



Lab introduction

Gemma Martínez Redondo & Karin Steffen

Many slides taken from Dr. Sophie Shaw

Daily workshop material, slides & exercises

<http://evomics.org/2024-workshop-on-phylogenomics-cesky-krumlov/>

← → ↺

🛡️ 🔗 🌐 evomics.org

☆ 🔍 Search

📧 📄 📅 📁 📌 📌 📌 📌 📌

EVOLUTION AND GENOMICS

Intensive and immersive training opportunities

WORKSHOPS

LEARNING

PEOPLE

APPLY

INFORMATION

2024 Workshop on Phylogenomics, Cesky Krumlov

2019 Workshop on Phylogenomics, Cesky Krumlov

2017 Workshop on Phylogenomics, Cesky Krumlov

Workshop on Genomics

Workshop on Population and Speciation Genomics

Workshop on Phylogenomics

Workshop on Molecular Evolution

Harvard University Workshops

Workshop on Microbiome and Transcriptome Analysis, Durban, South Africa

Advanced Topics

WORK

Immersive training opportunities in modern computational

UPCOMING WORKSHOPS

NEWS & ANNOUNCEMENTS

Daily workshop material, slides & exercises

EVOLUTION AND GENOMICS

Intensive and immersive training opportunities

WORKSHOPSLEARNINGPEOPLEAPPLYINFORMATION

Important information:


- Group photo!
- [Online booking](#) of your free time activities! (also available at the Info-Centre)

FIND YOUR WAY AROUND ČESKÝ KRUMLOV

- [Český Krumlov town information](#)
- [Map](#) of the 2024 changes in Český Krumlov due to the bridge reparation – **Check before arrival to choose the bus stop closest to your accomodation!!**
- [Map](#) of the **main locations in the schedule** (Save to your Google Maps!)
- [Map](#) of some of the main Český Krumlov **restaurants** (Save to your Google Maps!)
- [Map](#) of some of the main Český Krumlov **bars** (Save to your Google Maps!)
- [Map](#) of Český Krumlov **service points**: shops, pharmacy, ATM, hospital (Save to your Google Maps!)
- [Map](#) of the historic City Centre of Český Krumlov

FACULTY AND GENOMICS

- Click here to see our Faculty arrival and departure dates
- Faculty [Bios](#) and Teaching Assistant [Bios](#)
- **Faculty Lunches**: sign-up sheet
- Instance IPs



Daily workshop material, slides & exercises

EVOLUTION AND GENOMICS

Intensive and immersive training opportunities

WORKSHOPSLEARNINGPEOPLEAPPLYINFORMATION

- **Faculty Lunches:** sign-up sheet
- Instance IPs
- [Cheatsheet](#) – copying files from/to the instances

COMPETITIONS

- As a tradition we have a **T-shirt design competition!** Best T-shirt design will feature on the Workshop on Phylogenomics 2024 T-shirts! [Submit your design](#) by Friday 12th at the latest! (For inspiration, check out previous designs by our sister workshop [here!](#))
- BINGO!

CODE OF CONDUCT

- [Code of conduct](#), contact people Karin and Jacob

Week 1 : 21-27 January, 2024

DATE	DAY	TIME	PRESENTER	TOPIC	LOCATION
Jan 21	Sunday	18 – 22	Everyone	Reception	Hotel Zlaty Andel
Jan 22	Monday	09 – 12	Anna Karnkowska	Introduction & Orientation, City Information	Town Theatre
	Monday	14 – 17	Workshop Team	Lab introduction	House of Prelate
	Monday	19 – 22	Everyone	Scientific speed networking	Krumlov mill
Jan 23	Tuesday	09 – 12	Rosa Fernández	Introduction to Phylogenomics	Town Theatre
	Tuesday	14 – 17	Workshop Team	Alignment and Alignment Trimming	House of Prelate

Good workshop practice

- Work together
- Ask lots of questions
- Take breaks
- Use cheat sheets
- Have lots of fun



Cheat sheets

Unix/Linux Command Reference

File Commands	System Info
ls – directory listing ls -al – formatted listing with hidden files cd <i>dir</i> – change directory to <i>dir</i> cd – change to home pwd – show current directory mkdir <i>dir</i> – create a directory <i>dir</i> rm <i>file</i> – delete <i>file</i> rm -r <i>dir</i> – delete directory <i>dir</i> rm -f <i>file</i> – force remove <i>file</i> rm -rf <i>dir</i> – force remove directory <i>dir</i> * cp <i>file1 file2</i> – copy <i>file1</i> to <i>file2</i> cp -r <i>dir1 dir2</i> – copy <i>dir1</i> to <i>dir2</i> ; create <i>dir2</i> if it doesn't exist mv <i>file1 file2</i> – rename or move <i>file1</i> to <i>file2</i> if <i>file2</i> is an existing directory, moves <i>file1</i> into directory <i>file2</i> ln -s <i>file link</i> – create symbolic link <i>link</i> to <i>file</i> touch <i>file</i> – create or update <i>file</i> cat > <i>file</i> – places standard input into <i>file</i> more <i>file</i> – output the contents of <i>file</i> head <i>file</i> – output the first 10 lines of <i>file</i> tail <i>file</i> – output the last 10 lines of <i>file</i> tail -f <i>file</i> – output the contents of <i>file</i> as it grows, starting with the last 10 lines	date – show the current date and time cal – show this month's calendar uptime – show current uptime w – display who is online whoami – who you are logged in as finger <i>user</i> – display information about <i>user</i> uname -a – show kernel information cat /proc/cpuinfo – cpu information cat /proc/meminfo – memory information man <i>command</i> – show the manual for <i>command</i> df – show disk usage du – show directory space usage free – show memory and swap usage whereis <i>app</i> – show possible locations of <i>app</i> which <i>app</i> – show which <i>app</i> will be run by default
Process Management	Compression
ps – display your currently active processes top – display all running processes kill <i>pid</i> – kill process id <i>pid</i> killall <i>proc</i> – kill all processes named <i>proc</i> * bg – lists stopped or background jobs; resume a stopped job in the background fg – brings the most recent job to foreground fg <i>n</i> – brings job <i>n</i> to the foreground	tar cf <i>file.tar files</i> – create a tar named <i>file.tar</i> containing <i>files</i> tar xf <i>file.tar</i> – extract the files from <i>file.tar</i> tar czf <i>file.tar.gz files</i> – create a tar with Gzip compression tar xzf <i>file.tar.gz</i> – extract a tar using Gzip tar cjf <i>file.tar.bz2</i> – create a tar with Bzip2 compression tar xjf <i>file.tar.bz2</i> – extract a tar using Bzip2 gzip <i>file</i> – compresses <i>file</i> and renames it to <i>file.gz</i> gzip -d <i>file.gz</i> – decompresses <i>file.gz</i> back to <i>file</i>
File Permissions	Network
chmod <i>octal file</i> – change the permissions of <i>file</i> to <i>octal</i> , which can be found separately for user, group, and world by adding: <ul style="list-style-type: none">4 – read (r)2 – write (w)1 – execute (x) Examples: chmod 777 – read, write, execute for all chmod 755 – rwx for owner, rx for group and world For more options, see man chmod .	ping <i>host</i> – ping <i>host</i> and output results whois <i>domain</i> – get whois information for <i>domain</i> dig <i>domain</i> – get DNS information for <i>domain</i> dig -x <i>host</i> – reverse lookup <i>host</i> wget <i>file</i> – download <i>file</i> wget -c <i>file</i> – continue a stopped download
SSH	Installation
ssh <i>user@host</i> – connect to <i>host</i> as <i>user</i> ssh -p <i>port user@host</i> – connect to <i>host</i> on port <i>port</i> as <i>user</i> ssh-copy-id <i>user@host</i> – add your key to <i>host</i> for <i>user</i> to enable a keyed or passwordless login	Install from source: ./configure make make install dpkg -i <i>pkg.deb</i> – install a package (Debian) rpm -Uvh <i>pkg.rpm</i> – install a package (RPM)
Searching	Shortcuts
grep <i>pattern files</i> – search for <i>pattern</i> in <i>files</i> grep -r <i>pattern dir</i> – search recursively for <i>pattern</i> in <i>dir</i> <i>command</i> grep <i>pattern</i> – search for <i>pattern</i> in the output of <i>command</i> locate <i>file</i> – find all instances of <i>file</i>	Ctrl+C – halts the current command Ctrl+Z – stops the current command, resume with fg in the foreground or bg in the background Ctrl+D – log out of current session, similar to exit Ctrl+W – erases one word in the current line Ctrl+U – erases the whole line Ctrl+R – type to bring up a recent command !! – repeats the last command exit – log out of current session
	* use with extreme caution.

Cheat sheets



Stack overflow, ChatGTP,
Bard (Google).

Unix/Linux Command Reference

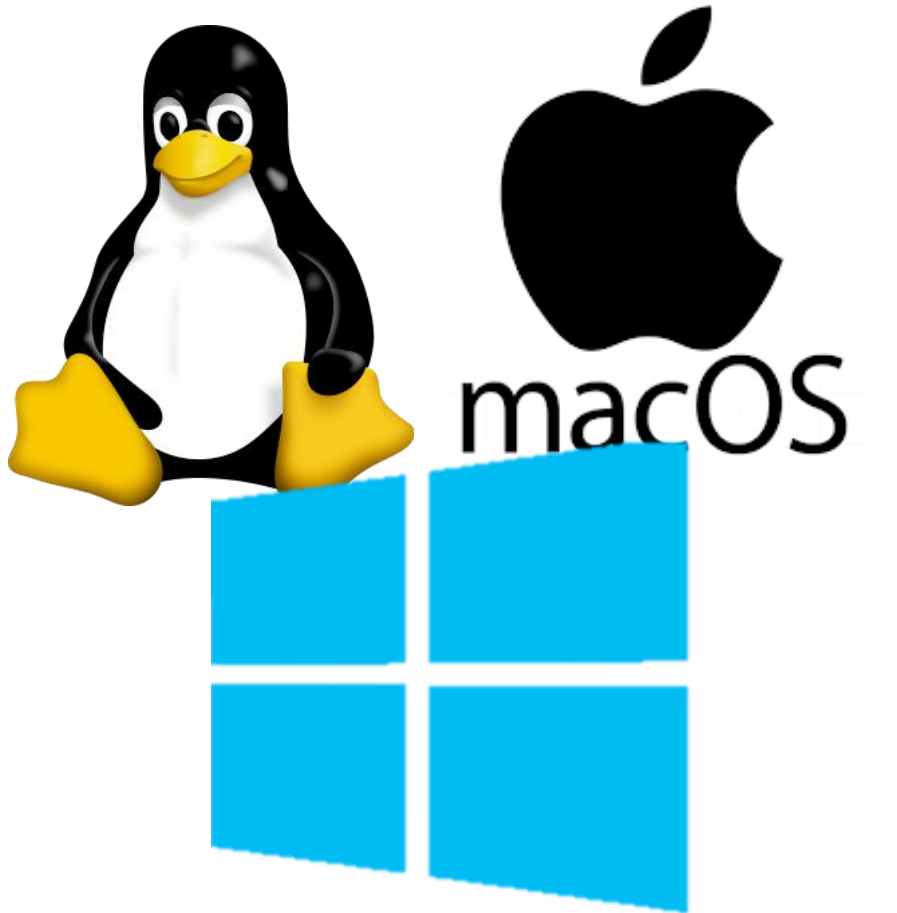
FOSSwire.com

File Commands	System Info
ls – directory listing ls -al – formatted listing with hidden files cd <i>dir</i> – change directory to <i>dir</i> cd – change to home pwd – show current directory mkdir <i>dir</i> – create a directory <i>dir</i> rm <i>file</i> – delete <i>file</i> rm -r <i>dir</i> – delete directory <i>dir</i> rm -f <i>file</i> – force remove <i>file</i> rm -rf <i>dir</i> – force remove directory <i>dir</i> * cp <i>file1 file2</i> – copy <i>file1</i> to <i>file2</i> cp -r <i>dir1 dir2</i> – copy <i>dir1</i> to <i>dir2</i> ; create <i>dir2</i> if it doesn't exist mv <i>file1 file2</i> – rename or move <i>file1</i> to <i>file2</i> if <i>file2</i> is an existing directory, moves <i>file1</i> into directory <i>file2</i> ln -s <i>file link</i> – create symbolic link <i>link</i> to <i>file</i> touch <i>file</i> – create or update <i>file</i> cat > <i>file</i> – places standard input into <i>file</i> more <i>file</i> – output the contents of <i>file</i> head <i>file</i> – output the first 10 lines of <i>file</i> tail <i>file</i> – output the last 10 lines of <i>file</i> tail -f <i>file</i> – output the contents of <i>file</i> as it grows, starting with the last 10 lines	date – show the current date and time cal – show this month's calendar uptime – show current uptime w – display who is online whoami – who you are logged in as finger <i>user</i> – display information about <i>user</i> uname -a – show kernel information cat /proc/cpuinfo – cpu information cat /proc/meminfo – memory information man <i>command</i> – show the manual for <i>command</i> df – show disk usage du – show directory space usage free – show memory and swap usage whereis <i>app</i> – show possible locations of <i>app</i> which <i>app</i> – show which <i>app</i> will be run by default
Process Management	Compression
ps – display your currently active processes top – display all running processes kill <i>pid</i> – kill process id <i>pid</i> killall <i>proc</i> – kill all processes named <i>proc</i> * bg – lists stopped or background jobs; resume a stopped job in the background fg – brings the most recent job to foreground fg <i>n</i> – brings job <i>n</i> to the foreground	tar cf <i>file.tar files</i> – create a tar named <i>file.tar</i> containing <i>files</i> tar xf <i>file.tar</i> – extract the files from <i>file.tar</i> tar czf <i>file.tar.gz files</i> – create a tar with Gzip compression tar xzf <i>file.tar.gz</i> – extract a tar using Gzip tar cjf <i>file.tar.bz2</i> – create a tar with Bzip2 compression tar xjf <i>file.tar.bz2</i> – extract a tar using Bzip2 gzip <i>file</i> – compresses <i>file</i> and renames it to <i>file.gz</i> gzip -d <i>file.gz</i> – decompresses <i>file.gz</i> back to <i>file</i>
File Permissions	Network
chmod <i>octal file</i> – change the permissions of <i>file</i> to <i>octal</i> , which can be found separately for user, group, and world by adding: <ul style="list-style-type: none">4 – read (r)2 – write (w)1 – execute (x) Examples: chmod 777 – read, write, execute for all chmod 755 – rwx for owner, rx for group and world For more options, see man chmod .	ping <i>host</i> – ping <i>host</i> and output results whois <i>domain</i> – get whois information for <i>domain</i> dig <i>domain</i> – get DNS information for <i>domain</i> dig -x <i>host</i> – reverse lookup <i>host</i> wget <i>file</i> – download <i>file</i> wget -c <i>file</i> – continue a stopped download
SSH	Installation
ssh <i>user@host</i> – connect to <i>host</i> as <i>user</i> ssh -p <i>port user@host</i> – connect to <i>host</i> on port <i>port</i> as <i>user</i> ssh-copy-id <i>user@host</i> – add your key to <i>host</i> for <i>user</i> to enable a keyed or passwordless login	Install from source: ./configure make make install dpkg -i <i>pkg.deb</i> – install a package (Debian) rpm -Uvh <i>pkg.rpm</i> – install a package (RPM)
Searching	Shortcuts
grep <i>pattern files</i> – search for <i>pattern</i> in <i>files</i> grep -r <i>pattern dir</i> – search recursively for <i>pattern</i> in <i>dir</i> <i>command</i> grep <i>pattern</i> – search for <i>pattern</i> in the output of <i>command</i> locate <i>file</i> – find all instances of <i>file</i>	Ctrl+C – halts the current command Ctrl+Z – stops the current command, resume with fg in the foreground or bg in the background Ctrl+D – log out of current session, similar to exit Ctrl+W – erases one word in the current line Ctrl+U – erases the whole line Ctrl+R – type to bring up a recent command !! – repeats the last command exit – log out of current session
	* use with extreme caution.



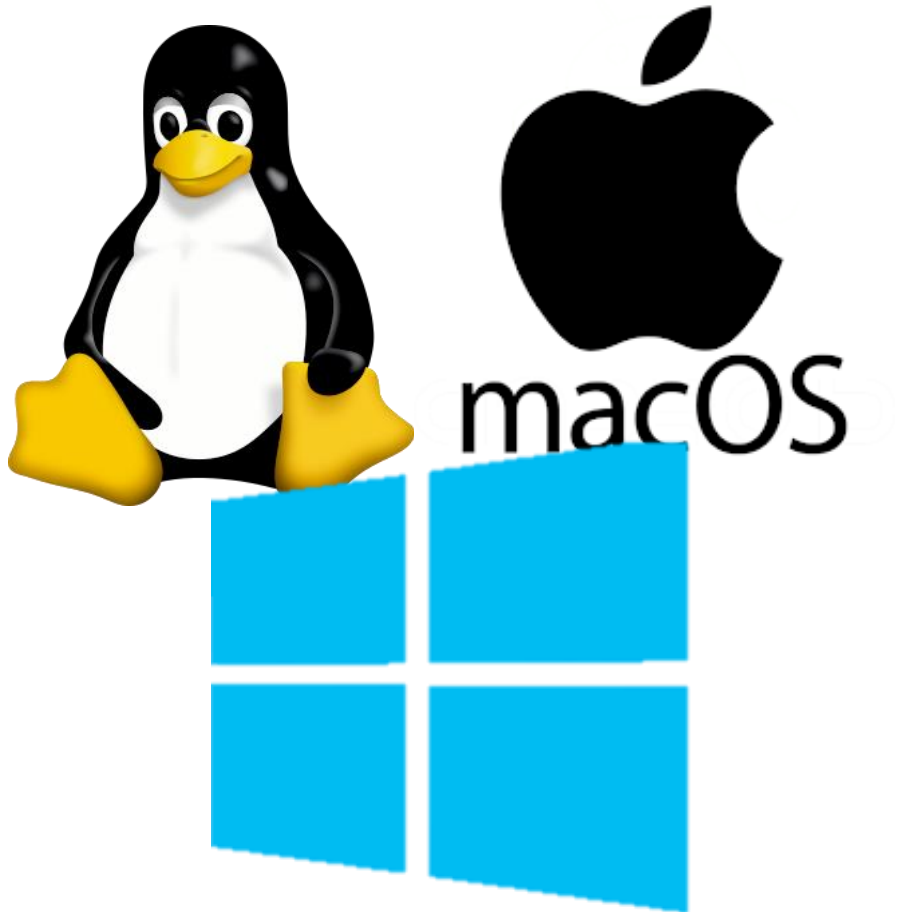
What is UNIX?

Operating system



What is UNIX?

Operating system

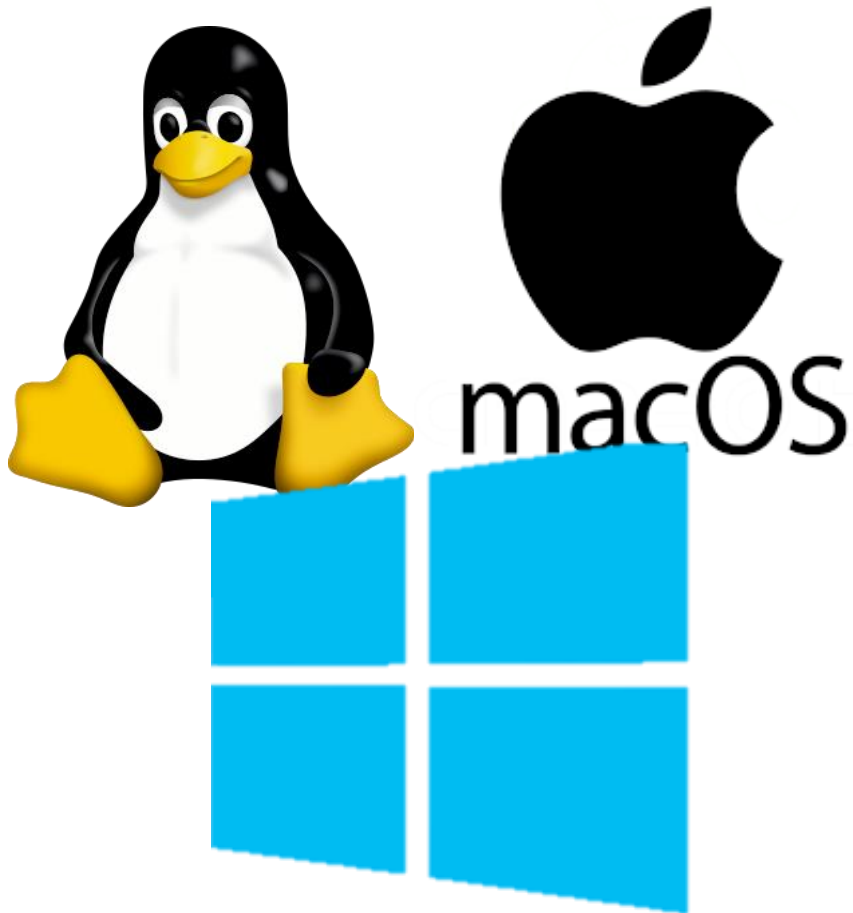


Why do we use it?

- Bioinformatics software designed to run on Unix platforms
- Large amounts of data
- Much faster than Windows PC

What is UNIX?

Operating system



Why do we use it?

- Bioinformatics software designed to run on Unix platforms
- Large amounts of data
- Much faster than Windows PC

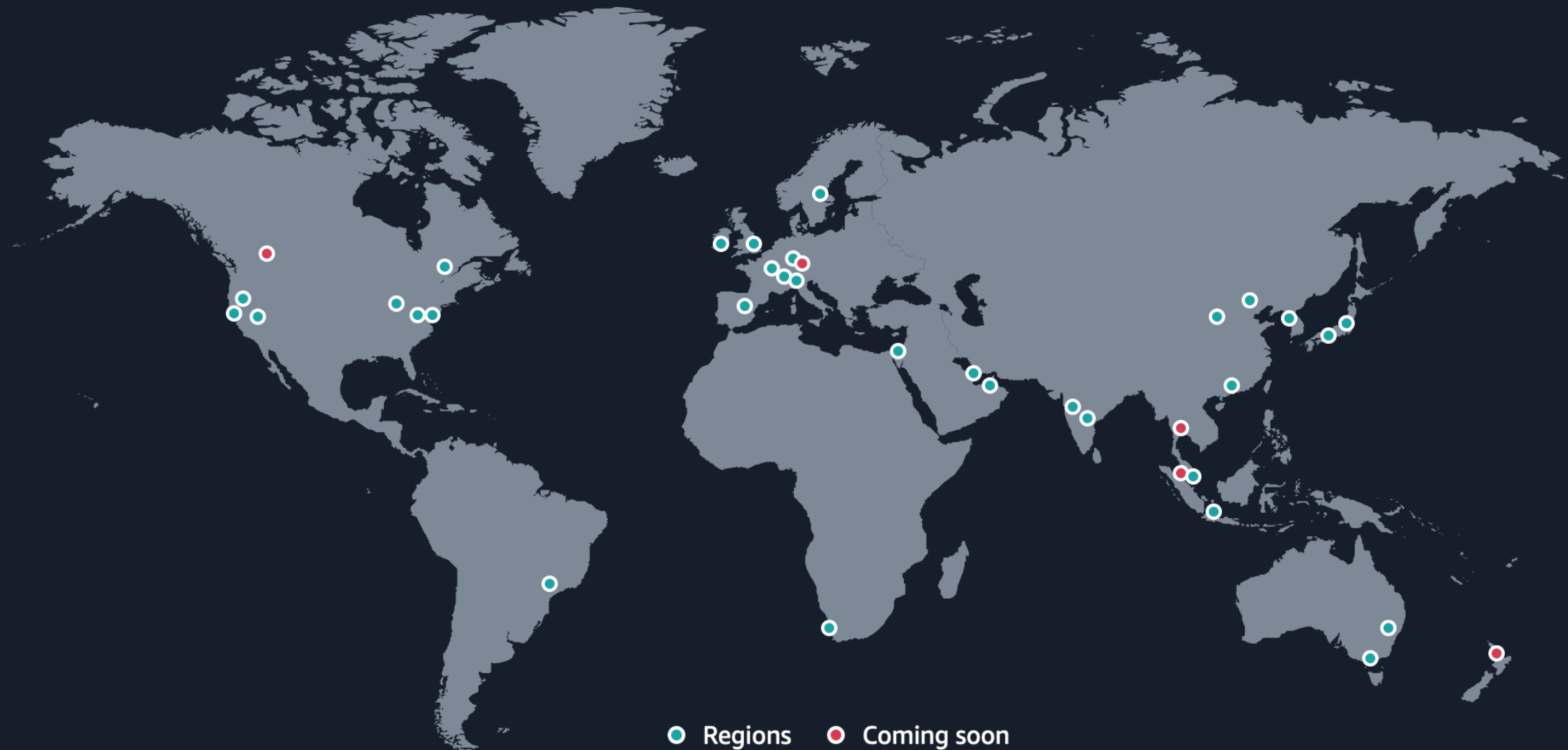
... And how?

- Linux computers or servers
- Computer clusters
- The cloud





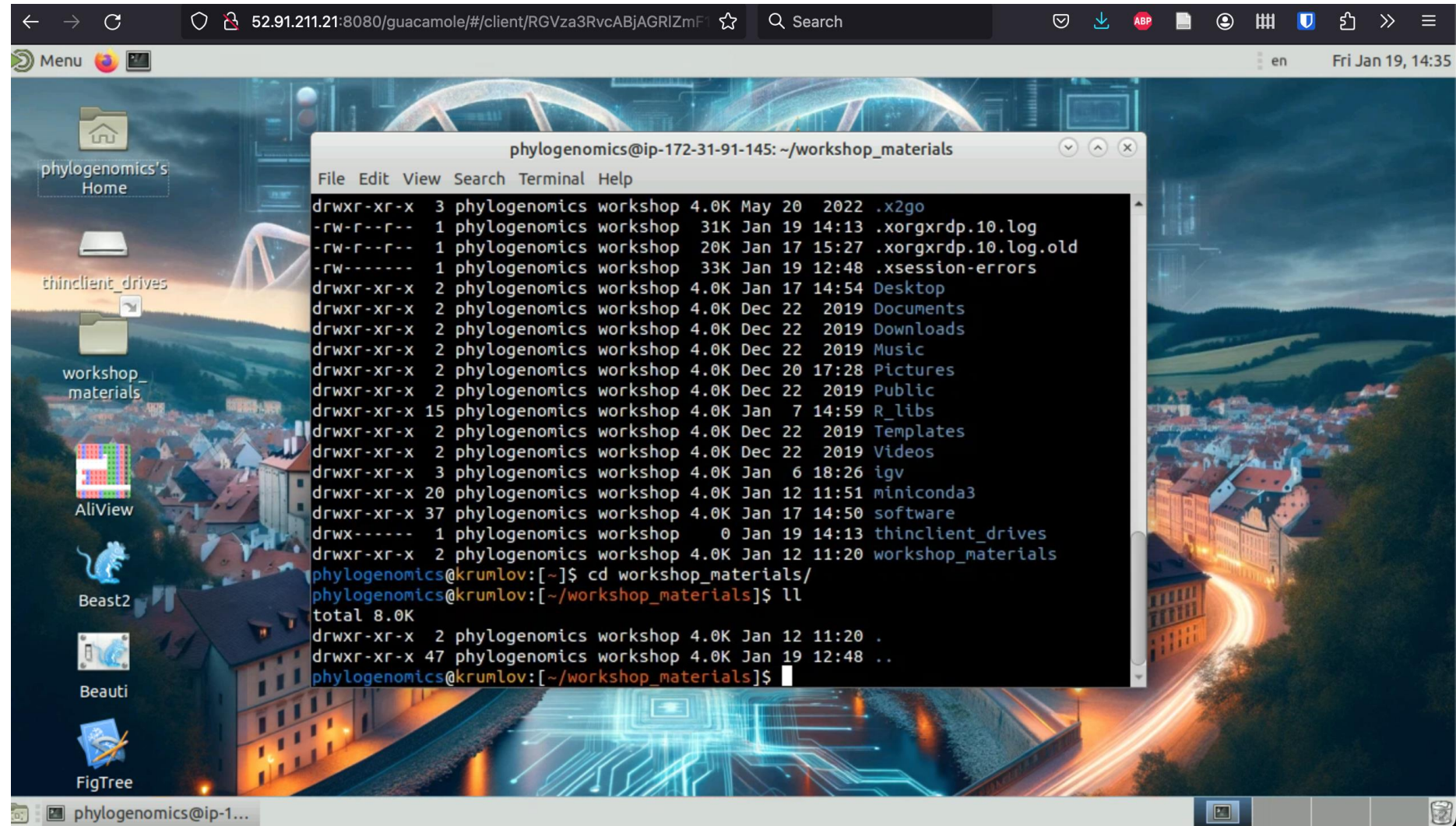
AWS availability zones



How it works

AMI
("Amazon Machine
Image")

Base computer
with all data and
software



How it works



Own copy of AMI = Instance (Virtual machine, VM)

Terminology

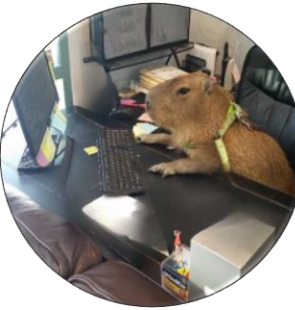
- Creating an instance – buying a brand new computer with software already installed.
- Starting an instance – turning that computer on.
- Stopping an instance – turning that computer off.
- Terminating an instance – setting that computer on fire and throwing it out of the window.

Watch & listen



Follow along on your
computer





Connecting to your instance

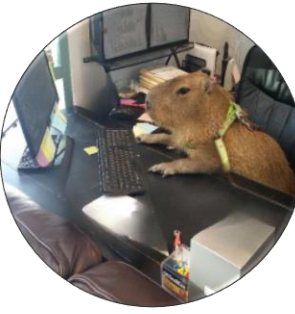
Windows: remote desktop
software

Guacamole, X2GO



Linux/Mac: Terminal
SSH (“Secure shell”)





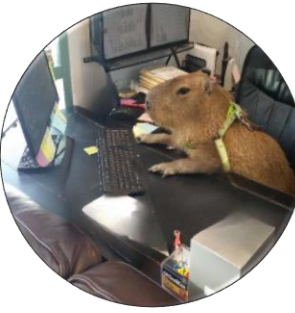
Instance addresses

<http://evomics.org/2024-workshop-on-phylogenomics-cesky-krumlov/>

FACULTY AND GENOMICS

- Click here to see our Faculty arrival and departure dates
- Faculty [Bios](#) and Teaching Assistant [Bios](#)
- **Faculty Lunches:** [sign-up sheet](#)
- [Instance IPs](#)
- [Cheatsheet](#) – copying files from/to the instances

Find your name and copy your IP address



Browser address bar: <https://docs.google.com/spreadsheets/d/1-vNj1eYBvvqzKp3c1K>

Spreadsheet Title: Workshop on Phyloenomics Instance List

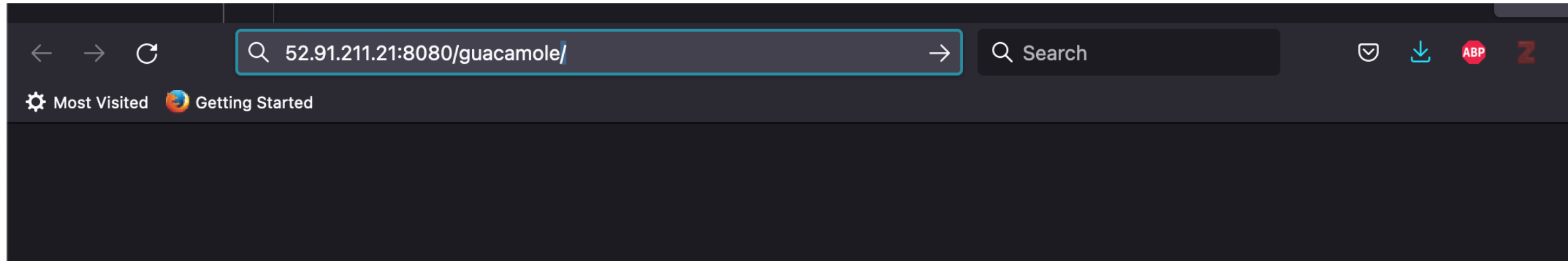
Menu: File Edit View Insert Format Data Tools Extensions Help

Toolbar: 100% | Kč % .0 .00 123 | Calibri | 12 | B I A | [Icons]

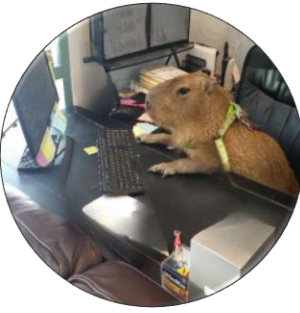
	A	B	C	D	E	F
	Name	IP address	Guacamole connection	ssh connection	RStudio server connection	Username: phyloge
1						Date:
2	Phylo-Karin	52.91.211.21	52.91.211.21:8080/guacamole	ssh genomics@52.91.211.21	52.91.211.21:8787	
3	Phylo-Marina	44.212.5.42	44.212.5.42:8080/guacamole	ssh genomics@44.212.5.42	44.212.5.42:8787	
4	Phylo-Jacob	3.89.92.242	3.89.92.242:8080/guacamole	ssh genomics@3.89.92.242	3.89.92.242:8787	
5	Phylo-Gemma	54.205.120.236	54.205.120.236:8080/guacamole	ssh genomics@54.205.120.236	54.205.120.236:8787	
6	Phylo-Michal	44.207.6.114	44.207.6.114:8080/guacamole	ssh genomics@44.207.6.114	44.207.6.114:8787	
7	Phylo-Rosa	3.94.82.239	3.94.82.239:8080/guacamole	ssh genomics@3.94.82.239	3.94.82.239:8787	
8			:8080/guacamole	ssh genomics@	:8787	
9			:8080/guacamole	ssh genomics@	:8787	
10			:8080/guacamole	ssh genomics@	:8787	
11			:8080/guacamole	ssh genomics@	:8787	
12			:8080/guacamole	ssh genomics@	:8787	
13			:8080/guacamole	ssh genomics@	:8787	
14			:8080/guacamole	ssh genomics@	:8787	




- Open your internet browser (e.g. Google Chrome)
- Paste the IP address followed by ‘:8080/guacamole’
- 52.91.211.21:8080/guacamole



Enter the username
“phylogenomics” and password

A screenshot of a web browser window showing the Apache Guacamole login page. The browser's address bar displays the URL "52.91.211.21:8080/guacamole/#/". The page features the Apache Guacamole logo, which is a green bowl with a yellow spoon. Below the logo, the text "APACHE GUACAMOLE" is displayed. There are two input fields: "Username" and "Password". At the bottom of the form is a dark grey button labeled "Login".

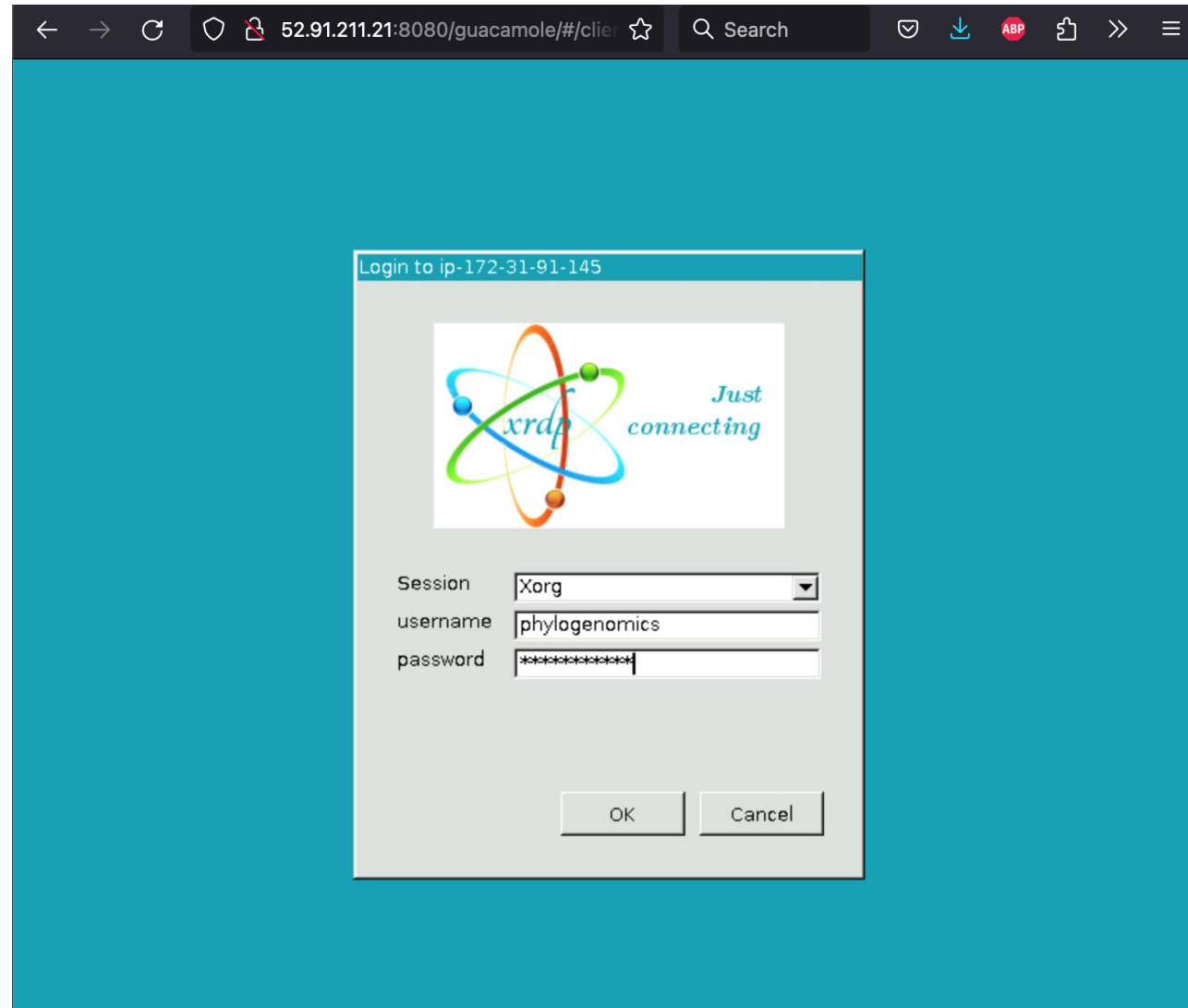
← → ↻ 🔒 52.91.211.21:8080/guacamole/#/ ☆ 🔍 Search 📧 ⬇️ ASP 📁 >> ☰



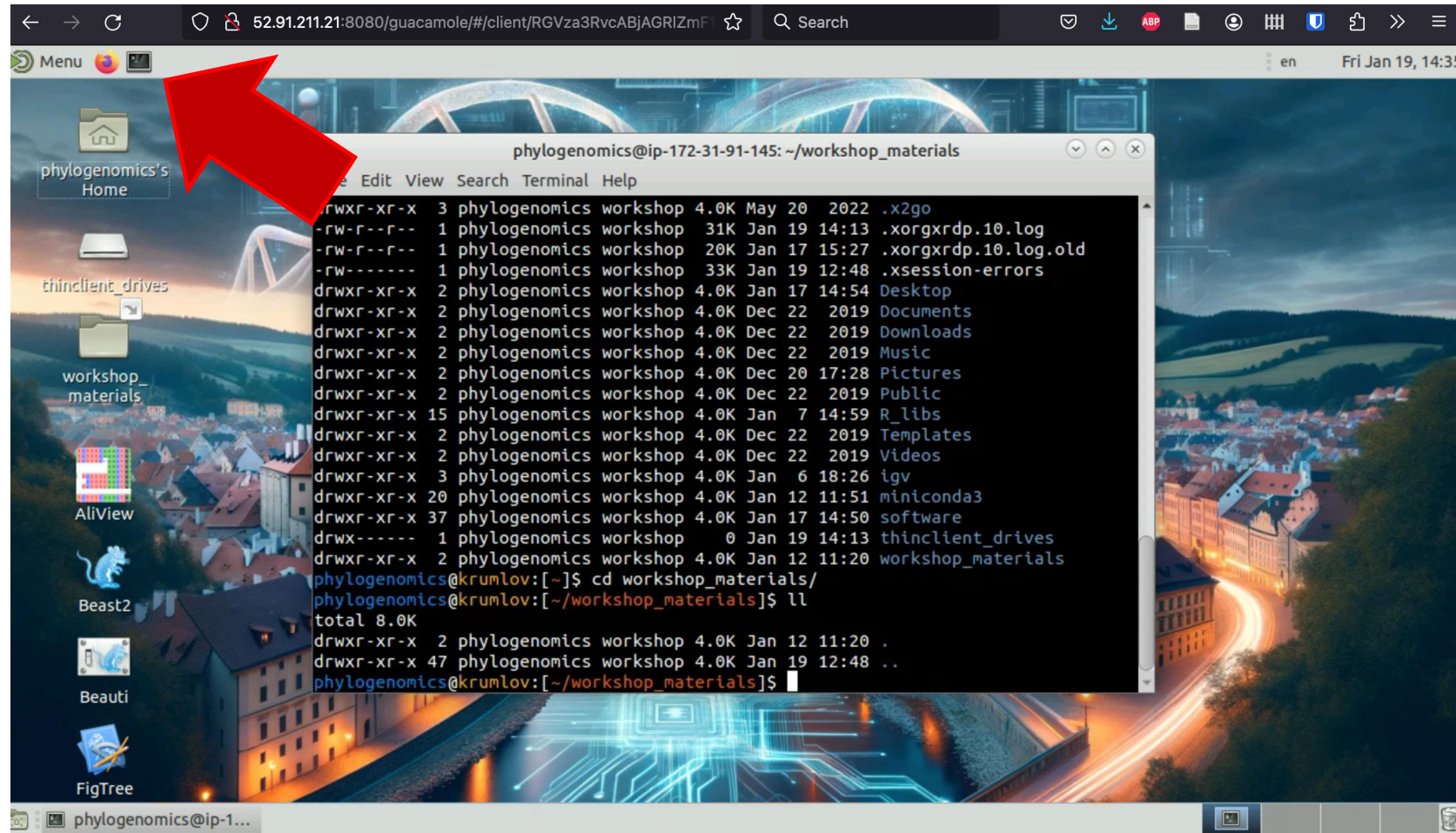
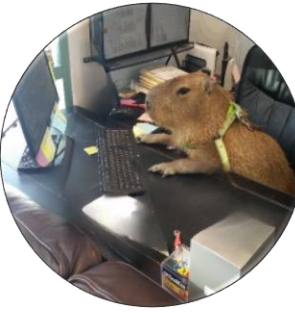
APACHE GUACAMOLE

Login

Select Desktop,
enter the same user name and pw again



Open terminal window using this icon



You're now connected and you're ready to learn some Unix!

Copy & Paste

AVOID COPYING AND PASTING
WHEREVER POSSIBLE!

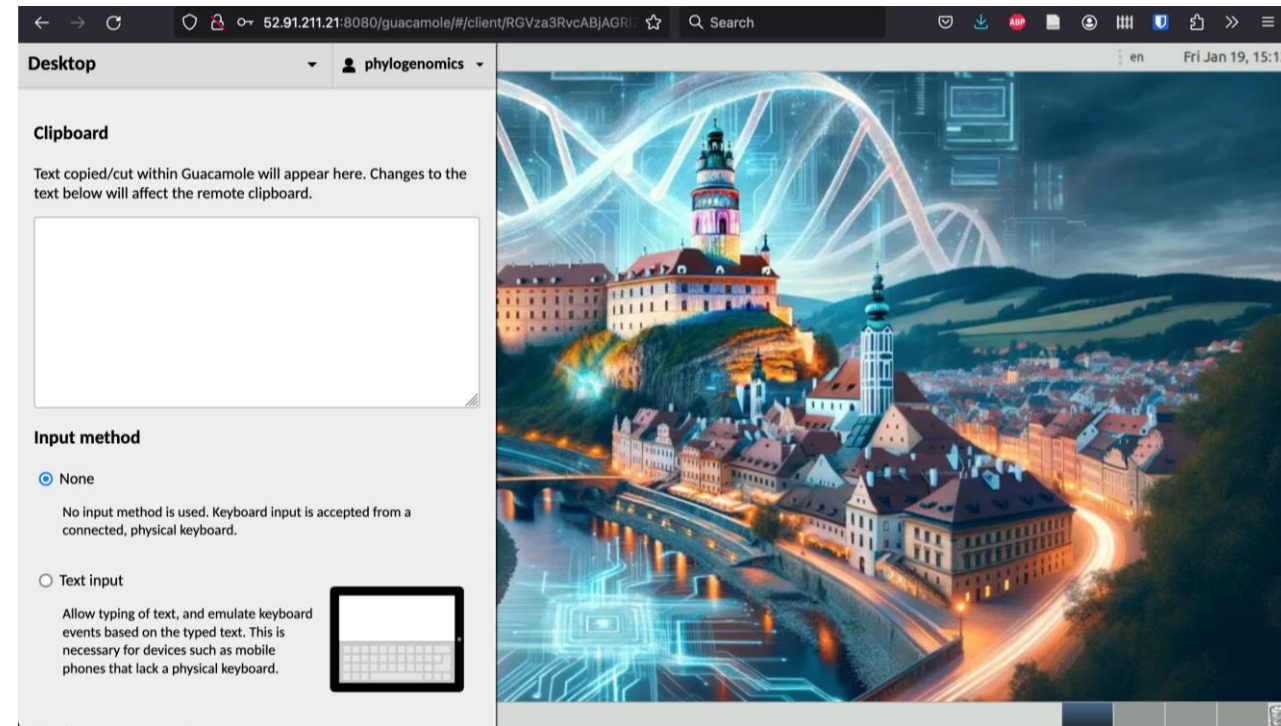
But if you do need to...

Press Ctrl+Alt+Shift

Paste the text into the box with right
click → Paste

Press Ctrl+Alt+Shift again

You can now paste into the instance
using right click





Your final task before we get started!

Make sure that typing tilde (~), backslash (\), pipe (|), and carat (^) in the terminal works.

Google search to find these on your computer if you don't know where they are.

Questions?





Commands overview

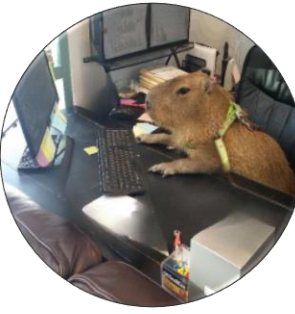
pwd	<i>print working directory</i>	cp	<i>copy</i>
~	<i>home</i>	gunzip	<i>unzip</i>
.	<i>here</i>	tar	<i>unarchive</i>
..	<i>one directory up</i>	head	<i>first (n=10) lines</i>
mkdir	<i>make new directory</i>	tail	<i>last (n=10) lines</i>
cd	<i>change directory</i>	cat	<i>concatenate</i>
touch	<i>create file</i>	wc	<i>word count</i>
ls	<i>list</i>	grep	<i>pattern search</i>
man	<i>manual</i>		<i>pipeline</i>
mv	<i>move</i>	sed	<i>stream editor</i>
rm	<i>remove</i>	chmod	<i>change file modes</i>

The terminal (command line, shell, prompt)



Where you see this “\$” followed by text, I want you to type the text on your command line

A screenshot of a terminal window. The title bar at the top reads "phylogenomics@ip-172-31-91-145: ~". Below the title bar is a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The main area of the terminal is black. The first line of text in the terminal is "phylogenomics@krumlov: [~]\$ " in a monospaced font, with a white cursor block at the end of the line.



Location is important

First task: Where am I?










```
$ pwd
```

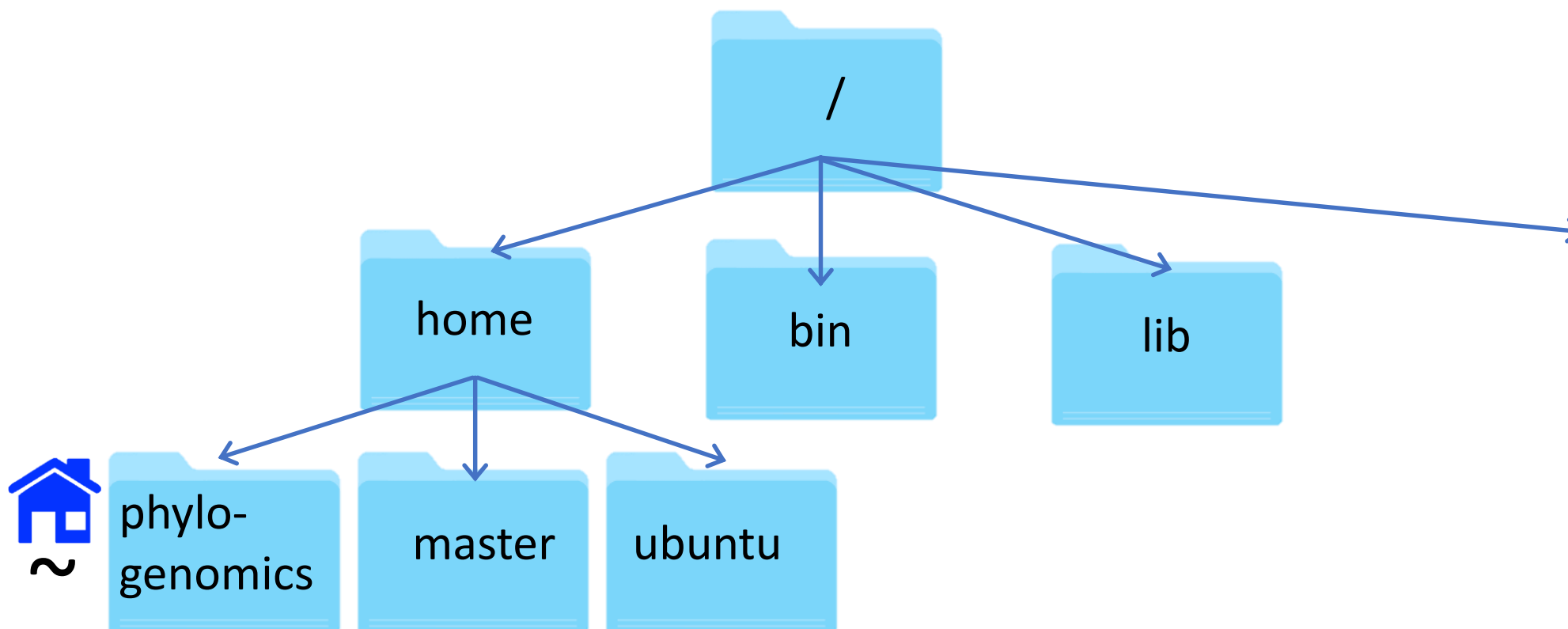
```
phylogenomics@ip-172-31-91-145: ~  
File Edit View Search Terminal Help  
phylogenomics@krumlov:~$ pwd  
/home/phylogenomics  
phylogenomics@krumlov:~$
```

This is your working directory, i.e. where you currently are.



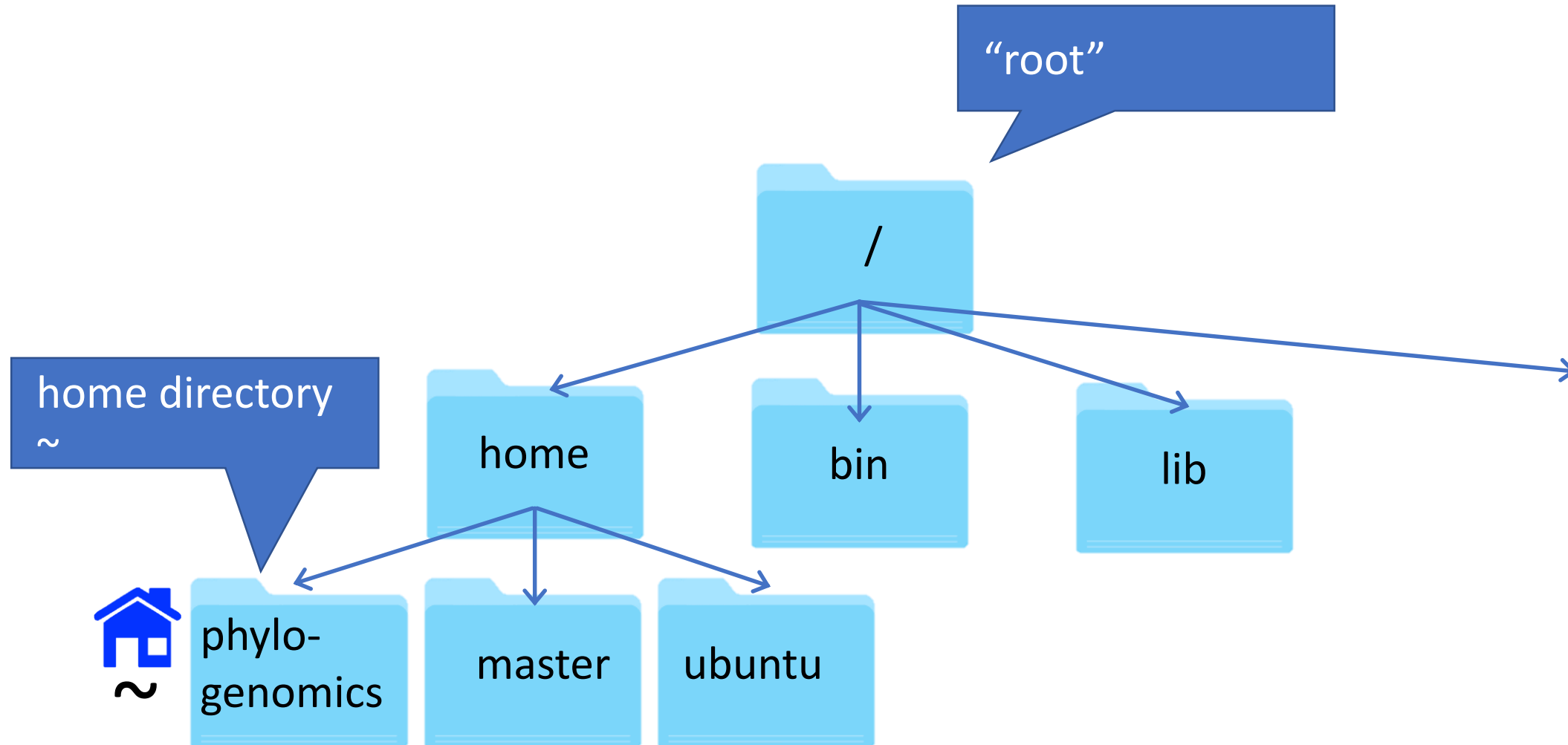
Creating and navigating directories

Folders	Folders	Folders	Folders
 / >	 bin >  home >  lib >	 master >  phylogenomics >  ubuntu >	 Data >  New >



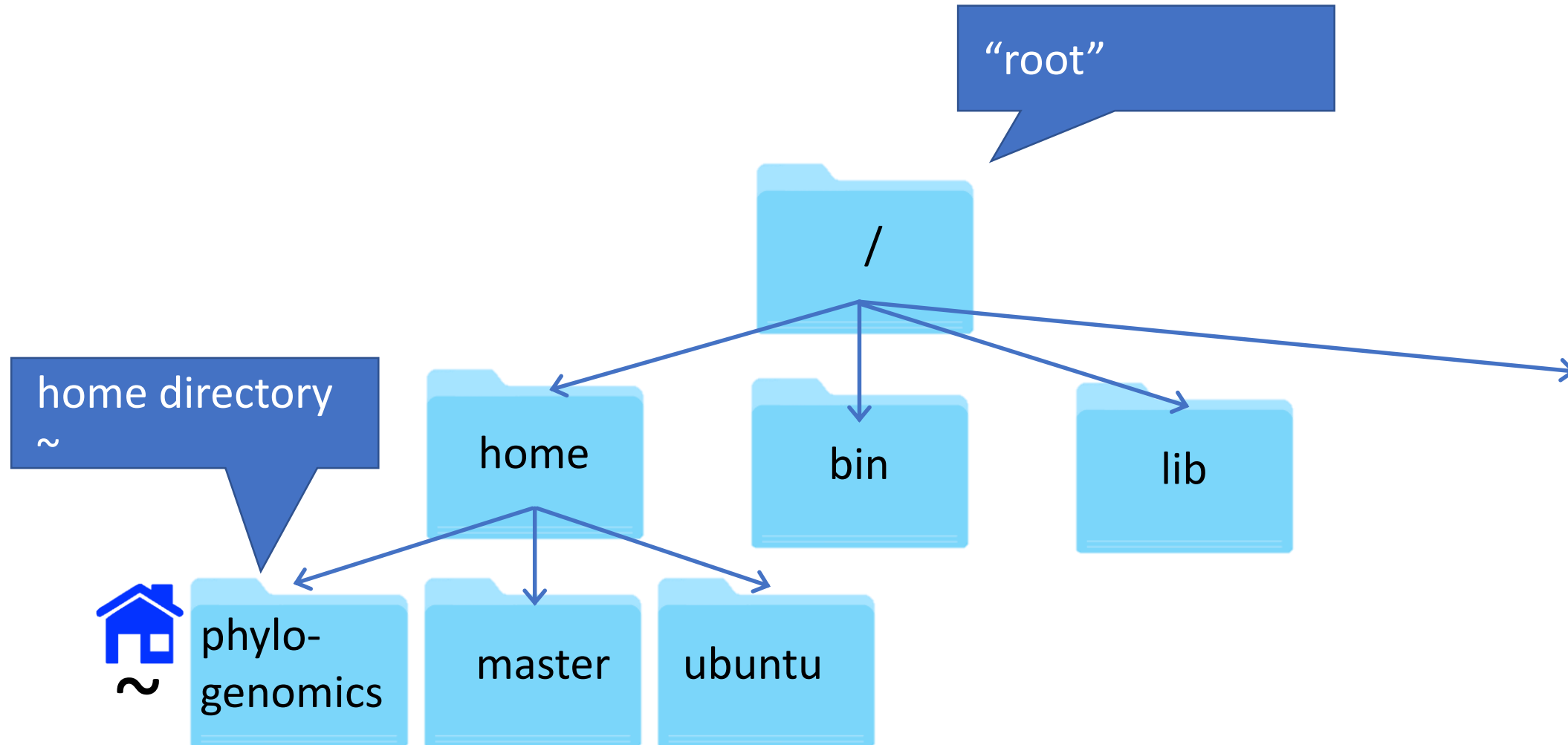


Creating and navigating directories





Creating and navigating directories





Creating and navigating directories

Create a new directory called “Data” in your current directory

```
$ mkdir ./Data
```

Change into the new directory

```
$ cd Data
```

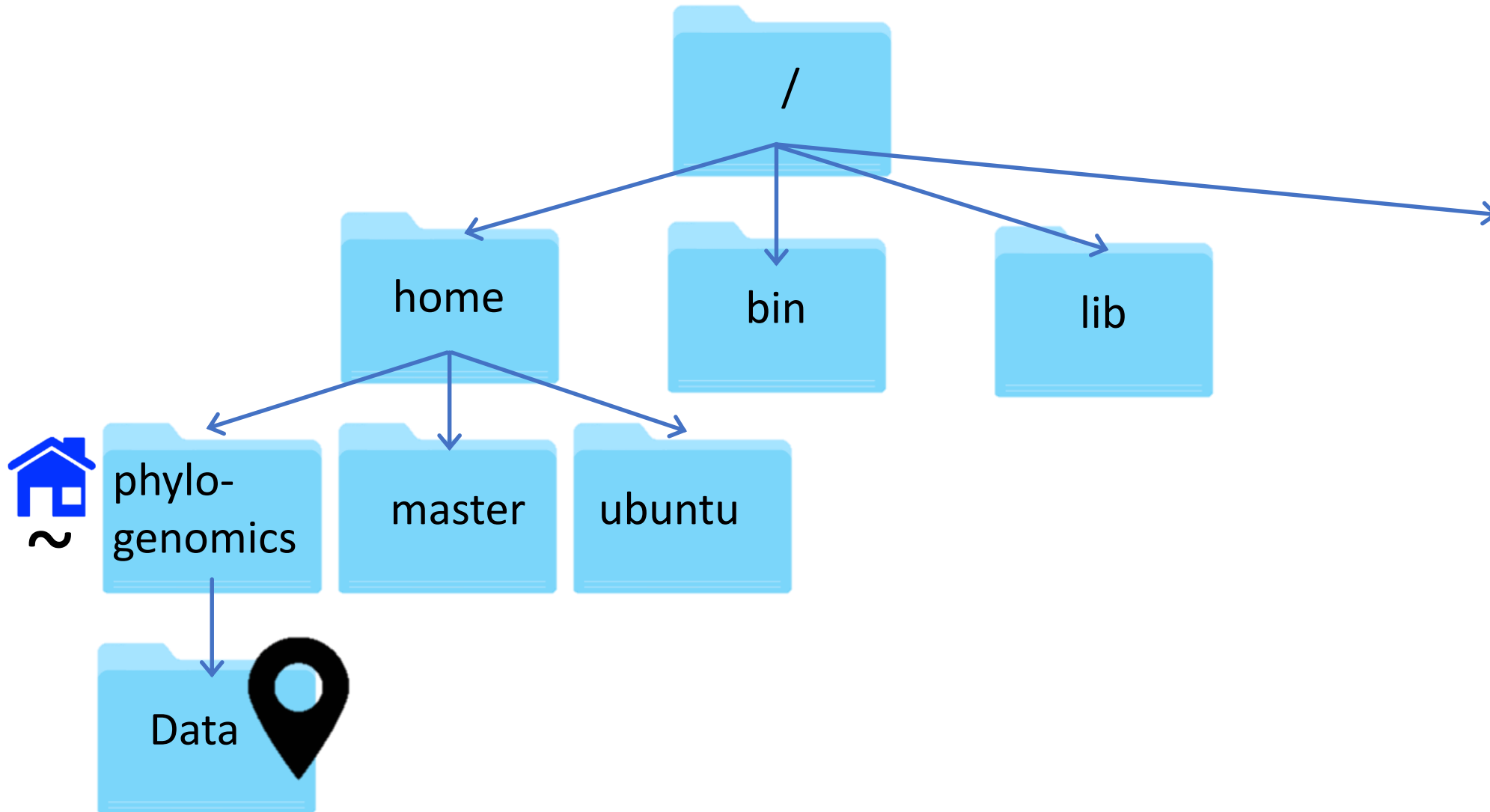
Where are you now? What is your present working directory?

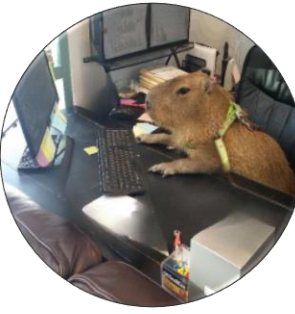
Directory names (and file names for the matter) can not contain spaces.*

Underscores are often used instead if you want to separate words.



Creating and navigating directories





Creating and navigating directories

Make an empty file "rags"

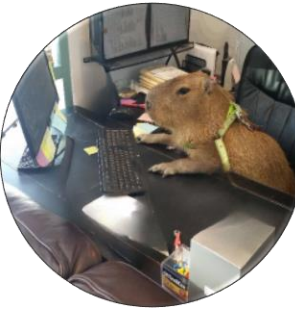
```
$ touch rags
```

And another two "Heaven" and "Earth"

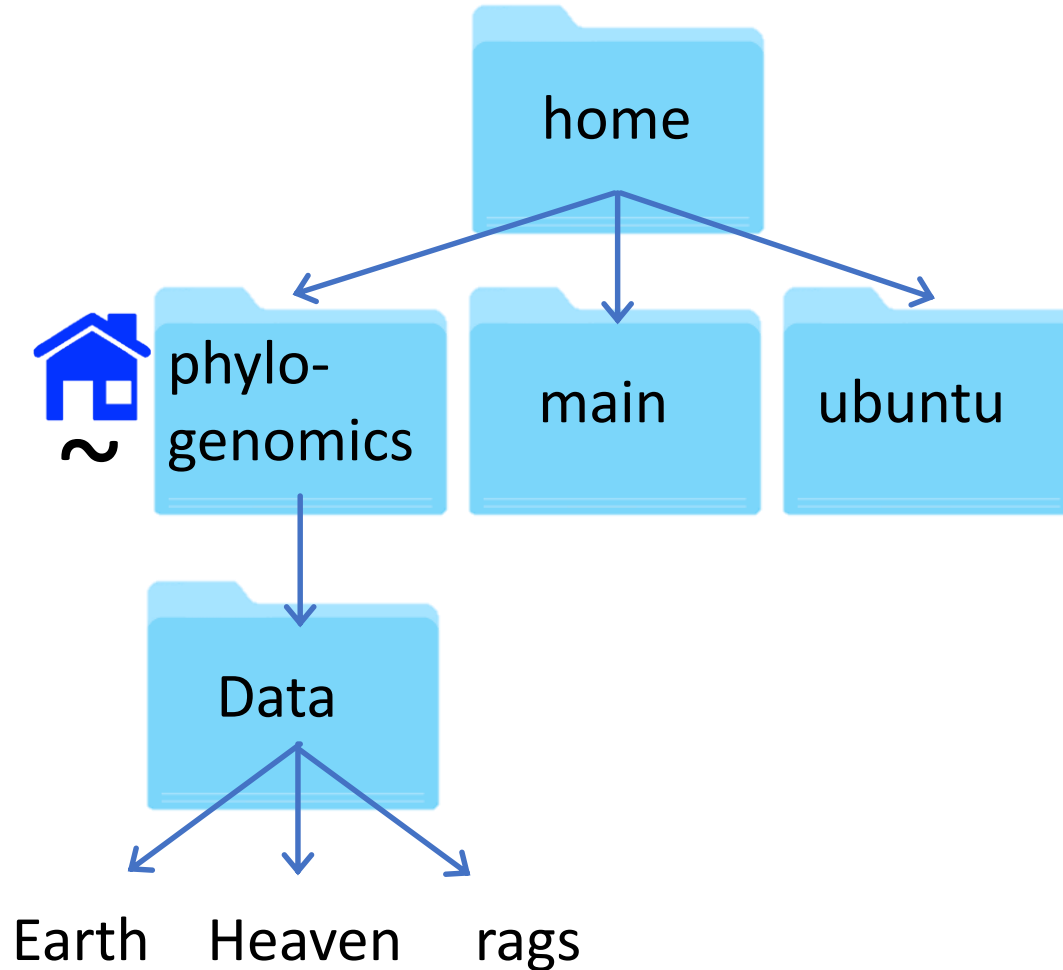
```
$ touch Heaven Earth
```

Now let's list the contents of the current directory (Data)

```
$ ls
```

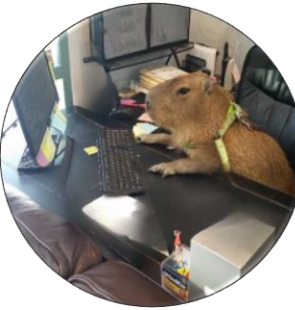


Creating and navigating directories

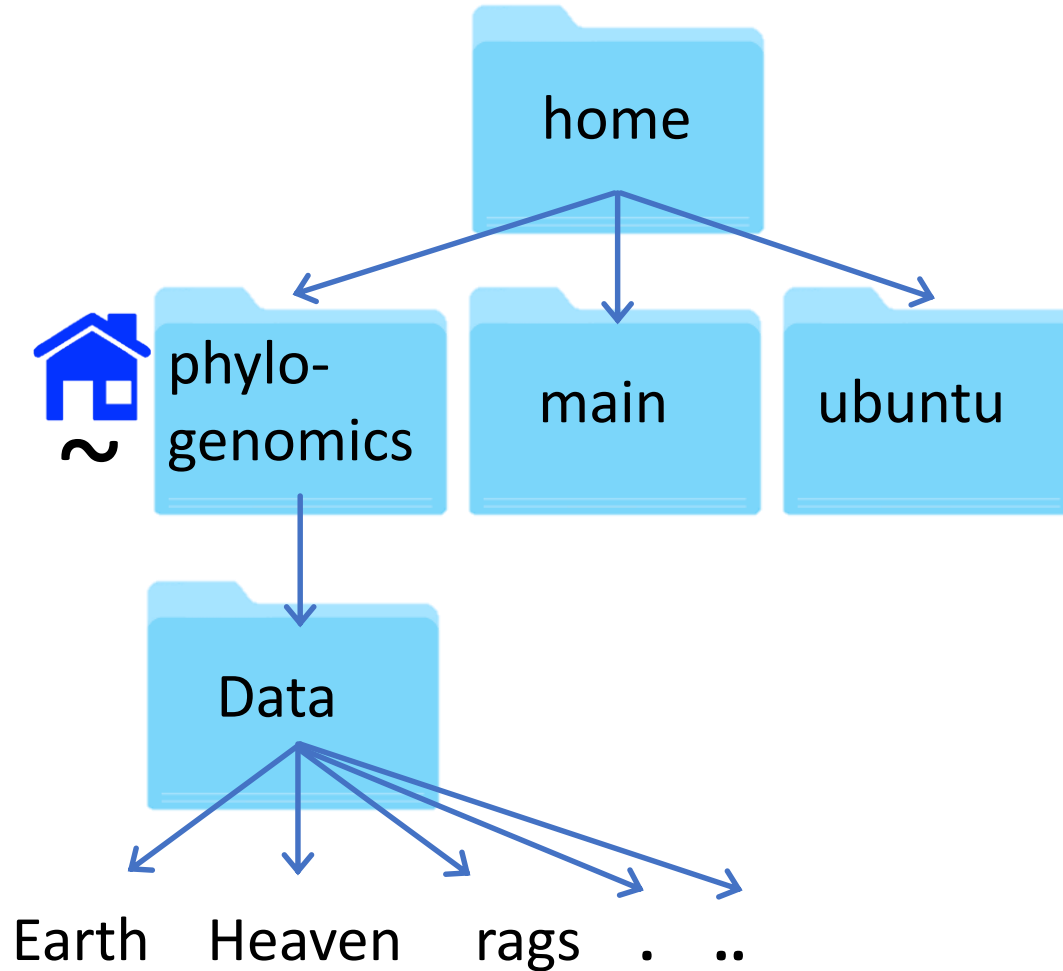


Now list **all** of the files in the directory

```
$ ls -a
```



Creating and navigating directories

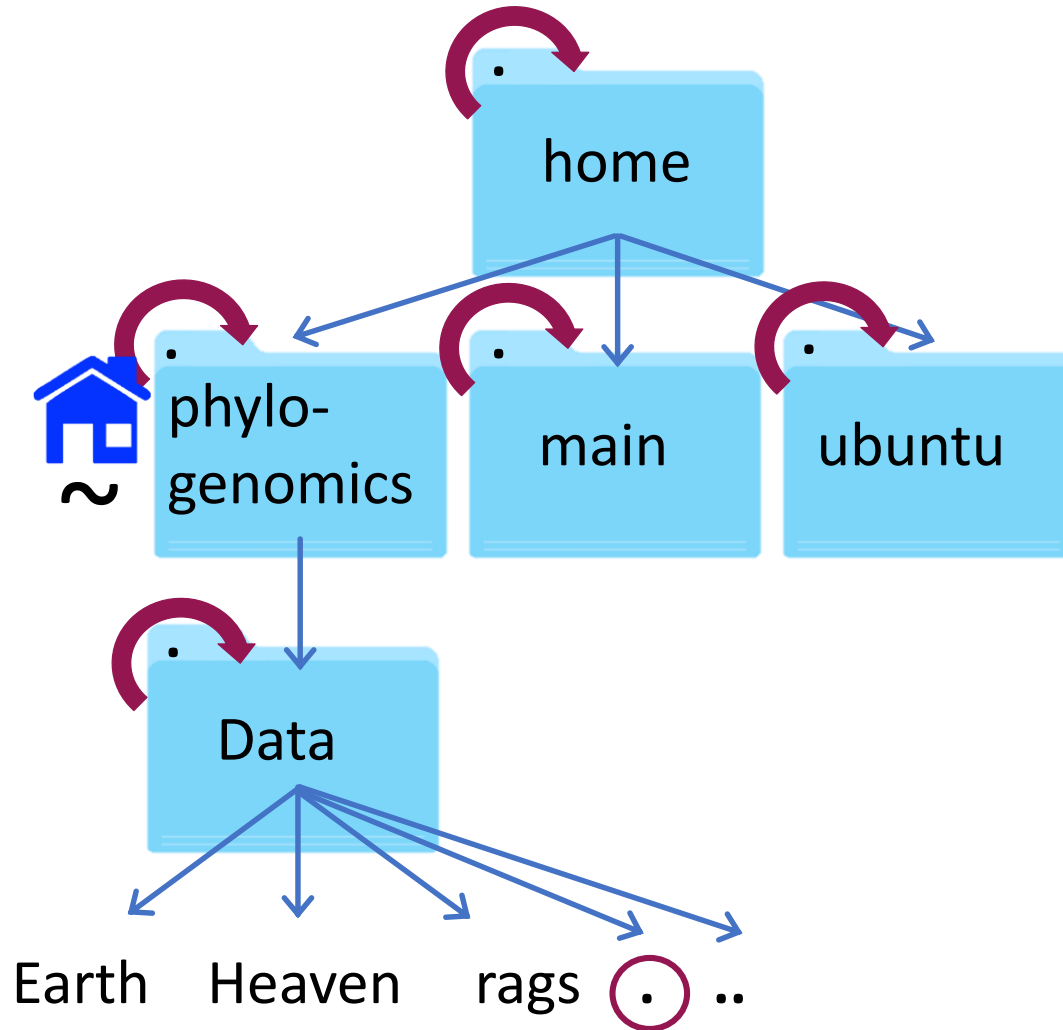


Now list **all** of the files in the directory

```
$ ls -a
```



Creating and navigating directories



Now list **all** of the files in the directory

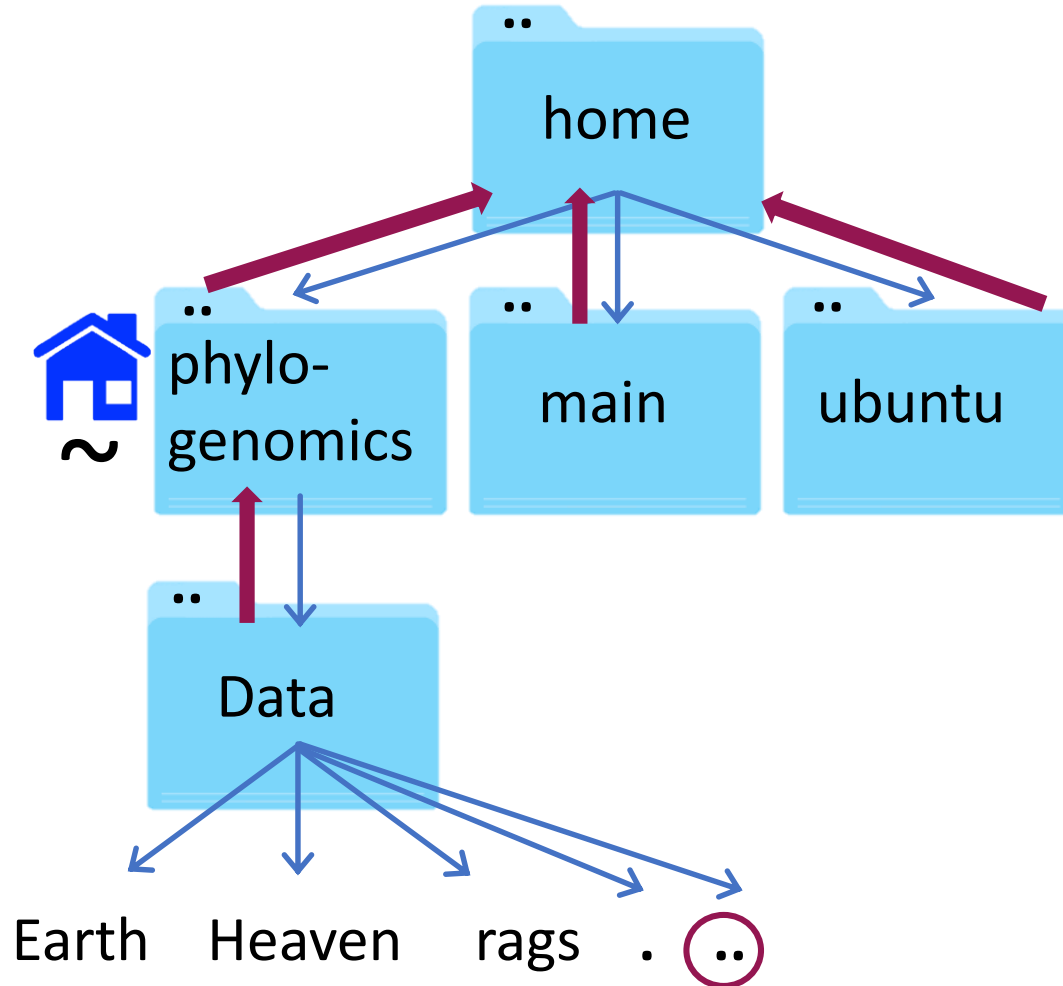
```
$ ls -a
```

. points to the current directory

.. points to one directory above



Creating and navigating directories



Now list **all** of the files in the directory

```
$ ls -a
```

. points to the current directory

.. points to one directory above



Creating and navigating directories

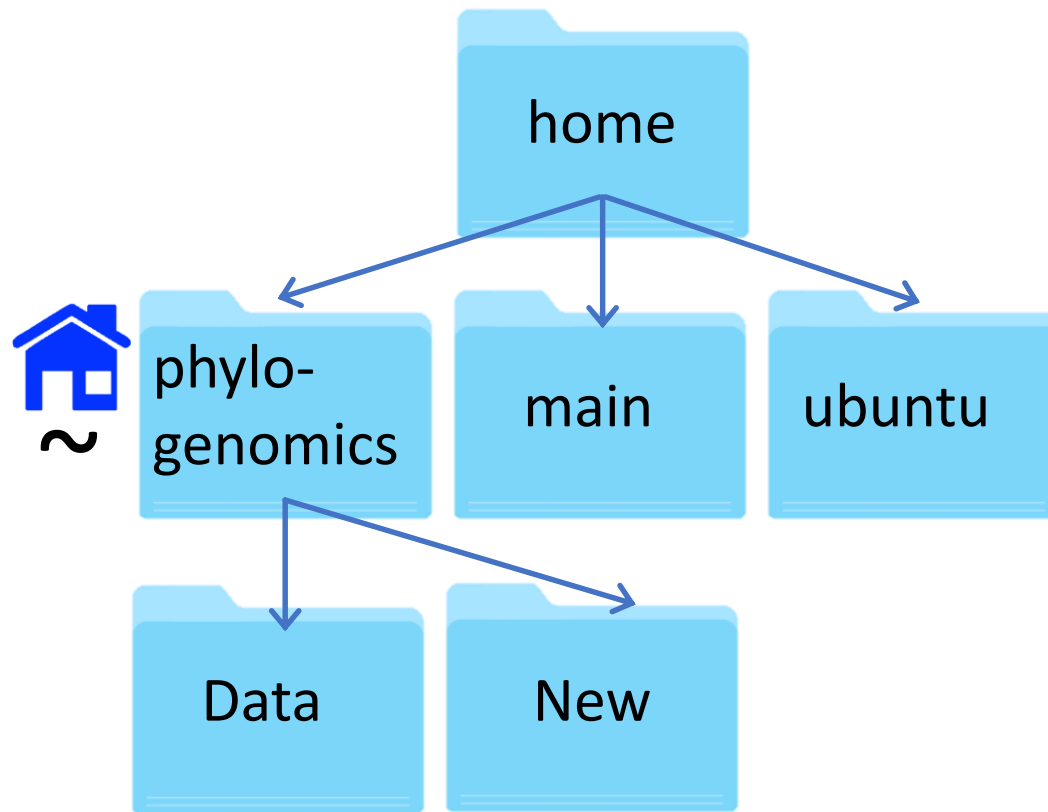
. and .. are used for specifying location

Whenever you do anything on Unix (move around, move a file, rename a file, run a program or script, etc...) you have to tell the system where that thing is using a path.

. and .. are part of RELATIVE paths



Creating and navigating directories

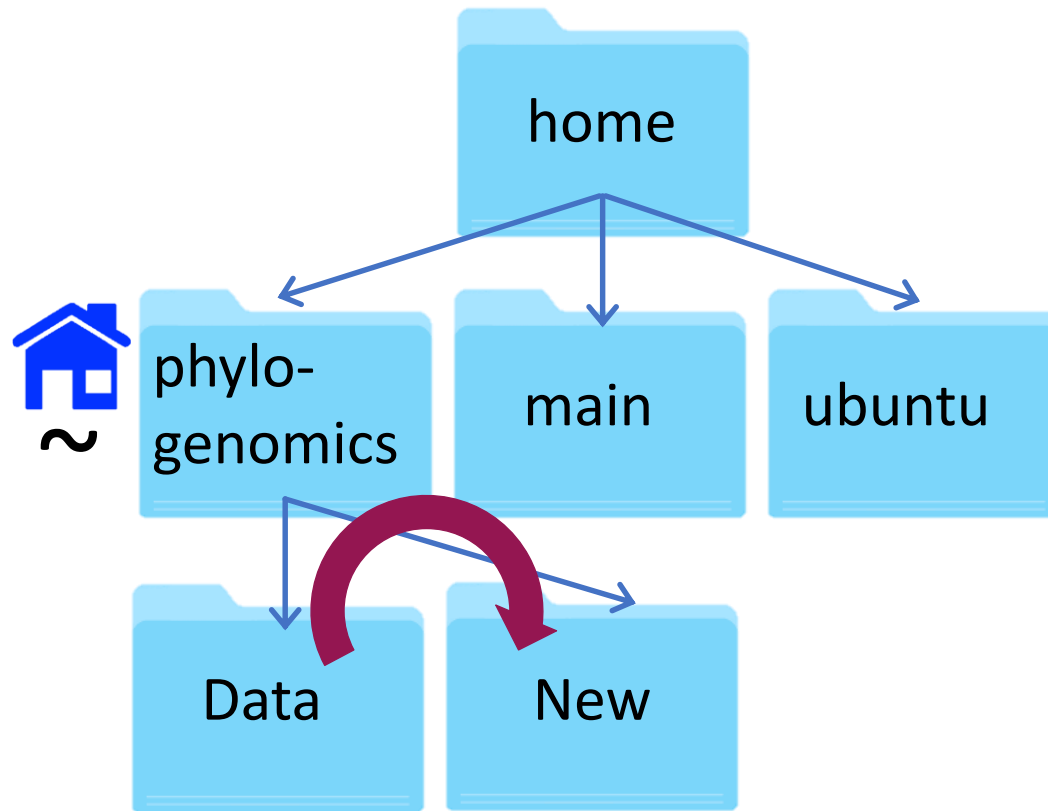


Create a directory called New within the phylogenomics directory using the RELATIVE PATH

```
$ mkdir ../New
```



Creating and navigating directories

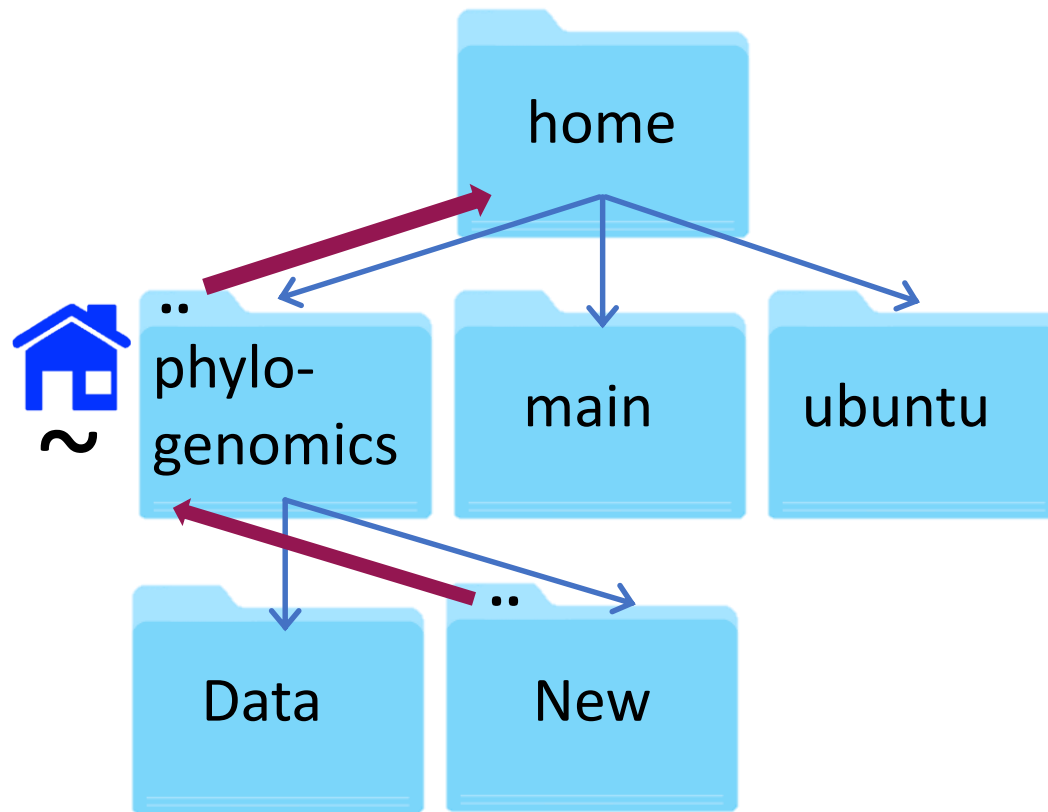


Move from Data to New
RELATIVE PATH

```
$ cd ../New
```




Creating and navigating directories



Move from New to home
RELATIVE PATH

```
$ cd ../../
```



Creating and navigating directories

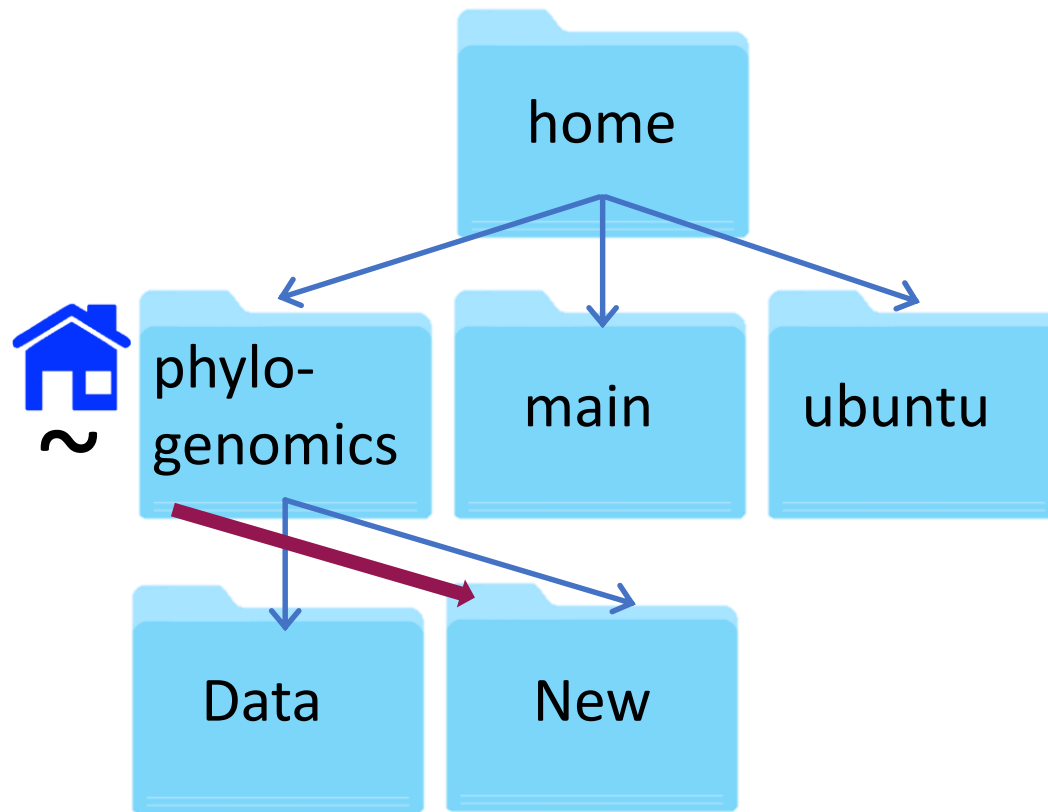
Relative paths will always change depending on your location.

The alternative is ABSOLUTE paths.

These always start from root and will never change.



Creating and navigating directories

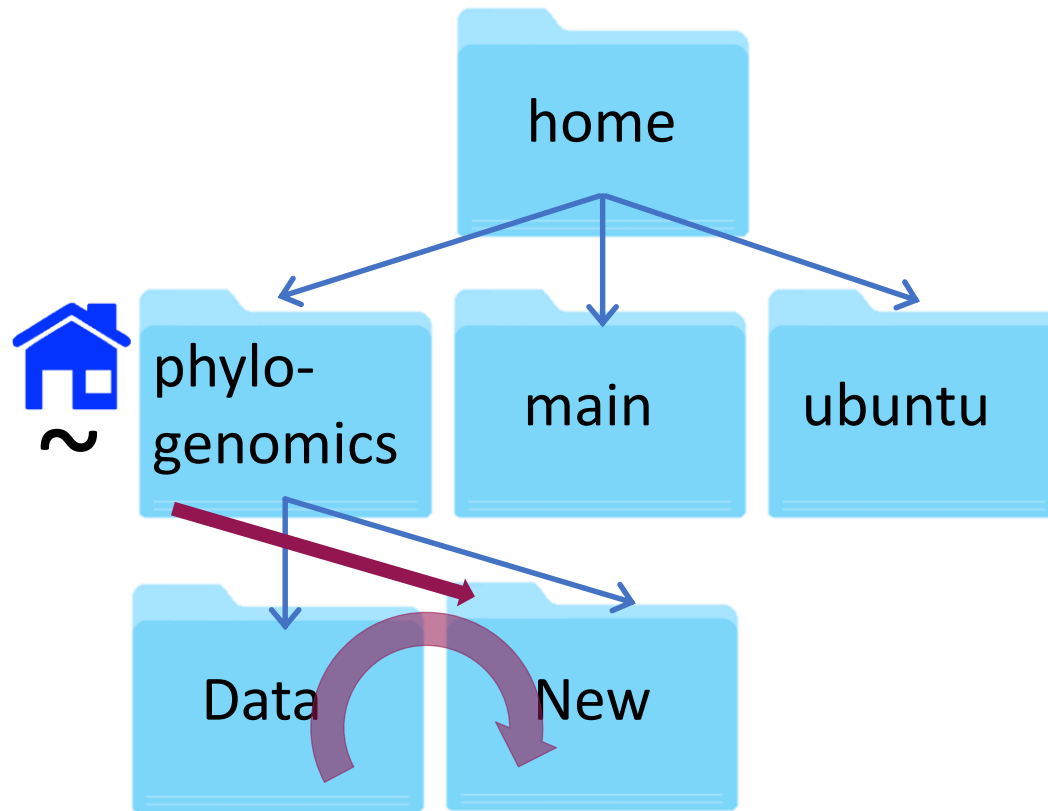


Move from your home directory to New
ABSOLUTE PATH

```
$ cd /home/phylogenomics/New
```



Creating and navigating directories



Move from your home directory to New
ABSOLUTE PATH

```
$ cd /home/phylogenomics/New
```

Move from Data to New
ABSOLUTE PATH

```
$ cd /home/phylogenomics/New
```




A note about . dot

. means in your (present) working directory

This command means “List everything that’s in the (present) working directory”

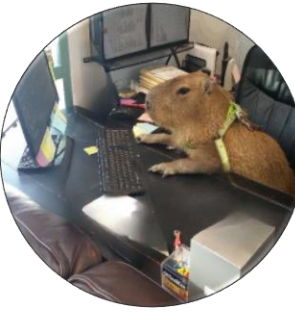
```
$ ls ./
```

This command means “List everything that’s in the working directory within a subdirectory called Data”

```
$ ls ./Data/
```

In most cases, people don’t use ./ at the beginning of a path. As long as the file/directory is within your working directory, the command will work.

```
$ ls ./Data/    =    $ ls Data
```



Let's practice

Where am I right now? (Should be the Data directory)

```
$ pwd
```

Change to the directory above

```
$ cd ../
```

Let's list the contents of the Data directory

```
$ ls ./Data
```

CHALLENGE 1!

1. Move into the Data directory and list the contents of your home directory.
2. In Data, make a new directory and move into this location.
3. From this new directory, move into your home directory IN ONE COMMAND and check your location.



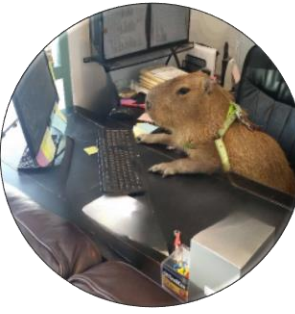
Work smarter, not harder!

Tab completion is a nice trick to save you typing paths

For this example we are going to list everything in the directory `/home/phylogenomics/workshop_materials/`. Start by typing

`$ ls /` followed by pressing tab twice quickly. This shows the contents of the root directory:

```
phylogenomics@ip-172-31-91-145: ~/workshop_materials
File Edit View Search Terminal Help
phylogenomics@krumlov: [~/workshop_materials]$ ls
ls          ls_release  lscpu      lshw       lspic       lslogins   lsmod      lsof        lspcmcia   lss3
lsattr      lsblk       lsdiff     lsinitramfs lslocks     lsmem      lsns       lspci       lspgpot    lsusb
phylogenomics@krumlov: [~/workshop_materials]$ ls
```



Work smarter, not harder!

Now type: `$ ls /h` followed by tab once.

The path to the `/home/` directory has filled in.

Now type: `$ ls /home/p` followed by tab once.

The path to the `/home/phylogenomics/` directory has filled in.

Finally type: `$ ls /home/phylogenomics/w` followed by tab once to finish the path, and then enter.

You've now listed that directory contents.

Tab complete will fill in paths, save you time in typing and **prevent typos!**



Work smarter, not harder!

Two more tricks for less typing!

The * (asterisk) represents any character

For example: `$ ls /home/phylogenomics/*.txt`

Will list everything in my home directory ending .txt

The up arrow can be used to re-run commands

Press your up arrow and see!

If you want all of your previous commands listed, simply type `$ history`

Questions?





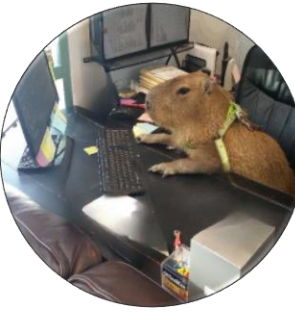
Binary programs

These are all programs installed on the Unix machine.

They can be found in /bin `$ ls /bin`

```
File Edit View Search Terminal Help
phylogenomics@ip-172-31-91-145: ~
h2xs          pod2html      zegrep
h5c++         pod2man      zeisstopnm
h5cc          pod2markdown zenity
h5fc          pod2readme   zfgrep
hardlink      pod2texi     zforce
hbf2gf        pod2text     zgrep
hbpldecode    pod2usage    zip
hciattach     podchecker   zipcloak
hciconfig     pdebcconf-display-po zipdetails
hcitool       pdebcconf-report-po zipgrep
hd            poff         zipinfo
head          pollinate    zipnote
helixturnhelix polydot      zipsplit
helpztags     pon         zjsdecode
hex2hcd       pooltype     zless
hexdump       post-grohtml zmore
hibagent      powerprofilesctl znew
hipercdecode  ppc         zoom2sam.pl
hipstopgm     ppdhtml     zstd
hmac256       ppci        zstdcat
hmmalign      ppdmerge    zstdgrep
hmmbuild      ppdpo       zstdless
hmmconvert    pphs        zstdnt
hmmemit       ppltotf
hmmfetch      ppm3d
```

These include pwd, mkdir, ls ...



Binary programs have manuals

To view the manual page, type `man` followed by the name of the program.

Open the manual page for `ls` `$ man ls`

Scroll through (enter) and find the options for:

long listing format (`-l`), human-readable file sizes (`-h`) and sort by modification time (`-t`).

Exit the manual page (type `q`) and give these `ls` options a go in your Data directory.

```
$ ls -l -t -h ./Data = $ ls -lth ./Data
```



PATH

The computer needs to know where a program is so that it can access the code to run the program.

The PATH environment variable is a list of locations your computer looks for programs.

You can either provide the path to the program you want to run

```
$ /usr/bin/mkdir
```




PATH

The computer needs to know where a program is so that it can access the code to run the program.

The PATH environment variable is a list of locations your computer looks for programs.

You can either provide the path to the program you want to run

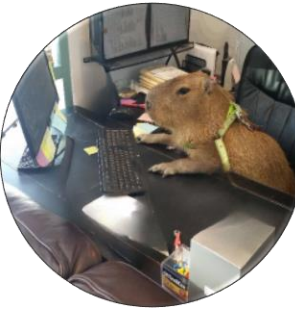
```
$ /usr/bin/mkdir
```

Or make sure the program is in your PATH environment variable

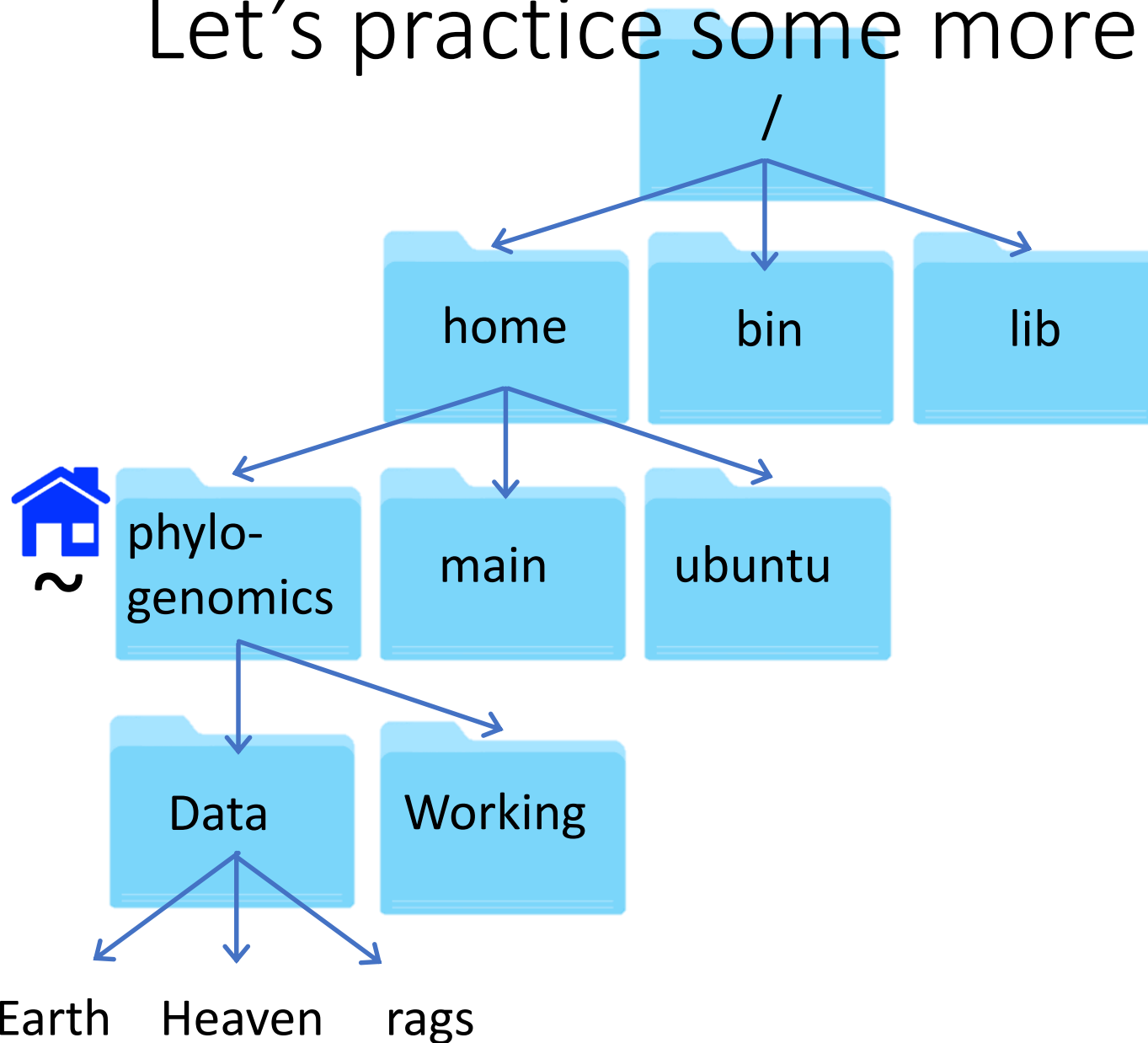
To view locations in your PATH environment variable:

```
$ echo $PATH
```

There are ways to add new locations to your PATH, but that is for another time.

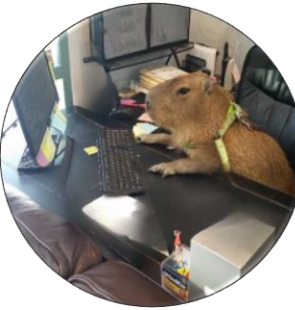


Let's practice some more

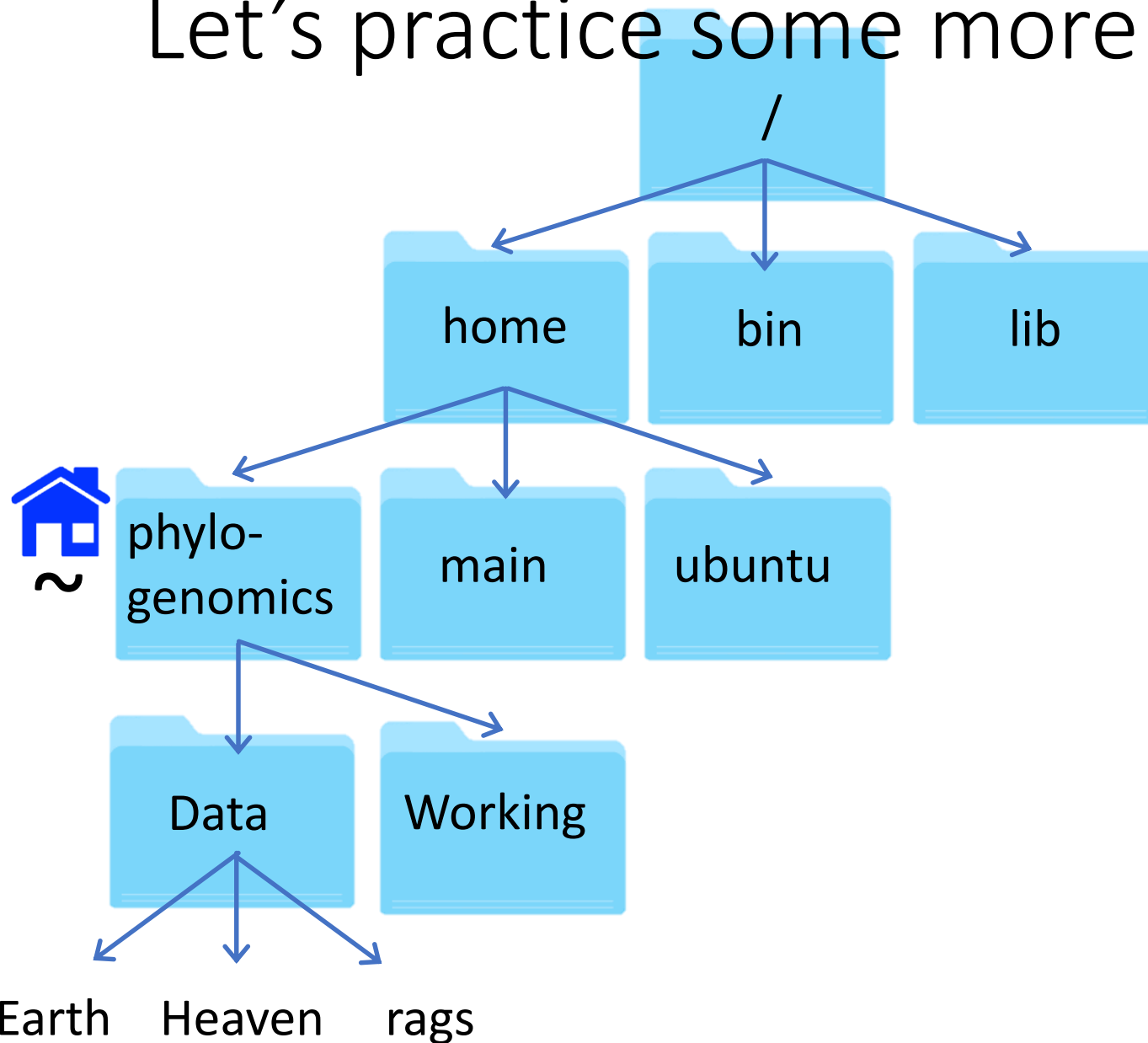


First I need you to make a new directory called “Working” within your home directory.

Afterwards your file structure should look like this.



Let's practice some more



Moving Files

Lets move Heaven and Earth from Data to Working

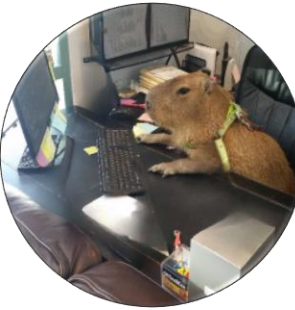
```
$ cd ~/Data  
$ mv ../Heaven ../Working
```

file path

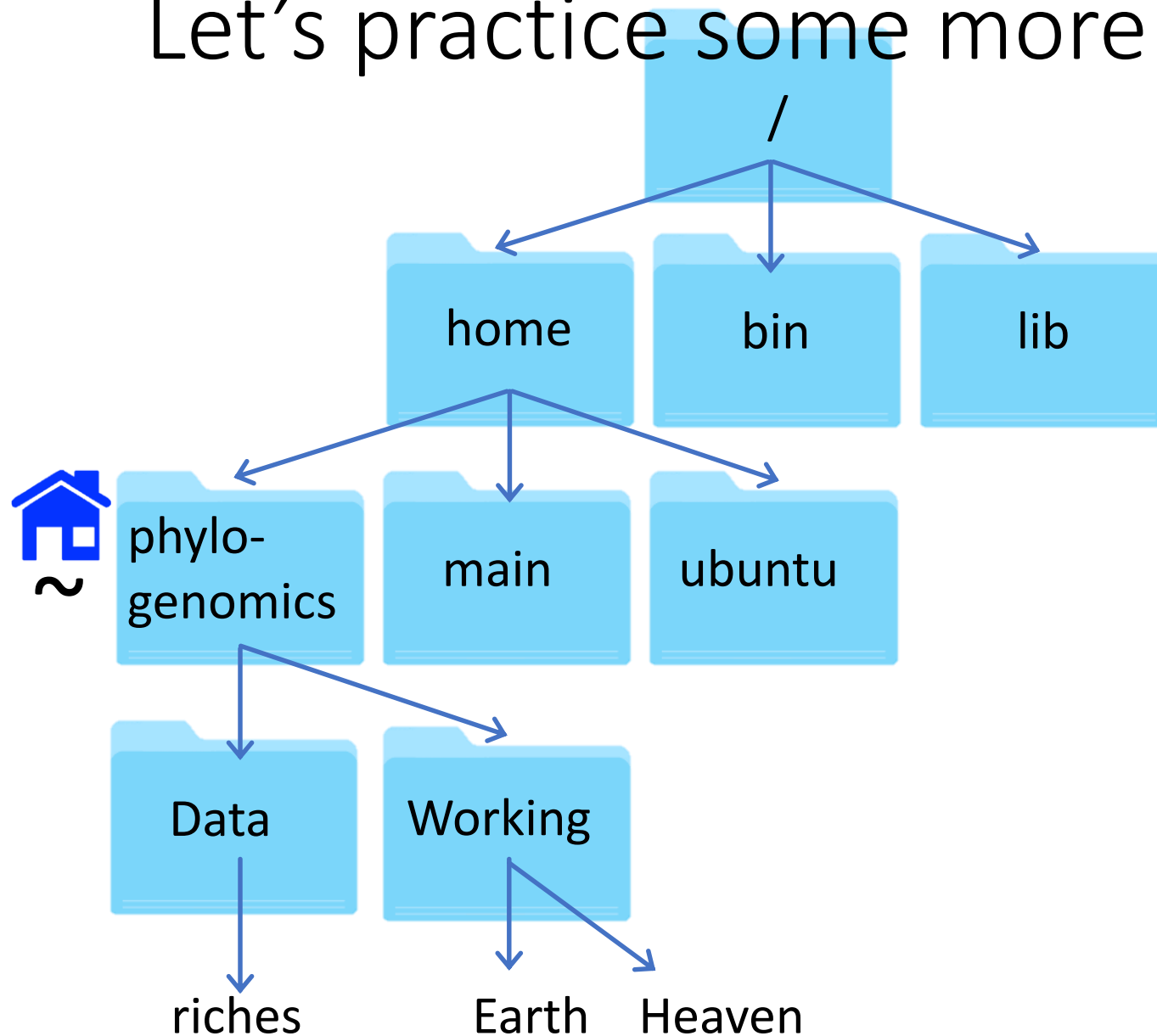
destination path

Now move Earth, too.

Remember to Tab complete!



Let's practice some more



Moving Files

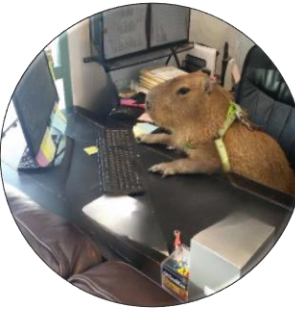
mv can also be used to rename files.

Let's change rags to riches.

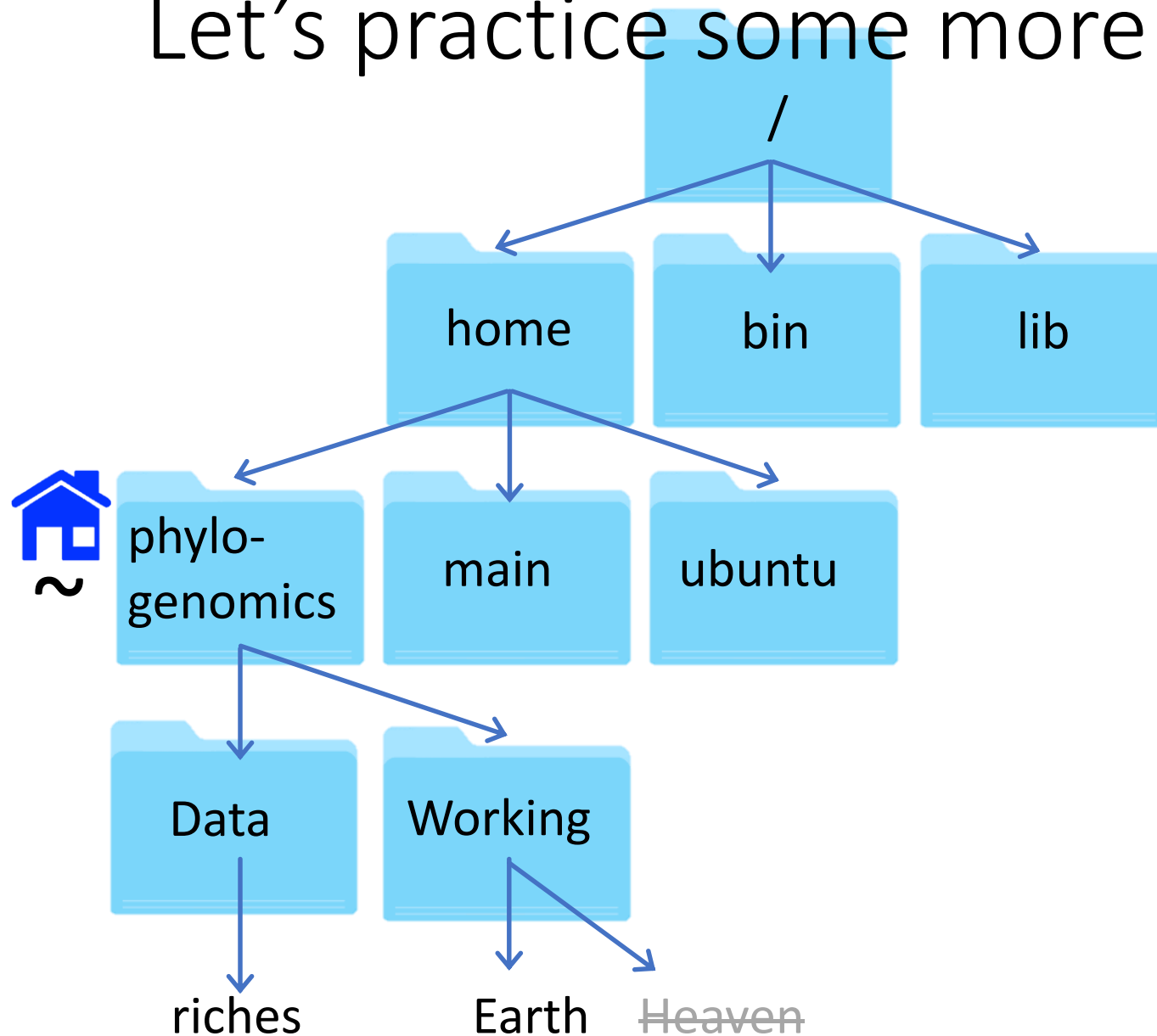
```
$ mv ./rags ./riches
```

file path

destination path



Let's practice some more



Deleting Files

Now let's delete Heaven

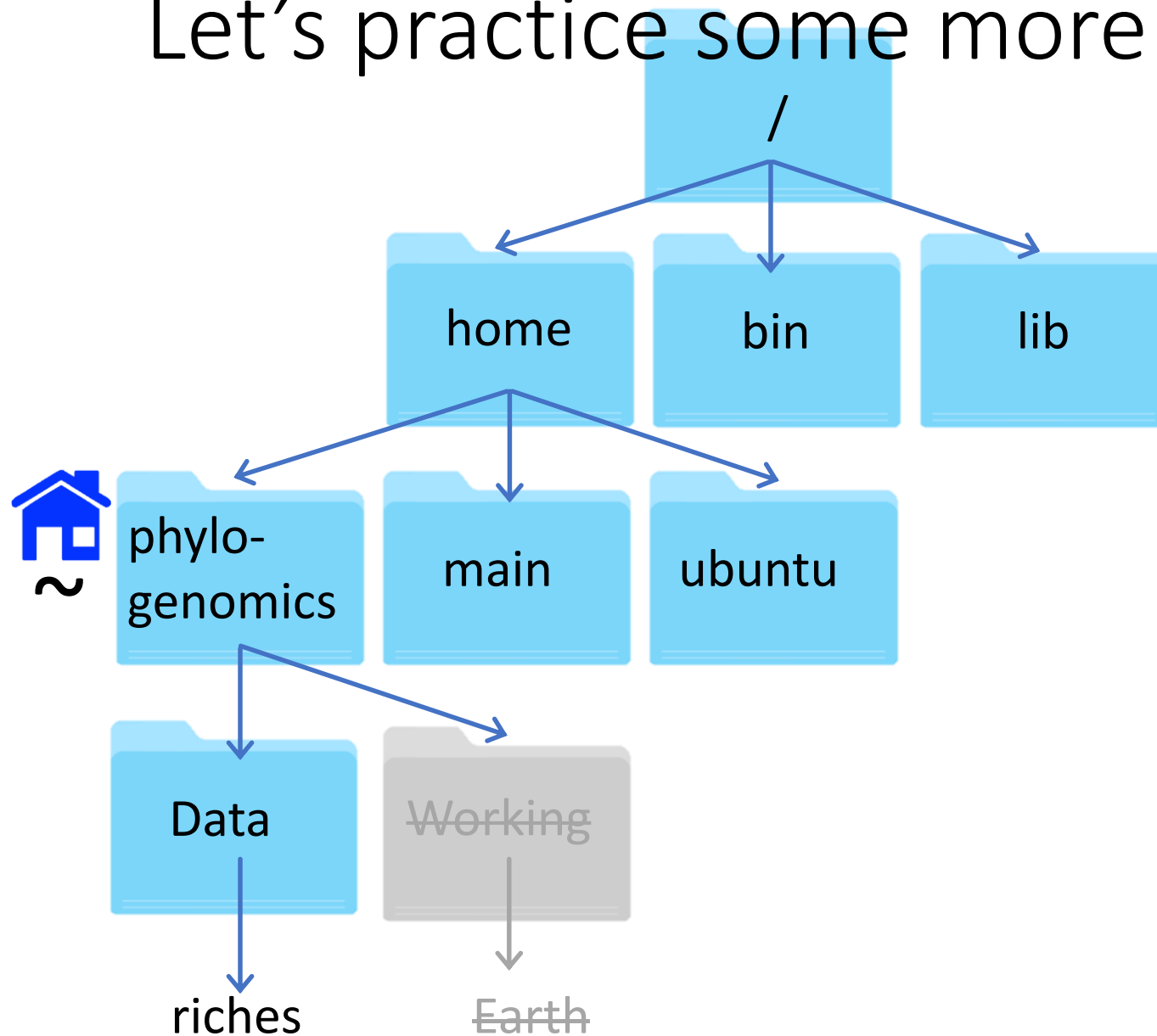
(Check your present working directory is Data)

```
$ rm -i ../Working/Heaven
```

When prompted type y for yes and press enter.



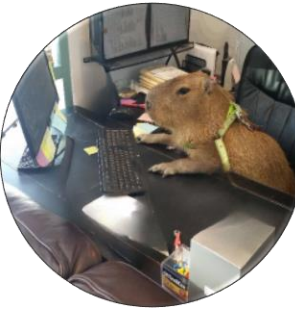
Let's practice some more



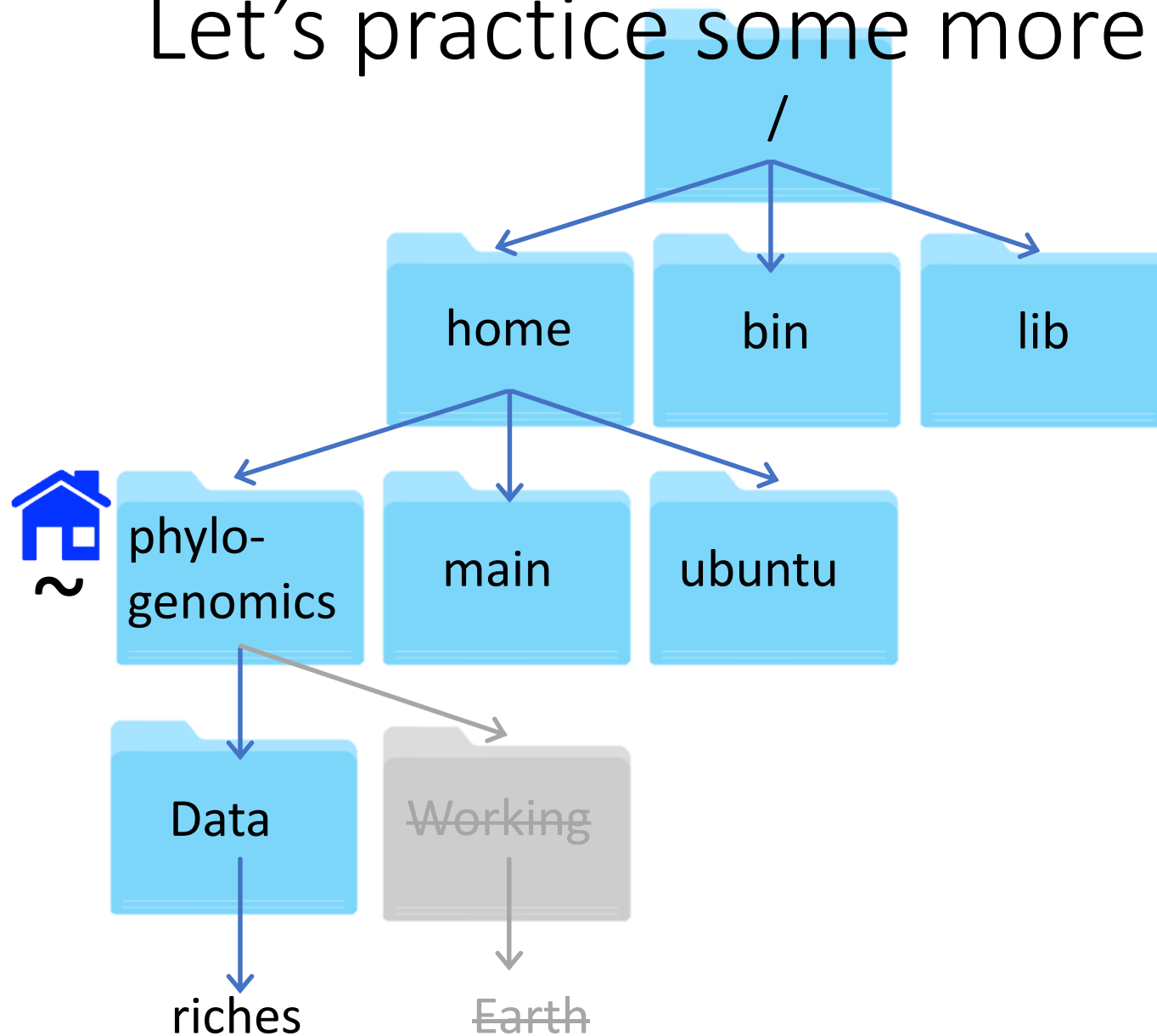
Deleting Files

Now let's delete the entire 'Working' directory including Earth.

```
$ rm -i ../Working/Heaven
```



Let's practice some more



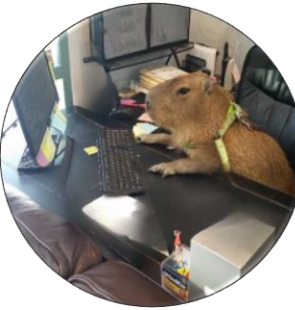
Deleting Files

There is no 'Trash' or 'Recycle Bin' in Unix!

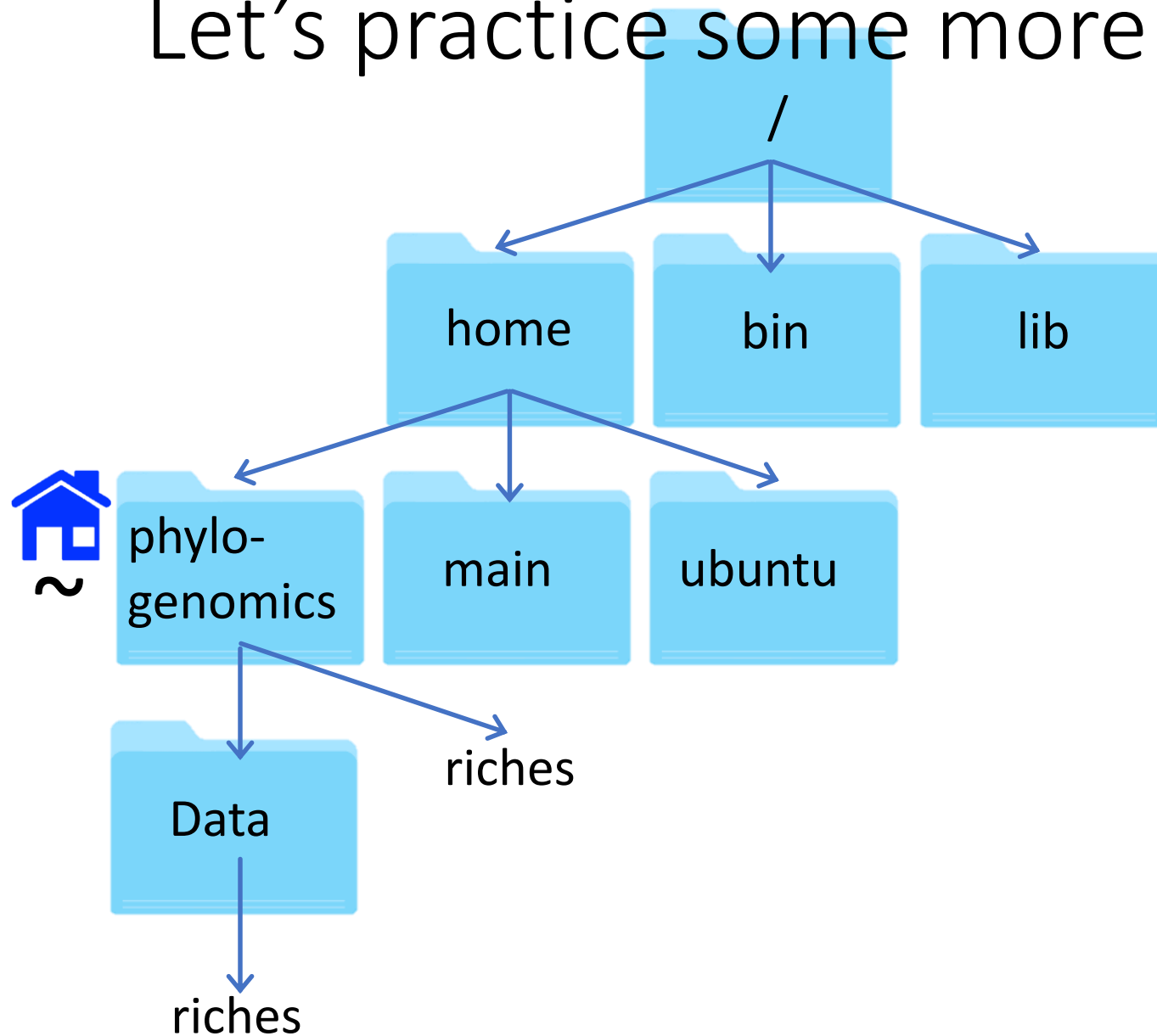
Once gone, files are gone forever!

Therefore try to ALWAYS use `rm -i`





Let's practice some more



Copying Files

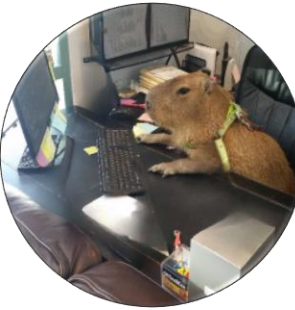
Let's make a copy of riches within the home directory.

(Make sure your present working directory is Data.)

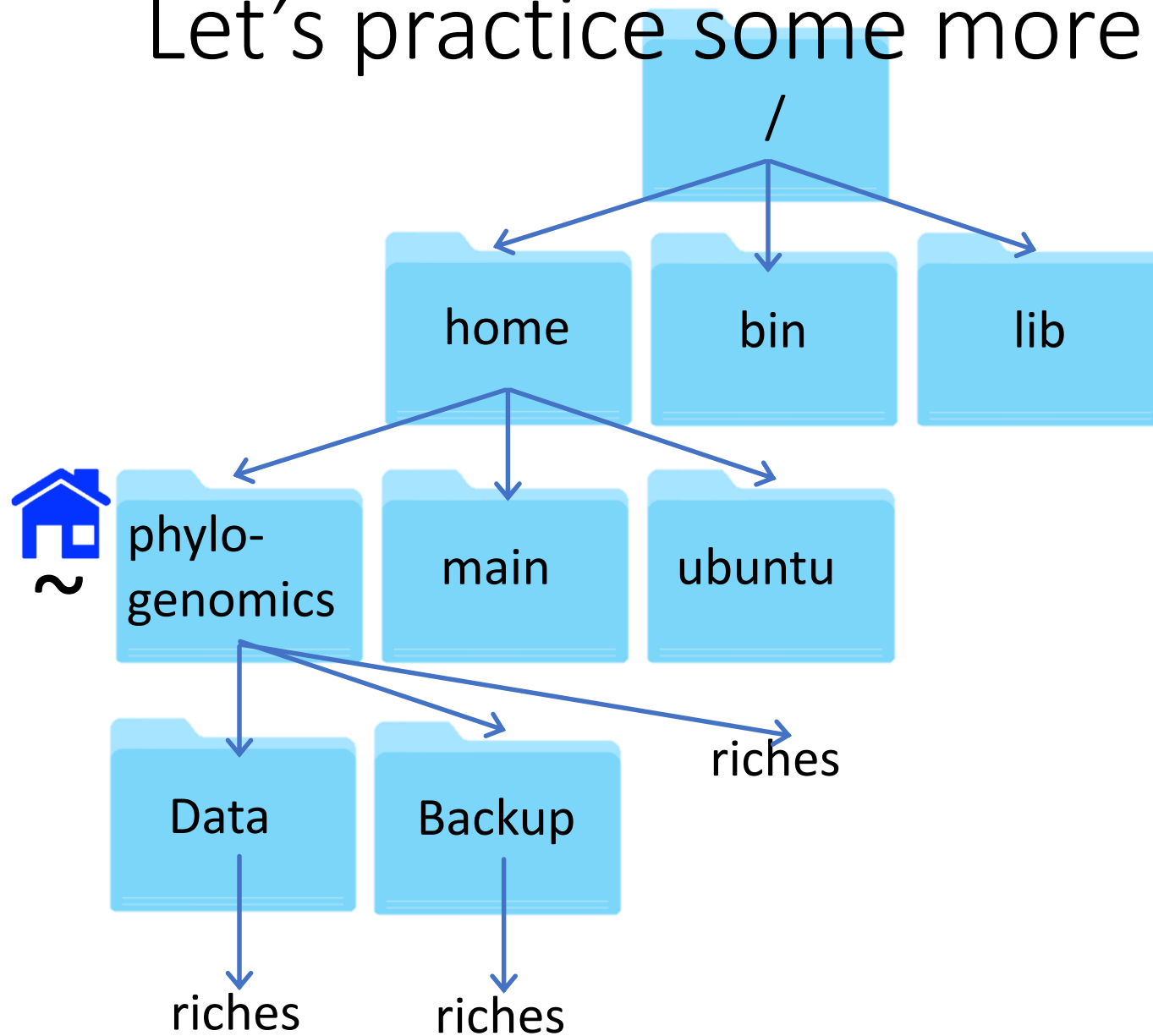
```
$ cp ./riches ../riches
```

file path

destination path



Let's practice some more



Copying Files

You can also copy entire directories and use this function to rename files/directories.

Move to home `$ cd ~`

Make a copy of the Data directory here and call it Backup.

```
$ cp -r ./Data ./Backup
```



Data management

Some files can become quite big so people will archive directories and compress large files so that they are easier to store or share. Here's an example: sequences.tar.gz

.tar – means that it is a tape archived directory

.gz – means that it is gzipped file

These can be used alone or in combination

To uncompress a tar archive (x = extract, v = verbose, f = all files)

```
$ tar -xvf <filename>
```

A Gzipped file

```
$ gunzip <filename>
```

A Gzipped Tar archive

```
$ tar -xzvf <filename>
```




Challenge 2

1. Change to the workshop_materials directory at the following path:
~/workshop_materials/unix

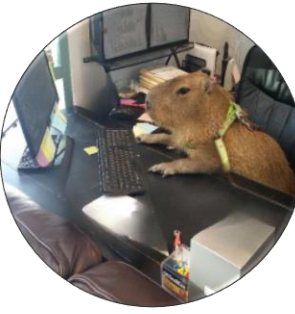
You should find a compressed directory: Sequences.tar.gz

2. Make a copy of this file in the Backup directory you created earlier
3. Un archive the original directory
4. Unzip the read files
5. Rename the unarchived files – sequence_1.fq and sequence_2.fq
6. Delete the original .tar file

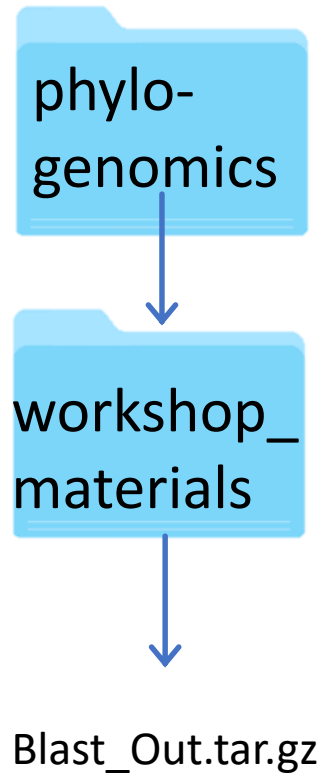
tar gunzip cp mv rm -i cd gunzip mv mkdir

Questions?





Viewing and manipulating files



Navigate to the workshop_materials directory

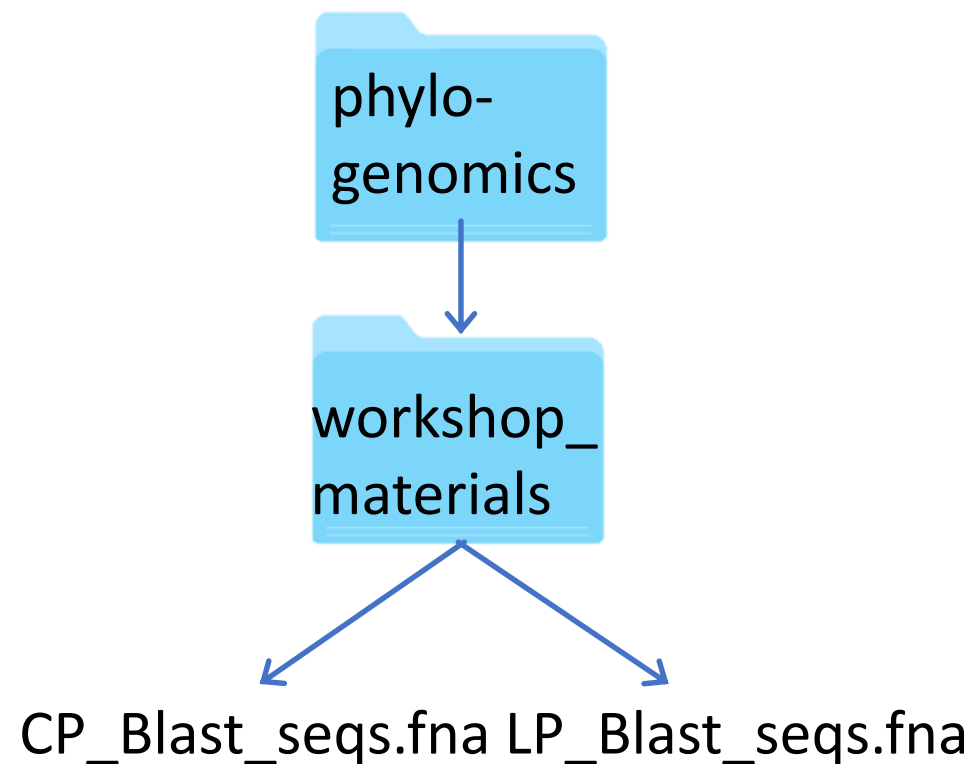
```
$ cd ~/workshop_materials
```

Unarchive the Blast_Out.tar.gz

```
$ tar -xzf Blast_Out.tar.gz
```



Viewing and manipulating files

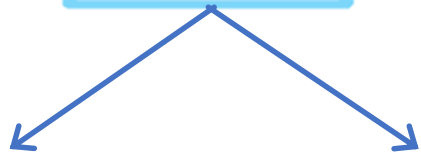
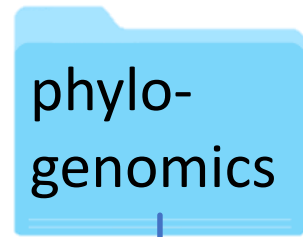


LINUX TERMINAL FOR BEGINNERS





Viewing and manipulating files



CP_Blast_seqs.fna LP_Blast_seqs.fna

View the first 10 lines of a file

```
$ head CP_Blast_seqs.fna
```

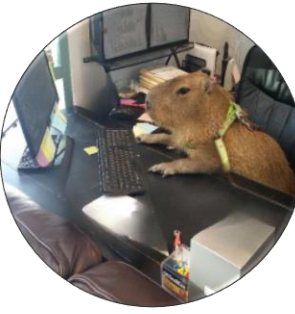
To view the first 30 lines of the file

```
$ head -n 30 CP_Blast_seqs.fna
```

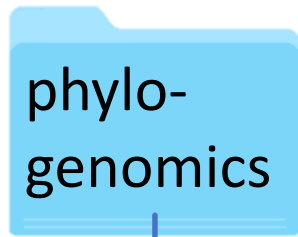
**LINUX TERMINAL
FOR BEGINNERS**

head





Viewing and manipulating files



CP_Blast_seqs.fna LP_Blast_seqs.fna

View the last 10 lines of a file

```
$ tail CP_Blast_seqs.fna
```

To view the last 30 lines of the file

```
$ tail -n 30 CP_Blast_seqs.fna
```

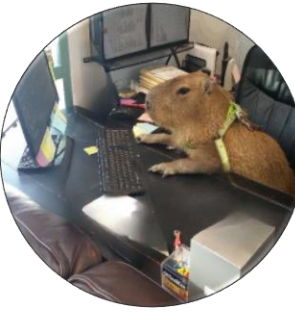
**LINUX TERMINAL
FOR BEGINNERS**

head

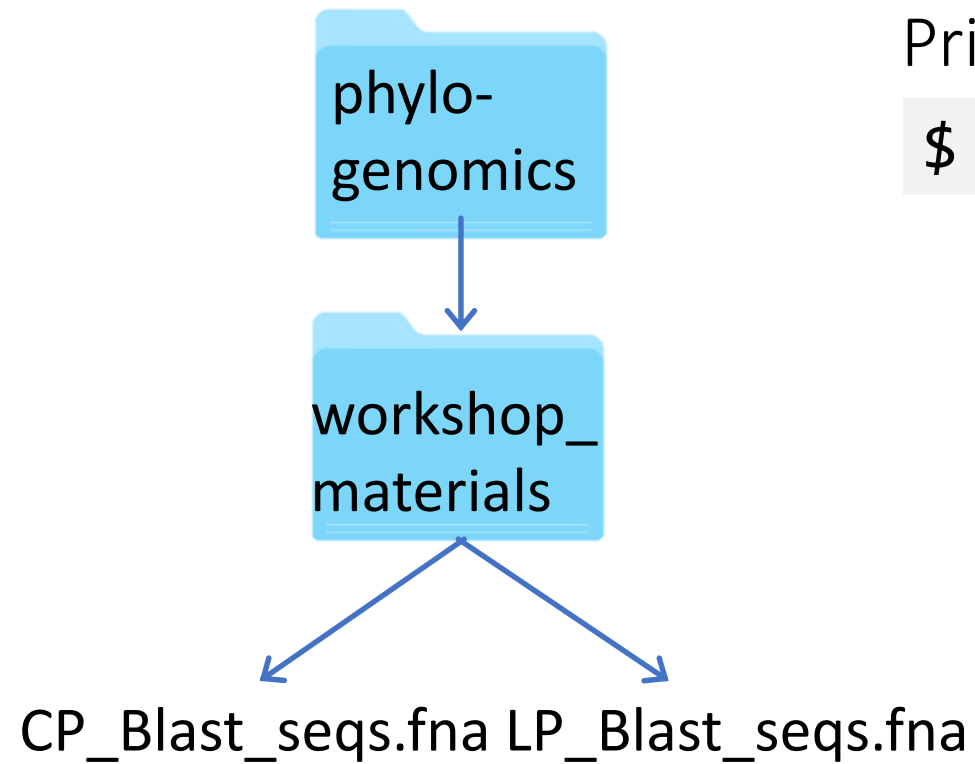


tail





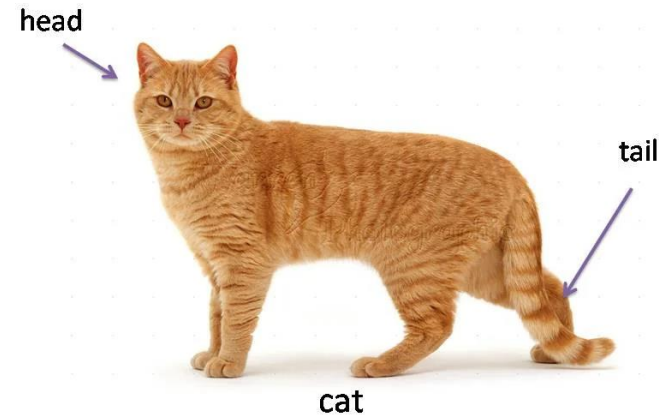
Viewing and manipulating files



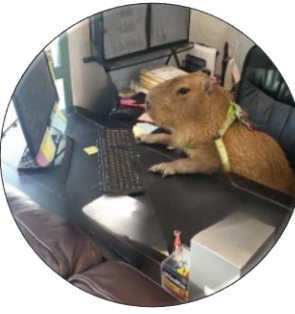
Print the entire file

```
$ cat CP_Blast_seqs.fna
```

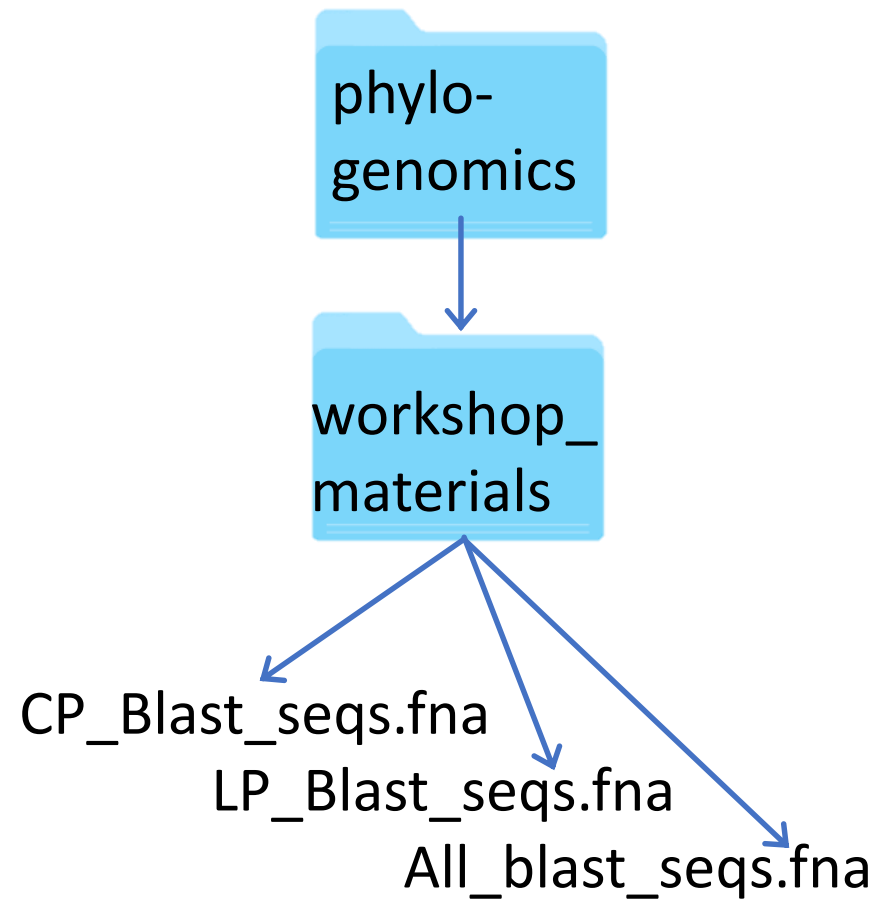
LINUX TERMINAL FOR BEGINNERS



Many files are too large to meaningfully view in terminal or to edit in a unix text editor.



Viewing and manipulating files

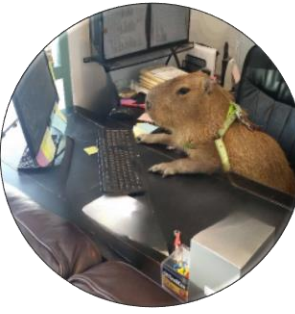


'cat' can also combine multiple files

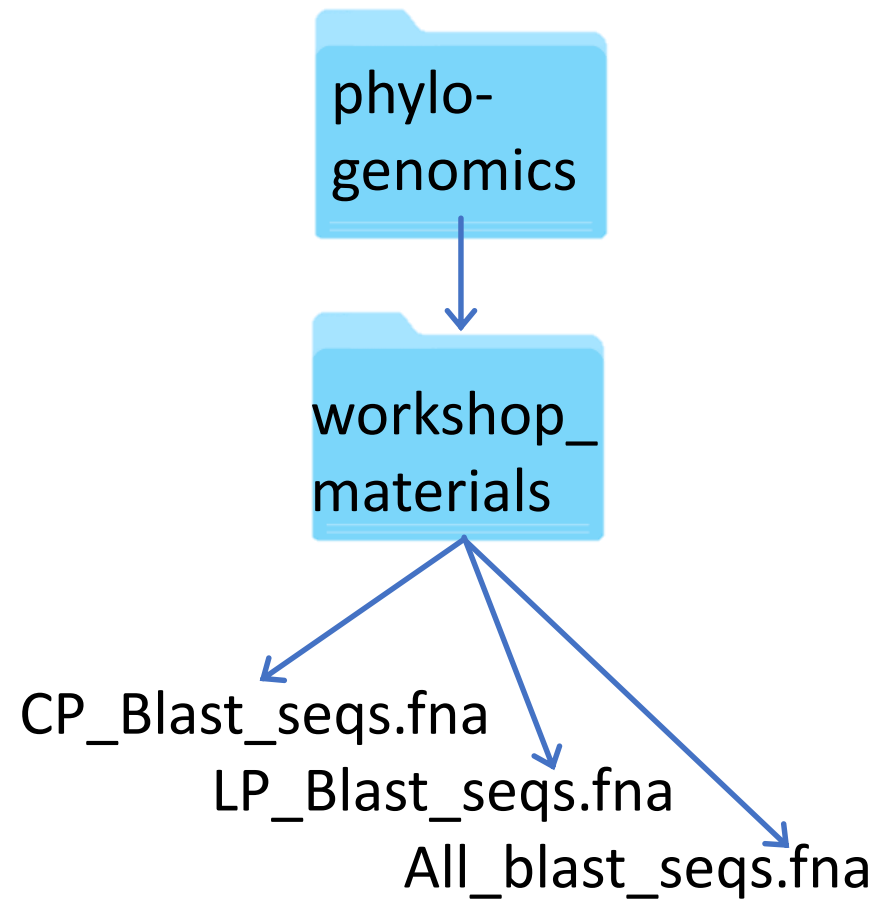
```
$ cat CP_Blast_seqs.fna LP_Blast_seqs.fna >  
All_blast_seqs.fna
```

And then count the number of lines in each fasta file to confirm they have been combined.

```
$ wc -l *.fna
```

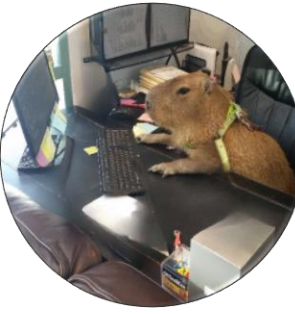


Viewing and manipulating files

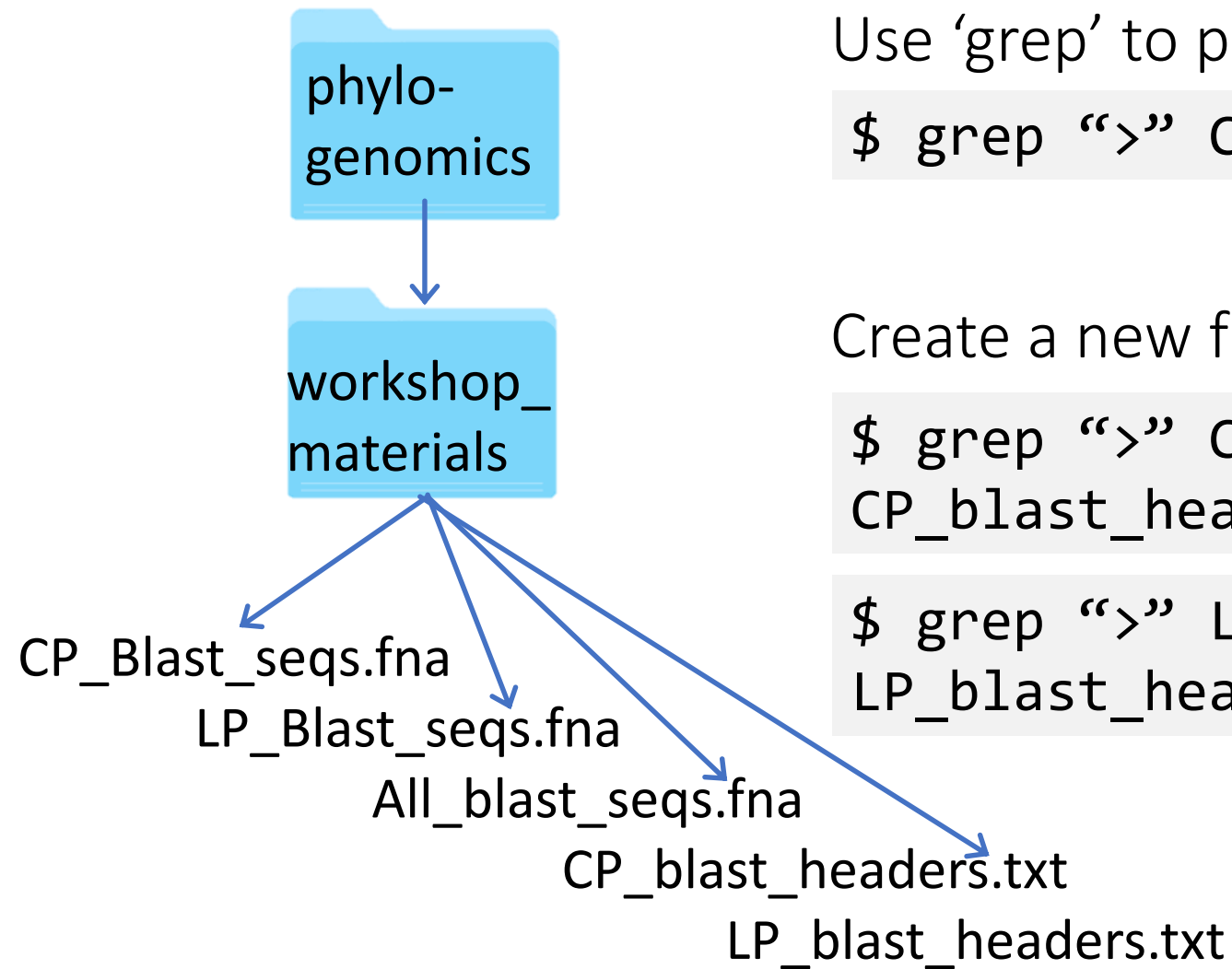


Use 'grep' to print occurrences of a pattern

```
$ grep ">" CP_Blast_seqs.fna
```



Viewing and manipulating files



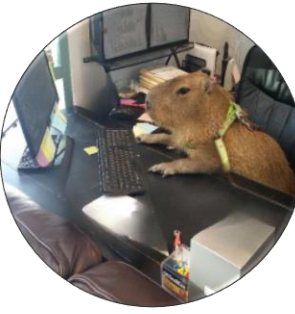
Use 'grep' to print occurrences of a pattern

```
$ grep ">" CP_Blast_seqs.fna
```

Create a new files of the fasta headers

```
$ grep ">" CP_Blast_seqs.fna >  
CP_blast_headers.txt
```

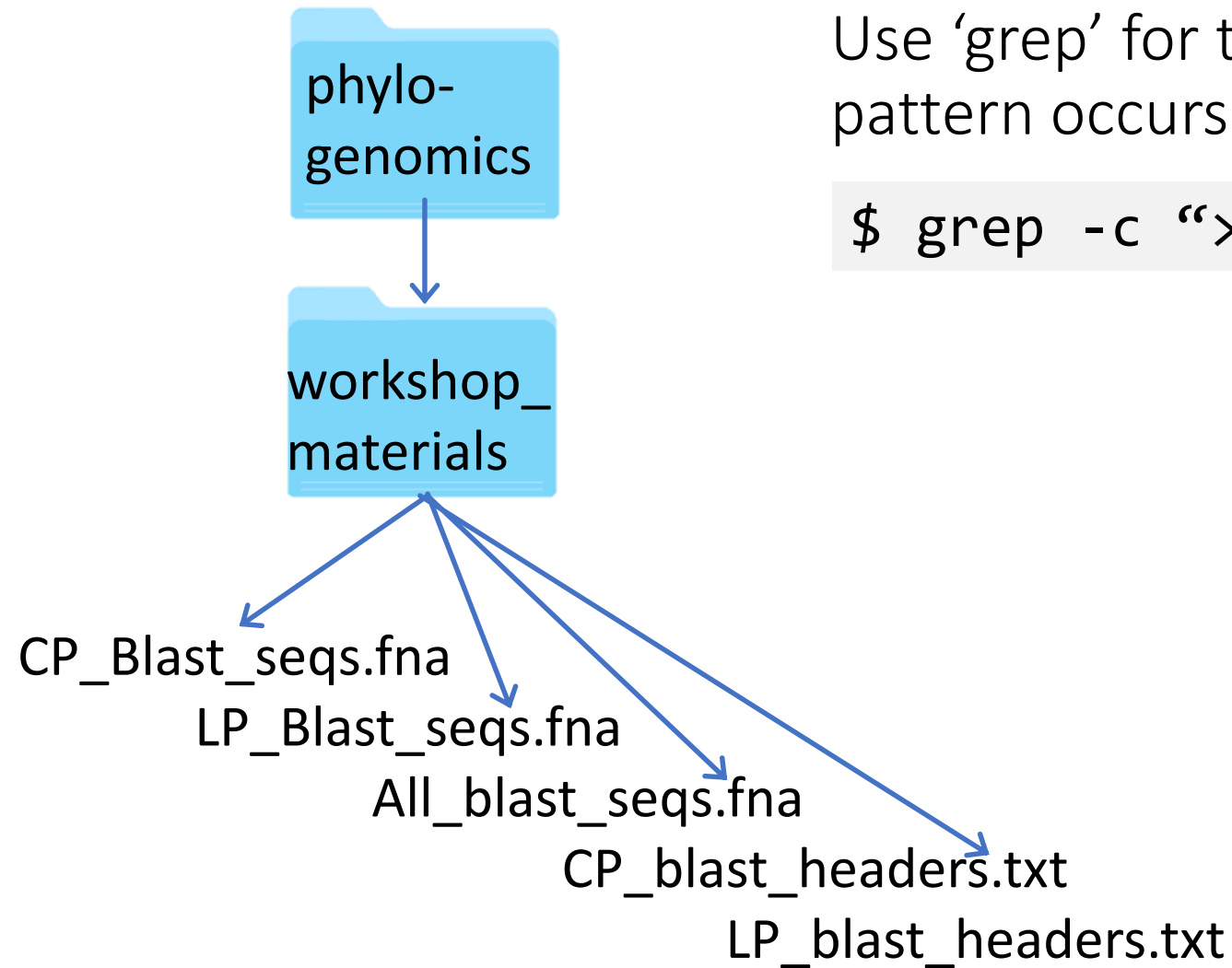
```
$ grep ">" LP_Blast_seqs.fna >  
LP_blast_headers.txt
```

Viewing and manipulating files

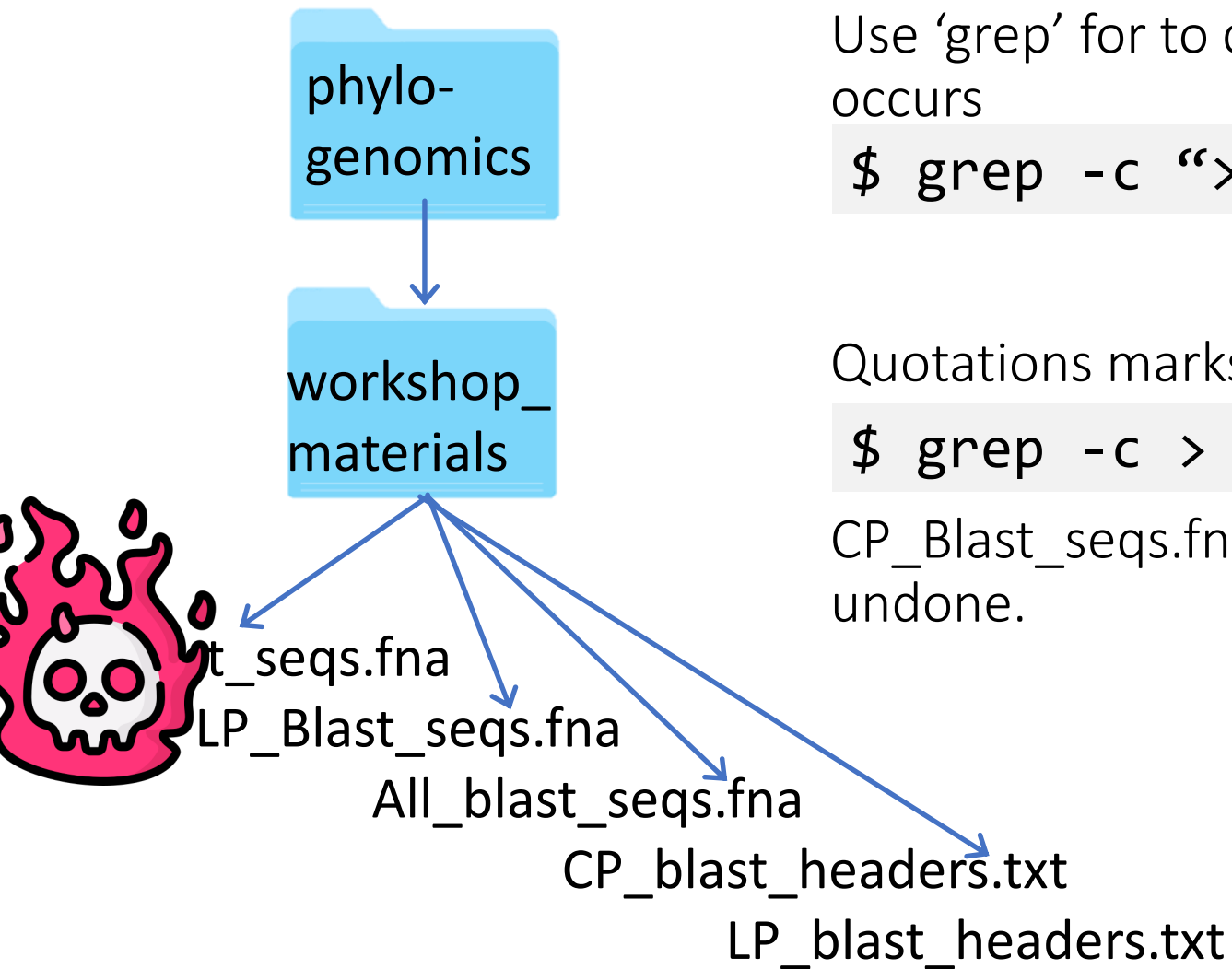
Use 'grep' for to count the number of times pattern occurs

```
$ grep -c ">" CP_Blast_seqs.fna
```





Viewing and manipulating files



Use 'grep' for to count the number of times pattern occurs

```
$ grep -c ">" CP_Blast_seqs.fna
```

Quotations marks are vital!

```
$ grep -c > CP_Blast_seqs.fna
```

CP_Blast_seqs.fna is now empty and this can't be undone.



Viewing and manipulating files

What can you do to your file to protect them? Change permissions

```
$ chmod 444 CP_Blast_seqs.fna
```



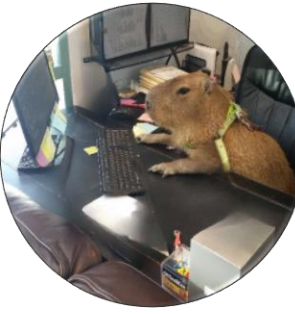
Viewing and manipulating files

What can you do to your file to protect them? Change permissions

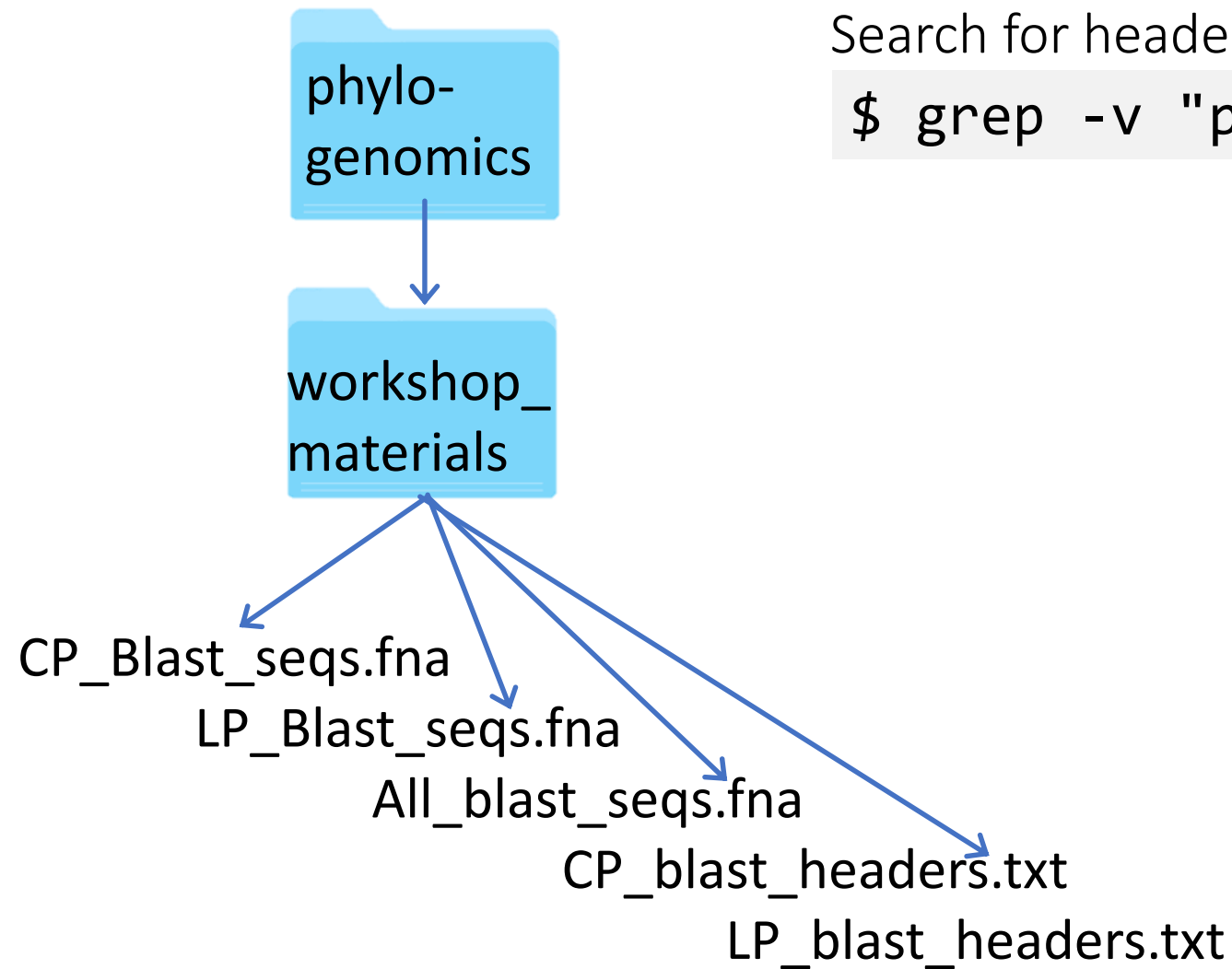
```
$ chmod 444 CP_Blast_seqs.fna
```

You might know and need this when you write scripts you want to execute. Sometimes you need to change permission to allow them to be executed.

```
$ chmod +x my_script.sh
```

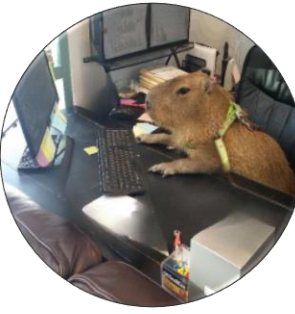


Viewing and manipulating files

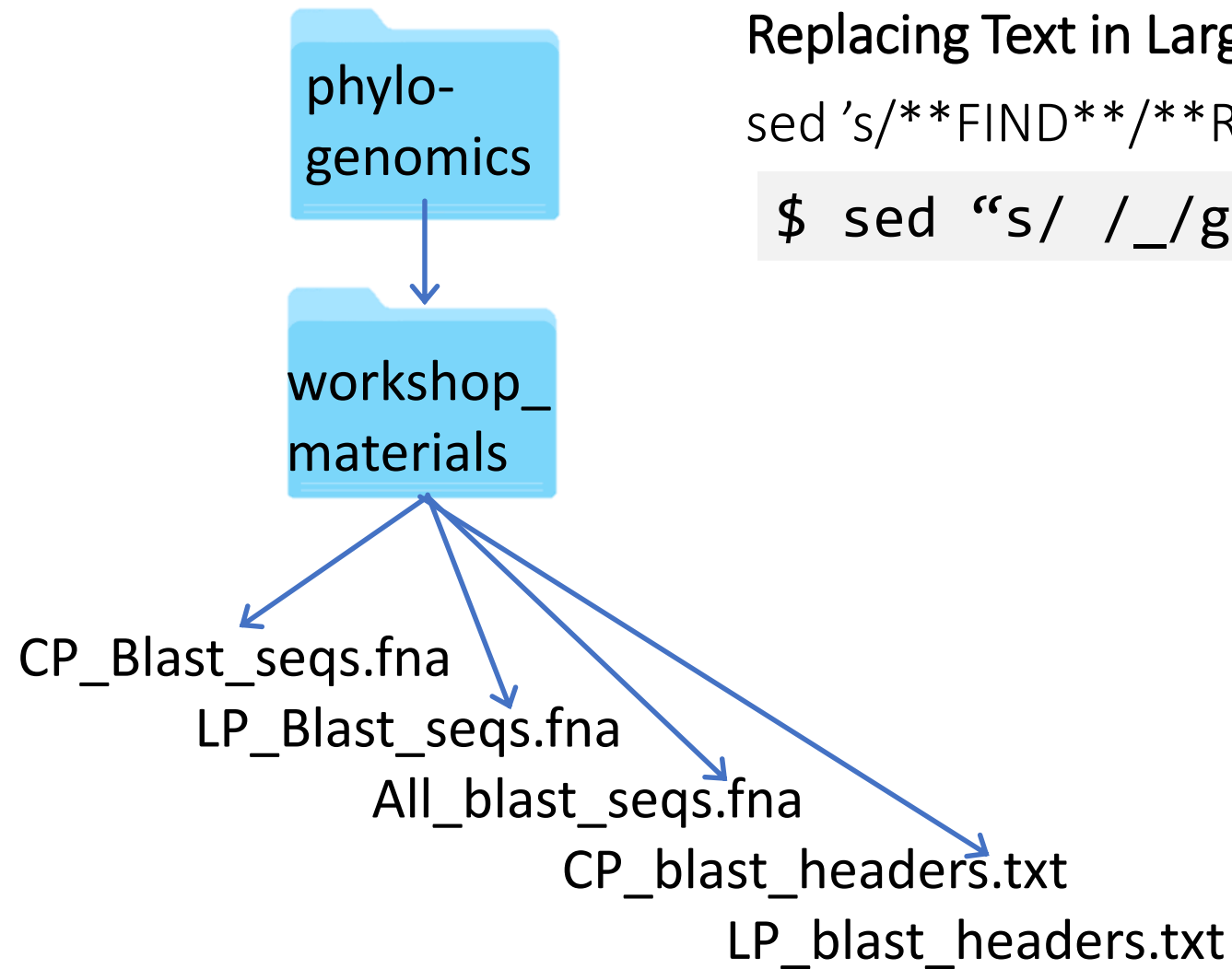


Search for headers that are not partial sequences

```
$ grep -v "partial" LP_headers.txt
```

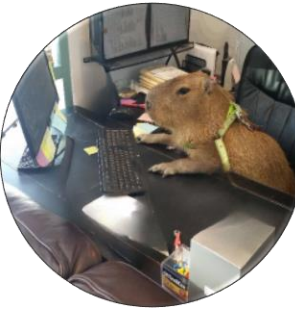
Viewing and manipulating files



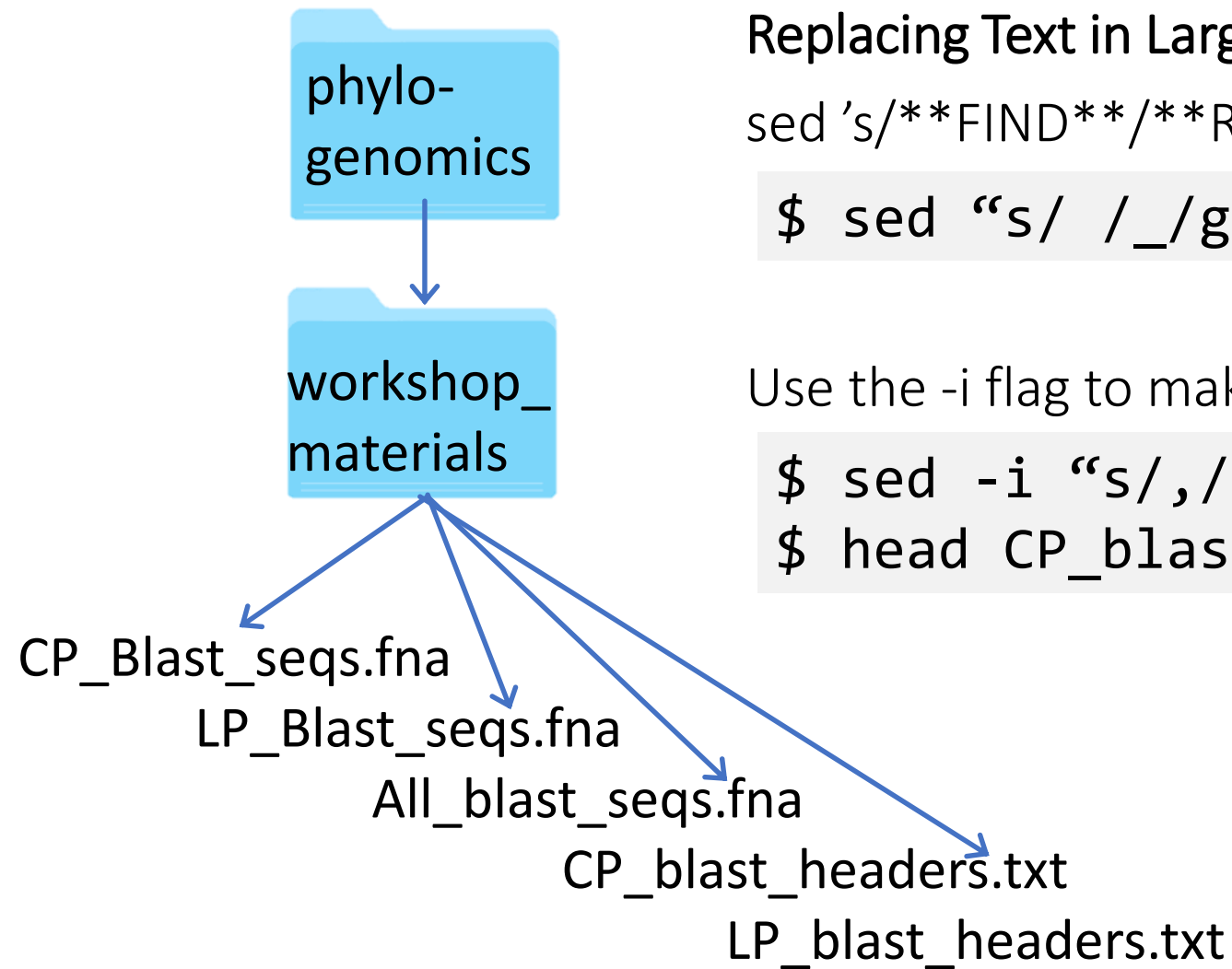
Replacing Text in Large files

```
sed 's/**FIND**/**REPLACE**/g' filename > output_file
```

```
$ sed "s/ /_/g" CP_blast_headers.txt
```



Viewing and manipulating files



Replacing Text in Large files

`sed 's/**FIND**/**REPLACE**/g' filename > output_file`

```
$ sed "s/ /_/g" CP_blast_headers.txt
```

Use the -i flag to make changes in place

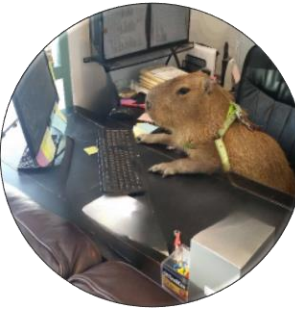
```
$ sed -i "s/,//g" CP_blast_headers.txt  
$ head CP_blast_headers.txt
```

Regular expressions



Expression	Modern equivalent	Pattern matched
.		a single character
.+		one or more characters
.*		zero or more characters
.?		maybe present
^		first in the line
\$		last in the line
[0-9]	\d	digits
[a-zA-Z]	\w	letters
' '	\s \t	space
{2}		exactly 2 characters long
{2,4}		between 2 and 4 characters long
[ACGT]		a specific set of characters

Regular expressions



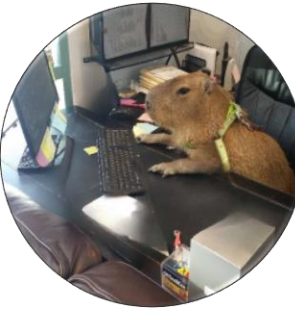
Use sed -E to use extended regular expressions

```
$ sed -E 's/(>[A-Z0-9.]+)(.+)/\1/' CP_blast_headers.txt
```

Store pattern in memory using parentheses

Print out only the GenBank accessions

Regular expressions



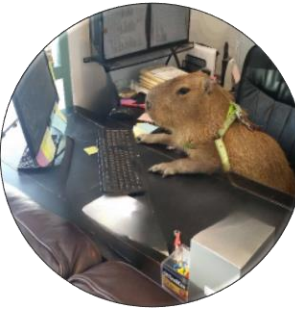
Use sed -E to use extended regular expressions

```
$ sed -E 's/(>[A-Z0-9.]+)(.+)/\2/' CP_blast_headers.txt
```

Store pattern in memory using parentheses

Print out the rest of the headers

Pipes |



Combine the fasta files from the blast output and identify how many sequences they contain

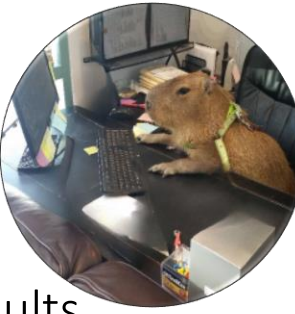
```
$ cat CP_Blast_seqs.fna LP_Blast_seqs.fna | grep ">" | wc -l
```

When building pipelines, it is useful to pipe to head to follow along

```
$ cat CP_Blast_seqs.fna LP_Blast_seqs.fna | head
```

```
$ cat CP_Blast_seqs.fna LP_Blast_seqs.fna | grep ">" | head
```

```
$ cat CP_Blast_seqs.fna LP_Blast_seqs.fna | grep ">" | wc -l
```

Pipes |

String together many commands to count the number of unique accessions from these blast results

```
$ cat *.fna
```

 Print out all fasta files

```
$ cat *.fna | grep ">"
```

 Find all the fasta headers

Extract accessions

```
$ cat *.fna | grep ">" | sed -E "s/(>[A-Z0-9.]+)(.+)/\1/"
```

Sort accessions

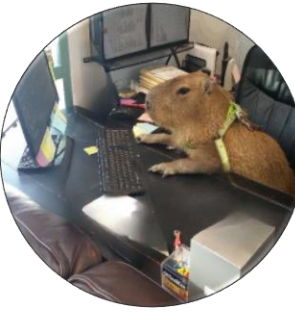
```
$ cat *.fna | grep ">" | sed -E "s/(>[A-Z0-9.]+)(.+)/\1/" | sort
```

Keep only unique accessions

```
$ cat *.fna | grep ">" | sed -E "s/(>[A-Z0-9.]+)(.+)/\1/" | sort | uniq
```

Count unique accessions

```
$ cat *.fna | grep ">" | sed -E "s/(>[A-Z0-9.]+)(.+)/\1/" | sort | uniq | wc  
-l
```



Challenge 3

Imagine you want to obtain the GenBank Accessions from the headers and you want to remove the version. Why is this command not working as expected?

```
$ sed -E "s/(>[A-Z0-9.]+)(.+)/\1/" CP_blast_headers.txt |  
sed "s/.1//g"
```

What would you need to change to make it work?



Loops

Iterate over e.g. files to execute a command repeatedly.

```
$ for i in *; do echo $i; done
```

Assign variable “i” to all files that end in .fna, then carry out command on all values of “i”.

```
$ for i in *.fna; do grep -c “CAT” $i; done
```

There is usually more than one way to do things. Try the following:

```
$ grep -c “CAT” *.fna
```



Downloading

From an internet URL

```
wget <url>
```

```
wget -P /path/to/where/the/download/should/be <url>
```

Save your data every day

We will launch new instance daily so everything you have done today will be gone tomorrow. Use scp or rsync to copy the notes you want to save to your own computer,

```
scp -r phylogenomics@<your.IP.address>:/Location/On/Instance /Local/path
```

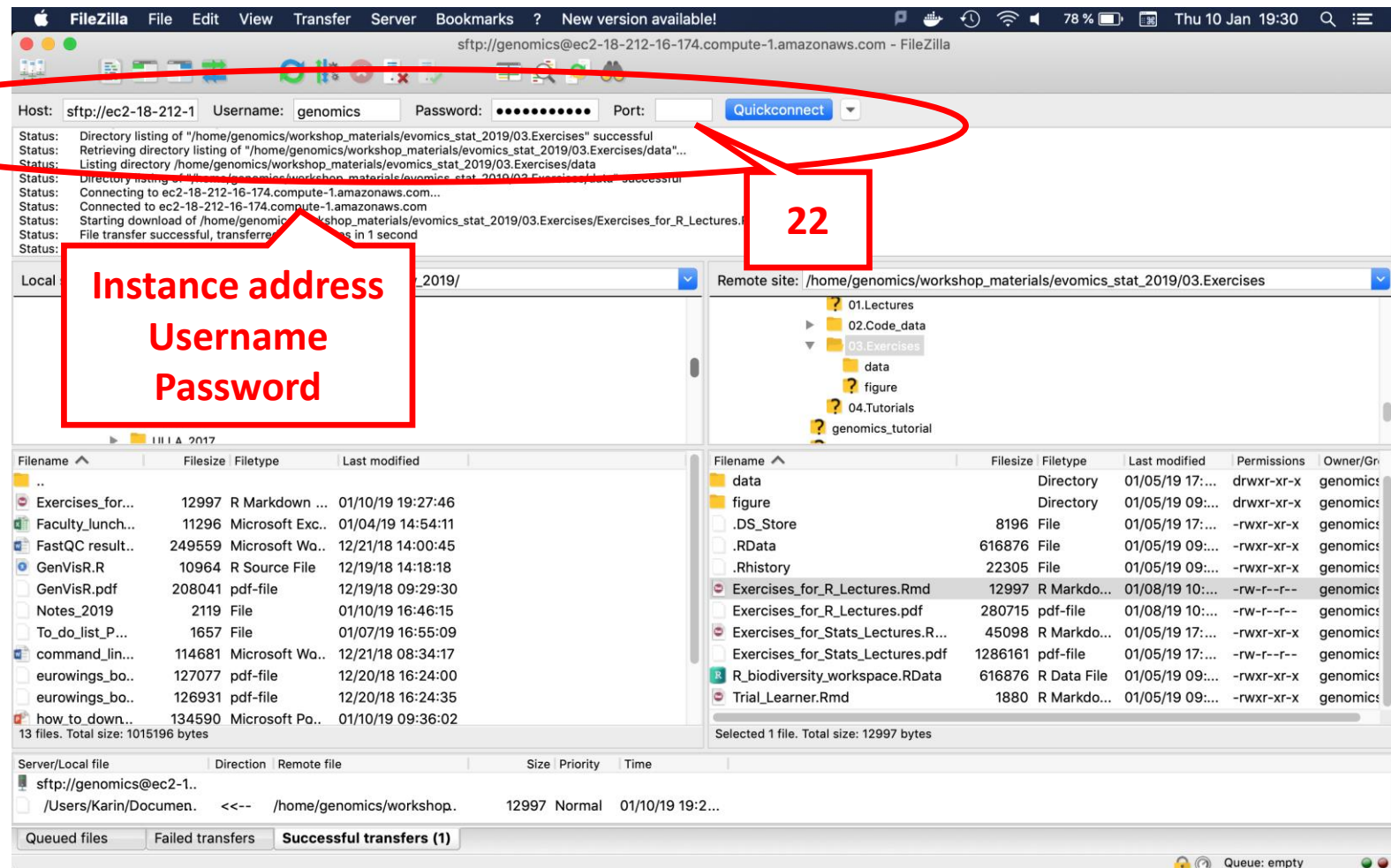
```
rsync -avz phylogenomics@<your.IP.address>:/Location/On/Instance /Local/path
```

```
$ rsync -avz phylogenomics@ec ... .
```

Downloading



GUI: Filezilla <https://sourceforge.net/projects/filezilla/>



Logging in to the instance through your local terminal

`ssh -Y phylogenomics@<your.IP.address>`

Accept and continue the connection with 'yes'

```
Karin — steffk1@gw344:/nobackup/rokaslab/steffk1/Shen_etal_2020/alternaria_phylo_inference/ma...
(anaconda3) IFB-FK0G-Karin:~ Karin$ ssh -Y genomics@52.91.211.21
The authenticity of host '52.91.211.21 (52.91.211.21)' can't be established.
ED25519 key fingerprint is SHA256:eH/xB39SR120qz5q/7nI+tWI04fYpw2IaxFuVB8RMLo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

```
Karin — phylogenomics@ip-172-31-91-145: ~ — ssh -Y phylogenomics@52.91.211.21 — 104x31
(anaconda3) IFB-FK0G-Karin:~ Karin$ ssh -Y phylogenomics@52.91.211.21
phylogenomics@52.91.211.21's password:

#####
##      Workshop on Phylogenomics 2024      ##
##      Cesky Krumlov                        ##
##      @evomics #evomics2024                ##
#####

Welcome to Ubuntu 22.04.3 LTS (6.2.0-1017-aws).

System information as of Fri Jan 19 16:10:20 CET 2024

System load: 0.27392578125      Processes:           234
Usage of /:  32.9% of 484.63GB  Users logged in:     0
Memory usage: 12%              IPv4 address for docker0: 172.17.0.1
Swap usage:  0%                IPv4 address for ens5:  172.31.91.145
Last login: Wed Jan 17 14:31:44 2024 from 194.228.207.2
phylogenomics@krumlov:~$
```

Questions?



Quests!

*Do these quests in whichever order you like.
Whatever sounds interesting to you.*

Easy

- open, modify and exit with saving a text file in an editor of your choice: nano, vim, emacs
- explore the commands: tr, cut, less, tree
- How do you append to a file without overwriting it?

You haven't heard about all of this. Use your creativity, neighbor and the internet to tackle these!

Fun

Beautiful figures: [Tidy Tuesday](#), [R graph gallery](#), [xkcd style](#), color choices ([Ghibli](#), [Wes Anderson](#), [I want hue](#), [Scico](#)))

Challenging

- what about the programming language 'awk'? :P

What are your favorite one-liners or UNIX tricks? (Collect them in a Google Docs!)