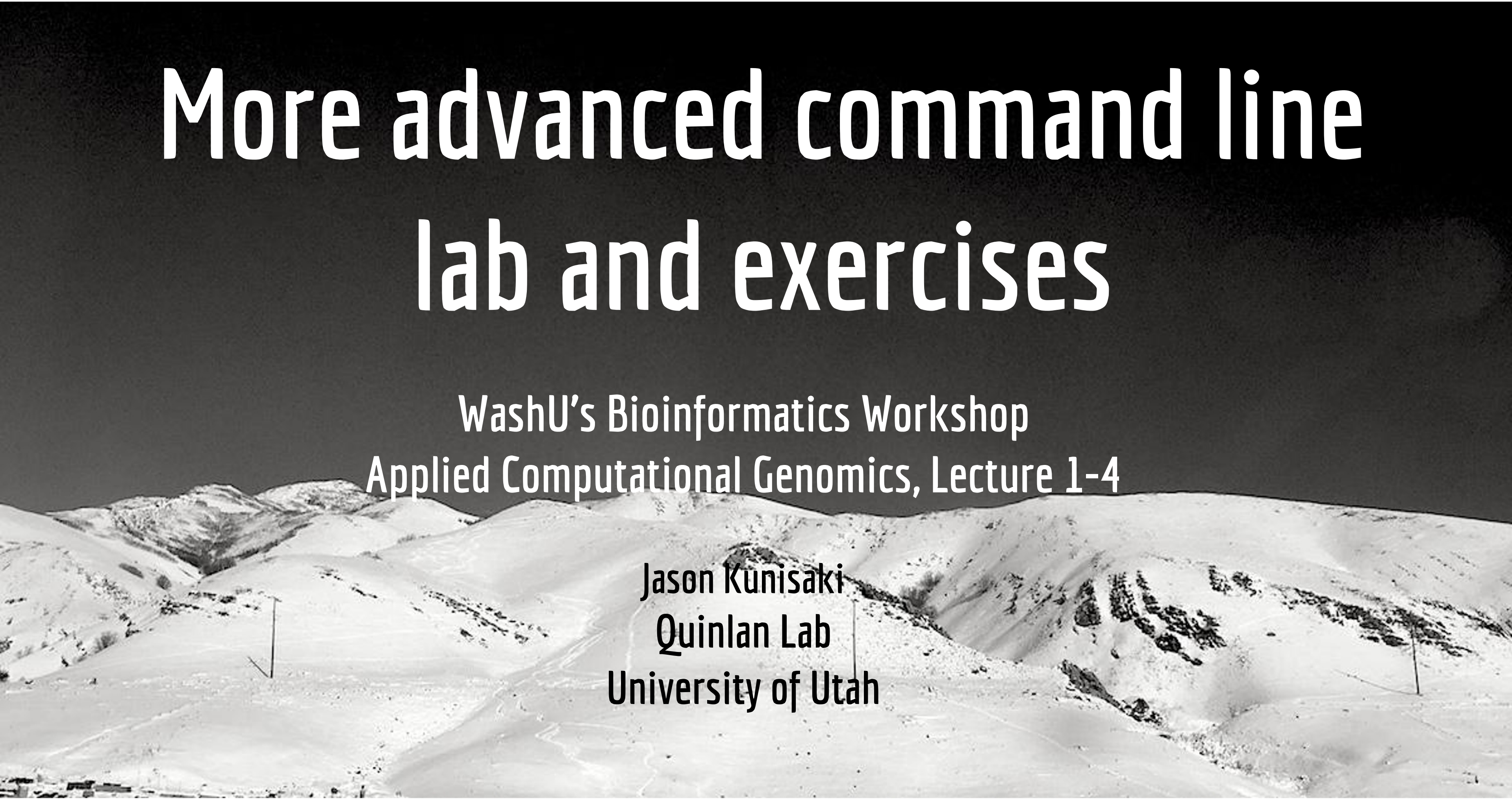
More advanced command line lab and exercises

WashU's Bioinformatics Workshop
Applied Computational Genomics, Lecture 1-4

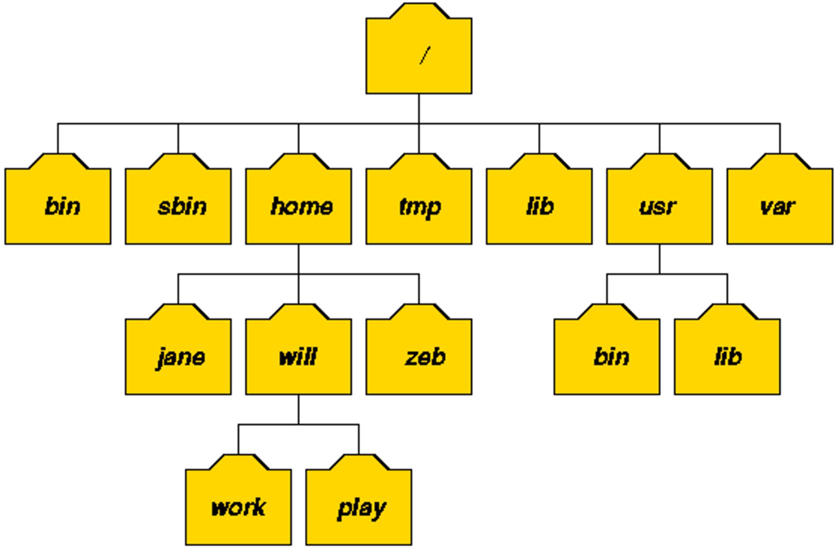
Jason Kunisaki
Quinlan Lab
University of Utah

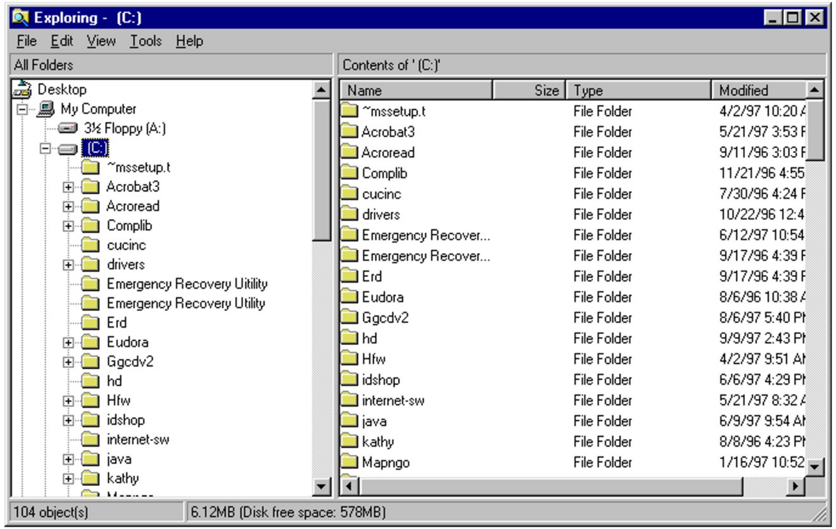


Navigating filesystems via the command line is essential

- ls
- wc
- pwd
- cd
- mkdir
- man
- rm
- touch
- mv
- echo
- less
- cat
- >> vs >
- grep
- sort
- Pipe |
- vim

Navigating filesystems via the command line is essential





Desktop
My Computer
3½ Floppy (A:)
(C:)
~mssetup.t
Acrobat3
Acroread
Complib
cucinc
drivers
Emergency Recovery Utility
Emergency Recovery Utility
Erd
Eudora
Ggcdv2
hd
Hfw
idshop
internet-sw
java
kathy

Name	Size	Type	Modified
~mssetup.t		File Folder	4/2/97 10:20 A
Acrobat3		File Folder	5/21/97 3:53 P
Acroread		File Folder	9/11/96 3:03 P
Complib		File Folder	11/21/96 4:55
cucinc		File Folder	7/30/96 4:24 P
drivers		File Folder	10/22/96 12:4
Emergency Recover...		File Folder	6/12/97 10:54
Emergency Recover...		File Folder	9/17/96 4:39 P
Erd		File Folder	9/17/96 4:39 P
Eudora		File Folder	8/6/96 10:38 A
Ggcdv2		File Folder	8/6/97 5:40 PI
hd		File Folder	9/9/97 2:43 PI
Hfw		File Folder	4/2/97 9:51 AI
idshop		File Folder	6/6/97 4:29 PI
internet-sw		File Folder	5/21/97 8:32 A
java		File Folder	6/9/97 9:54 AI
kathy		File Folder	8/8/96 4:23 PI
Mapngo		File Folder	1/16/97 10:52

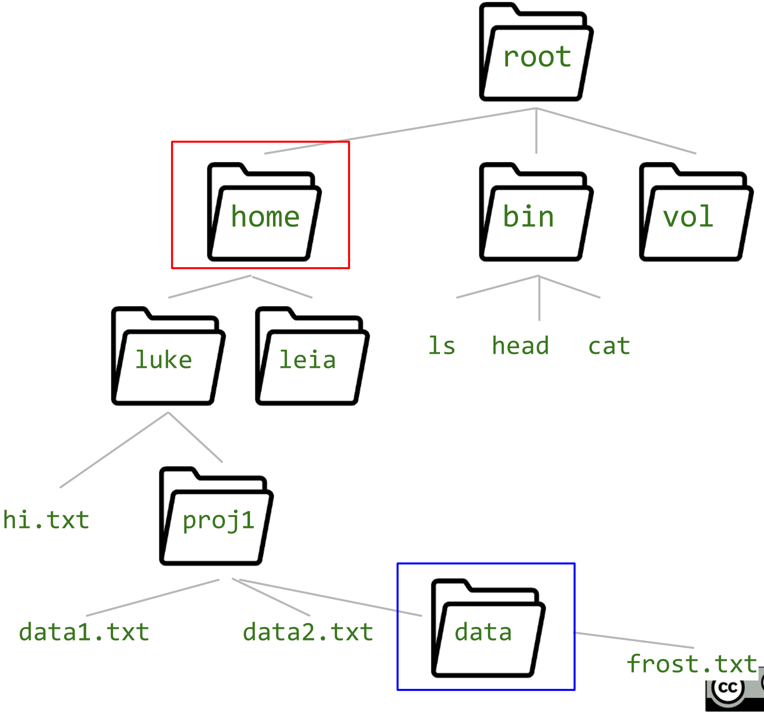
104 object(s)　6.12MB (Disk free space: 578MB)

Navigating filesystems via the command line is essential

Conceptual recap: You are in "home"

- How would you verify that?
- How do you move to a directory above/below home?
- How do you list the contents in the "data" folder

Example Unix file system (a "tree")



Commands we will use/reinforce in this session

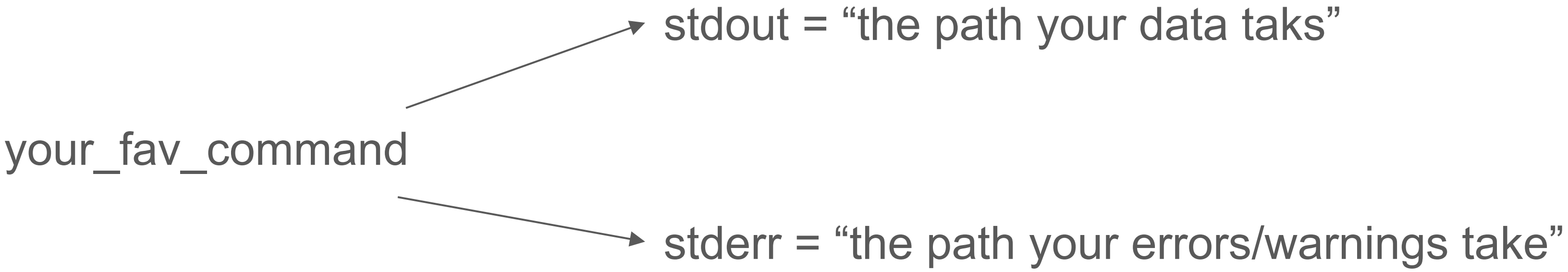
- ls
- wc
- pwd
- cd
- mkdir
- man
- rm
- touch
- mv
- echo

- less
- cat
- >> vs >
- grep
- sort
- Pipe |
- vim

Objectives:

- What does a command output?
- combining sort + uniq
- cut
- More advanced grep usage

“Show me the output of your command”



your_fav_command

stdout = “the path your data taks”

stderr = “the path your errors/warnings take”

Let’s see this in action with the `date` command

```
mkdir ~/command_line_lab

## Get the date
date
```

Let’s see this in action with the `date` command

```
mkdir ~/command_line_lab

## Get the date
date

## Use invalid parameter
date --asdf #error to stderr

## Store date in text file
date >file.txt
```

Let’s see this in action with the `date` command

```
mkdir ~/command_line_lab

## Get the date
date

## Use invalid parameter
date --asdf #error to stderr

## Store date in text file
date >file.txt

## Show date from text file
cat file.txt
```

Let’s see this in action with the `date` command

```
mkdir ~/command_line_lab

## Get the date
date

## Use invalid parameter
date --asdf #error to stderr

## Store date in text file
date >file.txt

## Show date from text file
cat file.txt

## Store the error/warning messages in a txt file
date --asdf 2>err.log
cat err.log
```

```
## What happens if we run this?
date 2> err.log

## And what about this?
date >file.txt
date >>file.txt
date >>file.txt
cat file.txt
```

Learning the **sort | uniq** dynamic duo

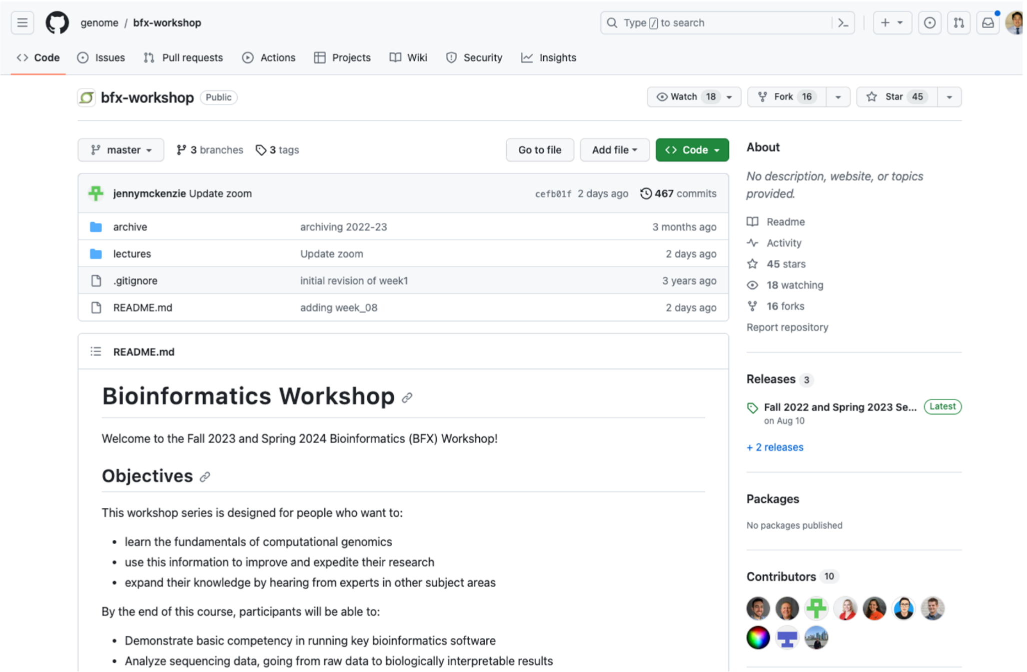
First, let’s download some files

```
mkdir ~/command_line_lab/workshop
```

```
cd ~/command_line_lab/workshop
```

```
git clone https://github.com/genome/bfx-workshop.git
```

```
cd bfx-workshop/lectures/week_02
```



jennymckenzie Update zoom cefb01f 2 days ago 467 commits

archive	archiving 2022-23	3 months ago
lectures	Update zoom	2 days ago
.gitignore	initial revision of week1	3 years ago
README.md	adding week_08	2 days ago

README.md

Bioinformatics Workshop

Welcome to the Fall 2023 and Spring 2024 Bioinformatics (BFX) Workshop!

Objectives

This workshop series is designed for people who want to:

- learn the fundamentals of computational genomics
- use this information to improve and expedite their research
- expand their knowledge by hearing from experts in other subject areas

By the end of this course, participants will be able to:

- Demonstrate basic competency in running key bioinformatics software
- Analyze sequencing data, going from raw data to biologically interpretable results

About

No description, website, or topics provided.

Readme
Activity
45 stars
18 watching
16 forks
Report repository

Releases 3

Fall 2022 and Spring 2023 Se... Latest
on Aug 10

+ 2 releases

Packages

No packages published

Contributors 10

https://github.com/genome/bfx-workshop

Working with sort and uniq

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

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

Working with sort and uniq

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

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

## Count number of “unique” genes
cat genes1.txt | uniq | wc -l
```

Working with sort and uniq

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

## Sort, unique, then count
sort genes1.txt | uniq | wc -l
```

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

## Count number of “unique” genes
cat genes1.txt | uniq | wc -l
```

```
Why is there a discrepancy on the left and right?
```

Working with sort and uniq

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

## Sort, unique, then count
sort genes1.txt | uniq | wc -l
```

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

## Count number of “unique” genes
cat genes1.txt | uniq | wc -l
```

Other useful uniq options:

- uniq -d --> reports only duplicate values
- uniq -u --> reports values that are in the file a single time
- uniq -c --> counts the number of occurrences for each value

Let’s count the number of gene occurrences in genes1.txt with/without sorting

Performing set operations with sort and uniq

genes1.txt = genes highly expressed in a tumor
genes2.txt = genes highly expressed in matching normal tissue

Question 1: how to find if a gene is present in both datasets?

Let’s first talk about the “pseudocode” AKA conceptual code

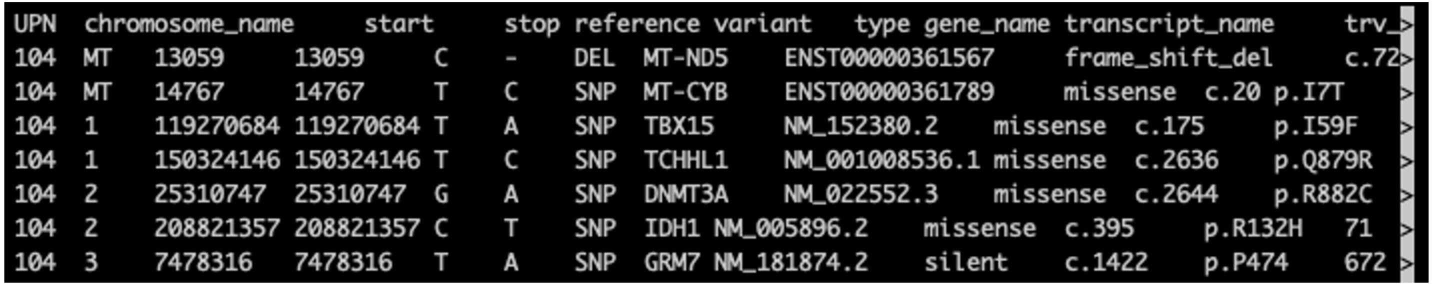
Performing set operations with sort and uniq

```
genes1.txt = genes highly expressed in a tumor
genes2.txt = genes highly expressed in matching normal tissue

Question 1: how to find if a gene is present in both datasets?

Question 2: how do we find genes only found in genes1.txt?
Hint: you will need to “manually” duplicate something.
```

More practical sorting practice with a tab-delimited file

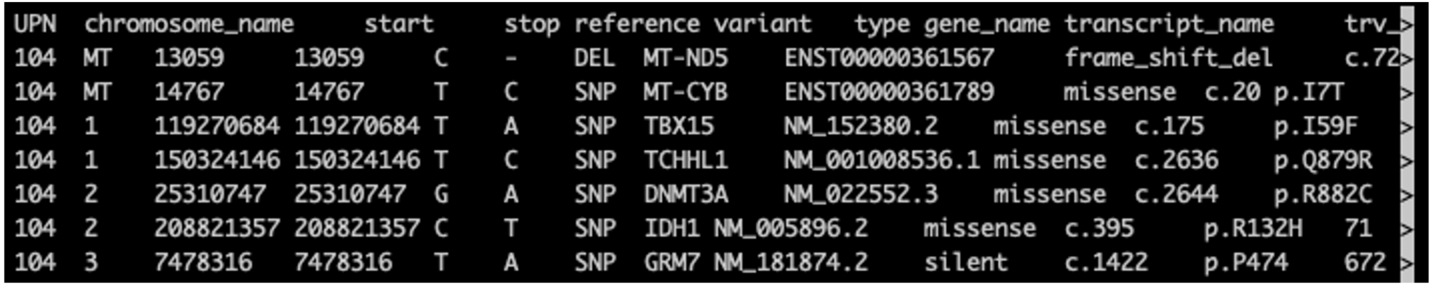


```
UPN chromosome_name     start     stop reference variant   type gene_name transcript_name     trv_>
104 MT  13059   13059   C   -   DEL MT-ND5  ENST00000361567 frame_shift_del c.72>
104 MT  14767   14767   T   C   SNP MT-CYB  ENST00000361789 missense c.20 p.I7T  >
104 1   119270684 119270684 T   A   SNP TBX15   NM_152380.2    missense c.175   p.I59F  >
104 1   150324146 150324146 T   C   SNP TCHHL1  NM_001008536.1 missense c.2636  p.Q879R >
104 2   25310747  25310747  G   A   SNP DNMT3A  NM_022552.3    missense c.2644  p.R882C >
104 2   208821357 208821357 C   T   SNP IDH1 NM_005896.2    missense c.395   p.R132H 71  >
104 3   7478316   7478316   T   A   SNP GRM7 NM_181874.2    silent   c.1422  p.P474  672 >
```

```
## Look at the first 15 rows of tcga.tsv
head -n 15 tcga.tsv

## View the file with the less command. While viewing, type -S and enter
## to toggle wrapping of lines. To increase width of tabs, type “-x20” and
## enter. You can press “q” to exit.
less tcga.tsv
```

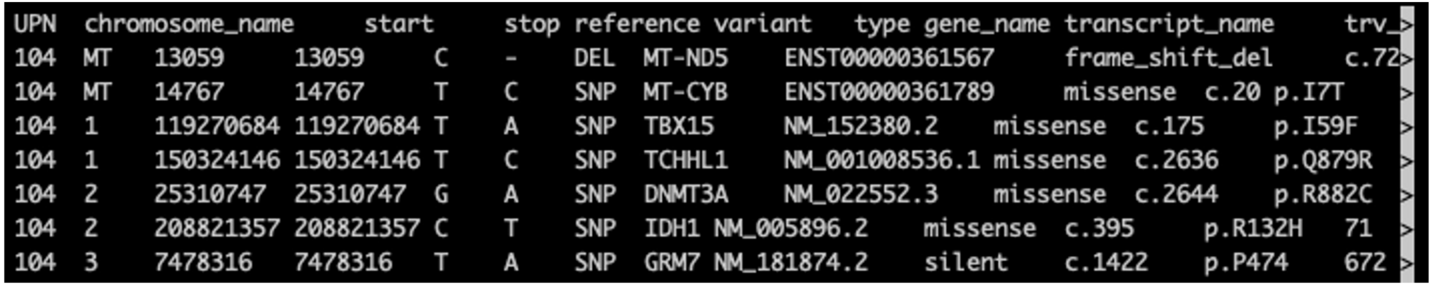
More practical sorting practice with a tab-delimited file



```
UPN  chromosome_name     start     stop reference variant   type gene_name transcript_name     trv_>
104  MT   13059     13059     C    -    DEL  MT-ND5    ENST00000361567     frame_shift_del     c.72>
104  MT   14767     14767     T    C    SNP  MT-CYB    ENST00000361789     missense  c.20 p.I7T    >
104  1    119270684 119270684 T    A    SNP  TBX15     NM_152380.2    missense  c.175     p.I59F   >
104  1    150324146 150324146 T    C    SNP  TCHHL1    NM_001008536.1 missense  c.2636    p.Q879R  >
104  2    25310747  25310747  G    A    SNP  DNMT3A    NM_022552.3    missense  c.2644    p.R882C  >
104  2    208821357 208821357 C    T    SNP  IDH1 NM_005896.2    missense  c.395     p.R132H   71  >
104  3    7478316   7478316   T    A    SNP  GRM7 NM_181874.2    silent    c.1422    p.P474    672 >
```

```
## Why might this be insufficient?
sort tcga.tsv | less
```

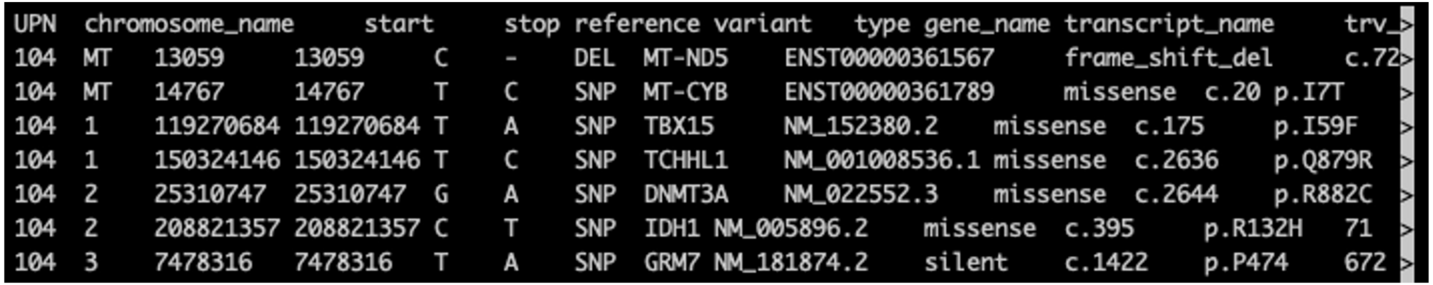
More practical sorting practice with a tab-delimited file



```
UPN  chromosome_name     start      stop reference variant   type gene_name transcript_name      trv_>
104  MT    13059       13059       C     -     DEL  MT-ND5     ENST00000361567      frame_shift_del      c.72>
104  MT    14767       14767       T     C     SNP  MT-CYB     ENST00000361789      missense  c.20 p.I7T     >
104  1     119270684 119270684 T     A     SNP  TBX15      NM_152380.2     missense  c.175      p.I59F      >
104  1     150324146 150324146 T     C     SNP  TCHHL1     NM_001008536.1 missense  c.2636     p.Q879R     >
104  2     25310747  25310747  G     A     SNP  DNMT3A     NM_022552.3     missense  c.2644     p.R882C     >
104  2     208821357 208821357 C     T     SNP  IDH1 NM_005896.2     missense  c.395      p.R132H    71  >
104  3     7478316   7478316   T     A     SNP  GRM7 NM_181874.2     silent     c.1422     p.P474     672 >
```

```
## Better to sort on a specific column of interest (chr)
sort -k 2 tcga.tsv | less
```

More practical sorting practice with a tab-delimited file

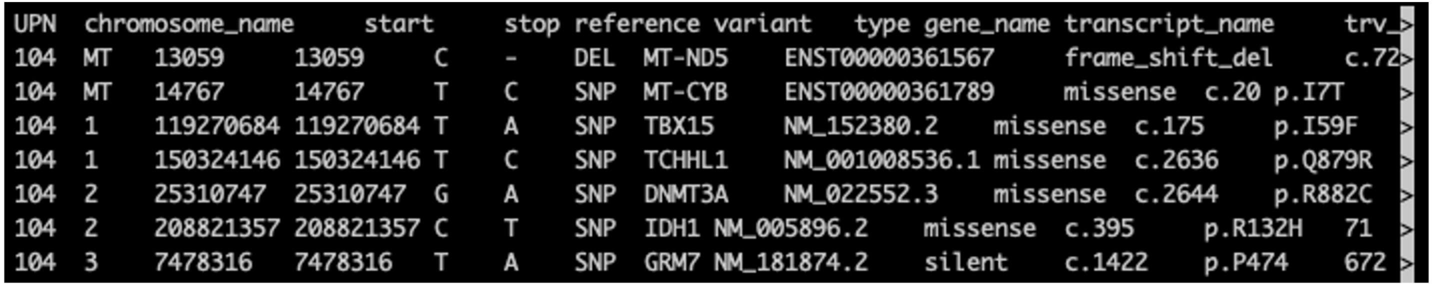


```
UPN  chromosome_name     start     stop reference variant   type gene_name transcript_name     trv_>
104  MT   13059     13059     C    -    DEL  MT-ND5    ENST00000361567     frame_shift_del     c.72>
104  MT   14767     14767     T    C    SNP  MT-CYB    ENST00000361789     missense  c.20 p.I7T    >
104  1    119270684 119270684 T    A    SNP  TBX15     NM_152380.2    missense  c.175     p.I59F   >
104  1    150324146 150324146 T    C    SNP  TCHHL1    NM_001008536.1 missense  c.2636    p.Q879R  >
104  2    25310747  25310747  G    A    SNP  DNMT3A    NM_022552.3    missense  c.2644    p.R882C  >
104  2    208821357 208821357 C    T    SNP  IDH1 NM_005896.2    missense  c.395     p.R132H   71  >
104  3    7478316   7478316   T    A    SNP  GRM7 NM_181874.2    silent    c.1422    p.P474    672 >
```

```
## Better to sort on a specific column of interest (chr)
sort -k 2 tcga.tsv | less

## Or event multiple columns of interest (chr, start)
sort -k 2,2 -k 3,3n tcga.tsv | less
```

More practical sorting practice with a tab-delimited file



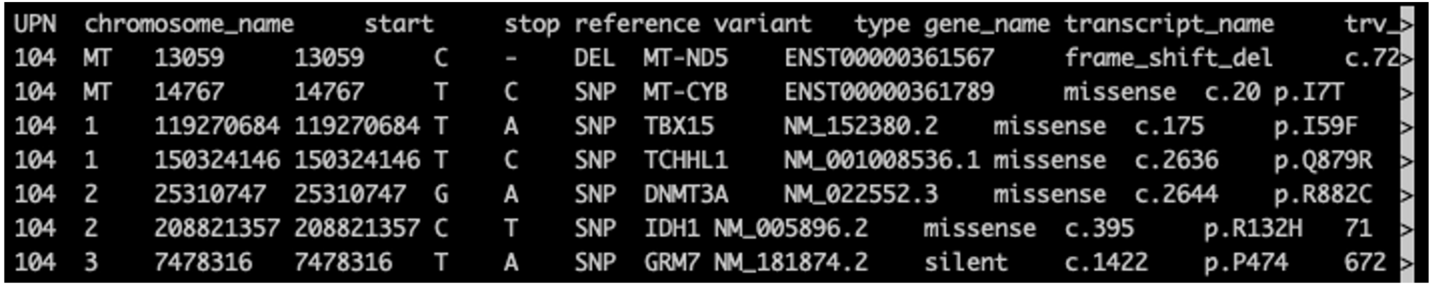
```
UPN  chromosome_name     start     stop reference variant   type gene_name transcript_name     trv_>
104  MT   13059     13059     C    -    DEL  MT-ND5    ENST00000361567     frame_shift_del     c.72>
104  MT   14767     14767     T    C    SNP  MT-CYB    ENST00000361789     missense  c.20 p.I7T    >
104  1    119270684 119270684 T    A    SNP  TBX15     NM_152380.2    missense  c.175     p.I59F   >
104  1    150324146 150324146 T    C    SNP  TCHHL1    NM_001008536.1 missense  c.2636    p.Q879R  >
104  2    25310747  25310747  G    A    SNP  DNMT3A    NM_022552.3    missense  c.2644    p.R882C  >
104  2    208821357 208821357 C    T    SNP  IDH1 NM_005896.2    missense  c.395     p.R132H   71  >
104  3    7478316   7478316   T    A    SNP  GRM7 NM_181874.2    silent    c.1422    p.P474    672 >
```

```
## Better to sort on a specific column of interest (chr)
sort -k 2 tcga.tsv | less

## Or event multiple columns of interest (chr, start)
sort -k 2,2 -k 3,3n tcga.tsv | less
```

```
Question: What is the difference between -k 2,2 and -k 2?
Hint run the following: cut -f 2,3 tcga.tsv | head
```

Using cut and piping to sort --> count unique events



```
UPN  chromosome_name     start     stop reference variant    type gene_name transcript_name     trv_>
104  MT   13059     13059     C    -    DEL  MT-ND5    ENST00000361567     frame_shift_del     c.72>
104  MT   14767     14767     T    C    SNP  MT-CYB    ENST00000361789     missense  c.20 p.I7T    >
104  1    119270684 119270684 T    A    SNP  TBX15     NM_152380.2    missense  c.175     p.I59F   >
104  1    150324146 150324146 T    C    SNP  TCHHL1    NM_001008536.1 missense  c.2636    p.Q879R  >
104  2    25310747  25310747  G    A    SNP  DNMT3A    NM_022552.3    missense  c.2644    p.R882C  >
104  2    208821357 208821357 C    T    SNP  IDH1 NM_005896.2    missense  c.395     p.R132H   71  >
104  3    7478316   7478316   T    A    SNP  GRM7 NM_181874.2    silent    c.1422    p.P474    672 >
```

Question 1: how many missense mutations are in the file?

Question 2: which gene is most frequently mutated in the file?

How many missense mutations are in the tcga.tsv file?

Commands we will need:

- `cut - isolates columns of interest`
- `sort - sorts data based on column(s) value`
- `uniq - identify “unique” values in dataset`

Sort mutation types by prevalence in descending order

Try this yourself/as a group!

Commands we will need:

- `cut - isolates columns of interest`
- `sort x2 - sorts data based on column(s) value`
- `uniq - identify “unique” values in dataset`

Getting fancy with **cat** and **grep** using a “fasta” file

Let’s download the fasta file

```
## Make new directory and change directory
cd ~/command_line_lab/workshop/bfx-workshop/lectures/week_02

## Download fasta file
curl http://hgdownload.cse.ucsc.edu/goldenPath/hg38/chromosomes/chr22.fa.gz > chr22.fa.gz

## Try to cat/less the fa.gz file
cat chr22.fa.gz

## Uncompress the gzipped file and rename
gzip -d chr22.fa.gz
mv chr22.fa hg.b38.chr22.fa
```

Searching for and counting patterns in genomes with grep

```
cat hg.b38.chr22.fa | head -n 10
>chr22
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
```

- First line is a header line to indicate which chromosome we are looking at

Searching for and counting patterns in genomes with grep

```
cat hg.b38.chr22.fa | head -n 10
>chr22
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
```

- First line is a header line to indicate which chromosome we are looking at

```
## Find line with the “>”
grep “>” hg.b38.chr22.fa
```

Searching for and counting patterns in genomes with `grep`

```
cat hg.b38.chr22.fa | head -n 10
>chr22
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
```

- First line is a header line to indicate which chromosome we are looking at

## Find line with the “>”

Searching for and counting patterns in genomes with grep

```
cat hg.b38.chr22.fa | head -n 10
>chr22
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
```

- First line is a header line to indicate which chromosome we are looking at

```
## Find line with the “>”
grep “>” hg.b38.chr22.fa
```

Searching for and counting patterns in genomes with grep

```
cat hg.b38.chr22.fa | head -n 10
>chr22
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
```

- First line is a header line to indicate which chromosome we are looking at

```
## Find line with the “>”
grep “>” hg.b38.chr22.fa

## What will this do?
grep > hg.b38.chr22.fa
```

Searching for and counting patterns in genomes with grep

```
cat hg.b38.chr22.fa | head -n 10
>chr22
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
```

- First line is a header line to indicate which chromosome we are looking at

```
## Find line with the “>”
grep “>” hg.b38.chr22.fa

## What will this do?
grep > hg.b38.chr22.fa

## Why does this return nothing?
grep -w “A” hg.b38.chr22.fa
```

How could we determine how many nucleotides are in chr22?

```
cat hg.b38.chr22.fa | head -n 10
>chr22
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
```

Need to remove the >chr22 line... How?

```
grep -v “>” hg.b38.chr22.fa
```

How could we determine how many nucleotides are in chr22?

```
cat hg.b38.chr22.fa | head -n 10
>chr22
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
```

```
Need to remove the >chr22
line... How?

grep -v “>” hg.b38.chr22.fa

## What does this do?
grep -v “>” hg.b38.chr22.fa |
wc -l
```

How could we determine how many nucleotides are in chr22?

Count characters with `wc -c`

```
grep -v ">" hg.b38.chr22.fa | wc -c
51834838
```

Wait - this is wrong... why? Because of hidden characters, which in this case, indicate a newline

What are hidden characters?

```
cat -t -e hg.b38.chr22.fa | head

>chr22$
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN$
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN$
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN$
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN$
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN$
```

How could we determine how many nucleotides are in chr22?

```
## Counts all characters in file
$ grep -v ">" hg.b38.chr22.fa | wc -c
51834838
```

How could we determine how many nucleotides are in chr22?

```
## Counts all characters (hidden + nucleotide) in file
$ grep -v ">" hg.b38.chr22.fa | wc -c
51834838

## Why do we perform this calculation?
$ grep -v ">" hg.b38.chr22.fa | wc -l
1016370

51834838 - 1016370 = 50818468
```

How many adenosines are there on chr22?

```
## Find A nucleotides in the file
grep -v ">" hg.b38.chr22.fa | grep "A"

## “Count” A nucleotides in the file
grep -v ">" hg.b38.chr22.fa | grep -c "A"
410249
```

Let’s sanity check our work.

1. We know that ~42% of the human genome is GC.
2. Therefore the AT content is ~58%
3. Thus we expect the A content to be ~58% / 2 = 14.5 million A nucleotides
1. But we see **`410249 / 50818468 = 0.8%`**
2. Fishy! What is going on here?

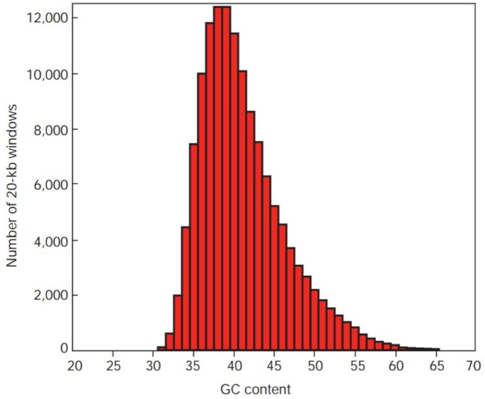


Figure 12 Histogram of GC content of 20-kb windows in the draft genome sequence.

Man page entry for grep -c

```
-c, --count
    Suppress normal output; instead print a
count of matching lines for each input file.
```

The -o option

```
-o, --only-matching
    Print only the matched (non-empty) parts of a matching line, with each such part on a separate output line.
```

```
grep -v ">" hg.b38.chr22.fa | grep -o "A"
```

The -n option

```
-n, --line-number
    Prefix each line of output with the 1-based
line number within its input file.
```

```
grep -v ">" hg.b38.chr22.fa | grep -o -n "A"
```

How many adenosines are there on chr22?

```
$ grep -v ">" hg.b38.chr22.fa | grep -o -n "A" | head -n 16
210201:A
210201:A
210201:A
210201:A
210201:A
210201:A
210201:A
210201:A
210201:A
210201:A
210201:A
210201:A
210201:A
210201:A
210201:A
210202:A
```

Line number:match

```
grep -v ">" hg.b38.chr22.fa | grep -o -n "A" | wc -l
4,583,339
```

Expected 14.5 million A’s so we are still missing something

We need to search for ”A” or ”a”
The name ”grep” stands for ”global regular expression print”.

```
$ grep -v ">" hg.b38.chr22.fa | grep -o -n "[A|a]" | less
```

Our first regular expression.
Match ”A” or (|) ”a”

How many adenosines are there on chr22?

```
$ grep -v ">" hg.b37.chr22.fa | grep -o -n "[A|a]" | wc -l
10382214
```

Why is our calculation still off?

- We know 29% of nucleotides are A’s
- We said there are 50818468 nucleotides on chr22
  - Includes A, T, G, C, and Ns
  - We need to exclude N’s

Let’s sanity check our work.

1. We learned in the last lecture that ~42% of the human genome is GC.
2. Therefore the AT content is ~58%
3. Thus we expect the A content to be ~58% / 2
4. But we see **`10382214 / 50818468 = 20.4%`**
5. Better, but still not what we expect. Why?

Need to remove gaps (Ns)! How?

Command Line Lab

Exercises

Question 1: What is the nucleotide sequence for the 542,560th line in the chr22 fasta file?

Question 2: How many G or C nucleotides are there on chr22?

Question 3: What is the GC content (% nucleotides that are G or C in the file)?

Question 4: How many lines in the chr22 fasta file have exactly 15 cytosines?

Bonus How many lines in the chr22 file have ≥ 15 cytosines (hint: may need to look up additional cut options to isolate the counts per line)