More Command Line

We’ve gone through many essential commands

- `ls`
- `wc`
- `pwd`
- `cd`
- `mkdir`
- `man`
- `rm`
- `touch`
- `mv`
- `echo`
- `less`
- `cat`
- `>> vs >`
- `grep`
- `sort`
- `| (pipe)`
- `nano`

| (pipes)

You cannot be a productive command line user until you really understand the power of pipes

```
grep TP53 genes.txt | grep “missense” | wc -l
```

This kind of construction allows you to get answers quickly!

I'm stuck!

- **Ctrl-C** to interrupt/kill a running process

- **q** quits some interactive commands (e.g. less)

- editing a file with vim?
  - press **Escape**
  - type **:q!**
  - press **Return**

It's not working!

- Did you check case?
  - capital vs lowercase matters!

- Are you in the right directory?
  - use `ls` all the time!

- typos
  - tab-complete is your friend!

First, let’s download some data

```
## make a directory, then move into it
mkdir ~/workspace/commandline
cd ~/workspace/commandline


## download the files from the links posted in slack
wget <URL>

## use ls to list the files
ls -l
```

Examine the files

```
genes1.txt = mutations identified in patient set 1
genes2.txt = mutations identified in patient set 1
```

Working with `sort | uniq`

```
## Sorts genes in genes1.txt
cat genes1.txt | sort

## ... is equivalent to ...
sort genes1.txt
```

Genes are sorted alphabetically, great!

CEBPA
CEBPA
DNMT3A
FLT3
IDH1
IDH2
IDH2
NPM1
NRAS
RUNX1
TET2
TP53

Working with `sort | uniq`

```
## Sorts genes in genes1.txt
cat genes1.txt | sort

## ... is equivalent to ...
sort genes1.txt

## Get the unique genes... right?
cat genes1.txt | uniq
```

DNMT3A
FLT3
NPM1
TET2
IDH2
RUNX1
TP53
IDH2
IDH1
CEBPA
NRAS
CEBPA

Why are IDH2 and CEBPA repeated?

Working with `sort | uniq`

```
## Sorts genes in genes1.txt
cat genes1.txt | sort

## ... is equivalent to ...
sort genes1.txt

## Sort then unique
cat genes1.txt | sort | uniq
```

CEBPA
DNMT3A
FLT3
IDH1
IDH2
NPM1
NRAS
RUNX1
TET2
TP53

This is the way!

(uniq only identifies matching values when they are immediately next to one another)

More `sort | uniq` combinations

```
## Report only duplicate values (uniq -d flag)
cat genes1.txt | sort | uniq -d

## Report values that are in the file a single time (uniq -u)
cat genes1.txt | sort | uniq -u

## Count the number of occurrences for each value (uniq -c)
cat genes1.txt | sort | uniq -c
```

Unsorted data:

DNMT3A
FLT3
NPM1
TET2
IDH2
RUNX1
TP53
IDH2
IDH1
CEBPA
NRAS
CEBPA

Performing set operations with `sort | uniq`

Question 1: how can we find if a gene is present in both genes1.txt and genes2.txt?

Question 2: how can we find genes only found in genes1.txt and not in genes2.txt?

Performing set operations with `sort | uniq`

Question 1: how to find if a gene is present in both genes1.txt and genes2.txt?

```
cat genes1.txt | sort | uniq –u > genes1.uniq.txt
cat genes1.txt | sort | uniq -u > genes2.uniq.txt
cat genes1.uniq.txt gene2.uniq.txt | sort | uniq -d
```

Question 2: how do we find genes only found in genes1.txt and not in genes2.txt?

Performing set operations with `sort | uniq`

Question 1: how to find if a gene is present in both genes1.txt and genes2.txt?

```
cat genes1.txt | sort | uniq –u > genes1.uniq.txt
cat genes1.txt | sort | uniq -u > genes2.uniq.txt
cat genes1.uniq.txt gene2.uniq.txt | sort | uniq -d
```

Question 2: how do we find genes only found in genes1.txt and not in genes2.txt?

```
cat genes1.uniq.txt genes1.uniq.txt gene2.uniq.txt | sort | uniq -u
```

Examine the mutation file

```
less tcga.tsv

#wrap long lines
less –S tcga.tsv

#wrap long lines and set tab spacing to 20 characters
less –S –x20 tcga.tsv
```

Examine the mutation file

```
less tcga.tsv

#wrap long lines
less –S tcga.tsv

#wrap long lines and set tab spacing to 20 characters
less –S –x20 tcga.tsv



type "q" to exit
```

Sort the mutation file

```
sort tcga.tsv | less

#sort by chromosome (second column)
sort -k 2 tcga.tsv | less

#sort by chromosome, and then position numerically
sort -k 2,2 -k 3,3n tcga.tsv | less
```

Extract info from the mutation file

```
#cut the 8th column (gene names)
cut –f 8 tcga.tsv | less

#cut multiple columns
cut -f 2-4,8,10 tcga.tsv | head

#find the most frequently mutated genes in this cohort
cut -f 8 tcga.tsv | sort | uniq -c | sort -nrk 1 | head -n 20
```

Some useful UNIX commands

- **`head`** print the first 10 lines of a file

- **`tail`** print the last 10 lines of a file
  getting fancy: **`tail -n +2`** (start at the second line of a file)

- **`wc`** count the number of characters/words/lines in a file
  `wc -l` for only lines

- **`less`** because you don't want 3 million lines scrolling through your terminal
  `q` to exit, **`-S`** to wrap lines (lots more useful options here)

- **`grep`** to search through a file (**`-v`** to search for lines *without* pattern)

Working with compressed data

- **`tar`** work with a “bundle” of data

  create: **tar -cvf output.tar infile1 infile2**
  extract: **tar -xvf output.tar**

- **`gzip`** compress a single file

  create: **gzip mydata.txt** (creates mydata.txt.gz)
  extract: **gunzip mydata.txt.gz** (creates mydata.txt)

Often these operations are combined

```
tar -czvf myfile.tar.gz <list of files>
  tar -xzvf myfile.tar.gz
```

sed and awk

sed is most commonly used for find and replace operations:

```
cat file.txt | sed 's/foo/bar/g' >file_fixed.txt
```

Awk can be used to reorder particular columns (here, third, first, then second):

```
awk '{print $3,$1,$2}' file.txt >file2.txt
```

Or to print only certain lines of a file - here, every third line, starting at line 0

```
awk 'NR % 3 == 0' file > file2.txt
```

(both are very powerful, if somewhat opaque tools, this is just scratching the surface!)

Working with FASTQs

- https://gist.github.com/chrisamiller/230cf13c1ee0ca10a5535279957f48a5