* 1. Exactly 5 characters

grep -E ‘\S{5}’ filename

* 1. Any letter followed by a number

grep -E ‘[A-Za-z][0-9]’ filename

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

grep -E f1-3 filename

* 1. The first 3 bases in a DNA sequence

grep ‘^[ACTG]\*\*’ filename

* 1. The last 3 bases in a DNA sequence

grep ‘[ACTG]\*\*&’ filename

* 1. Two numbers followed by 2 lower case letters

grep -E ‘[0-9]{2}[a-z]{2}’ filename

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

d refers to a number, and anything start with a digit and end with . three digits will match.

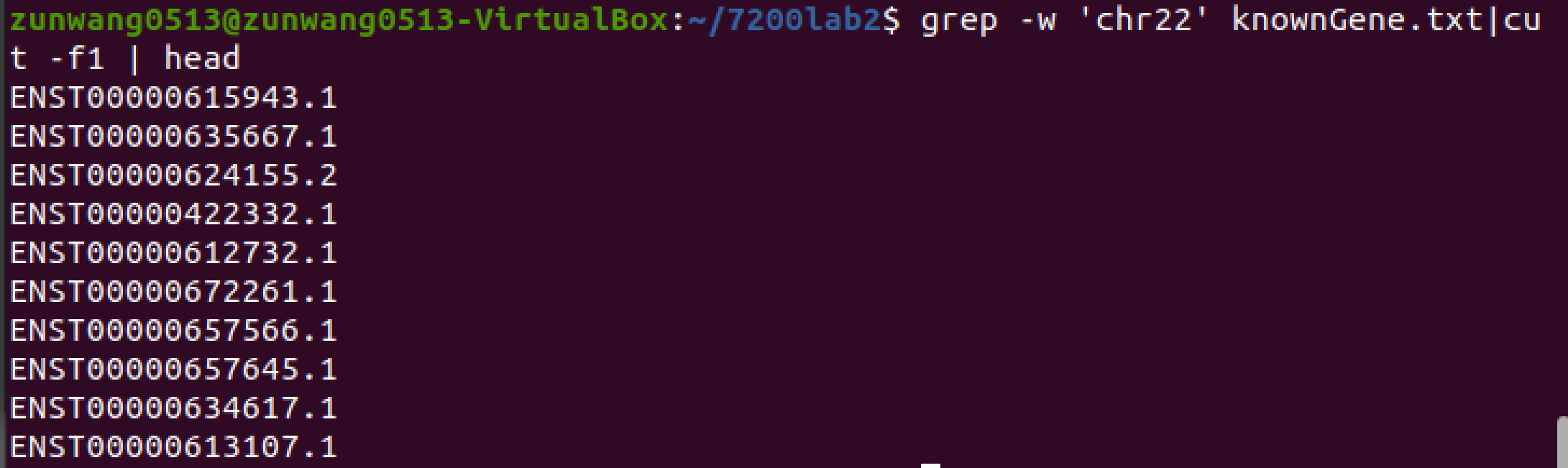
#### Regular expression command exercises

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.

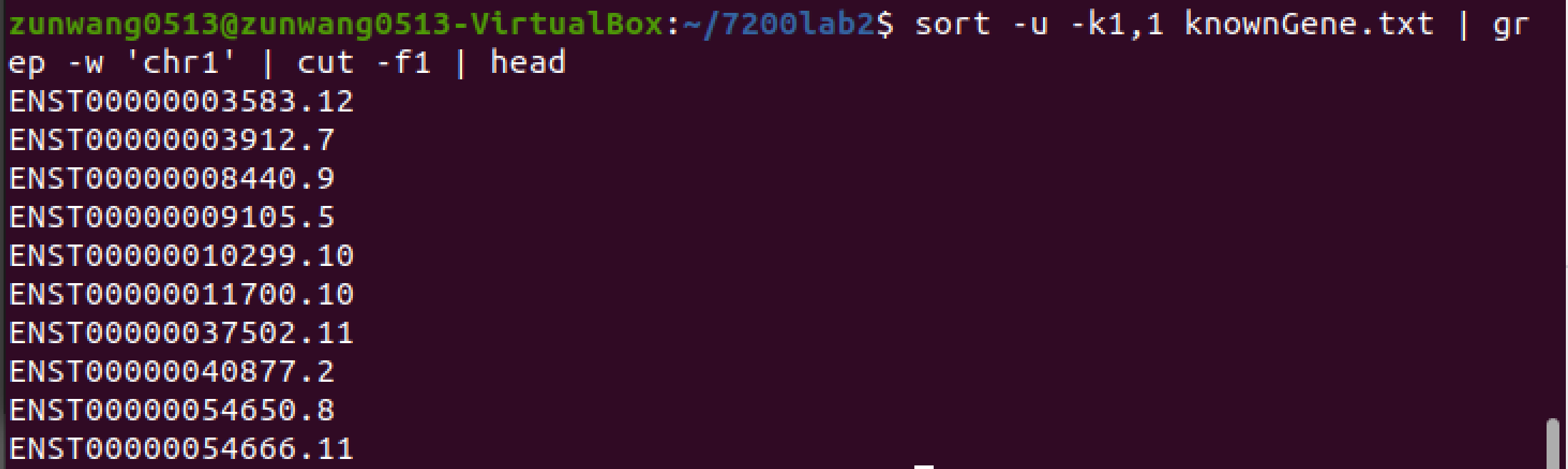


* 1. Use grep to get all genes on chr22

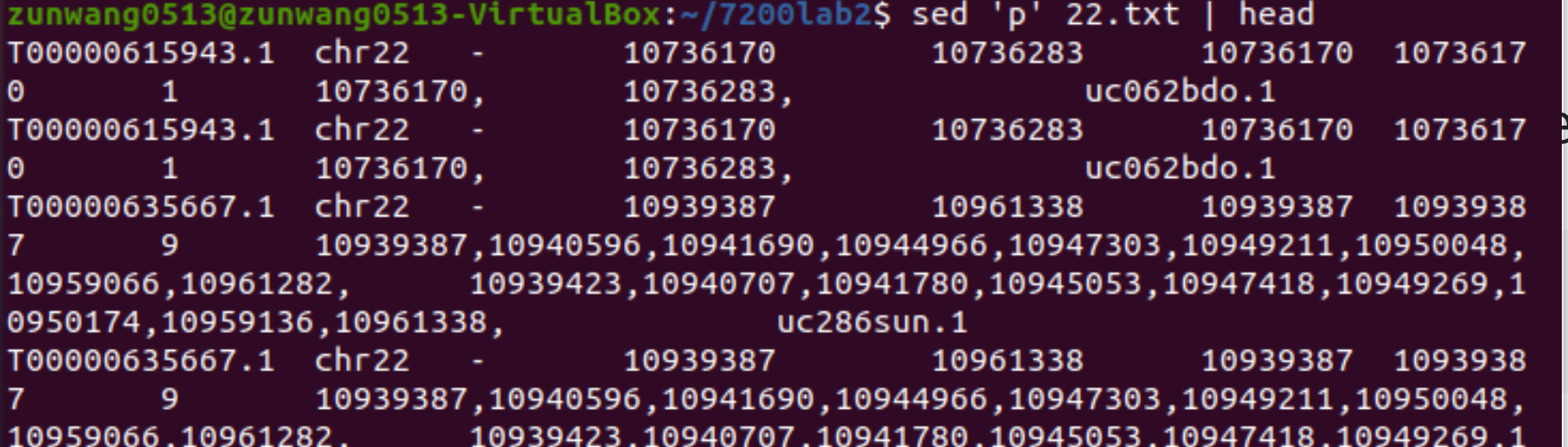
cat knownGene.txt | grep ‘chr22’ knownGene.txt



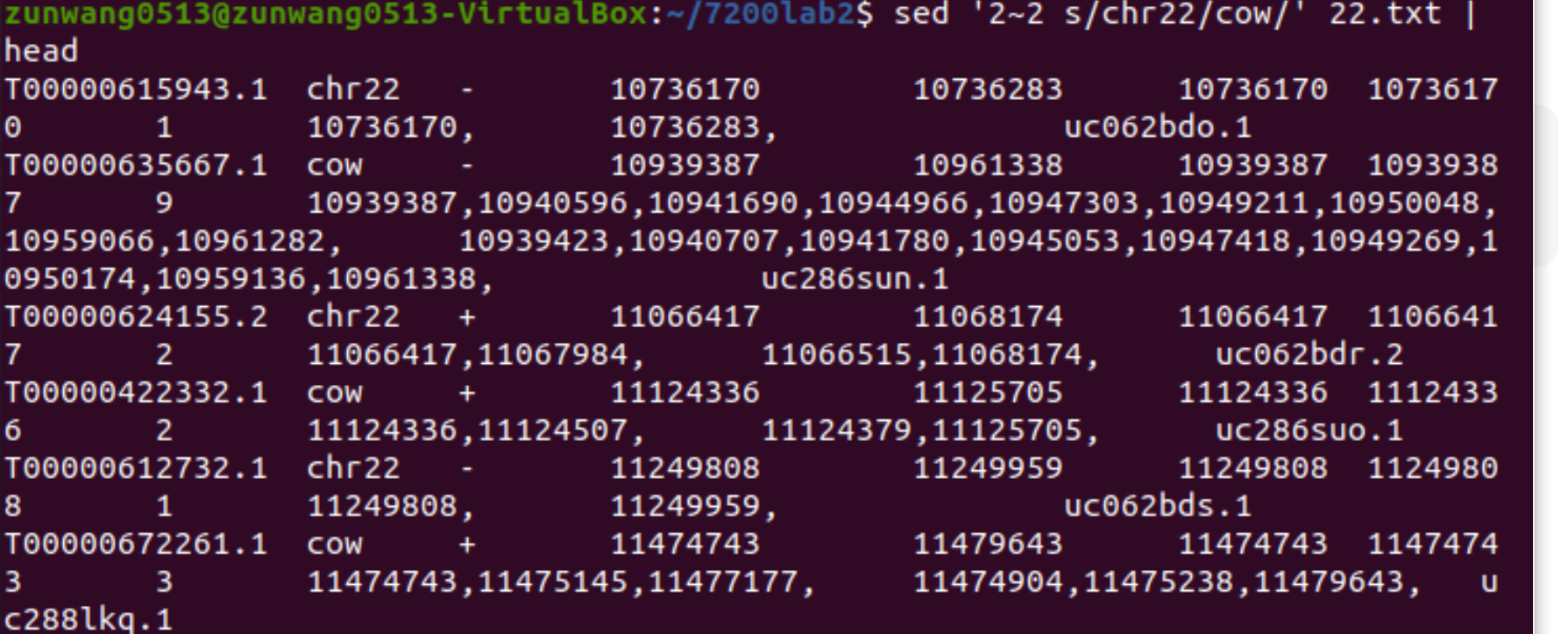
* 1. Use grep to get all and only those genes that occur on chr1



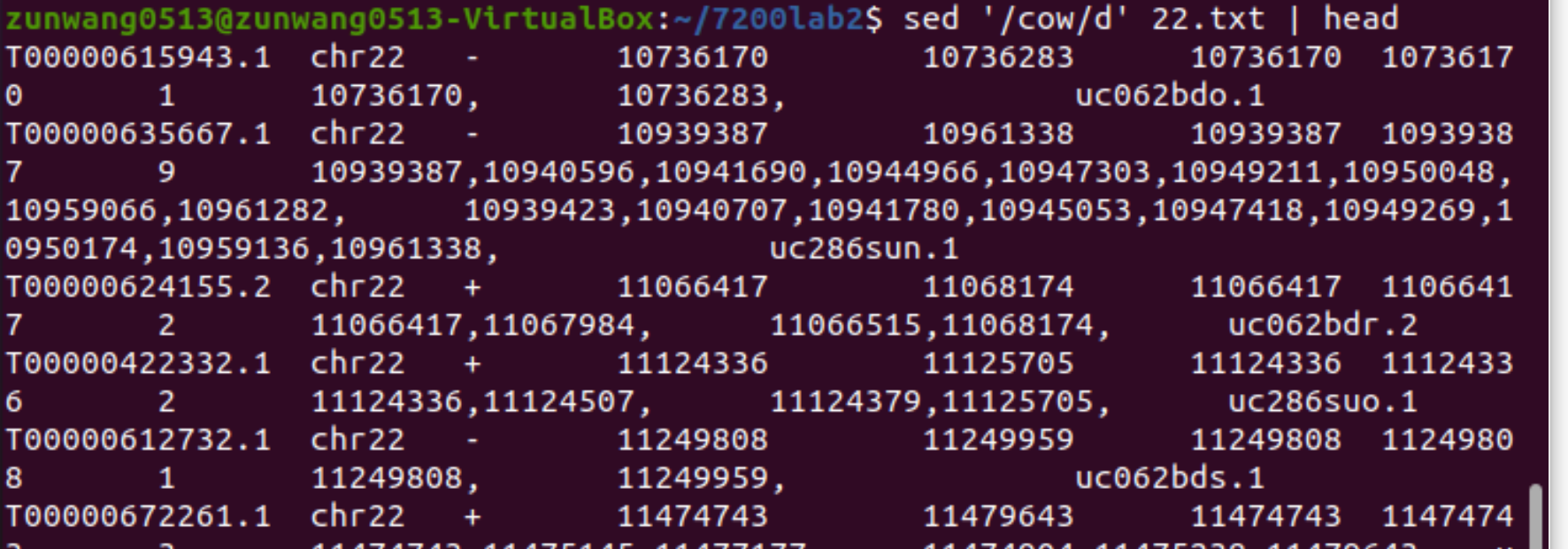
1. Editing data streams with sed
   1. Take the results from **2.2** and duplicate each line



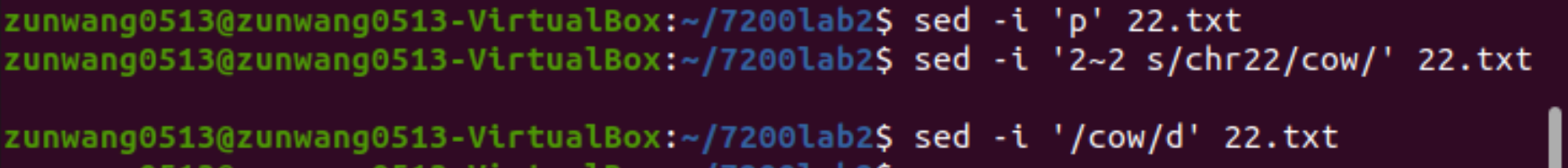
* 1. Change the chr position of every other line to cow



* 1. Delete the lines that have cow in them



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

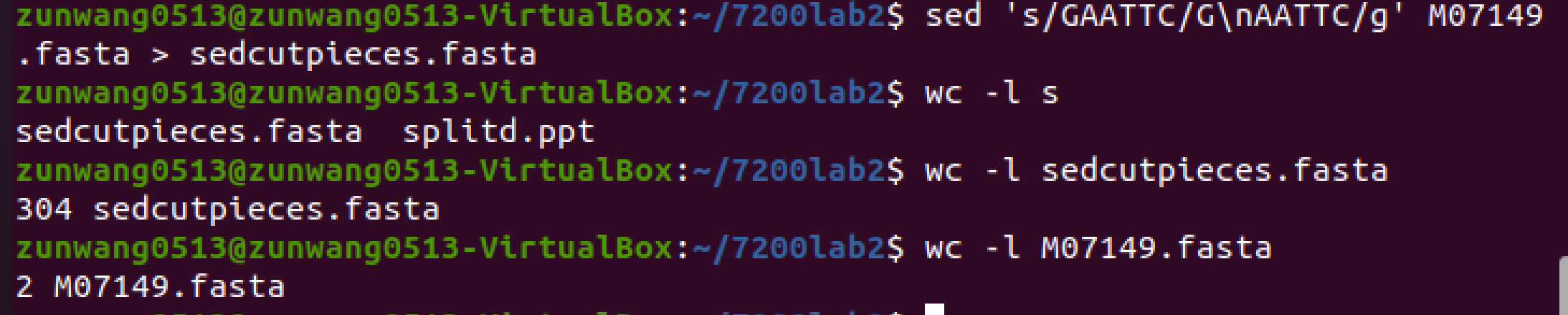


It will not printout the file and store the original file in the backup.

#### 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).

Since the original fasta contains a title, 304 lines means there are 303 pieces.

* 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:

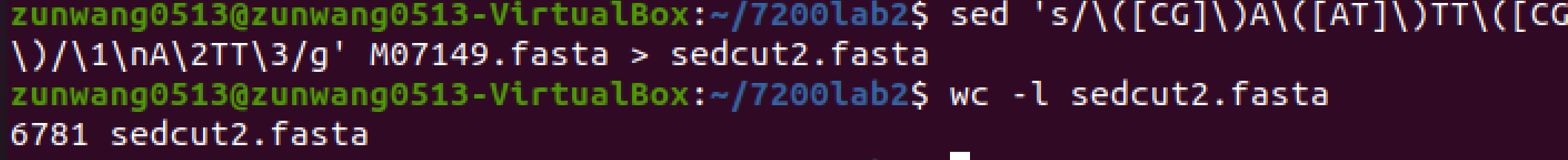
GAATTC, GAATTG,

GATTTC, GATTTG,

CAATTC, CAATTG,

CATTTC, CATTTG

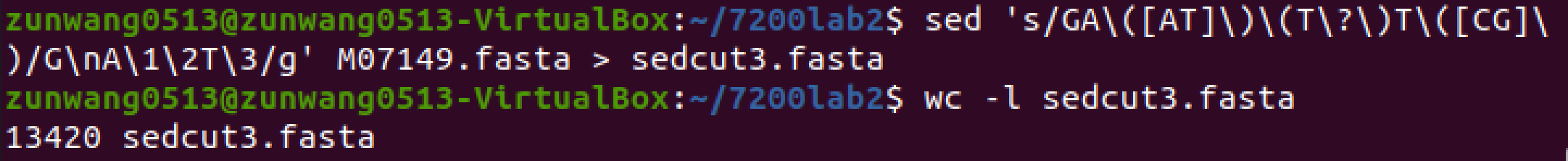
Update your pattern to cut the genome accordingly. How many pieces did you get this time?

 Exclude the title, there are 6780 pieces.

* 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?

Exclude the title line, there are 13419 pieces.

The question does not specifies if the first base is optional as in 4.2 so I assume it can only be G. Else the answer would be 25276.

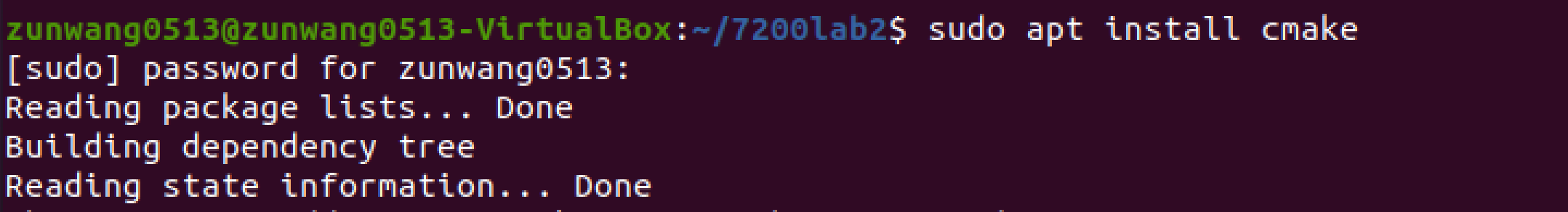


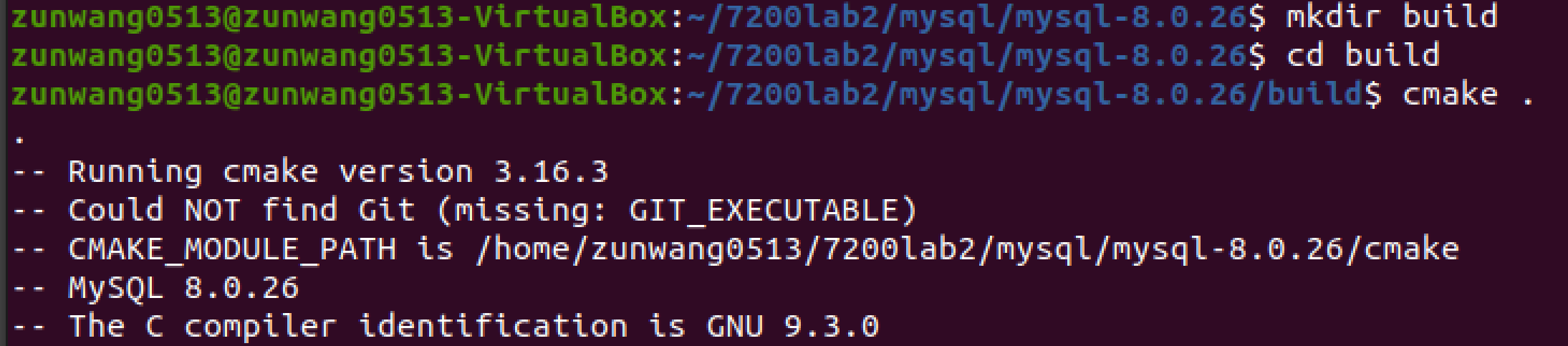
#### Harder installation problem

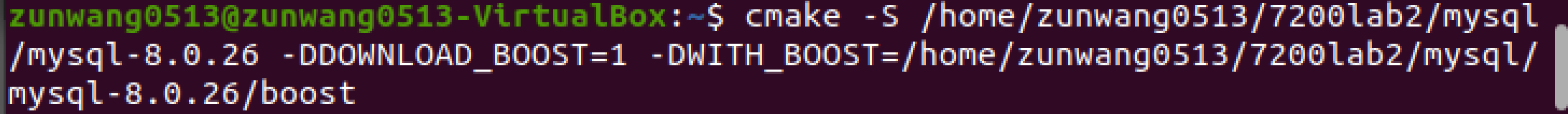
1. Continuing our installation discussion from last week, this week we will install MySQL without using root. MySQL is a relational database management system. If that doesn’t mean anything to you right now that’s okay, but databases are extremely useful in bioinformatics. I recommend relational databases (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.
3. Next step requires cmake. What is cmake?

Cmake in linux generate build tool files and install software through compilers.

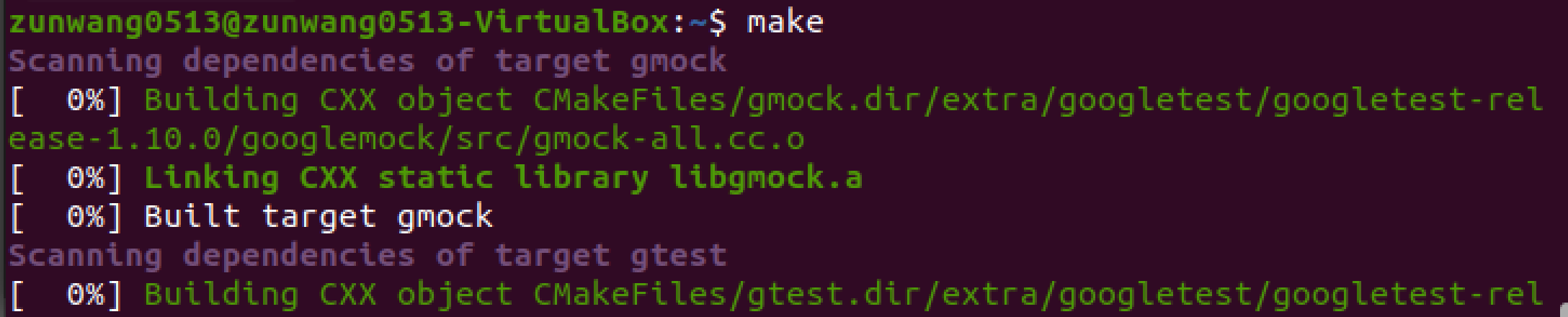
1. Unpack the source and run cmake . in the directory you just created. If you don’t have cmake in your system, get it using apt-get. Don’t attempt cmake install without root, it’s a harder install.



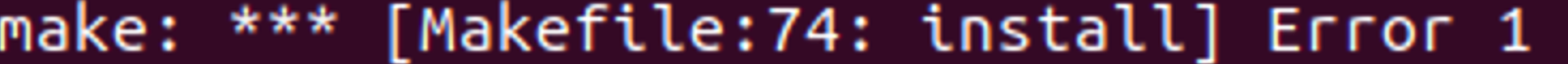




1. Build the MySQL executables with make



1. Try to install them with make install



It comes with error 1, file cannot create directory

1. That should have failed. Why?

Mysql requires permission from root and will be installed in directories other than home directory.

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 )

Sudo make install

set(CMAKE\_RUNTIME\_OUTPUT\_DIRECTORY "/usr/local')