Getting data from ftp servers and other useful commands

Getting data off the web.

Data files from an external source are often necessary to start an analysis. For example, one may align DNA sequence reads to a reference genome, but one needs the reference genome.

Genome sequences and annotations change over time. One should make sure to keep tabs on what, where, and when information is downloaded.

Some commands can download a lot of data. If data is being downloaded, remember you can kill a process using <control c>.
Also, you may want to delete large files that you can get easily.

Your browser, curl, ftp, and wget

I find data to download on an ftp server with an internet browser. As an example, we will capture some .gff files. GFF= general feature format. These files describe the locations and attributes of features on a nucleotide sequence. Let's get one of these files.

On a web browser, go to ensembl.org
Click "All genomes"
Click "Opposum" Monodelphis domestica

If you click on "Download DNA sequence," it will connect you to the genome sequence, split up into chromosomes, that you can download. If you click on "Download GFF3," you can download the annotation of that genome sequence split into chromosomes.

This address allows you to download a file from a web browser.

```
ftp://ftp.ensembl.org/pub/release-97/gff3/monodelphis_domestica/Monodelphis_domestica.monDom5.97.chromosome.7.gff3.gz
```

Unix can also be used to download, but you need to tell unix the path to the file you want to copy which can be difficult to find. On Mac firefox, you can copy paths by clicking and holding the file name, dragging it to another application, and releasing the mouse. In Windows, you can right click on the file name to copy the path and then paste.

wget works on compute Canada server but not on my macs. This command quickly downloads data from the command line. It can handle http links. It can handle ftp links.

```
wget ftp://ftp.ensembl.org/pub/release-97/gff3/monodelphis_domestica/Monodelphis_domestica.monDom5.97.chromosome.7.gff3.gz
```

Wildcard characters work with wget

```
wget
ftp://ftp.ensembl.org/pub/current_gff3/monodelphis_domestic
a/*.1.gff3.gz
```

Also character designations in [ ] and ? (see below)

wget also can recursively traverse a file hierarchy, delivering everything to you.
(don't do unless you want a lot of data)

```
wget -r
ftp://ftp.ensembl.org/pub/current_gff3/monodelphis_domestic
a/
```

-r means recursive. In this case, it will only copy this directory because there is no directory below it. There is a limit in terms of how deep in sub directories wget will travel, and this can be changed. The -A option will accept only certain filetypes.

wget does not work on Macs (or at least my macs) by default. (I've installed homebrew: "The missing package manager for macOS (or Linux)" to make wget and ftp available (See below).

curl is another command for transferring files

```
curl -O ftp://ftp.ensembl.org/pub/release-
97/gff3/monodelphis_domestica/Monodelphis_domestica.monDom5
.97.chromosome.7.gff3.gz
```

The issue with curl is that it does not take wild cards. There are workarounds- like making a list of desired files using a command such ls and looping through the list, but they are a bit complicated.

Connecting via ftp is another way of transferring files.

ftp is not available on Compute Canada machine I use and on one (newer) macs. But, it is available via homebrew and on the older mac. Ubuntu can use ftp.

```
ftp ftp.ensembl.org
```

For username, type: anonymous
For password: email address

If the files you're transferring are anything other than plain ASCII text files, specify a binary transfer before sending or receiving. To do so, at the FTP prompt, enter: binary.

You can travel through the ftp server's filesystem with cd and other commands you have learned. To get files, use mget.

```
mget *some.M*
mget *some.[78]*
mget *some.[7-9]*
mget *some.M?*
```

The asterisk (`*`) is a wildcard that tells FTP to match all files starting with any characters with "some.M" followed by other characters. You can also use a question mark (`?`) to match a single character and brackets to denote possible character classes.

When prompted, enter `y` to transfer each file. To turn this feature off, before you begin transferring files, at the FTP symbol, enter: prompt.

Compression
The files have a postcript .gz.

gzip and bzip2 are two data compression systems.
gzip is more often used.
gzip can compress files on disk in place. That is what we will do.

The command gzip compresses; the command gunzip decompresses.
Use ls -l and look at byte #s.

Now,

```
gunzip
Monodelphis_domestica.monDom5.97.chromosome.7.gff3.gz
```

Now, what is the size? Re-compress, if you want.

```
gzip Monodelphis_domestica.monDom5.97.chromosome.7.gff3
```

Some tools work with compressed files, so you don't need to unpack. I always like to take a look at new data. How can we do this with big files? A text editor (or even Word) can be used to take a look, though they may get bogged down by file size. More centrally, it is not a good idea to use a text editor because you usually don't want to change anything in your data files inadvertently.

less is a text viewer
less `Monodelphis_domestica.monDom5.97.chromosome.7.gff3`

```
Less commands:
b    up one page
<space> down one page
j down one line
k up one line
q quit
h access help page
g jump to start
G jump to end of file
/ start searching forward
? start searching back
```

Type G. How long is this chromosome?

Type g. What is the first feature?

We can go to certain line numbers.
10g goes to 10th line in the file. Try this a few times.

-N option uses line numbers
-n option removes them

We can search
/ start searching forward
? start searching back
Type / or ? followed by enter to jump to the next/ previous match.

head and **tail**

An even quicker way of getting a sense of a file is through head and tail

```
head -n100
Monodelphis_domestica.monDom5.97.chromosome.7.gff3
tail -n100
Monodelphis_domestica.monDom5.97.chromosome.7.gff3
```

Let's say we want to look at line 16000 of the file

One way it to take 16001 of the top lines and then take the last line.

```
head -n16000
Monodelphis_domestica.monDom5.97.chromosome.7.gff3 | tail -
n1
```

The bar in the statement is a "pipe." Instead of saving output from a command ('>' which we will cover later), and using this saved output to perform another command, we can connect commands with pipes. A pipeline is a set of commands that are linked together. One takes the output of one command and uses it as input for another.

cut
Our gff file has a header, but beyond it is tab delimited data. With data separated by a delimitor, cut is handy.

```
cut -f2 Monodelphis_domestica.monDom5.97.chromosome.7.gff3
| head
```

#what happened?

cut assumes space delimits columns. You can change this behaviour with option with -d and define a new delimitor, e,g, cut -f 2 -d ','.

wc
This is another useful command.
`wc` #print newline, word, and byte counts for each file

`wc -l` #lines only

sort
sort file sorts a file alphanumerically by line. In effect groups according to the first column. #by default uses spaces as delimitors. Could use -t ","

temp file is:

```
again  over   there
redo   hello  there
stop   saying over
echo   Over   there
```

```
sort -k3 temp
```

```
stop   saying over
again  over   there
echo   Over   there
redo   hello  there
```

```
sort -k3 -k2 temp
```

```
stop   saying over
redo   hello  there
again  over   there
echo   Over   there
```

#how to break ties.
#if still equal, sort will sort them according to entire line. This could change the order of lines that are equal. -s gets rid of this option

You can sort numerically with -k2,2n. If all columns are numeric, use the -n flag.

Here is a new file called temp.

```
a       21
a       1
c       5
b       6
```

```
sort -k2 temp
```

#sorting by 1st column, then the second. What happens if you don't use the n for the second column?

```
a       1
a       21
b       6
c       5
```

```
sort -k2n temp
```

```
c       5
b       6
a       21
a       1
```

Here is another new once again called temp.

```
a       10
a       10
b       15
b       13
b       15
```

uniq
uniq outputs all lines with *consecutive* duplicates removed.

```
sort –uk2n  temp
```
cut-f3 temp | uniq -c Gives us counts of occurrences.
cut -f3 temp | uniq -c

grep
Many text-based data files use a common syntax of one record per line. Grep searches lines for matches. We will discuss in more detail later.

grep [options] <pattern> file
#patterns can get complex.

Here is “temp”:

```
again   over    there
echo    Over    there
stop    saying  over
redo    hello   there
```

What lines contain there?
What lines contain there or There?
What lines don't?
How many different words are in the second column?
What lines comtain an o flanked by spaces?
What lines have two ls in a row.

grep [options] <pattern> file
#patterns can get complex.

grep Over temp

```
echo    Over    there
```

#note case sensitivity
#the -i flag means "ignore case"
grep -i Over temp

#what lines don't match?
#The -v option turns the logic of the grep command upside down; it shows lines that don't match the pattern.

```
grep -iv Over temp
redo     hello    there
```

We can search for do. #note how grep matched do even though it was in a word.

```
grep -i do temp
redo     hello    there
```

```
##do can be a word
##grep -iw do temp
```

```
# how many lines have a match to a pattern?
grep -c over temp
grep -ic Over temp
```

```
#what line numbers have a matching pattern?
grep -n over temp
#How many lines have either a T or a t in a row?
```

```
#grep -c [tT] temp
#grep -c [tT][h] temp
```

How about the beginning of a line?
^
```
grep ^again temp
```

Grep Options
-c prints a count of matching lines
-l lists the names of files with matching lines, but not the individual matched lines
- i ignore case
-A prints lines after the match
-B prints lines before the match
-c count
-v revert match- prints those lines that don't have pattern
-n prints line numbers
-f takes a list of patterns from file; one per line
-e allows multiple patterns to be matched
-E parse pattern as extended regular expression
- w is the word boundary-- i.e. 259 would match 259 not 2597.

cat

Note the cat command prints the entire contents of a file.
```
#Also useful for doing something line by line...
cat hello | tr '\n' '\t'
```

tr= translate characters.