ECOGEO 'Omics Training: 1.0 Unix and Bioinformatics Version 4

Benjamin Tully and Ken Youens-Clark

Abstract

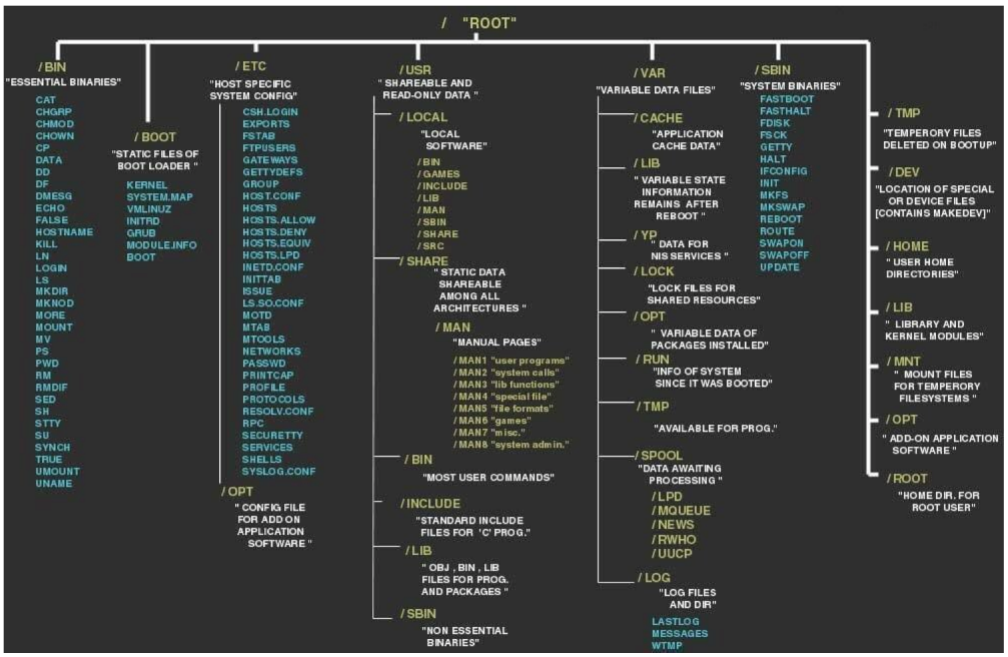
This protocol details the use of various unix commands commonly used in bioinformatics. Open this protocol inside the virtual machine (details in "Start Instructions") for easy copy, paste of commands into the command line terminal window.

Citation: Benjamin Tully and Ken Youens-Clark ECOGEO 'Omics Training: 1.0 Unix and Bioinformatics. **protocols.io** dx.doi.org/10.17504/protocols.io.fiubkew
Published: 18 Aug 2016

Guidelines

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	



Before start

Before starting, please visit the ECOGEO website for more information on this 'Introduction to Environmental 'Omics' training series. The site contains a pre-packaged virtual machine that can be downloaded and used to run all of the protocols in this protocols.io collection. In addition to the VM, the website contains video and presentations from our initial 'Intro to Env 'Omics' workshop held at the Univ. of Hawai'i at Manoa on 25-26 Jul 2016.

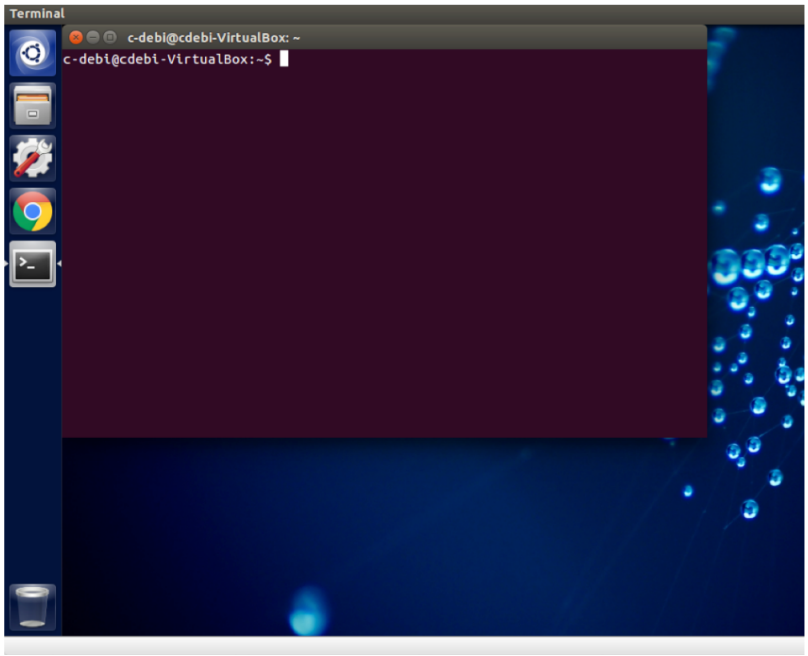
Please email ‘ecogeo-join@earthcube.org’ to join the ECOGEO listserv for future updates.

Protocol

The Start

Step 1.

Open terminal window



The Start

Step 2.

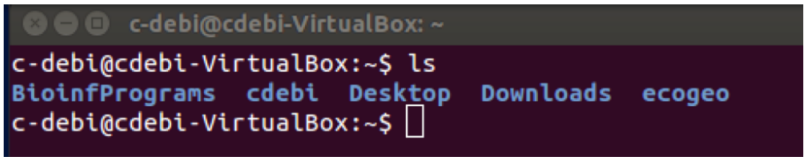
Use ls to list items in the current directory.

cmd COMMAND

```
ls
```

lists items in the current directory

EXPECTED RESULTS



The Start

Step 3.

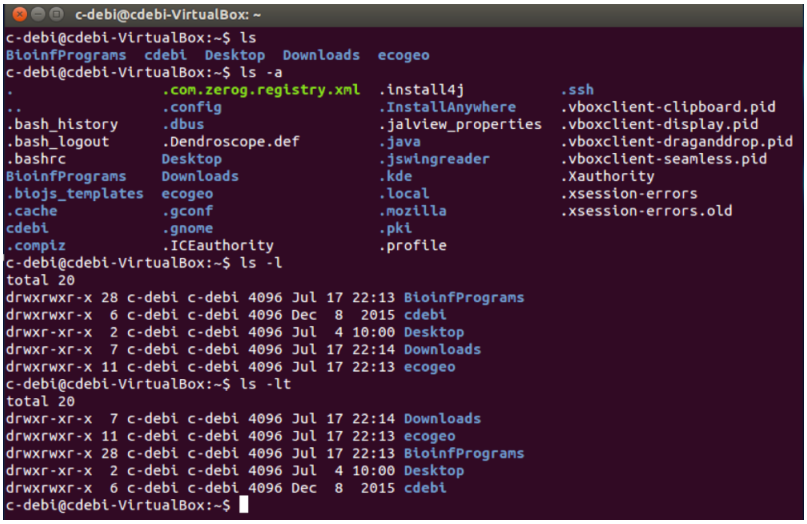
Many commands have additional options that can be set by a '-'

cmd COMMAND

```
ls -a
ls -l
ls -lt
```

lists all files/directories, including hidden files '.' lists the long format lists the long format, but ordered by date last modified

EXPECTED RESULTS



Directory System

Step 4.

cd - change directory

cmd COMMAND

```
cd ecogeo/
```

Directory System

Step 5.

pwd (present working directory) can be used to show the current directory.

cmd COMMAND

pwd

prints the path to the current directory

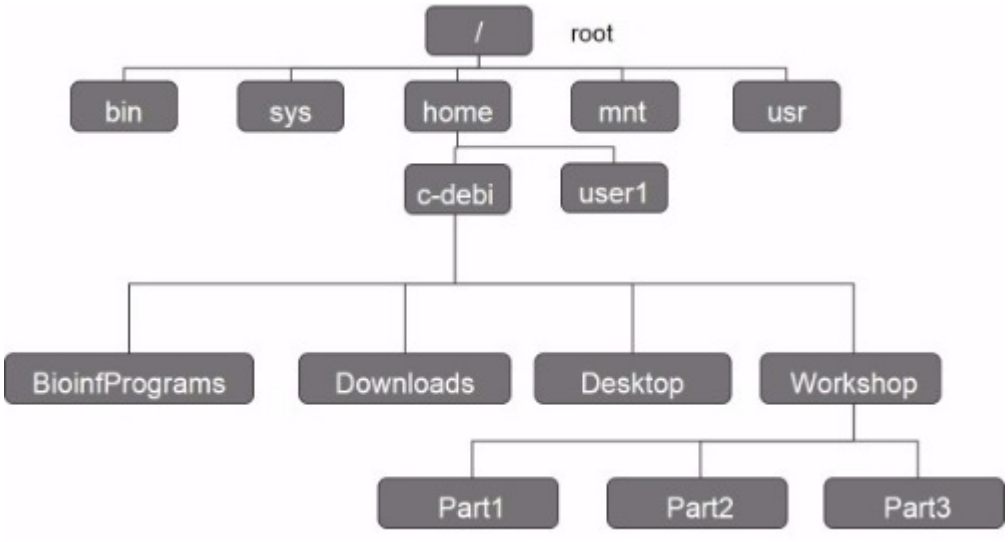
EXPECTED RESULTS

cd /home/c-debi/ecogeo/unix

Directory System

Step 6.

Move to the root directory.



cmd COMMAND

```
cd /
```

NOTES

Elisha Wood-Charlson 08 Aug 2016

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

Directory System

Step 7.

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**

NOTES

Elisha Wood-Charlson 08 Aug 2016

Tabs can be used to auto complete names.

Directory System

Step 8.

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

cmd COMMAND

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

Directory System

Step 9.

cd '..' allows you to step back up through the path directory. Display present working directory path.

cmd COMMAND

```
cd ..
pwd
```

moves back in the path directory

EXPECTED RESULTS

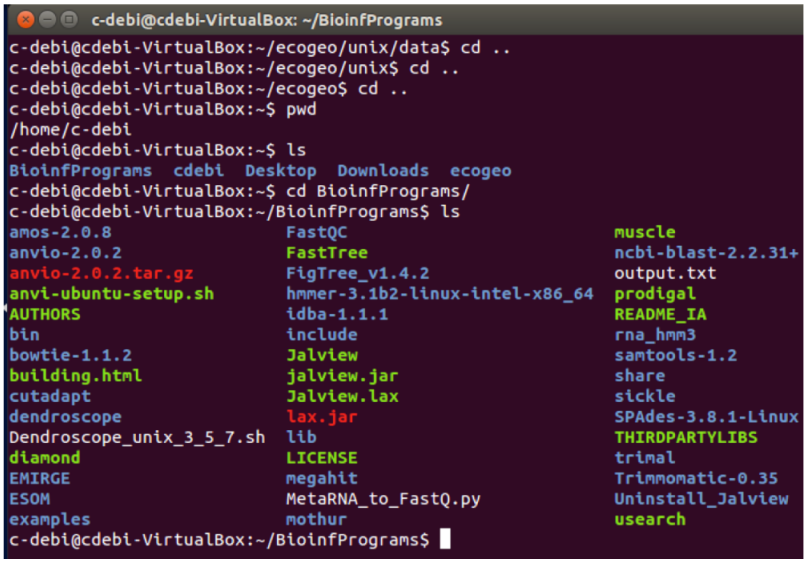
/home/c-debi/ecogeo/unix

Directory System

Step 10.

List contents of BioinfPrograms

EXPECTED RESULTS



Directory System

Step 11.

Make a directory named "storage".

cmd COMMAND

```
mkdir storage
```

Manipulating files

Step 12.

The 'touch' command allows you to create a blank file of the input name.

cmd COMMAND

```
touch temp.txt
```

creates a blank file of the input name

Manipulating files

Step 13.

The 'cp' command allows you to copy a file and can be used to move a copy of a file to a directory.

cmd COMMAND

```
$ cp
```

Manipulating files

Step 14.

The 'mv' or move command "destroys" the original and places the content elsewhere.

Manipulating files

Step 15.

Using copy:

cmd COMMAND

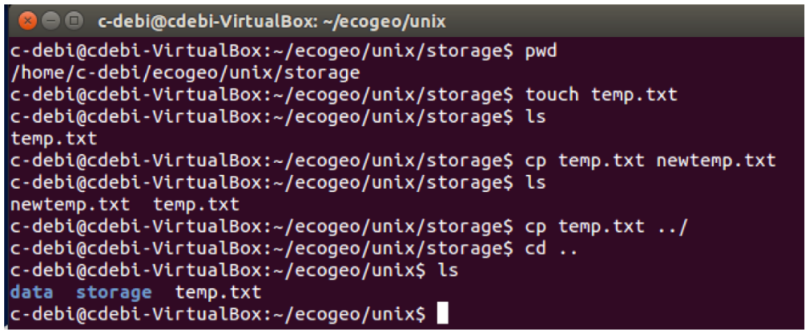
```
$ cp temp.txt newtemp.txt
$ cp temp.txt ../
```

Manipulating files

Step 16.

List contents.

EXPECTED RESULTS



Manipulating files

Step 17.

Utilize move command:

cmd COMMAND

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

Manipulating files

Step 18.

Remove **oldtemp.txt**

cmd COMMAND

```
$ rm oldtemp.txt
```

Manipulating files

Step 19.

Remove storage directory:

cmd COMMAND

```
$ rm -r storage
```

EXPECTED RESULTS

```
c-debi@cdebi-VirtualBox:~/ecogeo/unix/data$ cd ../storage/
c-debi@cdebi-VirtualBox:~/ecogeo/unix/storage$ ls
temp.txt
c-debi@cdebi-VirtualBox:~/ecogeo/unix/storage$ rm temp.txt
c-debi@cdebi-VirtualBox:~/ecogeo/unix/storage$ cd ..
c-debi@cdebi-VirtualBox:~/ecogeo/unix$ ls
data  storage  temp.txt
c-debi@cdebi-VirtualBox:~/ecogeo/unix$ rm -r storage/
c-debi@cdebi-VirtualBox:~/ecogeo/unix$ ls
data  temp.txt
c-debi@cdebi-VirtualBox:~/ecogeo/unix$
```

Manipulating files

Step 20.

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**

EXPECTED RESULTS

```
c-debi@cdebi-VirtualBox: ~/ecogeo/unix
c-debi@cdebi-VirtualBox:~/ecogeo/unix$ mkdir bestdirectoryever
c-debi@cdebi-VirtualBox:~/ecogeo/unix$ cd bestdirectoryever/
c-debi@cdebi-VirtualBox:~/ecogeo/unix/bestdirectoryever$ touch glam.txt
c-debi@cdebi-VirtualBox:~/ecogeo/unix/bestdirectoryever$ ls
glam.txt
c-debi@cdebi-VirtualBox:~/ecogeo/unix/bestdirectoryever$ mv glam.txt formerglam.txt
c-debi@cdebi-VirtualBox:~/ecogeo/unix/bestdirectoryever$ ls
formerglam.txt
c-debi@cdebi-VirtualBox:~/ecogeo/unix/bestdirectoryever$ rm formerglam.txt
c-debi@cdebi-VirtualBox:~/ecogeo/unix/bestdirectoryever$ cd ..
c-debi@cdebi-VirtualBox:~/ecogeo/unix$ ls
bestdirectoryever  data
c-debi@cdebi-VirtualBox:~/ecogeo/unix$ rm -r bestdirectoryever/
c-debi@cdebi-VirtualBox:~/ecogeo/unix$ ls
data
c-debi@cdebi-VirtualBox:~/ecogeo/unix$
```

Looking at the contents of a file

Step 21.

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.

Looking at the contents of a file

Step 22.

'head' will allow you to view the first 10 lines of a file.

cmd COMMAND

```
$ head [filename]
```

default displays the first 10 lines

Looking at the contents of a file

Step 23.

'tail' allows you to view the last 10 lines of a file.

cmd COMMAND

```
$ tail [filename]
```

default displays last 10 lines

Looking at the contents of a file

Step 24.

'less' allows you to scroll through a file using arrow keys or spacebar = advanced page | b = reverse page | q = quit

cmd COMMAND

```
$ less [filename]
```

Looking at the contents of a file

Step 25.

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**

NOTES

Elisha Wood-Charlson 09 Aug 2016

$ head -10 [filename]

$ head -5 [filename]

Looking at the contents of a file

Step 26.

grep - file pattern searcher

cmd COMMAND

```
$ grep
```

Looking at the contents of a file

Step 27.

wc - count the number of words, lines, characters

Looking at the contents of a file

Step 28.

Use grep on group12_contigs.fasta

cmd COMMAND

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

stdout prints all matches of “>” in the file

NOTES

Elisha Wood-Charlson 09 Aug 2016

All quotation marks should work for copy, paste into the VM. Also, you can use single '>' or double quotes ">" for any grep command

Looking at the contents of a file

Step 29.

How many? Combine grep and wc?

Use the "|" (pipe) symbol

cmd COMMAND

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

Looking at the contents of a file

Step 30.

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”?

cmd COMMAND

```
$ grep "47" group12_contigs.fasta
```

NOTES

Elisha Wood-Charlson 08 Aug 2016

Can also write as $ grep '>' group12_contigs.fasta | grep 47

Looking at the contents of a file

Step 31.

Redirecting output to file:

">" = retrieve

> = write to file

cmd COMMAND

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

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

Looking at the contents of a file

Step 32.

cat - has multiple functions:

cmd COMMAND

```
$ cat group12_ids_with_47
```

With a single input - prints file contents

Looking at the contents of a file

Step 33.

With ‘>’ cat has the same function as cp

cmd COMMAND

```
$ cat group12_ids_with_47 > temp1_ids
$ cp group12_ids_with_47 temp2_ids
```

Looking at the contents of a file

Step 34.

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

Looking at the contents of a file

Step 35.

Concatenate files with cat - most important function:

cmd COMMAND

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

Looking at the contents of a file

Step 36.

Check contents of duplicate_ids using less or cat

Looking at the contents of a file

Step 37.

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

cmd COMMAND

```
$ grep "51" group20_contigs.fasta > group20_ids_with_51
```

Looking at the contents of a file

Step 38.

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

cmd COMMAND

```
$ cat duplicate_ids group20_ids_with_51 > multiple_ids
```

Looking at the contents of a file

Step 39.

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

Compare **multiple_ids** before and after uniq

cmd COMMAND

```
$ uniq multiple_ids
```

Looking at the contents of a file

Step 40.

Why was there no change?

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

sort - sort lines in a file alphanumerically

cmd COMMAND

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

**note the version of sorting used by Unix

Looking at the contents of a file

Step 41.

Clear all present files with temp in title

cmd COMMAND

```
$ rm temp*
```

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

Looking at the contents of a file

Step 42.

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

-d = identify duplicates (temp1_ids)

-u = identify unique (temp2_ids)

cmd COMMAND

```
$ sort multiple_ids | uniq -d > temp1_ids
$ sort multiple_ids | uniq -u > temp2_ids
```

Looking at the contents of a file

Step 43.

temp1_ids = **group12_ids_with_47** &

temp2_ids = **group20_ids_with_51**

Looking at the contents of a file

Step 44.

sed - modify files a file based on the issued commands

Want a list of sequence IDs without the '>'?

cmd COMMAND

```
$ sed 's/C/c/' clean_ids
$ sed 's/_/./' clean_ids
$ sed 's/>//' clean_ids > newclean_ids
```

NOTES

Elisha Wood-Charlson 08 Aug 2016

sed 's/C/c/'

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

seqmagick

Step 45.

seqmagick

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

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

Discussed in video:

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

cmd COMMAND

```
$ seqmagick
```

seqmagick

Step 46.

Execute seqmagick convert:

cmd COMMAND

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

seqmagick

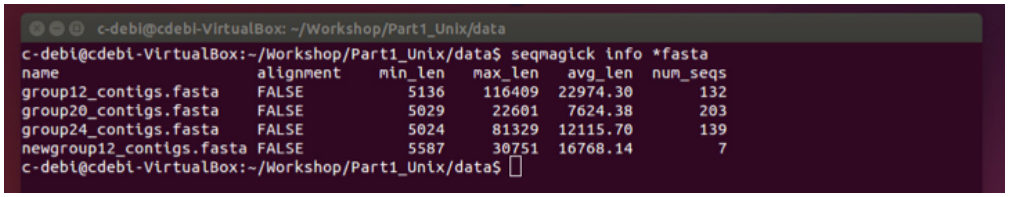
Step 47.

How many sequences are in **newgroup12_contigs.fasta**?

cmd COMMAND

```
$ seqmagick extract-ids newgroup12_contigs.fasta | wc
$ seqmagick info *fasta
```

EXPECTED RESULTS



seqmagick

Step 48.

Store the information generated by 'seqmagick info' in a new file

fasta_info

cmd COMMAND

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

cut - pulling out columns from a table file -d allows for the assignment of the type of delimiter between fields, if not TAB -f delineates which fields to preserve, starting at 1

Some additional tools

Step 49.

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

Some additional tools

Step 50.

nano - in window text editor

cmd COMMAND

```
$ 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 and then enter file name after Ctrl+X

Some additional tools

Step 51.

Simple bash scripts: Text file with a list of commands that can be executed as a batch. Look at the contents of **simplebashscript**

Some additional tools

Step 52.

Simple bash scripts: Text file with a list of commands that can be executed as a batch.

Look at the contents of **simplebashscript**

chmod - change file modes

cmd COMMAND

```
$ chmod 775 simplebashscript
```

Some additional tools

Step 53.

Plain text file -> executable text file.

cmd COMMAND

```
$ ./simplebashscript
```