Valentina Alaasam

Data Science II

Jan 25 2022

Homework 1

What is a Graphical User Interface (GUI) and how does that differ from the Command Line Interface (CLI)?

GUI is more user friendly and what we are used to interacting with. It gives you options and you can pick one. CLI is more like speaking a language, where you have to figure out how to put words/commands together to perform the tasks you want, but in this sense it is also much more flexible and creative.

What does the shell do?

Runs programs

What is the command used for listing things in a directory?

ls

What is the command used for changing directories?

cd

What command would you use to get your current working directory?

pwd

How do you get the manual for these commands?

man

What does the shell prompt look like?

$

How would you list things in a directory in chronological order?

ls -t

Name two ways to get to the ‘home’ or ‘root’ directory.

cd /Users/val

cd

or use relative path

What is the difference between an absolute path and a relative path?

Absolute is starting from the home root directory, is always the same to get to a given directory. Relative directs you forward or backwards depending on where you are currently.

What are the two relative path directories we talked about and what do they mean?

. = current directory

.. = upwards/backwards one directory

Why shouldn’t you put spaces in filenames?

Spaces between words are typically read as two different commands, so if there is a space in the file name it will only read the first word and think that is the whole name, moving onto the second word as if it were another command.

Name a way to have multiple readable words in a filename without spaces?

Using an underscore or a period between words

What is the program nano, what does it do? Do you use nano? If not what do you use?

Nano is a text editor. So you can use it to write scripts (kind of like writing a function in R) and then save them and execute them later. You can write any kind of script, python, bash, shell etc. I have tried to use Spyder (looks just like R studio), but then in the end I think it would be easier to just use R studio rather than having a whole separate parallel universe for python and command line scripts. Interested in trying to figure out Jupyter (is this even a text editor or am I confused…).

\* is a form of a wildcard. What does it mean?

Means it could represent any character or string of characters.

The following code would match what? ls \*.txt

It would list any file that ends in .txt

Why do we need to be careful with the mv command?

Moving files also deletes them from where they were and can re-write them/change the name in the process. So it’s important to pay attention to how we name things and where we are moving them to/from.

Why do we need to be careful with the rm command?

It immediately deletes things forever

What is the difference between > and >>?

“>” will overwrite a file if it exists while “>>” appends a file (edits it). Both will create a new file if it doesn’t exist

What does head do?

List the beginning of a file, first few lines

What is the purpose of | (pipe)?

Strings together a sequence of commands, so that you can do a lot of different things in sequence at the same time without printing an output at each step

Interpret the following command:

cat huge\_file.fasta | uniq | head -n 5 >proteins.fasta

read the file huge\_file.fasta, pull out just unique lines (no repeats), list the first 5 of the lines, and put these 5 into a file called proteins.fasta

What is a loop and when would you use it?

A command that cycles back on itself repeatedly until a given stop point. You would use it to do the same thing to a lot of lines or a lot of files.

In a directory with the following files. What would the following loop do?

$ for filename in \*.txt; do

> cat ${filename} >>fruits.txt

> done

If a file in the directory ends in .txt, read the file, and put the contents of the file in a new file called fruits.txt. Repeatedly go back and do this again for any file in the directory that ends in .txt.

What is a shell script?

A chunk of code that can be saved and run by shell whenever needed.

What are the benefits of writing and running a script over typing the code in?

Saves time and enforces consistency (prevents errors) if we are running the same code multiple times.

When looking at a script what does # mean? Why would you use one?

Comments out what follows, meaning it’s not read as code it is only there for us to read. We use # to annotate our code and describe what each line of code is meant to do so that other people (or our future selves) can understand it.

What does grep stand for and what does it do?

Searches for a string of characters

What does find do?

Locates files – will look within lower hierarchies of the directory

1. cd desktop

ls

1. usr/val

Diagram

Description automatically generated

#to get to homework directory

cd Desktop/Class/homework

#to get to Class

cd ..

#to get to ToRead

cd ../ToRead

1. mkdir testdirectory

cd testdirectory

print “I am amazing” > one.txt

print “So are you” > two.txt

echo \*.txt

1. cd

cd Desktop

mkdir hw1

#download Hutia file

#move to hw1 directory

cd ../Downloads

mv Hutia\_DNA.fasta ../Desktop/hw1

cd ../Desktop/hw1

ls

wc -l Hutia\_DNA.fasta  
# 2382004 lines in file

*# How many sequences are in the file?*

*# grep searches for a particular sequence (>) and ^ specifies searching the beginning of lines. wc -l then counts the number of lines grep pulled.*

grep ‘^>’ Hutia\_DNA.fasta | wc -l

# 1191002 sequences

1. # grep finds all lines that do NOT begin with “>” in my Hutia file. Then we pipe to head to count the first 100 lines, then copy these into MyOutputFile.fasta

grep '^[^>]' Hutia\_DNA.fasta | head -n 100 > MyOutputFile.fasta

1. ### writing shell script ###

### testscript.sh ###

#!/usr/bin/sh

for file in \*.txt #perform following action on all files ending in .txt

do cp $file subset\_directory #copy file to a directory called “subset\_directory”

done

sh testscript.sh

1. ### writing python script ###

### find\_taxa.py ###

### run using python3 ###

#!/usr/bin/sh

for file in \*.fasta txt #perform following action on all files ending in .fasta

do somethingcrazy #does something crazy

done

1. grep -c "GAGA" Hutia\_DNA.fasta

304806