University of Pennsylvania
BIOL4536 Fall 2023

HW#4 (SOLUTIONS)

(UNIX)

You should have received an email with information about your UNIX account, that you access it through a web browser.

In this HW you will be asked to construct some unix commands to perform certain tasks. There usually is no one right way to do things. But points can be deducted for doing them in an unreasonably long or inefficient way.

In your home directory, create a subdirectory called "hw4".

Move into that directory. Execute `pwd` to make sure.

Copy the file `biol4536_course_files.tar.gz` from the `/data` directory to the `hw4` dir. This is a `.tar` archive that has been gzipped. The class slides explain how to unzip and unpack such a file. Do that now.

ANSWER: The commands to unzip and unpack are:

```
> gunzip biol4536_course_files.tar.gz
> tar -xvf biol4536_course_files.tar
```

Do an `ls -ltr` to make sure the files are there.

The file `reads.fa` has 5,000 sequence reads in fastA format.

(1) (1/2 pt) Use `grep` to count the number of sequences that have five A's and five T's in them. Show your grep command and the answer.

ANSWER: The following grep command will work if there were no read IDs with five A's and five T's (as there are in this file):

```
> grep AAAAA reads.fa | grep TTTTT | wc -l
```

another possibility is:

```
> grep AAAAA reads.fa | grep -c TTTTT
```

and the answer is: 83

But this would not work if there were read IDs with five A's and five T's in them because those IDs would be included in the count. In that case we'd need to do the following to first get rid of the ID lines:

```
> grep -v ``>'' reads.fa | grep AAAAA | grep -c TTTTT
```

(2) (1/2 pt) Use `grep` to count the number of sequences that have five A's and five T's in them, but the A's occur before the T's. Show your grep command and the answer.

ANSWER:

```
> grep AAAAA.*TTTTT reads.fa | wc -l
```

or

```
> grep -c AAAAA.*TTTTT reads.fa
```

and the answer is: 43

(3) (1/2 pt) Use `grep` to count the number of sequences that have five A's and five T's in them, but the T's occur before the A's. Show your grep command and the answer.

ANSWER:

```
> grep TTTTT.*AAAAA reads.fa | wc -l
```

or

```
> grep -c TTTTT.*AAAAA reads.fa
```

and the answer is: 49

(4) (1 pt) The answer for #2 is the number of sequences with five A's upstream of five T's, and the answer to #3 is the number of sequences with five A's downstream of five T's. Meanwhile the answer to #4 is the number of sequences with five A's upstream *or downstream* of five T's. So why isn't the answer to #4 equal to the answer to #2 plus the answer to #3?

ANSWER: It's because four sequences actually have five A's upstream of five T's *and* at the same time have five A's downstream of five T's. You can count them by doing:

```
> grep AAAAA.*TTTTT.*AAAAA reads.fa | wc -l
```

Likewise, there are six sequences with five T's upstream and downstream of five A's, you can count them with:

```
> grep -c AAAAA.*TTTTT.*AAAAA reads.fa
```

There's even one on both of these lists, you can find it by:

```
> grep AAAAA.*TTTTT.*AAAAA.*TTTTT reads.fa
```

Advance to next page

Do the following to prepare for the next problem. We will make a simple “shell script” (as you’ve seen in the UNIX slides last topic).

Create a sub-directory of `hw4` called `temp`. Create a file called `makefiles.sh` with the following code in it:

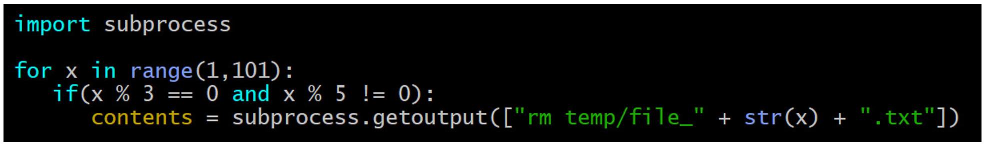
```
for i in {1..100}
do
    echo "This is the contents of file $i" > temp/file\detokenize{_}$i.txt
done
```

Now execute the command ’`source makefiles.sh`’ from the `hw4` directory.
The directory `temp` should now have 100 files in it. Do an `ls -l temp` to make sure.

(5) (2 pts) Write a python script to execute the unix commands to delete all files `file_N.txt` in the temp directory where *N* is divisible by 3 but not by 5.

Copy the python code to your write-up, or include as an attachment. It should not be too long.

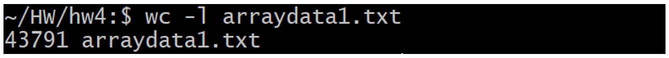
ANSWER: The following will do it. Execute this from the `hw4` directory.



```
import subprocess

for x in range(1,101):
   if(x % 3 == 0 and x % 5 != 0):
      contents = subprocess.getoutput(["rm temp/file_" + str(x) + ".txt"])
```

(6) (1/2 pt) Use a Unix command to count how many rows the file `arraydata1.txt` has. Give the answer and the command.

ANSWER:



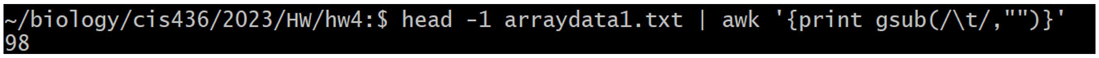
```
~/HW/hw4:$ wc -l arraydata1.txt
43791 arraydata1.txt
```

(7) (1/2 pt) You can use the following Unix command to count the number of columns in a tab-delimited file.

```
> head -1 <filename> | awk 'print gsub(//̑,"")'
```

This counts the number of tabs in the first row of the file. So add one to get the number of columns. Run this on `arraydata1.txt` to find out how many columns there are. Give your answer and the command.

ANSWER:

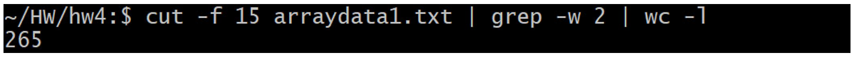


```
~/biology/cis436/2023/HW/hw4:$ head -1 arraydata1.txt | awk '{print gsub(/\t/,"")}'
98
```

Therefore, there are 99 columns in the file.

(8) (1/2 pt) Use a Unix command to count how many rows of `arraydata1.txt` have their 15th column equal to 2. Give the answer and the command.

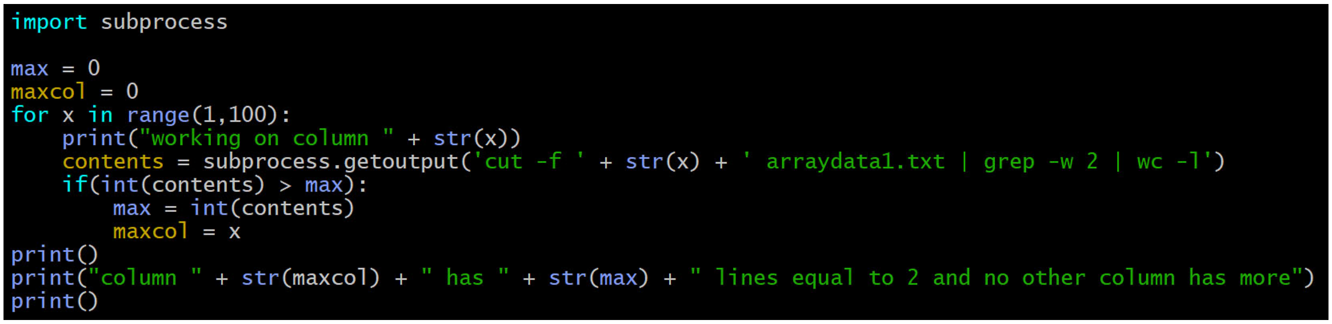
ANSWER:



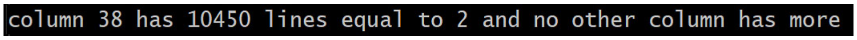
```
~/HW/hw4:$ cut -f 15 arraydata1.txt | grep -w 2 | wc -l
265
```

(9) (2 pts) Write a python program to run the unix command from the previous exercise for each column, and report the column with the greatest number of 2's. Copy your code into your write-up, or include as an attachment. *Note: we know how many columns there are from problem 7, you can hard code this into your script.*

ANSWER:



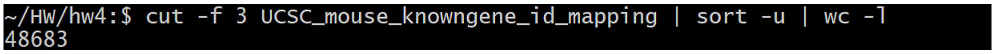
```
import subprocess

max = 0
maxcol = 0
for x in range(1,100):
    print("working on column " + str(x))
    contents = subprocess.getoutput('cut -f ' + str(x) + ' arraydata1.txt | grep -w 2 | wc -l')
    if(int(contents) > max):
        max = int(contents)
        maxcol = x
print()
print("column " + str(maxcol) + " has " + str(max) + " lines equal to 2 and no other column has more")
print()
```



```
column 38 has 10450 lines equal to 2 and no other column has more
```

(10) (1 pt) Column three of the file `UCSC_mouse_knowngene_id_mapping` has the so-called "refseq" gene identifier. Use `cut` and `sort` to figure out how many uniquely different ID's there are in that column. Report the answer and your unix command. *Note that the first row is the header row.*

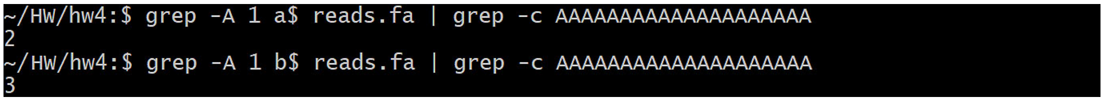
ANSWER: The following command also counts the header, so the answer is 48,682.



```
~/HW/hw4:$ cut -f 3 UCSC_mouse_knowngene_id_mapping | sort -u | wc -l
48683
```

(11) (1 pt) The file `reads.fa` has high-throughput sequencing reads of RNA. The ones with ID's that end in an "a" are the so-called "forward" reads and the ones that end in a "b" are the "reverse". We'll learn more about what that means later. But for now, all you need to know is `a`'s are foward and `b`'s are reverse. Suppose we are looking for polyadenylated genes, so we want to find polyA tails. Suppose to this end we want to find all sequences with 20 `A`'s in a row, or more. Out of the 5,000 reads in the file, how many foward reads are there with 20 `A`'s and how many reverses? Do this with `grep` commands, report the answer and the command.

ANSWER: The following commands will do it. The first part (before the pipe) `grep -A 1 a$ <filename>` grabs all rows with foward IDs and the following row with its sequence. Likewise for reverse. Two forwards and three reverse.



```
~/HW/hw4:$ grep -A 1 a$ reads.fa | grep -c AAAAAAAAAAAAAAAAAAAA
2
~/HW/hw4:$ grep -A 1 b$ reads.fa | grep -c AAAAAAAAAAAAAAAAAAAA
3
```