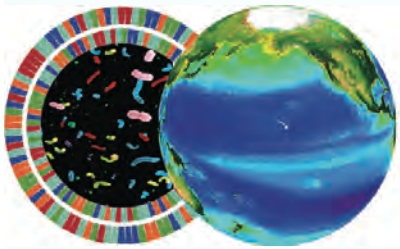
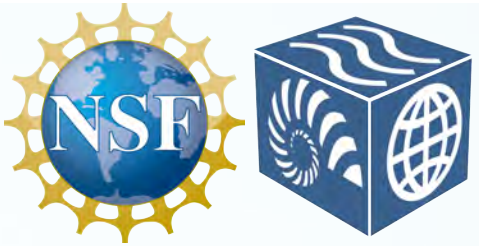


EarthCube Oceanography and Geobiology Environmental 'Omics

ECOGEO



ECOGEO Workshop 2: Introduction to Env 'Omics

Unix and Bioinformatics

Ben Tully (USC); Ken Youens-Clark (UA)

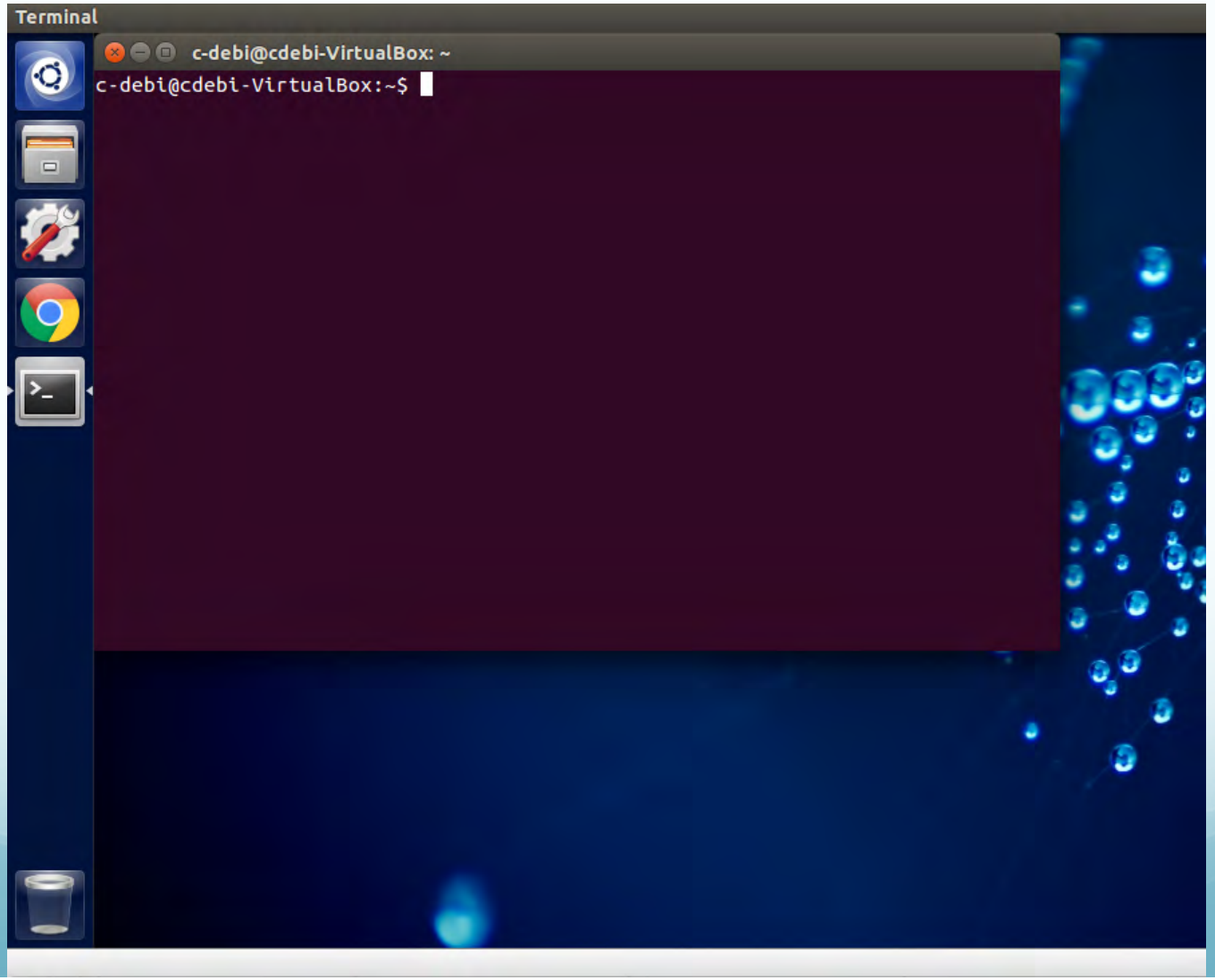
iMicrobe

Unix Commands

pwd	rm	grep	tail	install
ls	‘>’	sed	cut	
cd	cat	nano	top	
mkdir	‘<’	history	screen	
touch	‘|’	$PATH	ssh	
cp	sort	less	df	
mv	uniq	head	rsync/scp	

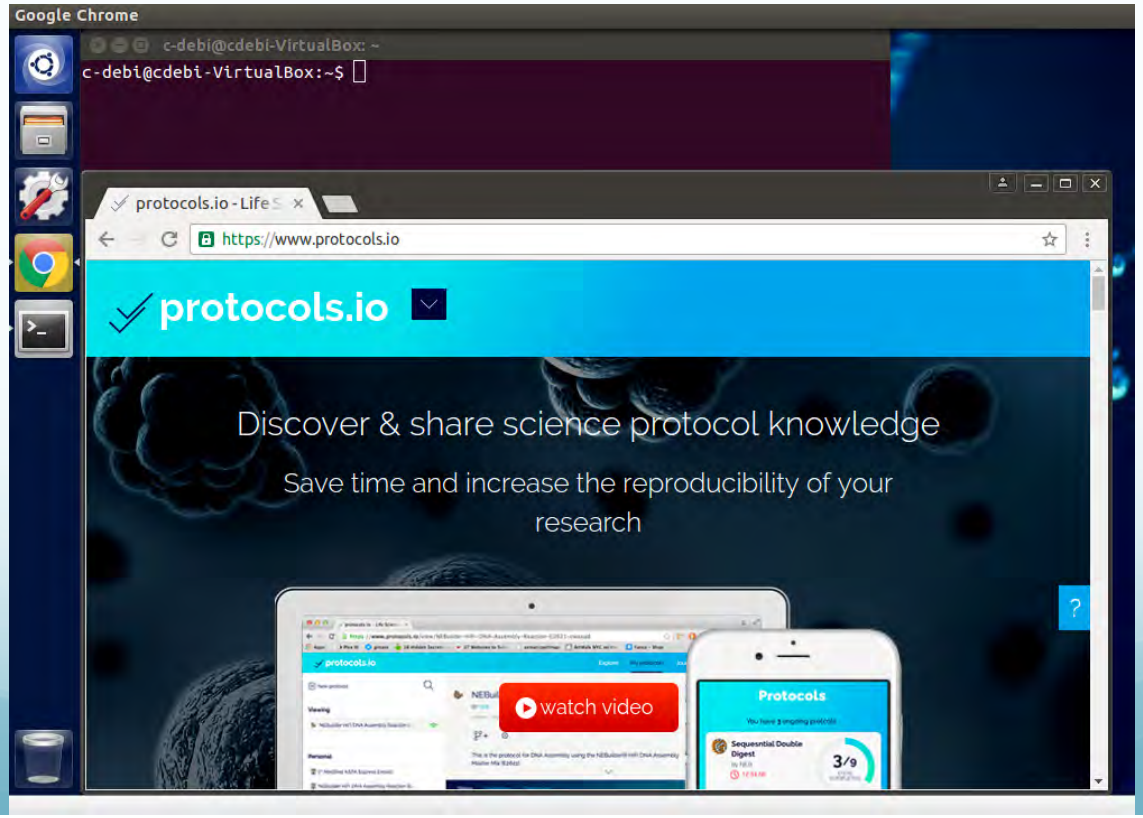
Unix Command Line

1. Open Terminal window



Unix Command Line

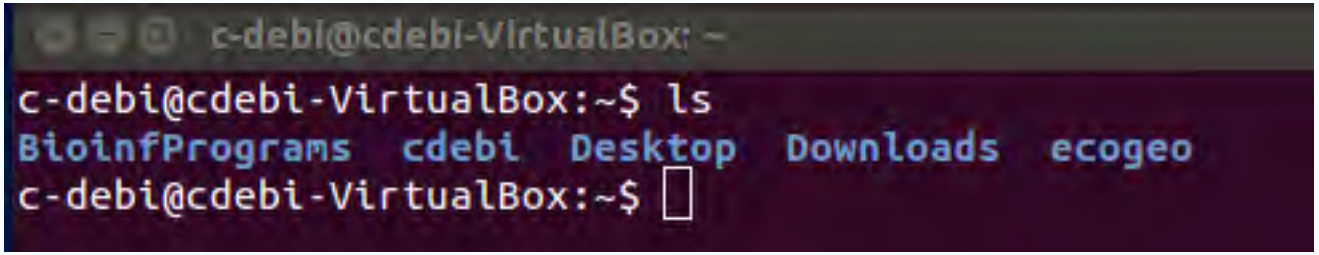
2. Open Chrome and navigate to Unix tutorial at Protocols.io
3. Group: ECOGEO
4. Protocol: ECOGEO Workshop 2: Unix Module
   - This will allow you to copy, paste Unix scripts into terminal window
   - ECOGEO Protocols.io for making copy, paste easier



Unix Command Line

```
$ ls
```

ls - lists items in the current directory



Many commands have additional options that can be set by a '-'

```
$ ls -a
```

Unix Command Line

```
$ ls -a
```

lists all files/directories, including hidden files ‘.’

```
$ ls -l
```

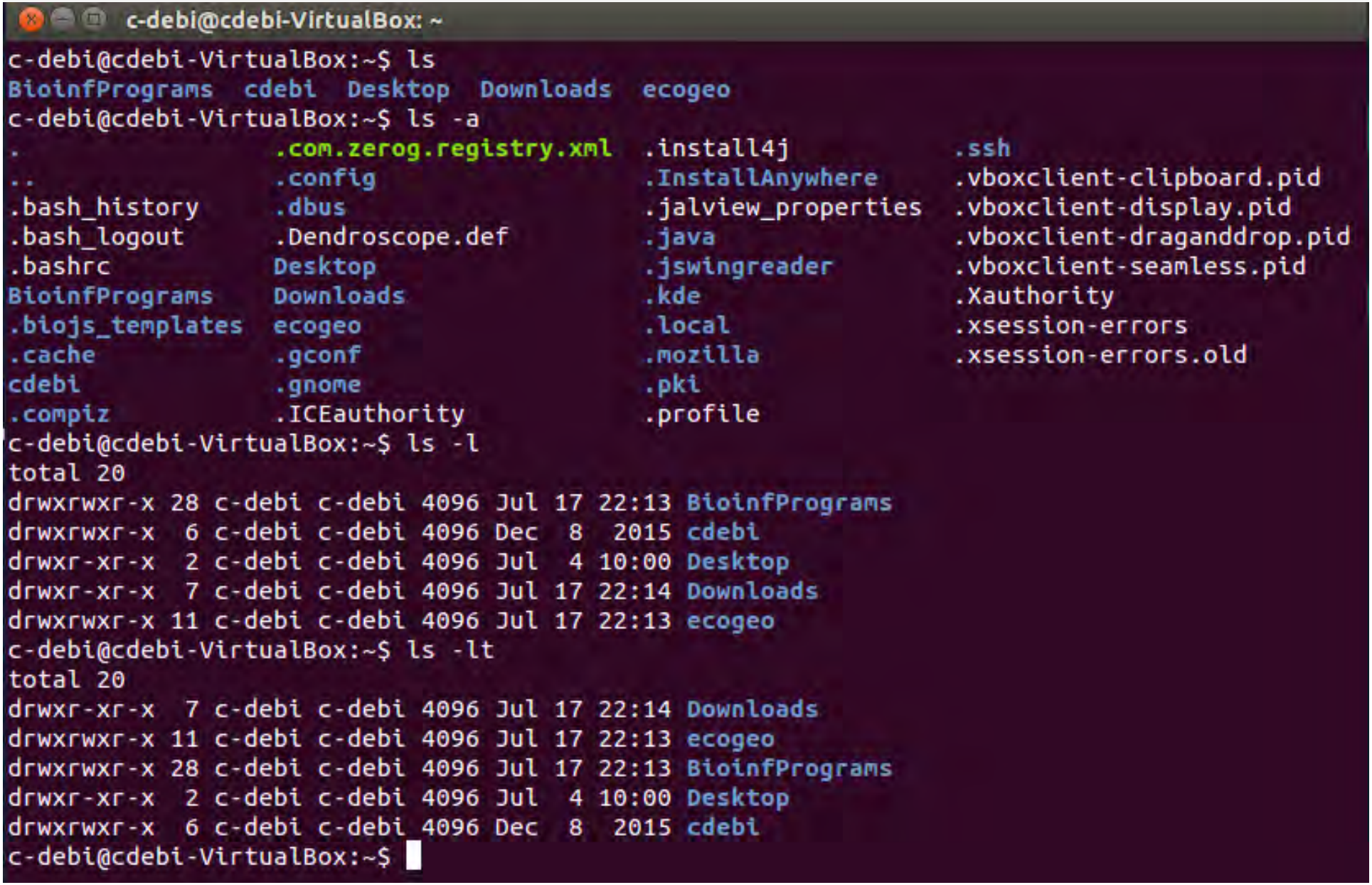
lists the long format

```
File Permissions | # Link | User | Group | Size | Last modified
```

```
$ ls -lt
```

lists the long format, but ordered by date last modified

Unix Command Line



Unix Command Line

```
$ cd ecogeo/
```

cd - change directory

List the contents of the current directory
Move into the directory called **unix**
List contents

```
$ pwd
```

pwd - present working directory

Unix Command Line

```
/home/c-debi/ecogeo/unix
```

When were we in the directory **home**?
Or **c-debi**?
Or **ecogeo**?

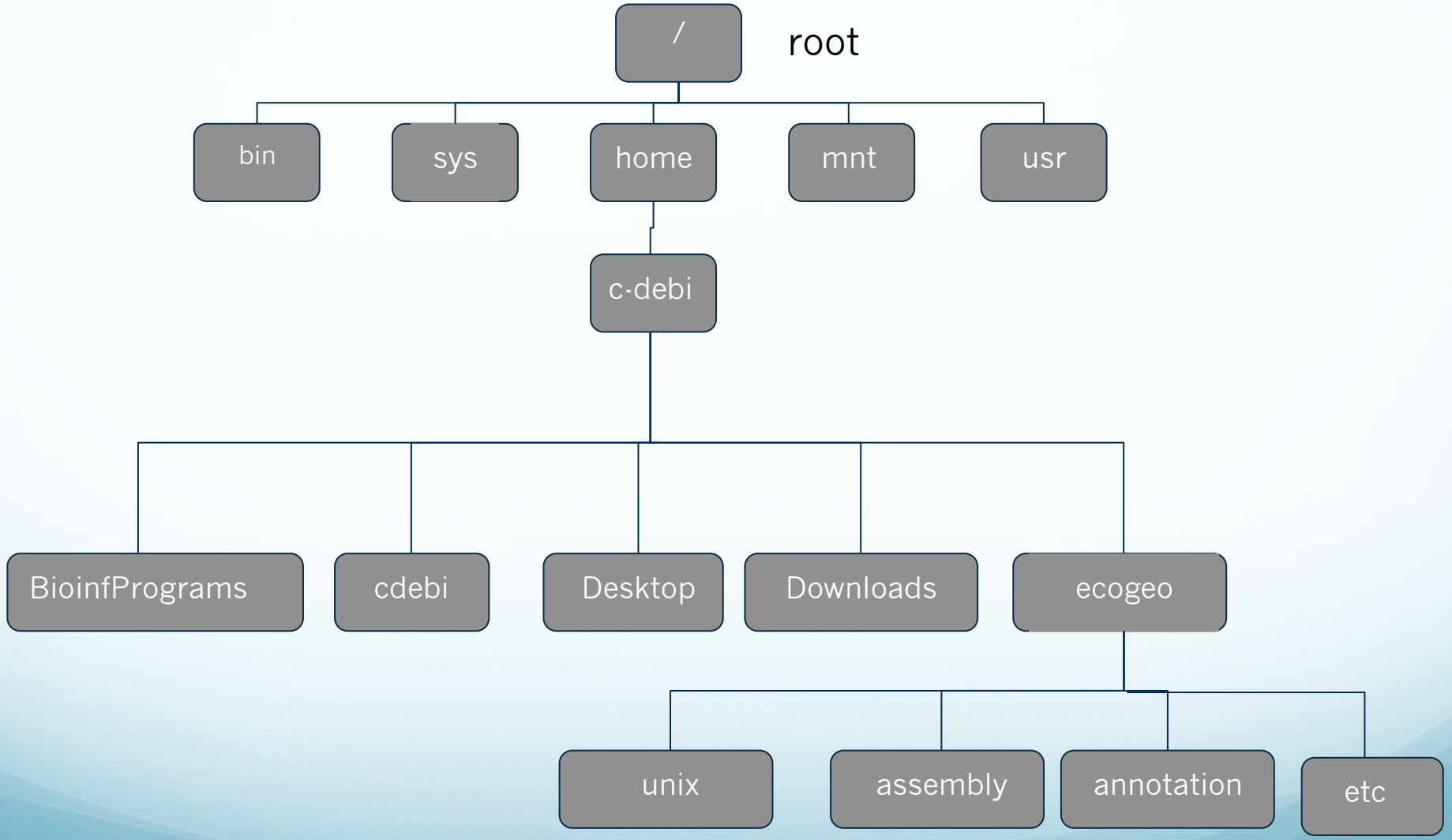
```
$ cd /
```

Navigates to **root** directory

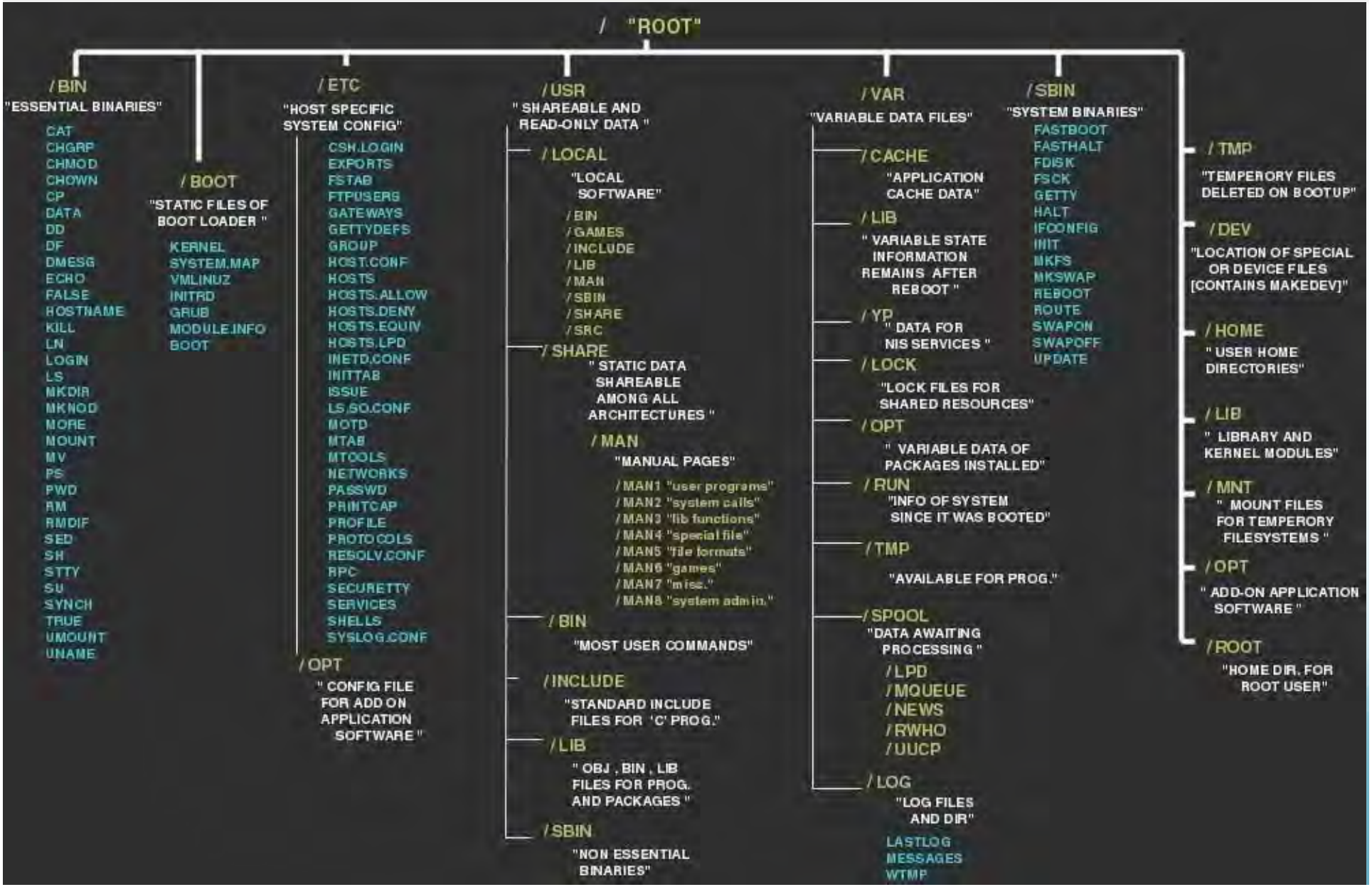
List contents of root directory

This where everything is stored in the computer
All the commands we are running live in **/bin**

Unix Command Line



Typical Unix Layout



Unix Command Line

Change directory to **home**
Change directory to **c-debi**
Change directory to **ecogeo**
Change directory to **unix**
List contents
Change directory to **data**
Change directory to **root**

Unix Command Line

Change directory to **unix/data** in one step

```
$ cd /home/c-debi/ecogeo/unix/data
```

Tab can be used to auto complete names

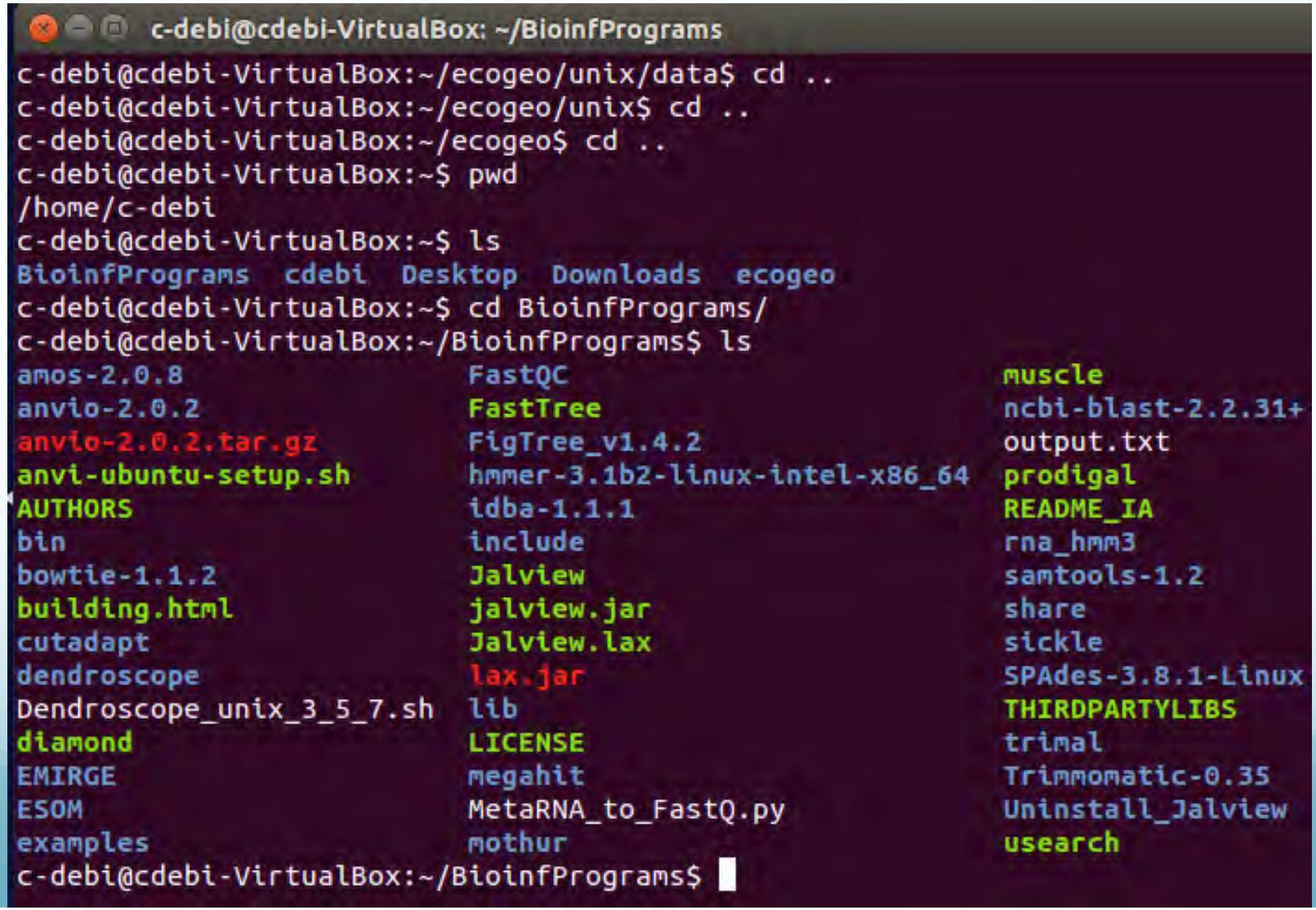
```
$ cd ..
```

cd '..' allows you step back up through the path directory

Display present working directory path
Step back up to the **c-debi** directory
Change directory to **BioinfPrograms**
List contents

Unix Command Line

List of all bioinformatic tools available



Unix Command Line

Change directory back to **unix/**

```
$ mkdir
```

mkdir - make directory

```
$ mkdir storage
```

List contents of directory
Change directory to **storage**

Unix Command Line

```
$ touch temp.txt
```

touch - creates a blank file of the input name

```
$ cp
```

cp - copy

```
$ mv
```

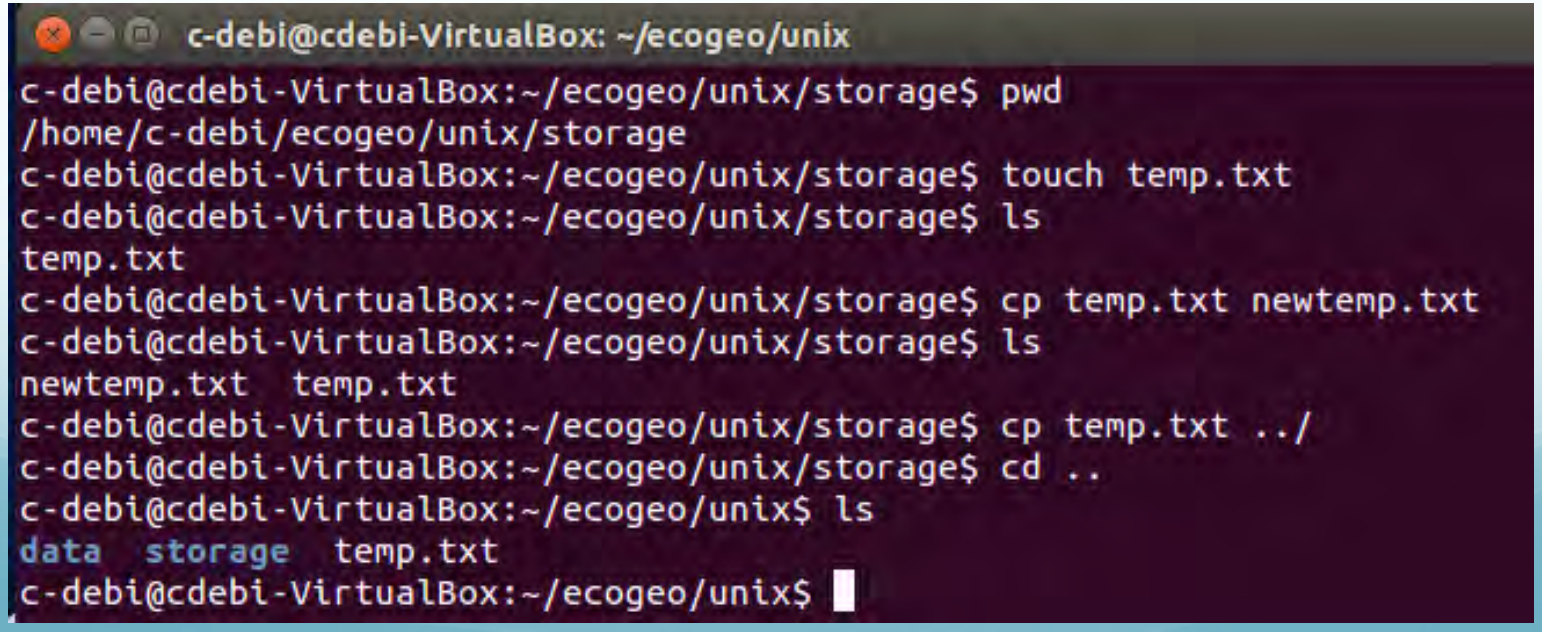
mv - move

Unix Command Line

```
$ cp temp.txt newtemp.txt
```

```
$ cp temp.txt ../
```

Change directory up a level
List contents



Unix Command Line

Change directory to storage

```
$ mv newtemp.txt oldtemp.txt
```

```
$ mv oldtemp.txt /home/c-debi/ecogeo/unix/data
```

Alternative command?
Change directory to data
List content

cp - will make a copy of a file and can be used to move a copy of a file(s) to a directory
mv - “destroys” the original and places the contents elsewhere

Unix Command Line

List current working directory

```
/home/c-debi/ecogeo/unix/data
```

List contents

```
$ rm
```

rm - removes a file PERMANENTLY

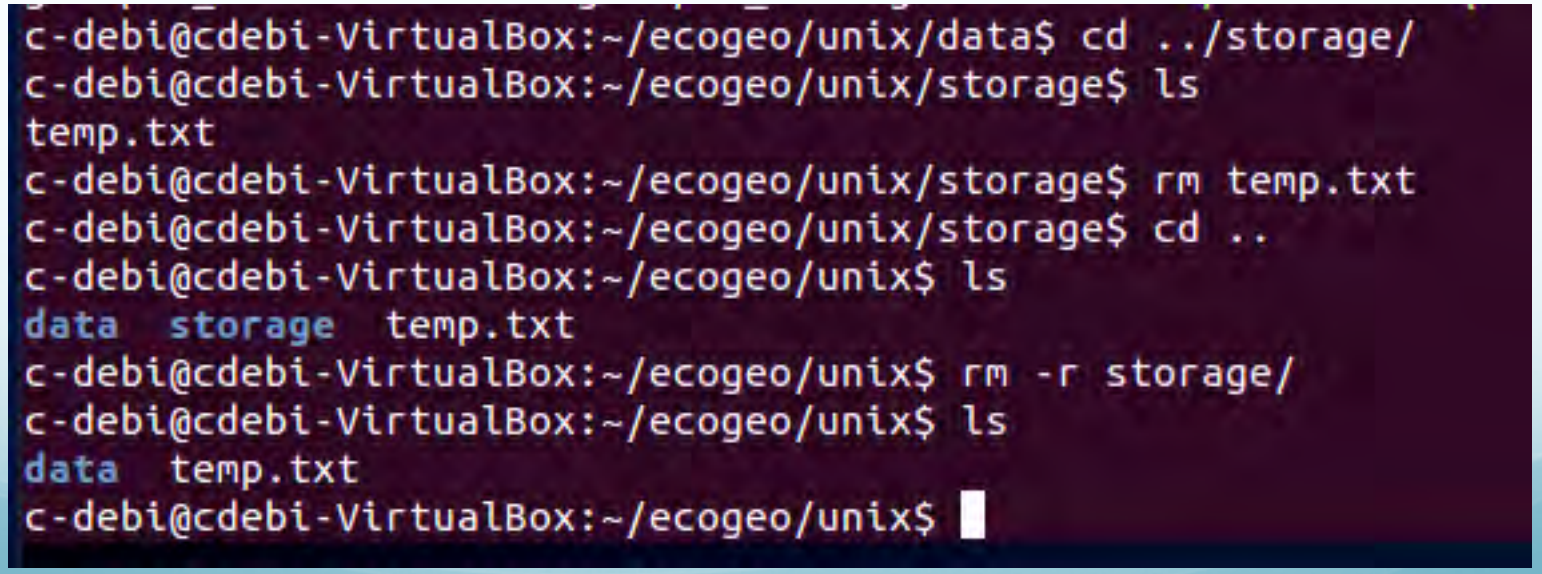
```
$ rm oldtemp.txt
```

List contents

Unix Command Line

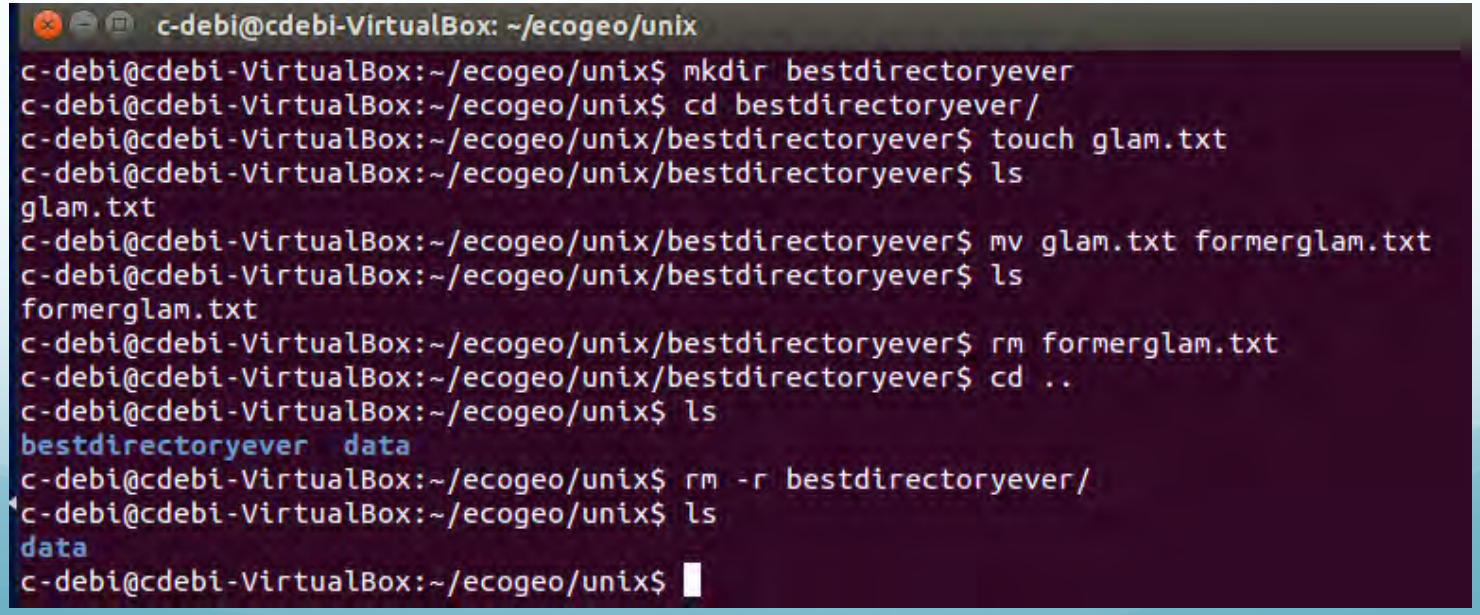
Change directory to **storage**
Remove **temp.txt**
Change directory to **unix**

```
$ rm -r storage
```



Unix Command Line

Create a directory called **bestdirectoryever**
Change directory to **bestdirectoryever**
Create a file called **glam.txt**
Change **glam.txt** to **formerglam.txt**
Remove **formerglam.txt**
Change directory to **unix**
Remove **bestdirectoryever**



Unix Command Line

Change directory to **data**
List contents
Remove **oldtemp.txt**

```
group12_contigs.fasta    group20_contigs.fasta    group24_contigs.fasta
```

FASTA files - specific format
> Header line, contains ID and information about…
ATGATAGCTAGCAGCAGCTA[...]80bp and then a newline

Unix Command Line

```
$ head [filename]
```

head - default displays the first 10 lines

```
$ tail [filename]
```

tail - default displays the last 10 lines

```
$ less [filename]
```

less - scroll through a file using arrow keys or
spacebar = advance page | b = reverse page | q = quit

Unix Command Line

Use head to display the first 10 lines of
group12_contigs.fasta
Display the first 5 lines of **group12_contigs.fasta**
Display the last 10 lines of **group12_contigs.fasta**
Display the last 5 lines of **group12_contigs.fasta**

Unix Command Line

```
$ grep
```

grep - file pattern searcher

```
$ wc
```

wc - count the number of words, lines, characters

Unix Command Line

```
$ grep “>” group12_contigs.fasta
```

Prints all matches of “>” in the file

How many? Combine grep and wc.
Use the “|” (pipe) symbol

```
$ grep “>” group12_contigs.fasta | wc
```

Repeat but at the option -l to wc

Unix Command Line

Use the same technique to determine the number of sequences in **group20_contigs.fasta**

What about the number of matches to “47” in **group12_contigs.fasta**?

Or “_47”?

Unix Command Line

```
$ grep “>” group12_contigs.fasta > group12_ids
```

> - redirects the output to a file
Look at the contents of **group12_ids**

```
$ grep “47” group12_contigs.fasta > group12_ids_with_47
```

Unix Command Line

```
$ cat group12_ids_with_47
```

cat - has multiple functions

- With a single input - prints file contents
- With '>' - has the same function as cp

```
$ cat group12_ids_with_47 > temp1_ids
```

```
$ cp group12_ids_with_47 temp2_ids
```

Double check to make sure **temp1_ids** = **temp2_ids**

Unix Command Line

cat - most important function

- Concatenate files

```
$ cat temp1_ids temp2_ids > duplicate_ids
```

Check contents of duplicate_ids using less or cat

Grab all of the contigs IDs from **group20_contigs.fasta** that contain the number “51”

Concatenate the new IDs to the duplicate_ids file in a file called **multiple_ids**

Unix Command Line

```
$ uniq
```

uniq - can be used to remove duplicates or identify lines with 1 occurrence or multiple occurrences

```
$ sort
```

sort - sort lines in a file alphanumerically

Unix Command Line

```
$ uniq multiple_ids
```

Compare **multiple_ids** before and after uniq

Why was there no change?

uniq has a weakness, can only identify duplicates in adjacent lines

```
$ sort multiple_ids | uniq > clean_ids
```

**note the version of sorting used by Unix

Unix Command Line

Clear all present files with temp in title

```
$ rm temp*
```

‘*’ - acts as a wildcard, so any file that starts with temp would be identified and removed, no matter the suffix

```
$ sort multiple_ids | uniq -d > temp1_ids
```

```
$ sort multiple_ids | uniq -u > temp2_ids
```

How do **temp1_ids** & **temp2_ids** compare?

Unix Command Line

```
$ sort multiple_ids | uniq -d > temp1_ids
```

Uniq -d identifies only duplicates

```
$ sort multiple_ids | uniq -u > temp2_ids
```

Uniq -u identifies only unique entries

temp1_ids = group12_ids_with_47 &
temp2_ids = group20_ids_with_51

Unix Command Line

Remove all present files with temp in title

```
$ sed
```

Sed - modify a file based on the issued commands
Want a list of sequence IDs without the '>'?

```
$ sed 's/C/c/' clean_ids
```

Between the single quotes, **s**ubstitute the occurrence of upper case C to lower case c

```
$ sed 's/_/./' clean_ids
```

```
$ sed 's/>//' clean_ids > newclean_ids
```

seqmagick

Wrapper designed to utilize built in Biopython modules to manipulate and change FASTA files

Requires Biopython

http://fhcrc.github.io/seqmagick/

Seqmagick

```
$ seqmagick
```

Discuss:

- convert - produce a modified new file
- mogrify - change the input file
- info - present information of files in a directory

Additionally: backtrans-align, extract-ids, quality-filter, and primer-trim

Seqmagick

```
$ seqmagick convert --include-from-file newclean_ids
group12_contigs.fasta newgroup12_contigs.fasta
```

How many sequences are in **newgroup12_contigs.fasta**?
Using grep '>'

```
$ seqmagick extract-ids newgroup12_contigs.fasta | wc -l
```

```
$ seqmagick info *fasta
```

```
c-debi@cdebi-VirtualBox:~/ecogeo/unix/data$ seqmagick extract-ids newgroup12_contigs.fasta | wc -l
7
c-debi@cdebi-VirtualBox:~/ecogeo/unix/data$ seqmagick info *fasta
name                     alignment    min_len   max_len   avg_len  num_seqs
group12_contigs.fasta    FALSE           5136    116409  22974.30       132
group20_contigs.fasta    FALSE           5029     22601   7624.38       203
group24_contigs.fasta    FALSE           5024     81329  12115.70       139
newgroup12_contigs.fasta FALSE           5587     30751  16768.14         7
c-debi@cdebi-VirtualBox:~/ecogeo/unix/data$
```

Seqmagick

Store the information generated by ‘seqmagick info’ in a new file **fasta_info**

```
$ cut
```

cut - pulling out columns from a table file

Seqmagick

```
$ cut -f 2 fasta_info
```

```
$ cut -f 2,4 fasta_info
```

```
$ cut -f 2-4 fasta_info
```

-d allows for the assignment of the type of delimiter between fields, if not TAB

-f delineates which fields to preserve, starting at 1

Unix Command Line

Some additional tools

history - prints a sequential list of all commands in the current session

echo $PATH - lists the directories for which the OS is checking for commands and data

nano - in window text editor

Unix Command Line

```
$ nano fasta_info
```

Additional text can be entered like any text editor

To close out - Ctrl+X, hit ‘Y’, then ENTER

Create a new file – nano, then enter file name after Ctrl+X

Unix Command Line: Bash Scipts

Text file with a list of commands that can be executed as a batch

Look at the contents of **simplebashscript**

```
$ chmod 775 simplebashscript
```

chmod - change file modes
Plain text file -> executable text file

```
$ ./simplebashscript
```

Executes simplebashscript

FIND A SERVER OR HPC!

Most bioinformatic research is going to require more power and time then is available on a laptop

1. Buy a high-end server or lab computer
2. Collaborate with a lab or group who have computer space to share
3. Contact and work with your university/college High Performance Computing centers