# EEB 177 Lecture 4

Topics

- Advanced shell

# Office Hours

Tues, Wednesday 11-12

Hershey Courtyard (inside Hershey Hall)

Terasaki 2149 on rainy days

# Hacking Sessions

Tuesdays from 6-8 in LS 3209

## Preliminaries

- Start gedit: `$ gedit` and save the file "classwork-Thursday-1-21.txt" to your class-assignments directory
- push this to your remote repository
- you can write answers to today's exercises in this file.

# Challenge 1

Append all answers to your class-exercises file.

- Go to the datadirectorywithinCSB/unix.
- How many lines are in file `Marra2014_data.fasta`?
- Create the empty file toremove.txt in the CSB/unix/sandbox without leaving the current directory.
- List the content of the directory unix/sandbox.
- Remove the file toremove.txt.

# Wildcards

We will go through examples from CSB section 1.6.5 together. Wildcards are placeholders for one or more symbols. You used wildcards in your lab. In the shell we can use the `*` wildcard (match 0 or more characters except for a leading `.`) to find specific file types.

For example, to see only text files we could type:

```
ls *.txt
```

1.6.5 shows other powerful applications of this wildcard.

# Printing and modulating files

We will go through examples from CSB section 1.6.4 together. The following commands will help you learn to manipulate text files within the shell

# less

this command lets you examine the contents of large text files. You can move through these pages with `ctrl-f` and `ctrl-b`. Exit `less` with `q`; `h` for more commands.

Try this

examine the file

`Marra2014_data.fasta` in the `/CSB/unix/data directory`

# cat

`cat file1 file 2...` concatenates and prints files

Try this

concatenate the following files (in the `/CSB/unix/data directory`)

`Marra2014_about.txt` `Buzzard2015_about.txt`

## wc

gives line, word, and byte count of a file. Look at the `-w -l -c -m` options in the manual. What do they do?

Try this

how man words in are the file `Marra2014_about.txt` ?

# sort

sorts lines of a file alphabetically or numberically (with `-n`). TO choose a specific column for sorting, choose `-k`. For revese sort use `-r`.

Try this

numerically sort the file `Gesquiere2011_data.csv` by the second column.

# head and tail

these commands show the beginning and end of a text file. Use `-n` to specify the number of lines to show.

Try this

show the first and last five lines of the file `Gesquiere2011_data.csv`

show everything but the first line of `Gesquiere2011_data.csv` (hint: see the manual on how to start from a specific line)

# Redirection and pipes

On Tuesday I introduced you to the `>` command which redirects screen output to a file and the `>>` which *appends* output to a file.

```
ls > current_dir_contents.txt
```

```
cat history >> myhomework.txt
```

The `echo` command prints a string to the screen. Tell the shell to print your name. Now tell the shell to print your name to a file called `name.txt`

# pipe and redirect example

Lets use the commands you have learned already to avoid a tedious task. Imagine that you want to know the number files within the folder `Saavedra2013`. How could you do this?

# pipe and redirect example

One way woud be to go to the folder and then count by hand. But this would be tedious!

# pipe and redirect example

We can use the shell to do this in two steps by creating a text file that contains all of the file names and then counting the length of that file.

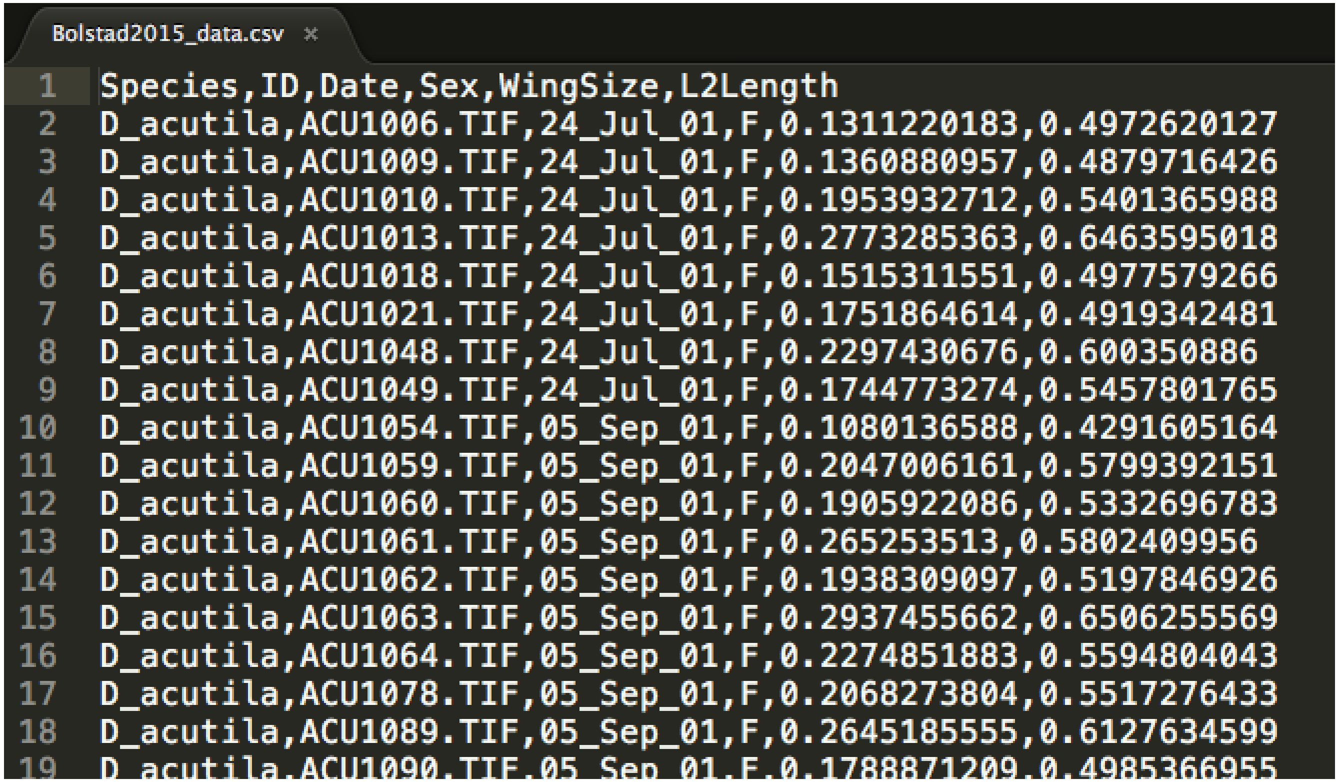
```
ls ../data/Saavedra2013 >> filelist.txt
wc -l filelist.txt
```

But we can do even better using the pipe command, `|`. Pipe says take the output on the left side and send it as input to the right side. So, for example, we can do the above in one line:

```
ls ../data/Saavedra2013 | wc-l
```

# csv files

One of the most common and useful formats for tabular data is .csv (Comma Separated Values) where columns are separated by a comma or other delimiter.



```
Species,ID,Date,Sex,WingSize,L2Length
D_acutila,ACU1006.TIF,24_Jul_01,F,0.1311220183,0.4972620127
D_acutila,ACU1009.TIF,24_Jul_01,F,0.1360880957,0.4879716426
D_acutila,ACU1010.TIF,24_Jul_01,F,0.1953932712,0.5401365988
D_acutila,ACU1013.TIF,24_Jul_01,F,0.2773285363,0.6463595018
D_acutila,ACU1018.TIF,24_Jul_01,F,0.1515311551,0.4977579266
D_acutila,ACU1021.TIF,24_Jul_01,F,0.1751864614,0.4919342481
D_acutila,ACU1048.TIF,24_Jul_01,F,0.2297430676,0.600350886
D_acutila,ACU1049.TIF,24_Jul_01,F,0.1744773274,0.5457801765
D_acutila,ACU1054.TIF,05_Sep_01,F,0.1080136588,0.4291605164
D_acutila,ACU1059.TIF,05_Sep_01,F,0.2047006161,0.5799392151
D_acutila,ACU1060.TIF,05_Sep_01,F,0.1905922086,0.5332696783
D_acutila,ACU1061.TIF,05_Sep_01,F,0.265253513,0.5802409956
D_acutila,ACU1062.TIF,05_Sep_01,F,0.1938309097,0.5197846926
D_acutila,ACU1063.TIF,05_Sep_01,F,0.2937455662,0.6506255569
D_acutila,ACU1064.TIF,05_Sep_01,F,0.2274851883,0.5594804043
D_acutila,ACU1078.TIF,05_Sep_01,F,0.2068273804,0.5517276433
D_acutila,ACU1089.TIF,05_Sep_01,F,0.2645185555,0.6127634599
D_acutila,ACU1090.TIF,05_Sep_01,F,0.1788871209,0.4985366955
```

# working with csv files in the shell

We can use several commands you have learned already plus the `cut` command to easily manipulate csv files.

First, take a look at `Pacifici2013_data.csv` using your text editor. Then move to the containing diretory and use a unix command to view the first line (only) of that file.

What is the delimiter in this file?

```
head -n 1 Pacifici2013_data.csv
```

We can use `cut` to extract specific fields by specifying the delimiter with `-d` and the desired columns with `-f` argument.

```
head -n 1 Pacifici2013_data.csv | cut -d ’;’ -f 1-4
```

If we wanted to list rows of data without the header, we can pipe the results of cut to `tail` (remember `tail -n +2` will show the contents of a file or stream starting from the second line.

```
cut -d ’;’ -f 2 | head -n 5| tail -n +2
```

## Challenge 2

Show the Order of the first 5 species in the data set. Append this to your class-exercises files for today.

(hint: you will need `cut`, `|`, `tail`, and `head`)

## Challenge 3

use what you know plus the `uniq` command to count the number of unique families in this file.hint: you will need to sort your data before you apply `uniq`. Append the line "There are X unique families:" (fill in the value for x) to your exercise file. Then append the list of unique families to your exercise file.

# Reformatting a csv file

We will now work through example 1.7.3 to create a data file with the following fields: Order, Family, Genus, Scientific_Name and AdultBodyMass_g with the following properties

- no headers
- data are sorted by size from large to small
- delimter is a space

We will need to introduce the `tr` command to translate characters.

#grep

We will work through example 1.7.4 together to explore grep. Grep is a powerful pattern matching command that can be combined with the regular expressions you used in lab. Useful grep options: `-c` to count lines, `-w` to match words, `-i` to make case insensitive, `-n` to show line number of match.

# Permissions

We will go through examples from CSB section 1.6.6 together.