Bioinformatics Lessons Schedule

NOTE: It’s already changed 3 times, so it will continue to change as we go forward.

Date	Subject
10-22	Terminus
10-29	Server Basics
11-5	Server Basics, continued
11-12	Server Basics, continued
11-19	How to run software on the server
11-26	No lesson, week of Thanksgiving
12-03	Basic Git
12-10	Start running through RRBS (or RNA-seq)
12-17	Continue from the previous week
12-24	Christmas break
12-31	Christmas break
01-07	Resume lessons

How to Use the Server

Command-Line Basic Navigation (continued)

Wildcards

Wildcards

- * = match all characters

```
$ ls

f1.txt f1.log f2.txt f2.log f3.txt
f3.log f4.txt f4.log f5.txt f5.log

$ ls *.txt

f1.txt f2.txt f3.txt f4.txt f5.txt
```

Wildcards

- * = match all characters
- ? = match a single character

```
$ ls

f1.txt f1.log f2.txt f2.log f3.txt
f3.log f4.txt f4.log f5.txt f5.log

$ ls *.txt

f1.txt f2.txt f3.txt f4.txt f5.txt

$ ls f?.log

f1.log f2.log f3.log f4.log f5.log
```

Wildcards

- * = match all characters
- ? = match a single character
- [] = match a range of values

```
$ ls

f1.txt f1.log f2.txt f2.log f3.txt
f3.log f4.txt f4.log f5.txt f5.log

$ ls *.txt

f1.txt f2.txt f3.txt f4.txt f5.txt

$ ls f?.log

f1.log f2.log f3.log f4.log f5.log

$ ls f[2-4]*

f2.txt f2.log f3.txt f3.log f4.txt
f4.log
```

Wildcards

- * = match all characters
- ? = match a single character
- [] = match a range of values
- ! = NOT (just like R)

```
$ ls

f1.txt f1.log f2.txt f2.log f3.txt
f3.log f4.txt f4.log f5.txt f5.log

$ ls f[!2-4]*

f1.txt f1.log f5.txt f5.log
```

Wildcards

- * = match all characters
- ? = match a single character
- [] = match a range of values
- ! = NOT (just like R)
- {} = match a list of values
- You can use as many wildcards as you can think of

```
$ ls

f1.txt f1.log f2.txt f2.log f3.txt
f3.log f4.txt f4.log f5.txt f5.log

$ ls f[!2-4]*

f1.txt f1.log f5.txt f5.log

$ ls *[1-3]?{txt,log}

f1.txt f1.log f2.txt f2.log f3.txt
f3.log
```

PRACTICE
BREAK

Miscellaneous Useful Things

Miscellaneous Useful Things

- history = remembers (up to, on our system) the 1,000 previous commands

Miscellaneous Useful Things

- history = remembers (up to, on our system) the 1,000 previous commands
- up arrow = if you hit up, it fills in the previous command; can do this multiple times

Miscellaneous Useful Things

- history = remembers (up to, on our system) the 1,000 previous commands
- up arrow = if you hit up, it fills in the previous command; can do this multiple times
- clear / ctrl + L = if you want a blank screen, type “clear” or hit ctrl + L to clear the screen

Miscellaneous Useful Things

- history = remembers (up to, on our system) the 1,000 previous commands
- up arrow = if you hit up, it fills in the previous command; can do this multiple times
- clear / ctrl + L = if you want a blank screen, type “clear” or hit ctrl + L to clear the screen
- There is tab completion (like in Rstudio) for command names and file paths (tab completion is your friend)

PRACTICE BREAK

Input and Output (I/O)

More File Wrangling

```
$ cat wrong_alphabet.txt
L is for llama
D is for donkey
O is for octopus
R is for raccoon
X is for x-ray tetra
T is for tarantula
N is for narwhal
V is for vicuna
Y is for yak
G is for Galapagos tortoise
P is for penguin
M is for mongoose
E is for elephant shrew
W is for warthog
H is for hippopotamus
J is for jackal
F is for flamingo
K is for kangaroo
A is for aardvark
C is for chihuahua
Z is for zebra
I is for iguana
Q is for quetzal
B is for bumblebee
U is for unicorn
S is for squirrel
Z is for zebra
K is for kangaroo
Y is for yak
A is for aardvark
Q is for quetzal
N is for narwhal
C is for chihuahua
G is for Galapagos tortoise
B is for bumblebee
U is for unicorn
```

More File Wrangling

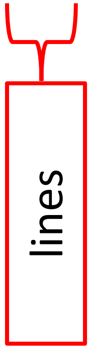
- wc = “word count”

```
$ wc wrong_alphabet.txt
36 148 649 alphabet.txt
```

More File Wrangling

- wc = “word count”

```
$ wc wrong_alphabet.txt
36 148 649 alphabet.txt
```

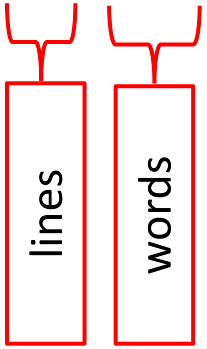


lines

More File Wrangling

- wc = “word count”

```
$ wc wrong_alphabet.txt
36 148 649 alphabet.txt
```



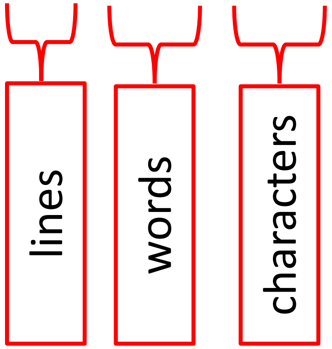
lines

words

More File Wrangling

- wc = “word count”

```
$ wc wrong_alphabet.txt
36 148 649 alphabet.txt
```



More File Wrangling

- wc = “word count”
  - -l = give line count only
  - -w = give word count only
  - -c = give character count only

```
$ wc wrong_alphabet.txt

36 148 649 alphabet.txt

$ wc –l wrong_alphabet.txt

36 file1.txt
```

More File Wrangling

- wc = “word count”
  - -l = give line count only
  - -w = give word count only
  - -c = give character count only
- sort (like arrange() in R)

```
$ wc wrong_alphabet.txt

36 148 649 alphabet.txt

$ wc –l wrong_alphabet.txt

36 file1.txt

$ sort wrong_alphabet.txt
A is for aardvark
A is for aardvark
B is for bumblebee
B is for bumblebee
C is for chihuahua
```

More File Wrangling

- wc = “word count”
  - -l = give line count only
  - -w = give word count only
  - -c = give character count only
- sort (like arrange() in R)
- cut = print selected parts of files
  - -f = select columns to print
  - -d = let cut know what the delimiter is

```
$ $ wc wrong_alphabet.txt

36 148 649 alphabet.txt

$ wc –l wrong_alphabet.txt

36 file1.txt

$ sort wrong_alphabet.txt
A is for aardvark
A is for aardvark
B is for bumblebee
B is for bumblebee
C is for chihuahua
$ cut –d ‘ ‘ –f 4 wrong_alphabet.txt
ardvark
ardvark
umblebee
umblebee
Chihuahua
```

More File Wrangling

- wc = “word count”
  - -l = give line count only
  - -w = give word count only
  - -c = give character count only
- sort (like arrange() in R)
- cut = print selected parts of files
  - -f = select columns to print
  - -d = let cut know what the delimiter is
- uniq = “unique”
  - -c = count

```
$ uniq wrong_alphabet.txt
L is for llama
D is for donkey
O is for octopus
R is for raccoon
X is for x-ray tetra
```

More File Wrangling

- wc = “word count”
  - -l = give line count only
  - -w = give word count only
  - -c = give character count only
- sort (like arrange() in R)
- cut = print selected parts of files
  - -f = select columns to print
  - -d = let cut know what the delimiter is
- uniq = “unique”
  - -c = count
- grep
  - -v = NOT

```
$ uniq wrong_alphabet.txt
L is for llama
D is for donkey
O is for octopus
R is for raccoon
X is for x-ray tetra
$ uniq –c wrong_alphabet.txt
      1 L is for llama
      1 D is for donkey
      1 O is for octopus
      1 R is for raccoon
      1 X is for x-ray tetra
```

More File Wrangling

- wc = “word count”
  - -l = give line count only
  - -w = give word count only
  - -c = give character count only
- sort (like arrange() in R)
- cut = print selected parts of files
  - -f = select columns to print
  - -d = let cut know what the delimiter is
- uniq = “unique”
  - -c = count
- grep
  - -v = NOT

```
$ uniq wrong_alphabet.txt
L is for llama
D is for donkey
O is for octopus
R is for raccoon
X is for x-ray tetra
$ uniq –c wrong_alphabet.txt
      1 L is for llama
      1 D is for donkey
      1 O is for octopus
      1 R is for raccoon
      1 X is for x-ray tetra
$ grep ‘yak’ wrong_alphabet.txt
Y is for yak
Y is for yak
```

More File Wrangling

- wc = “word count”
  - -l = give line count only
  - -w = give word count only
  - -c = give character count only
- sort (like arrange() in R)
- cut = print selected parts of files
  - -f = select columns to print
  - -d = let cut know what the delimiter is
- uniq = “unique”
  - -c = count
- grep
  - -v = NOT

```
$ uniq wrong_alphabet.txt
L is for llama
D is for donkey
O is for octopus
R is for raccoon
X is for x-ray tetra
$ uniq –c wrong_alphabet.txt
      1 L is for llama
      1 D is for donkey
      1 O is for octopus
      1 R is for raccoon
      1 X is for x-ray tetra
$ grep ‘yak’ wrong_alphabet.txt
Y is for yak
Y is for yak
$ grep –v ‘llama’ wrong_alphabet.txt
D is for donkey
O is for octopus
R is for raccoon
X is for x-ray tetra
T is for tarantula
```

PRACTICE BREAK