

IBiS Computational Biology Bootcamp

September 8: Cook 3118, 12 - 5 PM

September 9: Annenberg G02, 12 - 5 PM

September 10: Annenberg G02, 9 AM - 1 PM

September 11: Annenberg G02, 9 AM - 1 PM



Unsolicited advice about computational work



Don't be afraid



It will be rewarding



It works



Stop using Excel

Use our wiki

The screenshot shows a GitHub repository page for `AndersenLab/IBiS-Bootcamp`. The top navigation bar includes links to various services like Apple, Google Maps, YouTube, Wikipedia, Gmail, Calendar, NU mail, NU PO system, iBuyNU, myNU, NUHR, CapitalOne360, Vault – Document Manager, iCloud, Facebook, Twitter, Wikipedia, Yahoo!, News, Popular, and a Reader link.

The repository header shows `AndersenLab / IBiS-Bootcamp`, a profile picture for `synmuv`, and buttons for Unwatch (6), Star (0), and Fork (0).

The main content area displays the repository's activity: 32 commits, 1 branch, 0 releases, and 2 contributors. The `branch: master` dropdown is selected. A list of commits is shown:

- Fixed DL238 script! by `daniellecook` (authored 18 hours ago, latest commit `5e7988d6c6`)
- Add sorter data (2 days ago)
- atom selections example (3 days ago)
- Fixed DL238 script! (18 hours ago)
- Initial commit (a month ago)
- Update README.md (6 days ago)

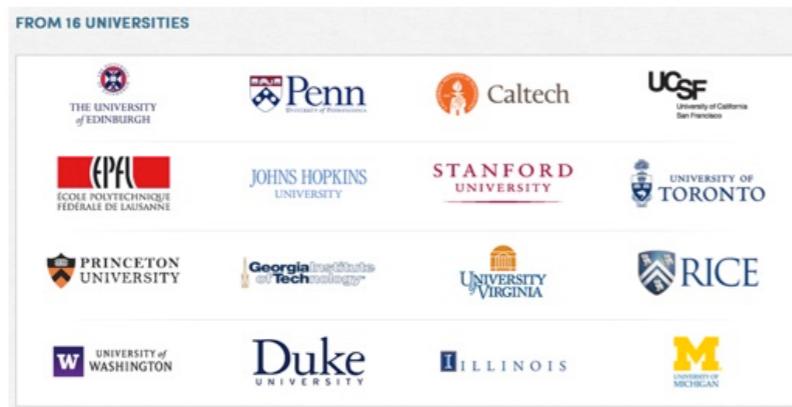
A section titled `IBiS-Bootcamp` contains the text: "This repository will store files and scripts for the 2014 IBiS Computational Biology Bootcamp." Below this, a link "Check out our Wiki" is circled in red.

The right sidebar provides links to Code, Issues (0), Pull Requests (0), Wiki, Pulse, Graphs, and Settings. It also includes an HTTPS clone URL field with the value `https://github.com/` and buttons for "Clone in Desktop" and "Download ZIP".

[www.github.com/AndersenLab/IBiS-Bootcamp](https://github.com/AndersenLab/IBiS-Bootcamp)

The web is filled with free books,
instructional tools, courses, etc.

Coursera



Data Science Specialization

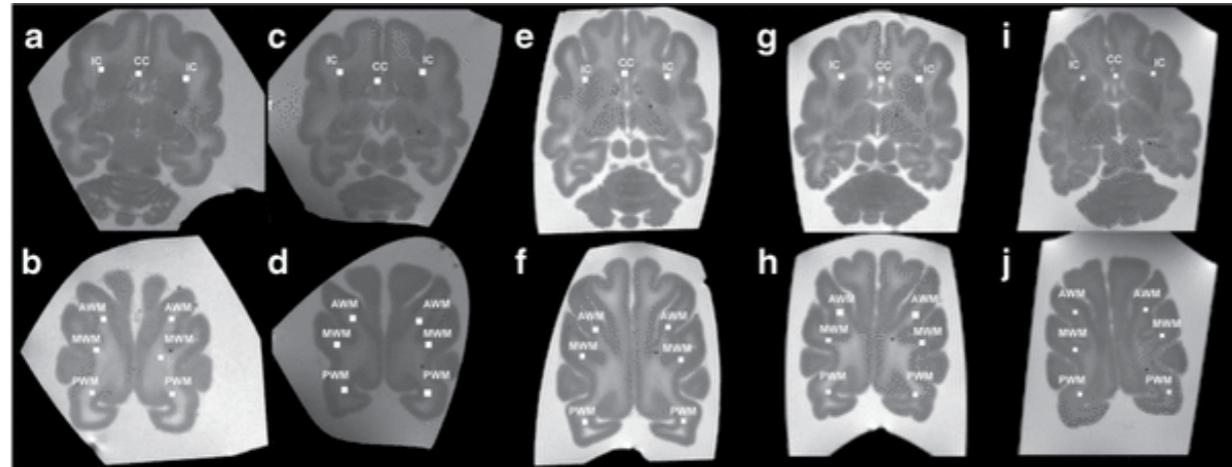


codecademy

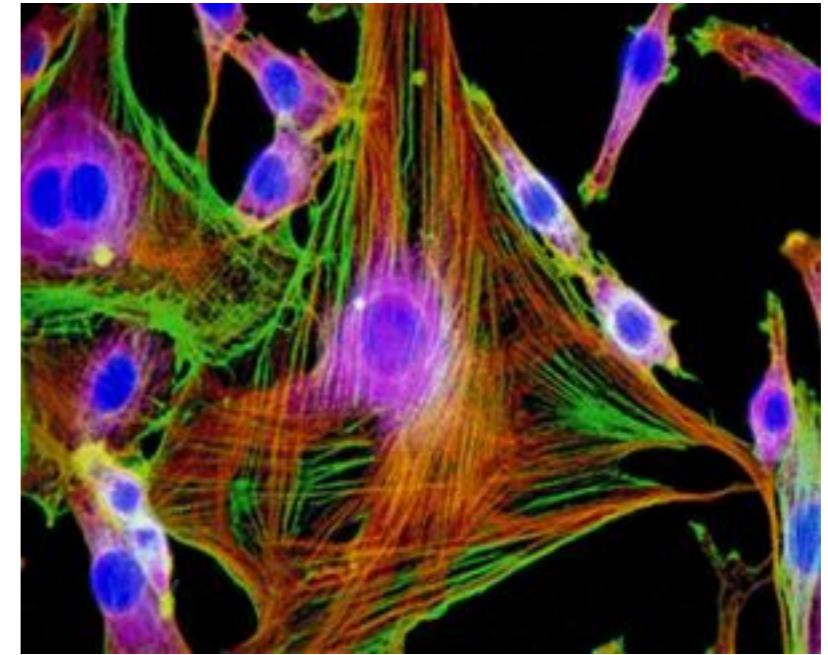
ROSALIND

software carpentry

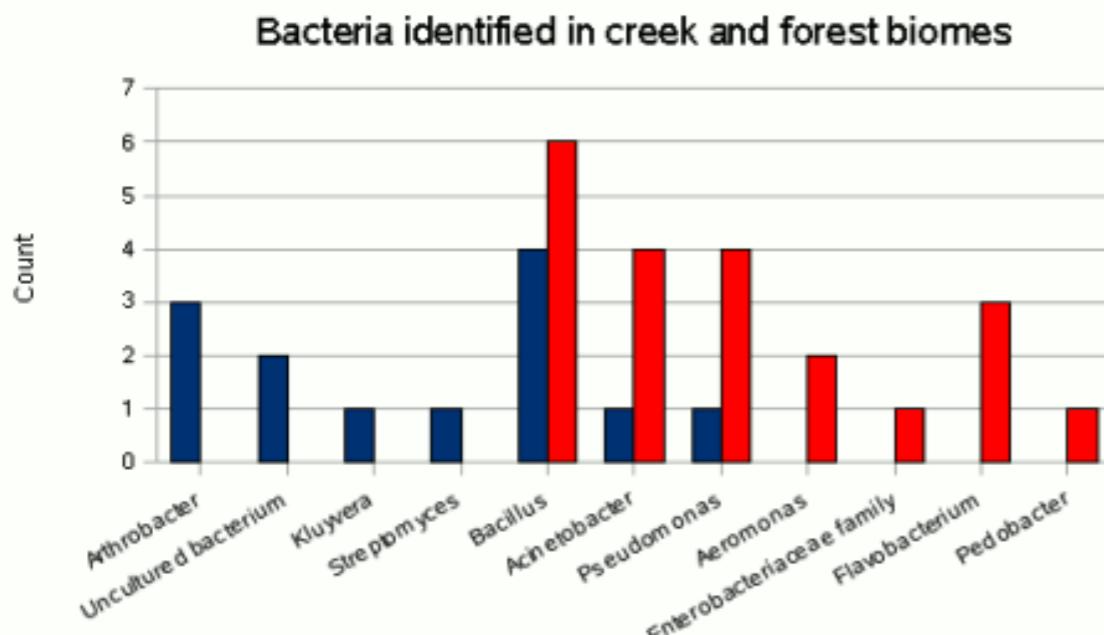
The days of descriptive biology are [almost] over



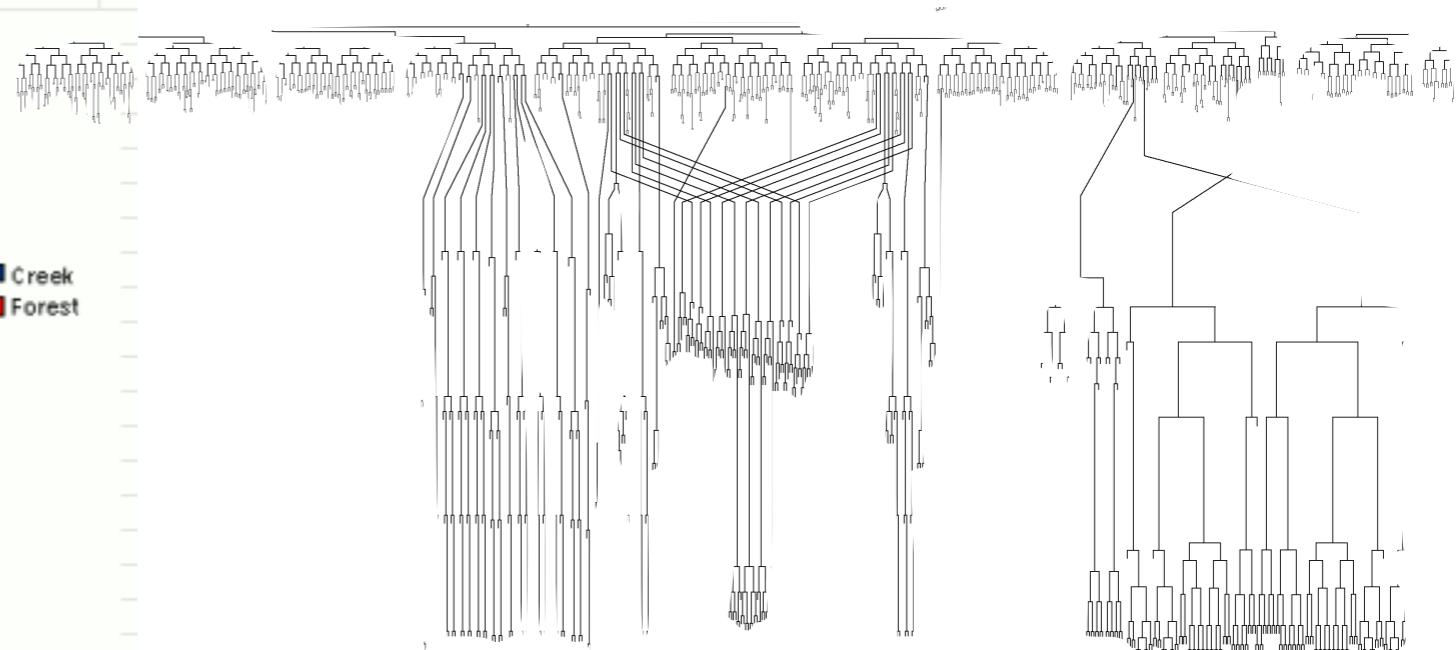
Reproducibility?



Sampling?



Error?



Automation?

NOW is the most exciting time to be a biologist



Any biologist can collect data.

Few biologists can process and analyze them.



Bootcamp outline

Day #1: Basic command line interface,
reproducible research, and sequence alignment

Day #2: Intro to R and RStudio, data input, and
cleaning

Day #3: R and data manipulation, processing

Day #4: R and data plotting, presentations

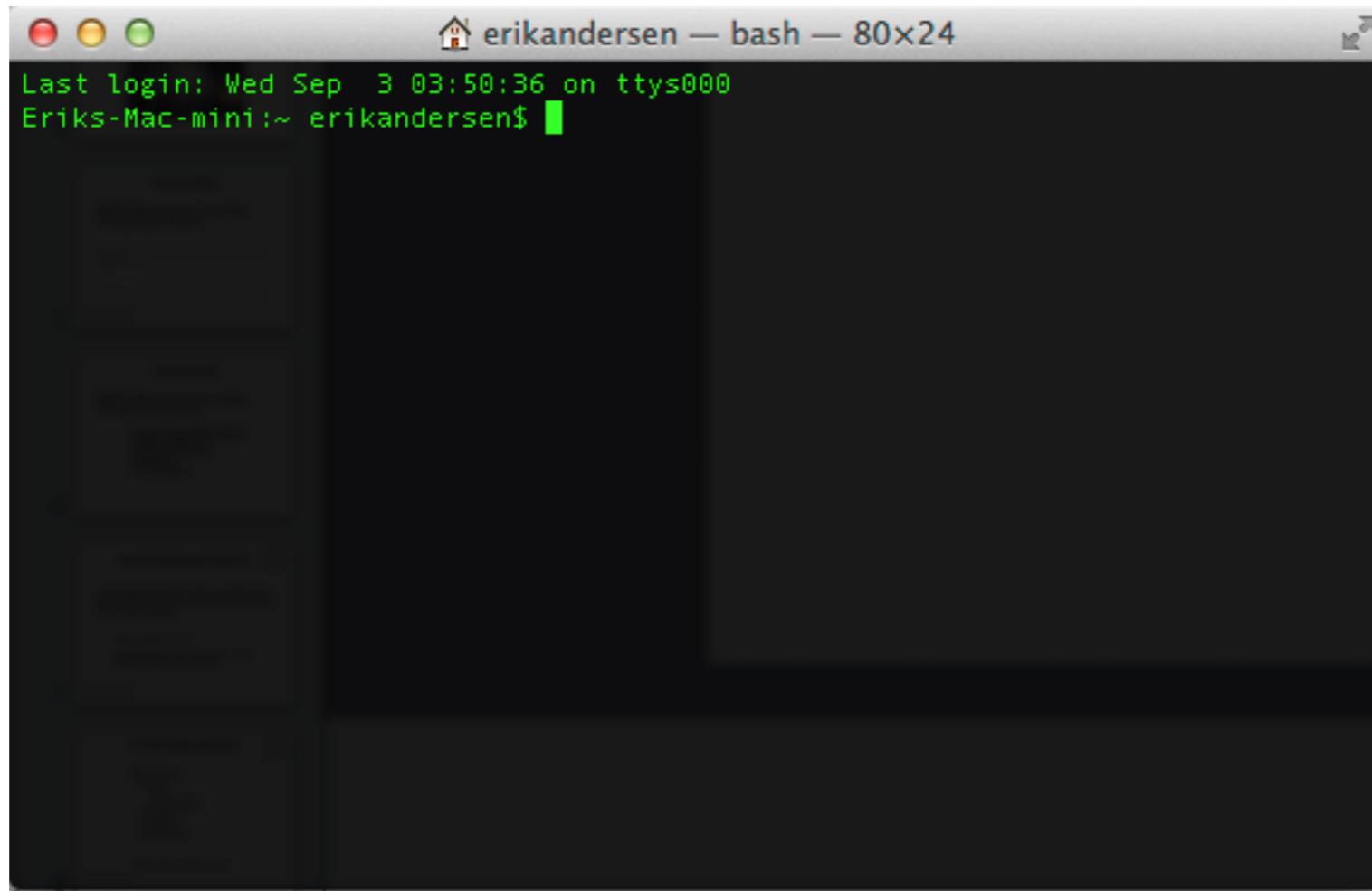
Bootcamp outline

Day #1: Basic command line interface, reproducible research, and sequence alignment

- **Intro to reproducible research**
- **Basic command line**
- **Directory structures**
- **Markdown**
- **Git and github**
- **More command-tool utilities**
- **Sequence alignment**

Notes vs what you type

Command-line interface on a Mac



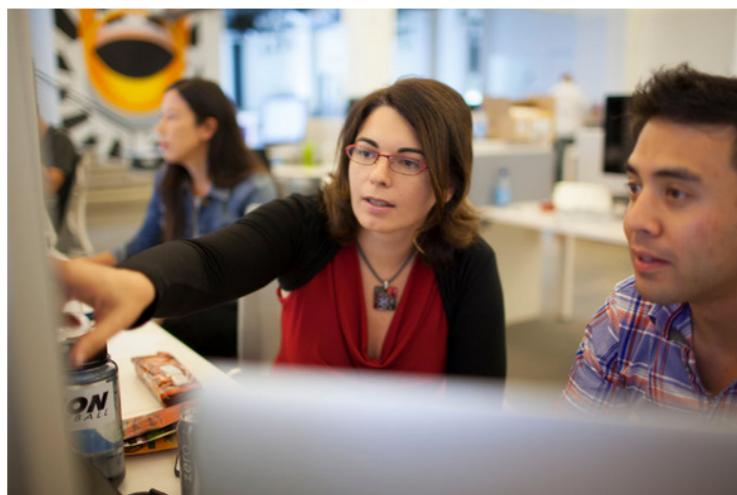
Open Terminal (in your dock)

Working with biological data requires the ability to clean data.



For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights

By STEVE LOHR AUG. 17, 2014



Monica Rogati, Jawbone's vice president for data science, with Brian Wilt, a senior data scientist.
Peter DaSilva for The New York Times



Big data's dirty problem

by Verne Kopytoff

@vkopytoff

JUNE 30, 2014, 10:58 AM EDT



Science is in a reproducibility crisis: How do we resolve it?

Sep 20, 2013 by Fiona Fidler & Ascelin Gordon, The Conversation



About Projects Press Contact

Validating key experimental results
via independent replication

Learn more »

Reproducibility initiatives will invite increased scrutiny into data cleaning methods.

What is reproducible research?

The concept that scientific claims are published with data and code so that others can verify and build on those findings.

Please experience more at:

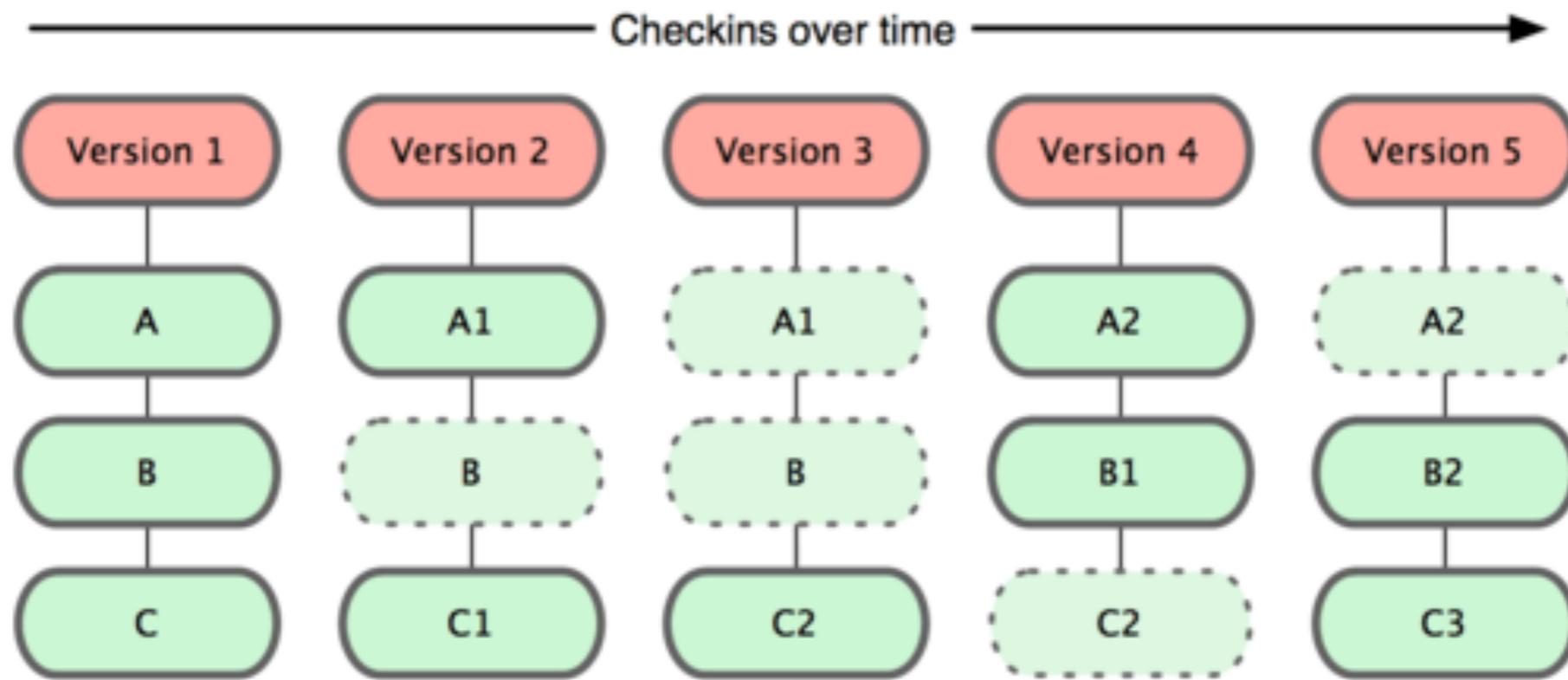
www.coursera.org Reproducible research course
GitHub.com Help and training pages



git

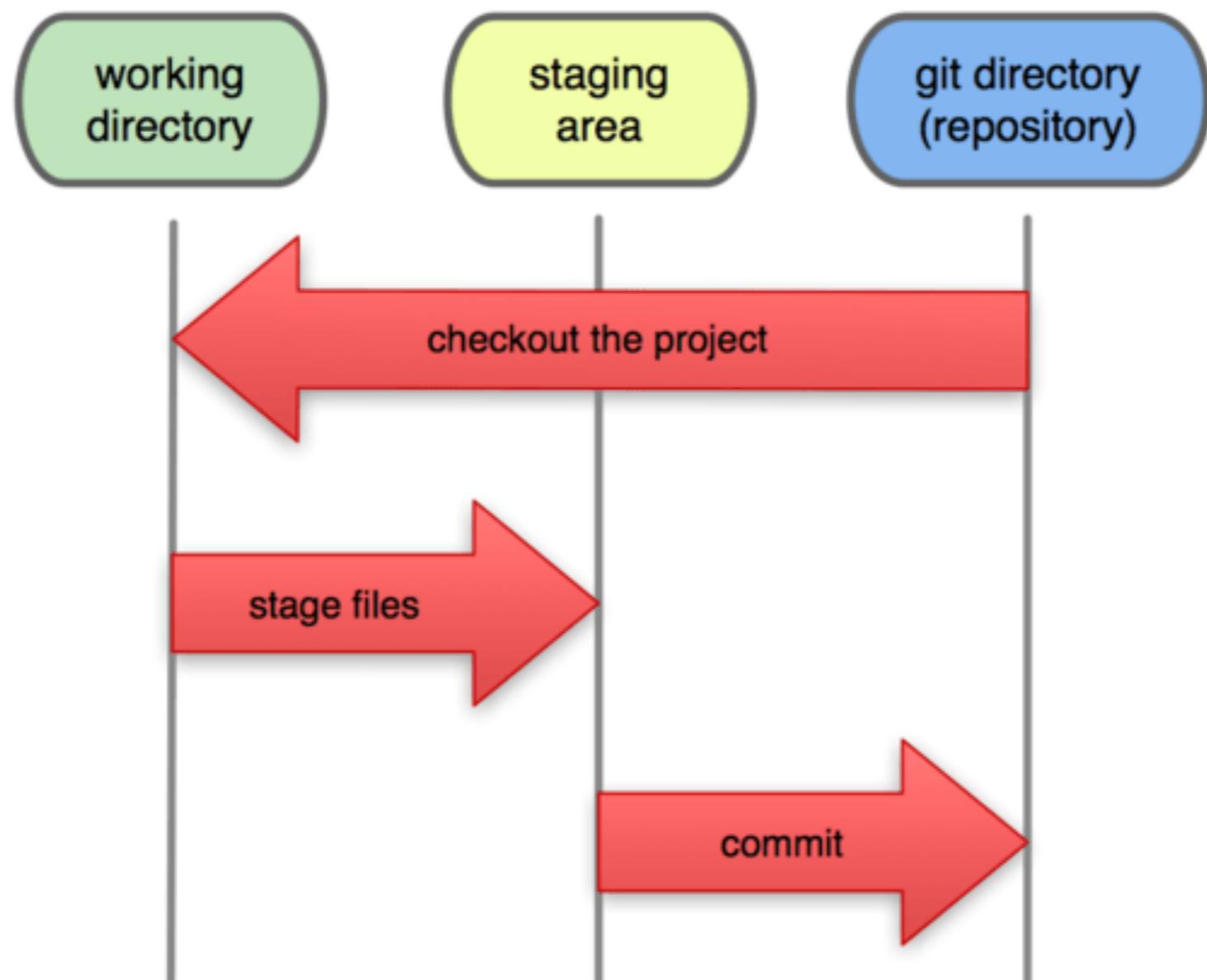
is a version control system

git tracks changes and files over time



git tracks local changes

Local Operations

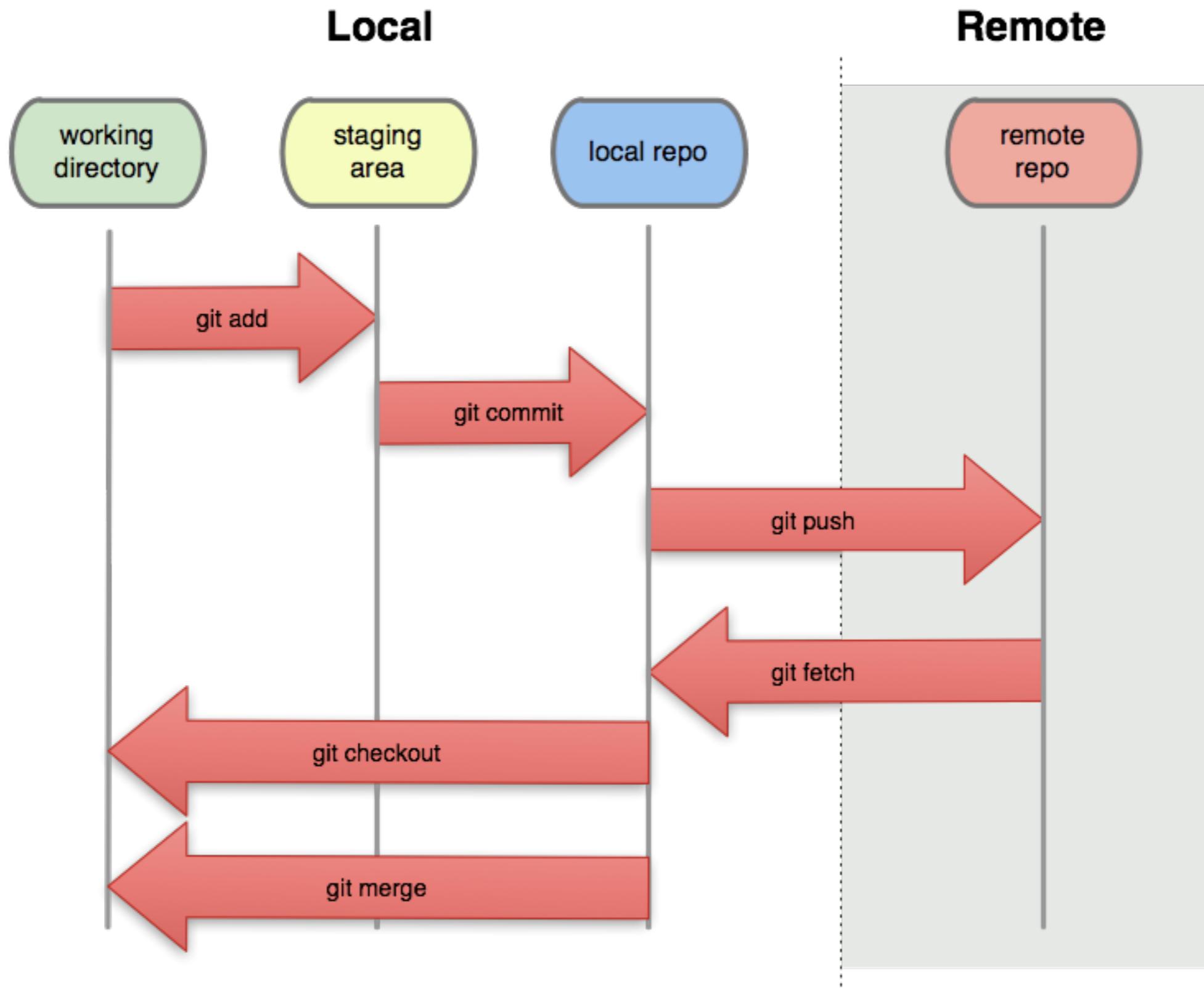




Github makes git super powerful

Check out: www.github.com/andersenlab

github takes git from local to the world



Let's clone the IBiS Bootcamp repository

 AndersenLab / **IBiS-Bootcamp** Unwatch ▾ 6 Star 0 Fork 0

<http://www.andersenlab.org> — Edit

11 commits 1 branch 0 releases 1 contributor

branch: master / +

Update README.md

 **synmuv** authored 5 minutes ago latest commit `7d0459551e`

| File | Description | Time |
|-----------|---|---------------|
| data | Deleted the wrong file. | 26 days ago |
| scripts | Scripts and data for variant calling excercise. | 26 days ago |
| LICENSE | Initial commit | a month ago |
| README.md | Update README.md | 5 minutes ago |

 README.md

IBiS-Bootcamp

This repository will store files and scripts for the 2014 IBiS Computational Biology Bootcamp.

Code

- Issues 0
- Pull Requests 0
- Wiki
- Pulse
- Graphs
- Settings

HTTPS clone URL 

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

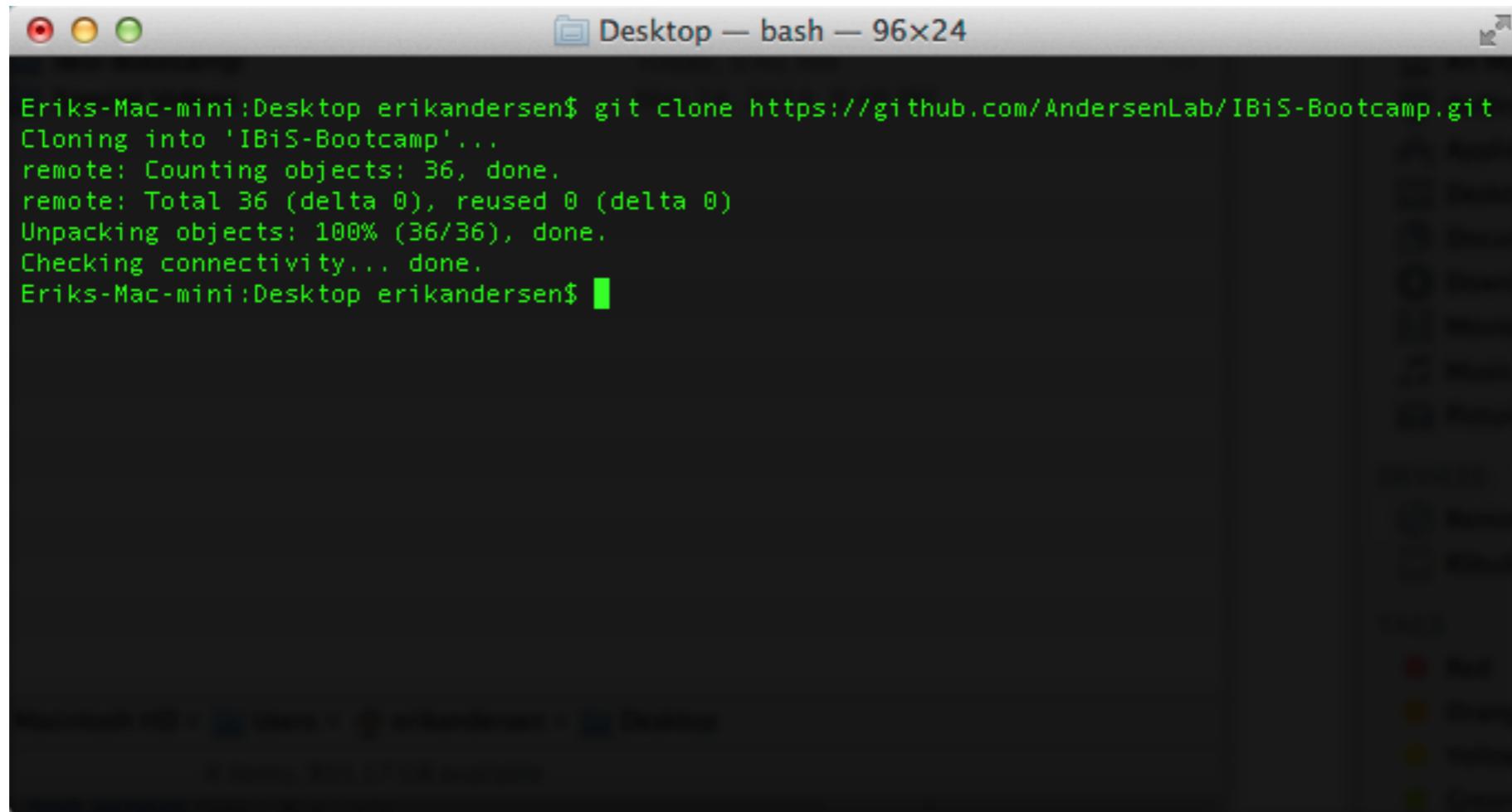
 [Clone in Desktop](#)

 [Download ZIP](#)



Let's clone the IBiS Bootcamp repository

git clone invokes git to clone a repository



```
Eriks-Mac-mini:Desktop erikandersen$ git clone https://github.com/AndersenLab/IBiS-Bootcamp.git
Cloning into 'IBiS-Bootcamp'...
remote: Counting objects: 36, done.
remote: Total 36 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (36/36), done.
Checking connectivity... done.
Eriks-Mac-mini:Desktop erikandersen$
```

1. Go to home directory. Type `cd ~`
2. Type `git clone` and then paste directory after `clone`.

Git/Github Commands

Move files to staging area `git add [file/s]`

Check status of local repo `git status`

Commit files to local repo `git commit -m "short message"`

Push files to remote repo `git push origin master`

Further experiences:
<http://git.io/vGoaJ>

It's time to get organized...

```
MyProject/  
  data/  
    raw/  
    processed/  
  scripts/  
  results/  
  readme.md
```

```
MyProject/data/raw/
```

Basic directory commands for UNIX

man = read the manual for a command

ls = list the contents of a directory

Try typing **man ls**

Try typing **man ls -a**

Further experiences:

<http://freeengineer.org/learnUNIXin10minutes.html>
<http://www.ee.surrey.ac.uk/Teaching/Unix/>
or just google: “unix command line tools”

Basic directory commands for UNIX

man = read the manual for a command

ls = list the contents of a directory

pwd = display the present working directory

cd = change directory

mkdir = make a directory

rmdir = remove a directory

Further experiences:

<http://freeengineer.org/learnUNIXin10minutes.html>

<http://www.ee.surrey.ac.uk/Teaching/Unix/>

or just google: “unix command line tools”

Keyboard shortcuts for UNIX

Press *tab* to complete a filename

Navigation shortcuts

ctrl + a : go to beginning of line

ctrl + e : go to end of line

alt + <- (->) : skip between delimiters

alt + delete : delete previous word

command + up (down) arrow: beginning (or end) of text file

Selection shortcuts

option + shift + <- (->): highlight text to next delimiter

shift + up (down) arrow : highlight previous (next) line

command + shift <- (->) : highlight to beginning (or end of line)

command + shift + up (down) arrow : highlight to beginning (or end of text)

Further experiences:
<http://goo.gl/ukS77X>

Let's each make the directory structure using the command line.

```
MyProject/  
    data/  
        raw/  
        processed/  
        scripts/  
        results/  
        mistakes_happen/
```

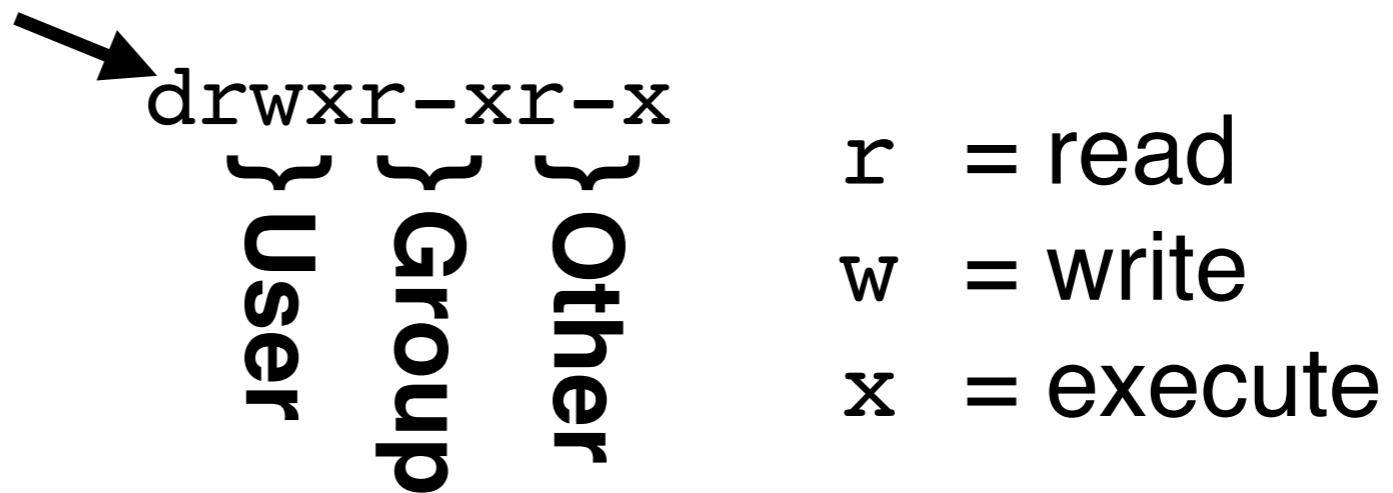
| | |
|-------|---|
| man | = read the manual for a command |
| ls | = list the contents of a directory |
| pwd | = display the present working directory |
| cd | = change directory |
| mkdir | = make a directory |
| rmdir | = remove a directory |

cd .. allows you to go back one directory in tree

Permissions (modes) matter.

Go to your data directory, type `ls -la`

Filetype (- , d, or c)



| Symbolic Notation | Octal Notation | English |
|-------------------|----------------|------------------------|
| ----- | 0000 | no permissions |
| ---x--x--x | 0111 | execute |
| --w--w--w- | 0222 | write |
| --wx-wx-wx | 0333 | write & execute |
| -r--r--r-- | 0444 | read |
| -r-xr-xr-x | 0555 | read & execute |
| -rw-rw-rw- | 0666 | read & write |
| -rwxrwxrwx | 0777 | read, write, & execute |

`chmod` changes
the mode

What about that `readme.md` file?

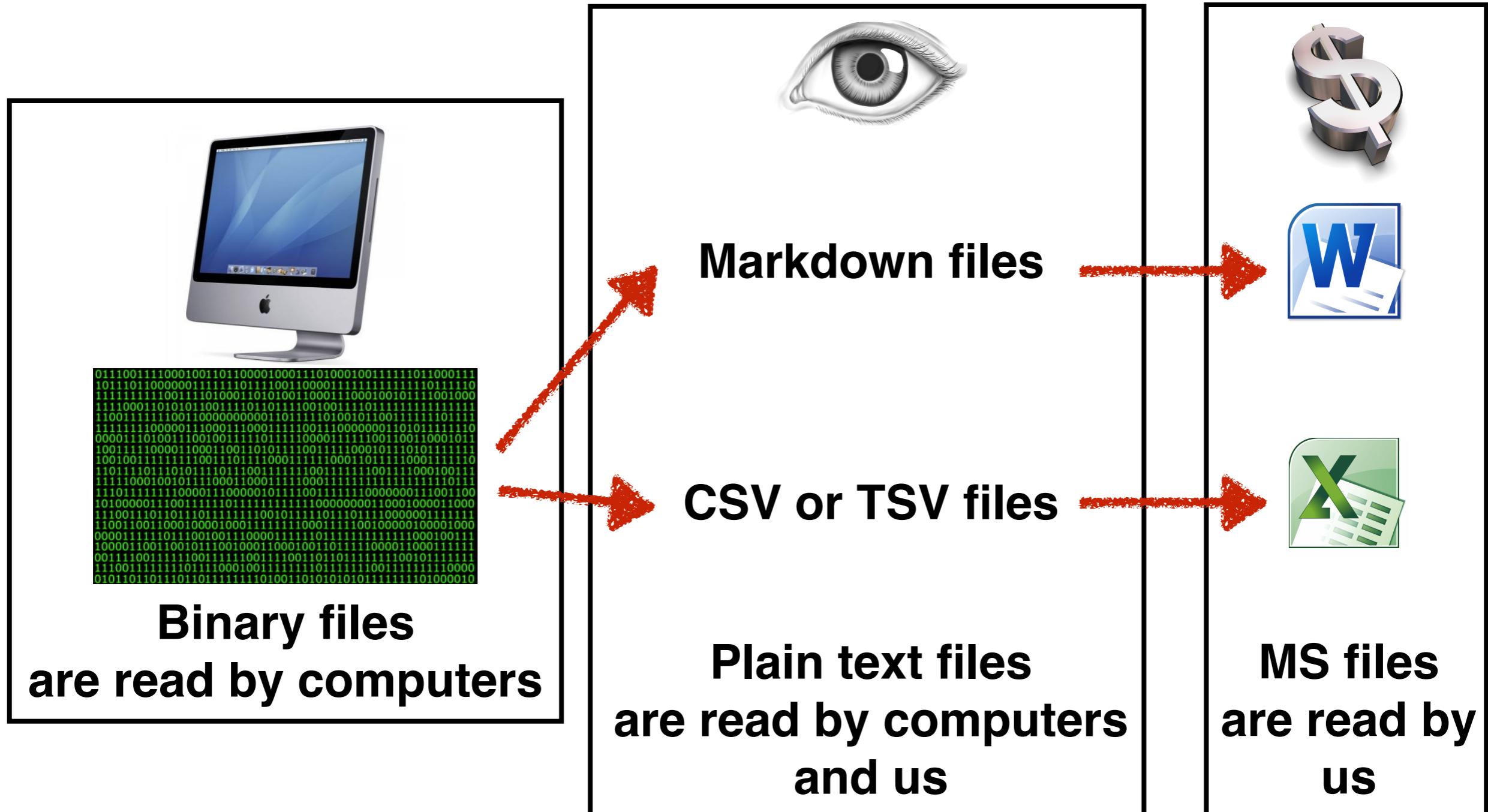
`md` = markdown

Markdown is an emerging standard for software documentation.

Easily written, read, and translated.

Portable to HTML, PDF, etc.

We need files that the computer and we can read.



What about that readme.md file?

md = markdown

what you type

```
Heading  
=====  
  
Sub-heading  
-----  
  
h3. Traditional html title  
  
Paragraphs are separated  
by a blank line.  
  
Let 2 spaces at the end of a line to do a  
line break  
  
Text attributes *italic*,  
**bold**, `monospace`.  
  
A [link](http://example.com).  
<<< No space between ] and ( >>>  
  
Shopping list:  
  
* apples  
* oranges  
* pears  
  
Numbered list:  
  
1. apples  
2. oranges  
3. pears  
  
The rain---not the reign---in  
Spain.
```

what is rendered

Heading

Sub-heading

Traditional html title

Paragraphs are separated by a blank line.

Let 2 spaces at the end of a line to do a
line break

Text attributes *italic*, **bold**, `monospace`.

A [link](#).

Shopping list:

- apples
- oranges
- pears

Numbered list:

1. apples
2. oranges
3. pears

The rain—not the reign—in Spain.

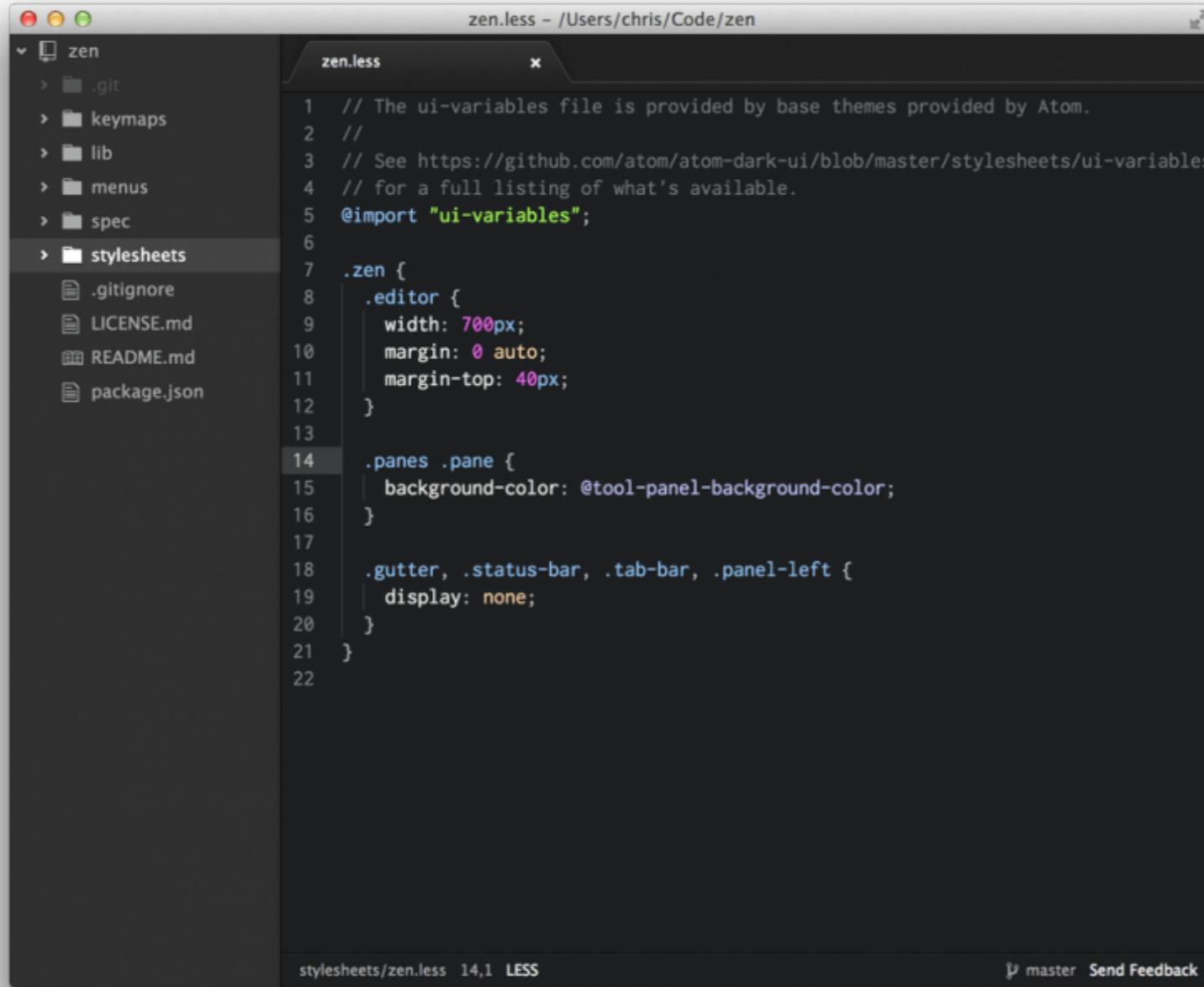
Let's make `readme.md` from the command line

1. Go to MyProject directory
2. Type `nano readme.md`
3. Type:

This example `*project*` will teach us `**a lot**` about basic computational biology.

4. Press control O to save. Press enter.
5. Press control X to exit.

Further experiences:
http://git.io/cnT_vA



A screenshot of the Atom text editor interface. The window title is "zen.less - /Users/chris/Code/zen". The left sidebar shows a file tree with "zen" as the root directory, containing ".git", "keymaps", "lib", "menus", "spec", "stylesheets" (which is selected), ".gitignore", "LICENSE.md", "README.md", and "package.json". The main editor area displays the following LESS code:

```
// The ui-variables file is provided by base themes provided by Atom.  
// See https://github.com/atom/atom-dark-ui/blob/master/stylesheets/ui-variables  
// for a full listing of what's available.  
@import "ui-variables";  
  
.zen {  
  .editor {  
    width: 700px;  
    margin: 0 auto;  
    margin-top: 40px;  
  }  
  
.panes .pane {  
  background-color: @tool-panel-background-color;  
}  
  
.gutter, .status-bar, .tab-bar, .panel-left {  
  display: none;  
}  
}
```

The status bar at the bottom shows "stylesheets/zen.less 14,1 LESS" and "master".

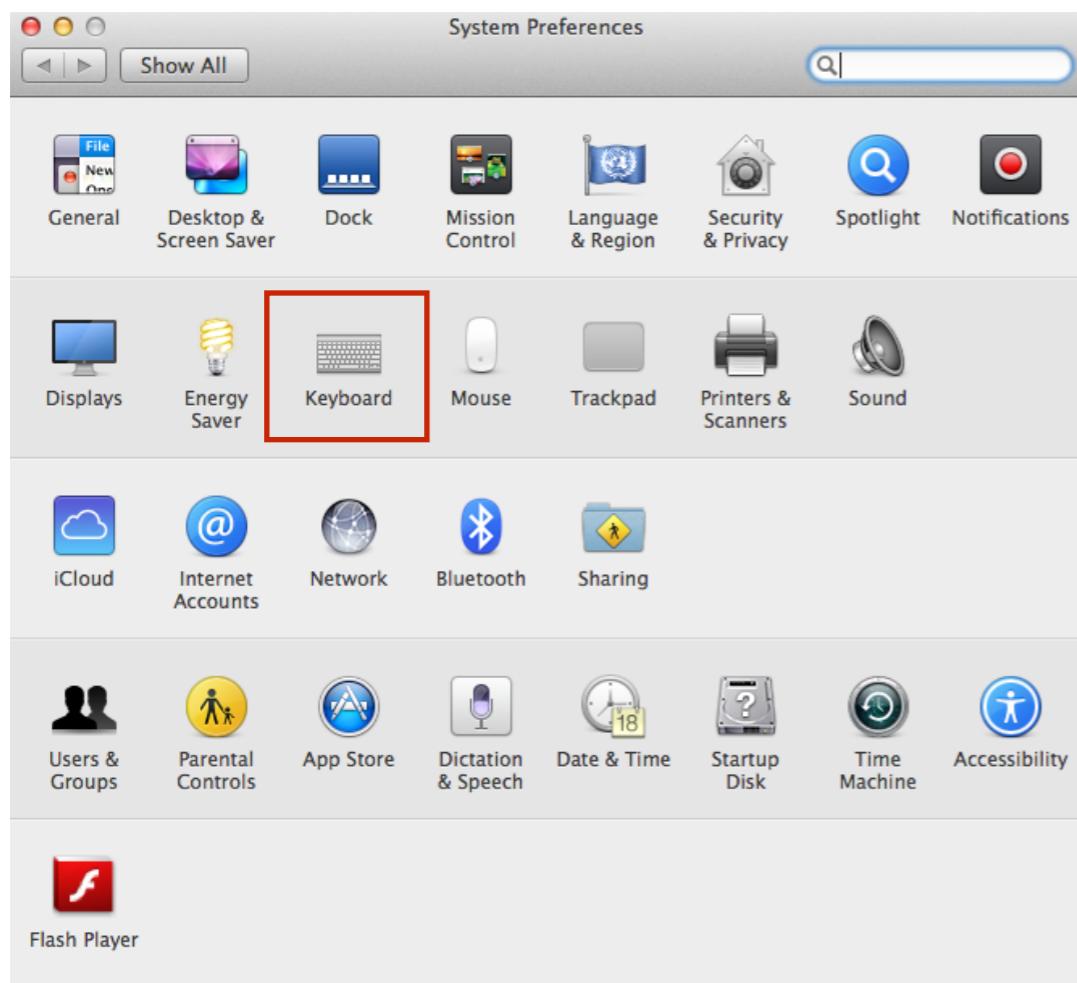
A plain text editor.
Very different from
Microsoft Word.



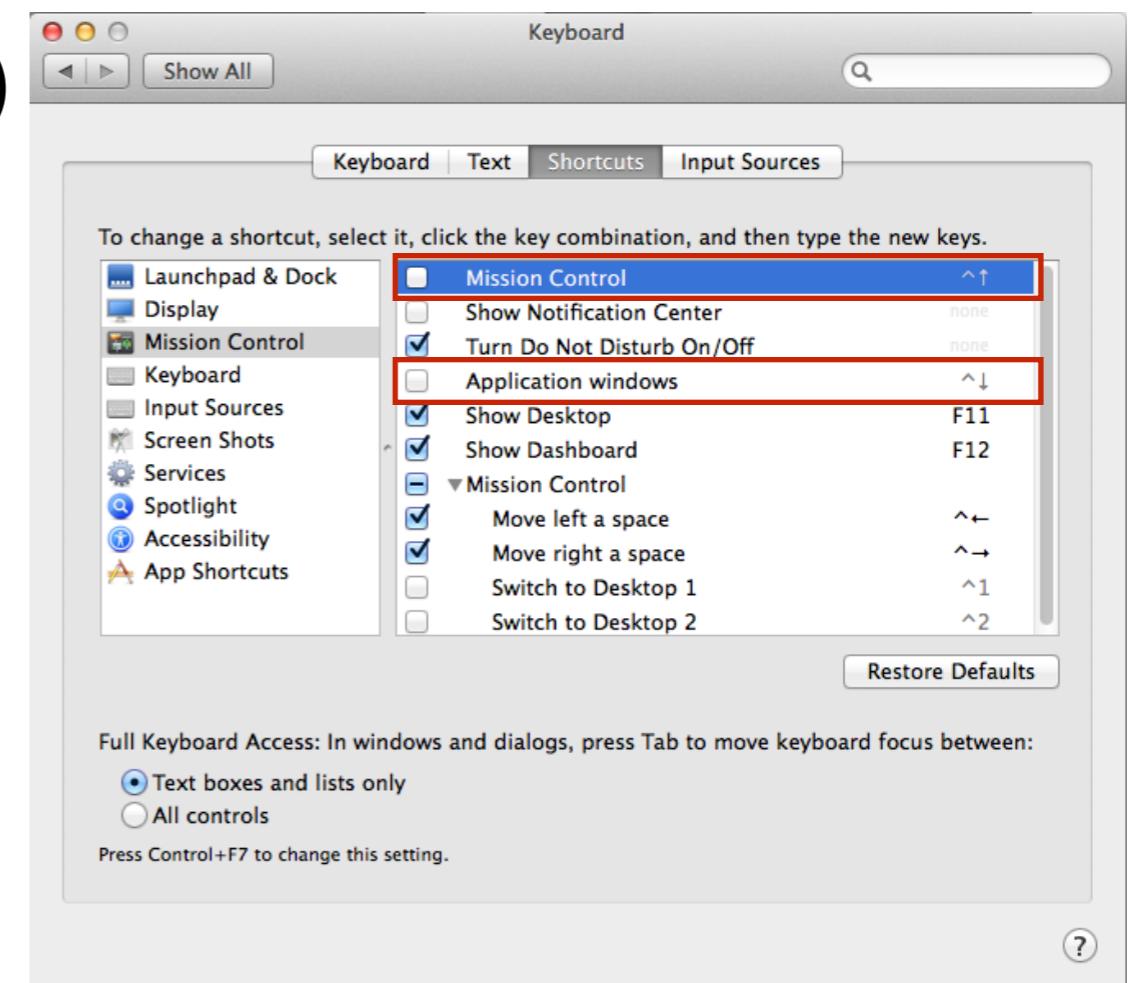
1)



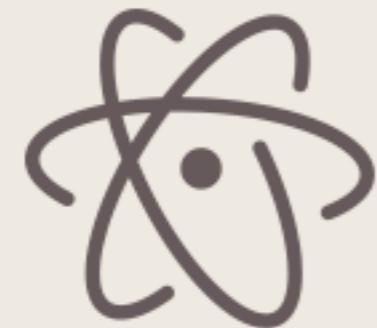
2)



3)



unchecked these two shortcuts



Selection

| | |
|-----------------------------------|-----|
| Add Selection Above | ⌃⇧↑ |
| Add Selection Below | ⌃⇧↓ |
| Single Selection | ◎ |
| Split into Lines | ⇧⌘L |
| Select to Top | ⇧⌘↑ |
| Select to Bottom | ⇧⌘↓ |
| Select Line | ⌘L |
| Select Word | ⌃⇧W |
| Select to Beginning of Word | ⌥⇧B |
| Select to Beginning of Line | ⌥⇧B |
| Select to First Character of Line | ⇧⌘← |
| Select to End of Word | ⌥⇧F |
| Select to End of Line | ⇧⌘→ |

Open IBiS_bootcamp/resources/atom_selections_example.txt

Try the exercises, then try some of the options in the selection menu.

These functions and keyboard shortcuts are designed to *save you time*.



ATOM is customizable!

The image displays two screenshots of the Atom application. On the left, the Atom application menu is shown, featuring a blue header with the word "Atom" and a white main menu area. The menu includes options like "About Atom", "View License", "Version 0.125.0", and "Check for Update". Below these, a "Preferences..." option is highlighted in blue, accompanied by a keyboard shortcut icon. Further down, there are sections for "Open Your Config", "Open Your Init Script", "Open Your Keymap", "Open Your Snippets", and "Open Your Stylesheet". At the bottom, there are "Install Shell Commands" and "Hide Atom" (with a keyboard shortcut of ⌘H), "Hide Others" (with a keyboard shortcut of ⌄⌘H), and "Show All". The "Quit" option is at the very bottom with a keyboard shortcut of ⌘Q. A black arrow points from the "Preferences..." menu item towards the right screenshot. On the right, the Atom settings interface is shown. It has a blue header with the word "ATOM" and a white main area. The "Settings" tab is selected, showing sections for "Core Settings" and "Editor Settings". Under "Core Settings", there are checkboxes for "Audio Beep", "Destroy Empty Panes", and "Exclude Vcs Ignored Paths". There is also a section for "Ignored Names" with a text input field containing ".git, .svn, .DS_Store". Under "Project Home", the "Default" path is set to "/Users/daniel/github". Under "Editor Settings", there are checkboxes for "Auto Indent" and "Confirm Checkout Head Revision". There are also fields for "Font Family" and "Font Size" (set to 14). A red circle highlights the "Themes" option under the "Core Settings" section. At the bottom of the settings interface, there is a button labeled "Open ~/atom".



ATOM is customizable!

 **Choose a Theme**

 You can also style Atom by editing [your stylesheet](#)

UI Theme Atom Light → Overall look of program
This styles the tabs, status bar, tree view, and dropdowns

Syntax Theme Proton Bat → Color of text in the editor
This styles the text inside the editor

Overall look of program

Color of text in the editor



ATOM is extensible!

A screenshot of the Atom settings interface. The sidebar on the left shows options like "Settings", "Keybindings", "Packages" (which is highlighted with a red oval), and "Themes". The main area shows "Core Settings" with checkboxes for "Audio Beep", "Destroy Empty Panes", and "Exclude Vcs Ignored Paths". It also includes sections for "Ignored Names" (containing ".git, .svn, .DS_Store") and "Project Home" (set to "Default: /Users/daniel/github"). Below that is "Editor Settings" with checkboxes for "Auto Indent" and "Confirm Checkout Head Revision", and fields for "Font Family" and "Font Size" (set to 14). At the bottom left is a button labeled "Open ~/atom".

ATOM

Settings

Keybindings

Packages

Themes

proton

Proton Bat
jdsimcoe

Open ~/atom

Core Settings

- Audio Beep
- Destroy Empty Panes
- Exclude Vcs Ignored Paths

Ignored Names

.git, .svn, .DS_Store

Project Home

Default: /Users/daniel/github

Editor Settings

- Auto Indent
- Confirm Checkout Head Revision

Font Family

Font Size

14

Install ‘sort lines’



ATOM is extensible!

Edit

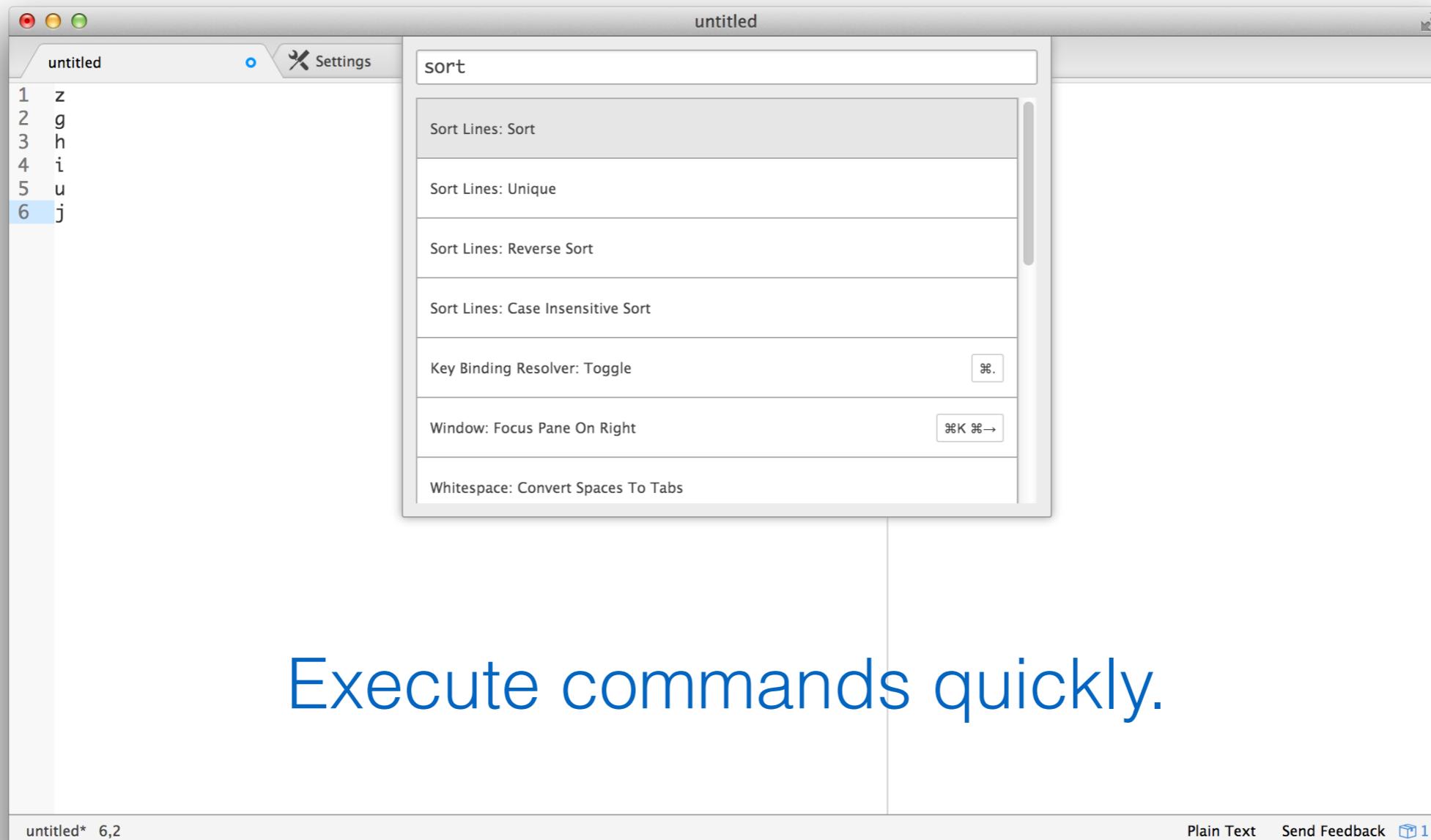
- Undo ⌘Z
- Redo ⌘Y
-
- Cut ⌘X
- Copy ⌘C
- Copy Path ⌘⇧C
- Paste ⌘V
- Select All ⌘A
-
- Toggle Comments ⌘/
- Lines ▶**
- Text ▶
- Folding ▶
- Reflow Selection ⌘⌥Q
- Bookmark ▶
- Go to Line ⌘G
- Select Grammar ⌘⇧L
-
- Start Dictation...
- Special Characters... ⌘Space

- Indent ⌘]
- Outdent ⌘[
- Auto Indent
-
- Move Line Up ⌘↑
- Move Line Down ⌘↓
- Duplicate Lines ⌘D
- Delete Line ⌘⌫
- Join Lines ⌘J**
- Sort
- Reverse Sort
- Unique
- Sort (Case Insensitive)

New functions!



ATOM is has “command palette”! Command-shift-p



Execute commands quickly.

Let's make readme.md from ATOM

1. Open Atom
2. Check out the welcome.md
3. Go to the untitled pane.
4. Type:

This example *project* will teach us **a lot** about basic computational biology.

5. Press command-shift-p
6. Type “markdown” and press enter
7. Check it out!
8. Close the markdown preview
9. Press control-shift-m
10. Check it out!
11. Save as “readme.md” into MyProject folder

Further experiences:

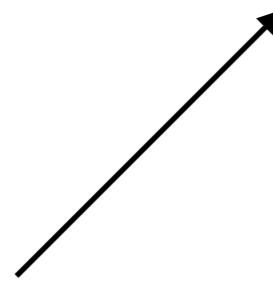
<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>
<http://daringfireball.net/projects/markdown/basics>

Make an ibis.md file

1. Title: IBiS is awesome
2. “awesome” in bold
3. Make it a first level header with a line underneath it
4. Add normal text below: “Our webpage is here.”
5. Add the link for the IBiS webpage to “here”.
6. Make a list of the top three reasons you love IBiS
7. Add the IBiS image to the bottom

The command line can be personalized!

.bash_profile



A hidden file in your home directory

- Your *bash profile* is run every time you open your terminal.
- You can use your bash profile to store common settings, functions, modify commands, or personalize your terminal.

Shortcuts can be added with aliases

```
alias refresh="source ~/.bash_profile"  
alias dbx="cd ~/Dropbox/"
```

Typing ‘refresh’ will refresh your bash profile.

Typing ‘dbx’ will move you to the dropbox directory.

Functions can be used to execute common tasks.

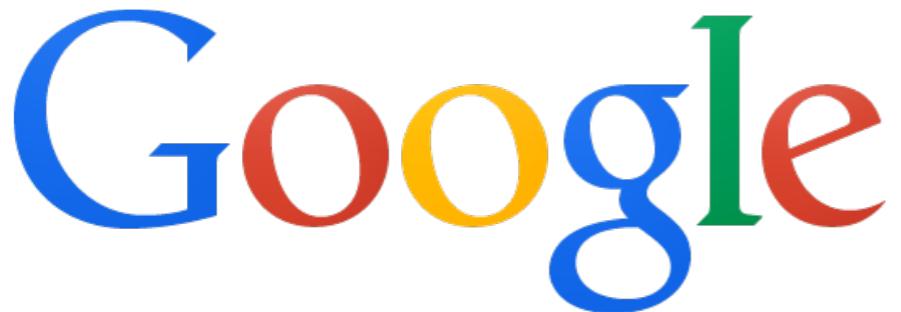
```
new_project() {  
    mkdir $1  
    touch $1/Readme.md  
    mkdir $1/Data  
    mkdir $1/Data/Raw  
    mkdir $1/Data/Processed  
    mkdir $1/Scripts  
    mkdir $1/Results  
}
```

This function will create a project directory structure.

Example:

```
new_project chip-seq-analysis
```

What to do when you are lost?



Let's have some fun!

1. *Cool computer voice*: Type say and whatever you want. Make sure your sound is on
2. *Get your fortune*: Type brew install fortune. Then type: fortune
3. *Download youtube videos*: Type brew install youtube-dl.
4. Then type: youtube-dl cHK428vSIz8
5. *Watch an asci movie*: Type: telnet towel.blinkenlights.nl 23

A few additional commands for UNIX

time = time how long an operation takes

screen = allows multiplexing of terminal windows

nohup = no hang ups, process errors go to a log

sed = search and edit text

awk = search, extract, or transform text

Also, check out datamash utilities

Further experiences:

<http://freeengineer.org/learnUNIXin10minutes.html>

<http://www.ee.surrey.ac.uk/Teaching/Unix/>

or just google: “unix command line tools”

Remote commands for UNIX

ssh = secure shell, remote access to UNIX systems

rsync = remote synchronization, remote copy

Let's play with data

1. Open terminal
 2. Go to data folder in IBiS_Bootcamp
 3. Look at directory contents
 4. .gz is a g-zipped file. Type `gunzip` file to unzip it
 - 5. Type “atom” to open a text editor for this project - File naming use date YYYYMMDD_file for clarity**
 - 6. Comment your code!!!**

head, tail, clear

Explore the data

1. How big is the file?
2. What are the permissions?
3. Can we open it with atom?

less

head, tail, clear
head, tail, clear, less

Illumina sequencing generates FASTQ files



**Each paired-end lane has
four FASTQ files**

FASTQ files contain the sequence and quality data, along with sequencer, index, and lane information.

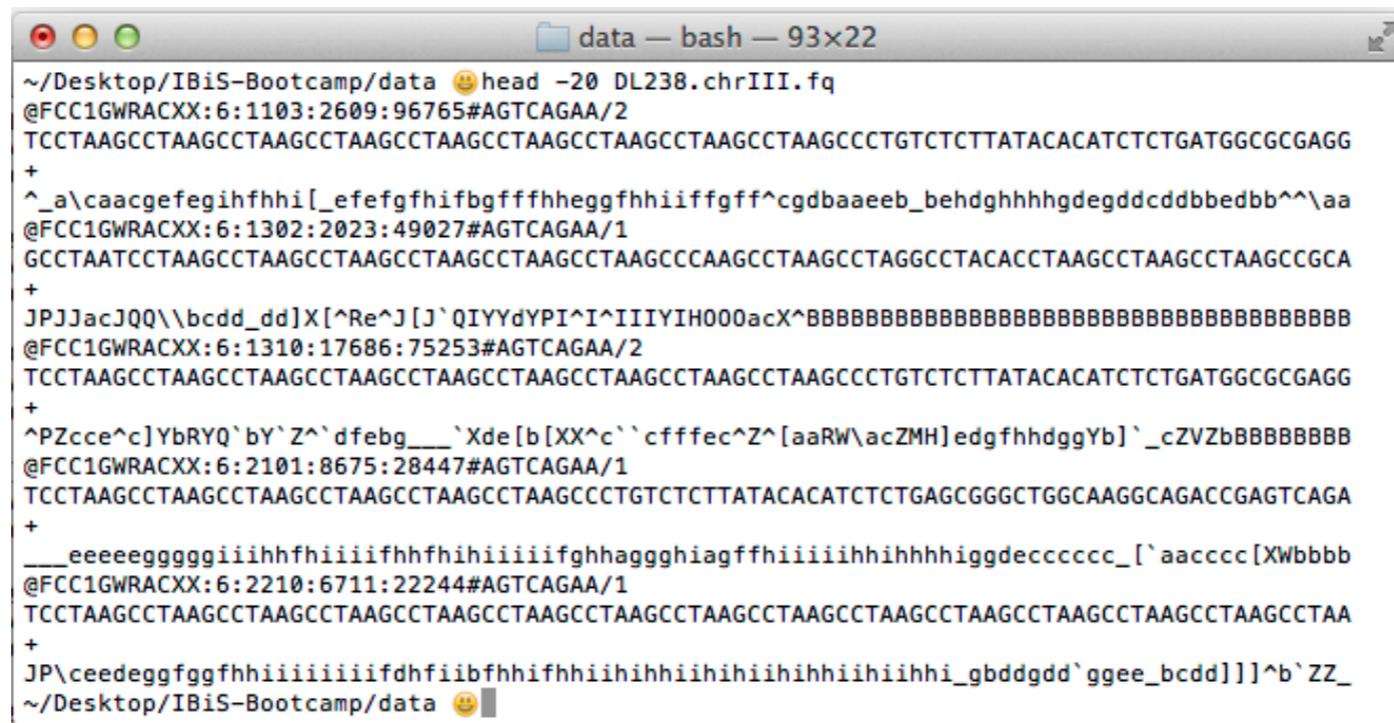
OK. It's an Illumina sequence file. Now, what?

cp DL238.chrIII.fq seq.fq

copy DL238.chrIII.fq to seq.fq

head, tail, clear, less

OK. It's an Illumina sequence file. Now, what?



The terminal window shows the command `head -20 DL238.chrIII.fq` being run. The output displays 20 lines of sequence data from the file. Each line starts with a header containing sequencing information like "@FCC1GWRACXX:6:1103:2609:96765#AGTCAGAA/2" followed by a sequence of nucleotides (A, T, C, G) and a quality score (Q). The sequence data is highly compressed and contains many 'N' characters.

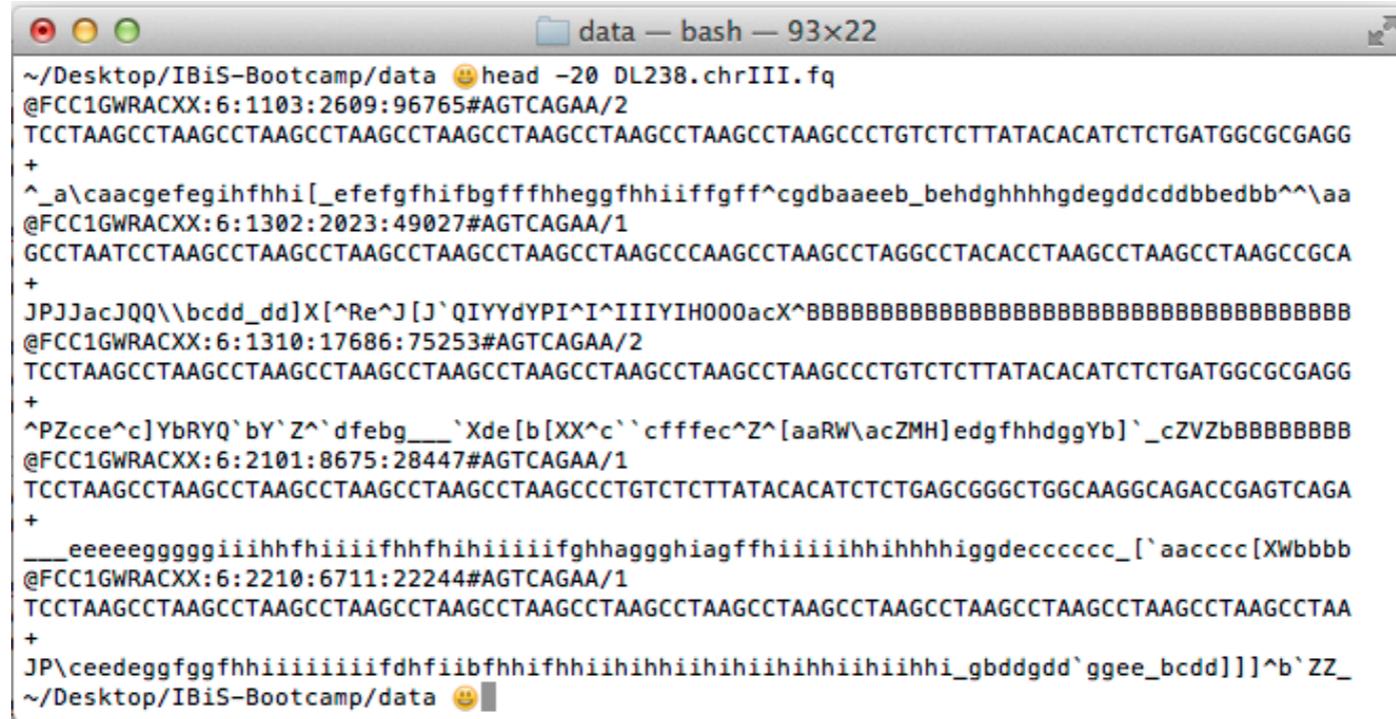
cat DL238.chrIII.fq > seq2.fq

catenate DL238.chrIII.fq to seq2.fq

the **>** sign tells catenate to put the contents into the file that follows it

head, tail, clear, less

OK. It's an Illumina sequence file. Now, what?



```
~/Desktop/IBiS-Bootcamp/data 😊 head -20 DL238.chrIII.fq
@FCC1GWRACXX:6:1103:2609:96765#AGTCAGAA/2
TCCTAAGCCTAACGCTAACGCCTAACGCCTAACGCCTAACGCCTAACGCCGTGCTCTTATACACATCTGATGGCGCGAGG
+
^_a\caacgefegihfhhi[_efefgfhifbgffffheggfhhifffgff^cgdbaaeeb_behdghhhgdegddcdedbba^\\aa
@FCC1GWRACXX:6:1302:2023:49027#AGTCAGAA/1
GCCTAACCTAACGCCTAACGCCTAACGCCTAACGCCAACGCTAACGCCTAGGCCACACCTAACGCTAACGCCGCA
+
JPJJacJQQ\\bcdd_dd]X[^Re^J[J`QIYYdYPI^I^IIIIH000acX^BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
@FCC1GWRACXX:6:1310:17686:75253#AGTCAGAA/2
TCCTAACGCCTAACGCCTAACGCCTAACGCCTAACGCCGTGCTCTTATACACATCTGATGGCGCGAGG
+
^PZcce^c]YbRYQ`bY`Z^`dfebg__`Xde[b[XX^c``cfffec^Z^[aaRW\acZMH]edgfhhdggyb]`_cZVzbBBBBBBB
@FCC1GWRACXX:6:2101:8675:28447#AGTCAGAA/1
TCCTAACGCCTAACGCCTAACGCCTAACGCCGTGCTCTTATACACATCTGAGCGGGCTGGCAAGGCAGACCGAGTCAGA
+
____eeeeeggggiiihhfiiifhhfhihiiifghhaggghiagffhiiiihhhhiggdecccccc_`aacccc[XWbbbb
@FCC1GWRACXX:6:2210:6711:22244#AGTCAGAA/1
TCCTAACGCCTAACGCCTAACGCCTAACGCCTAACGCCTAACGCCTAACGCCTAACGCCTAACGCCTAACGCCTAACGCCTAA
+
JP\ceedeggfggfhiiiiiiifdhfiibfhhifhiihiihiihiihiihiihiihiihiihiihiihiihiihiihiihiih
~/Desktop/IBiS-Bootcamp/data 😊
```

rm seq2.fq

remove file seq2.fq

head, tail, clear, less

OK. It's an Illumina sequence file. Now, what?

gzip DL238.chrIII.fq

zip DL238.chrIII.fq to make it smaller

head, tail, clear, less

Let's look for specific data within the FASTQ file

Let's get all the lines in the file that start with '@'.

```
grep '^@' seq.fq > at_lines.txt
```

globally search a regular expression and print

Press ctrl-c to stop a process

What HiSeq machines were used for sequencing?

```
head, tail, clear, less, cp, mv, cat
```

`cut` allows you to pull out columnar data from a text file.

delimiter - A character used to separate pieces of data.

field - A column formed by delimiters.

Example

```
@FCC1GWRACXX:6:1103:2609:96765#AGTCAGAA/2  
@FCC1GWRACXX:6:1302:2023:49027#AGTCAGAA/1
```

If we make the delimiter a colon (:), then the corresponding fields will look like below.

| <i>Field</i> | 1 | 2 | 3 | 4 | 5 | | | | |
|--------------|--------------|---|---|---|------|---|------|---|------------------|
| | @FCC1GWRACXX | : | 6 | : | 1103 | : | 2609 | : | 96765#AGTCAGAA/2 |
| | @FCC1GWRACXX | : | 6 | : | 1302 | : | 2023 | : | 49027#AGTCAGAA/1 |

cut

Field 1 2 3 4 5
@FCC1GWRACXX : 6 : 1103 : 2609 : 96765#AGTCAGAA/2
@FCC1GWRACXX : 6 : 1302 : 2023 : 49027#AGTCAGAA/1

| Command | Translation | Output |
|---|--|--|
| <pre>cut -f 1 -d ":" at_lines.txt</pre> | "cut the first field formed using the colon delimiter." | @FCC1GWRACXX @FCC1GWRACXX |
| Save output to machine.txt | | |
| <pre>cut -f 1,3,5 -d ":" at_lines.txt</pre> | "cut the 1st , 3rd , and 5th fields formed using the colon delimiter." | @FCC1GWRACXX:1103:96765#AGTCAGAA/2 @FCC1GWRACXX:1302:49027#AGTCAGAA/1 |

Let's look for specific data within the FASTQ file

We'd like to know how many sequencers were used.

```
sort machine.txt > sorted_machine.txt
```

First, we have to sort the data and save to sorted_machine.txt

```
uniq sorted_machine.txt
```

How many lines of data do we have?

```
wc -l sorted_machine.txt
```

```
head, tail, clear, less, cp, mv, cat, grep, cut
```

**It's time to put it all together.
Enter the pipe operator.**

```
grep '^@' seq.fq | cut -f 1 -d ':' | sort | uniq
```

Challenge:

Find all unique barcodes in the seq data
and save to a file called barcode.txt

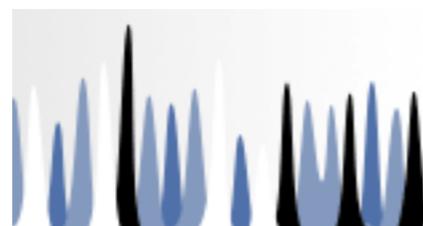
```
grep '^@' seq.fq | cut -f 2 -d '#' | cut -f 1 -d '/' | sort | uniq > bc.txt
```

head, tail, clear, less, cp, mv, cat, grep, cut, sort, uniq, wc

What to do when you are lost?



www.biostars.org



SEQanswers
the next generation sequencing community

www.seqanswers.com

Day #1 Homework

1. Sign up for github
2. Star our class repository
3. Download Github desktop and login with your Github login
4. Go and explore github! Find some cool code.

The screenshot shows the GitHub Explore page with a light gray header containing the GitHub logo, a search bar, and navigation links for Explore, Gist, Blog, and Help. On the right side of the header is a user profile for 'synmuv' with a dropdown menu and other account options. Below the header, the main navigation includes 'Explore GitHub' on the left and 'All' (selected), Showcases, Trending, and Stars on the right. The main content area is titled 'Explore' with the subtitle 'Browse interesting projects, solving all types of interesting problems.' It features a grid of eight cards:

- Writing**: A green card with a pattern of circles. It has a link to 'Open Journalism'.
- Open Journalism**: A purple card with a pattern of circles.
- Projects with great wikis**: A teal card with a wavy pattern. It says 'These projects all use GitHub Wikis to share documentation and helpful r...' and shows 8 forks and 5 stars.
- Web games**: A red card with a pattern of triangles. It says 'Who says that coding has to be boring? Have some fun with these open sou...' and shows 6 forks and 2 stars.
- Text editors**: A large green card with a pattern of chevrons. It has a link to 'Productivity tools'.
- Productivity tools**: A blue card with a pattern of squares.
- Projects that power GitHub**: A green card with a geometric pattern.
- Game off winners**: A pink card with a checkered pattern.

At the bottom center of the page is a 'See all ›' link.



- Package managers make it easy to manage the software installed on your computer.
- Package ≈ Software ≈ App
- Software can have a lot of dependencies, or *other* required pieces of software. Homebrew takes care of this.
- *1000s* of packages available.



Software installation

Homebrew - a package manager for command line

1. In your browser, go to brew.sh
2. Copy the link at the bottom
3. Go to your Terminal window
4. Paste the link and press enter
5. Follow the prompts to install
6. Run `brew doctor`

Homebrew Commands

Install a package

`brew install package_name`

Uninstall a package

`brew uninstall package_name`

List installed packages

`brew list`

Search for packages

`brew search package_name`

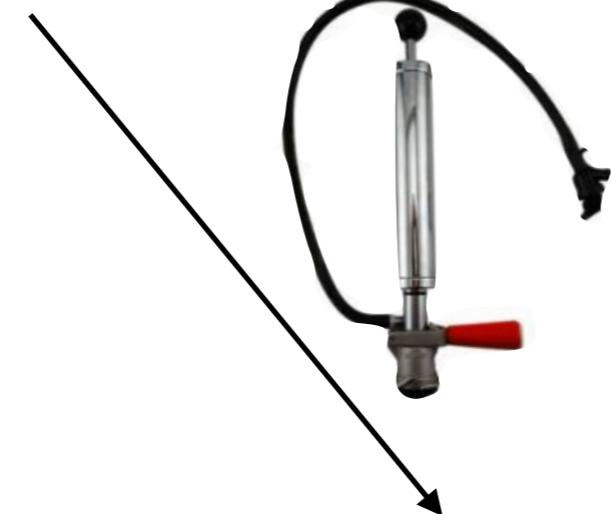
Update Software List

`brew update`

Browse packages at braumeister.org

Homebrew

The missing package manager for OS X



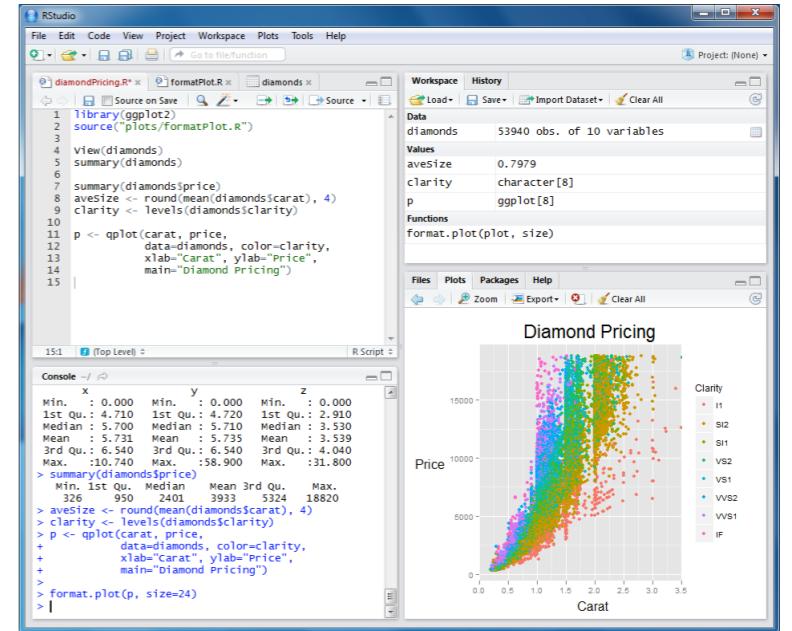
General Purpose software
is already included.

To install science related software, we have to ‘tap’ into the science-software repository.

Science Software

`brew tap homebrew/science`

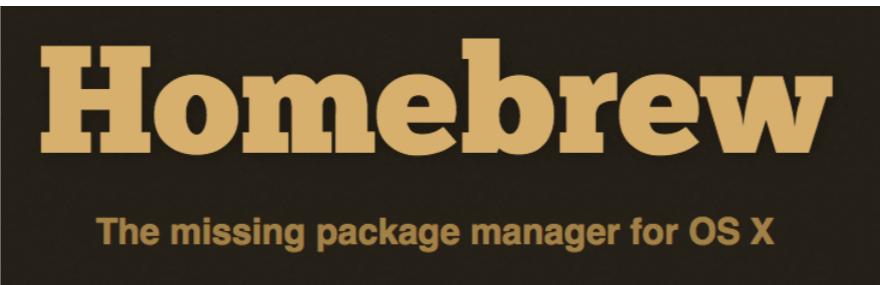
Graphical User Interface (GUI)



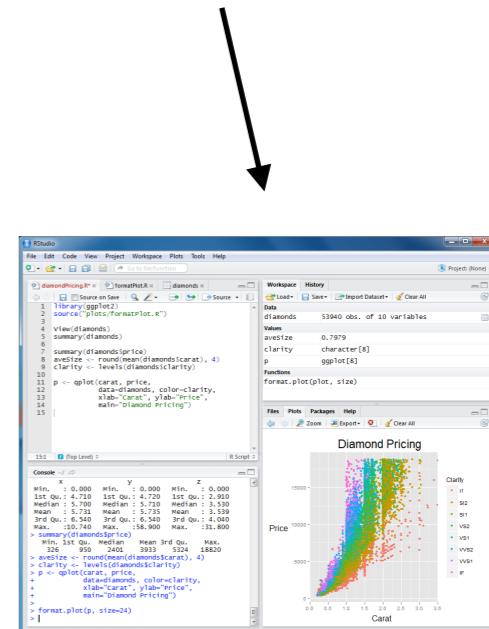
“Point and Click”

But Wait! It is possible to manage GUI software with ‘cask’, an extension of homebrew.

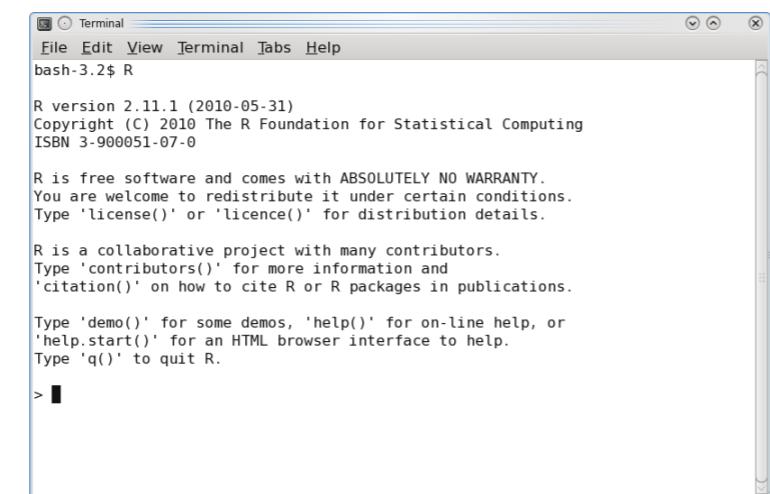
```
brew install caskroom/cask/brew-cask
```



General
Purpose



GUI software



Command line software

Software installation

Go to [www.GitHub.com/AndersenLab/IBiS-Bootcamp/Wiki/Installing-Required-Software](https://www.github.com/AndersenLab/IBiS-Bootcamp/Wiki/Installing-Required-Software)

```
1  #!/bin/bash
2
3  # This script install the required software!
4
5  # Install Homebrew, if you haven't as a part of the Lecture
6  #ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/homebrew/go/install)"
7  #brew tap homebrew/science
8
9  # Cask
10 #brew install caskroom/cask/brew-cask
11
12 # Sequence Analysis
13 brew install --devel samtools
14 brew install bcftools
15 brew install bwa
16 brew install igv
17
18 # wget
19 brew install wget
20
21 # Atom
22 brew cask install atom --appdir=/Applications
23
24 # ql markdown
25 brew cask install qlmarkdown
26
27 # R + Rstudio
28 brew install R
29 brew cask install rstudio --appdir=/Applications
```

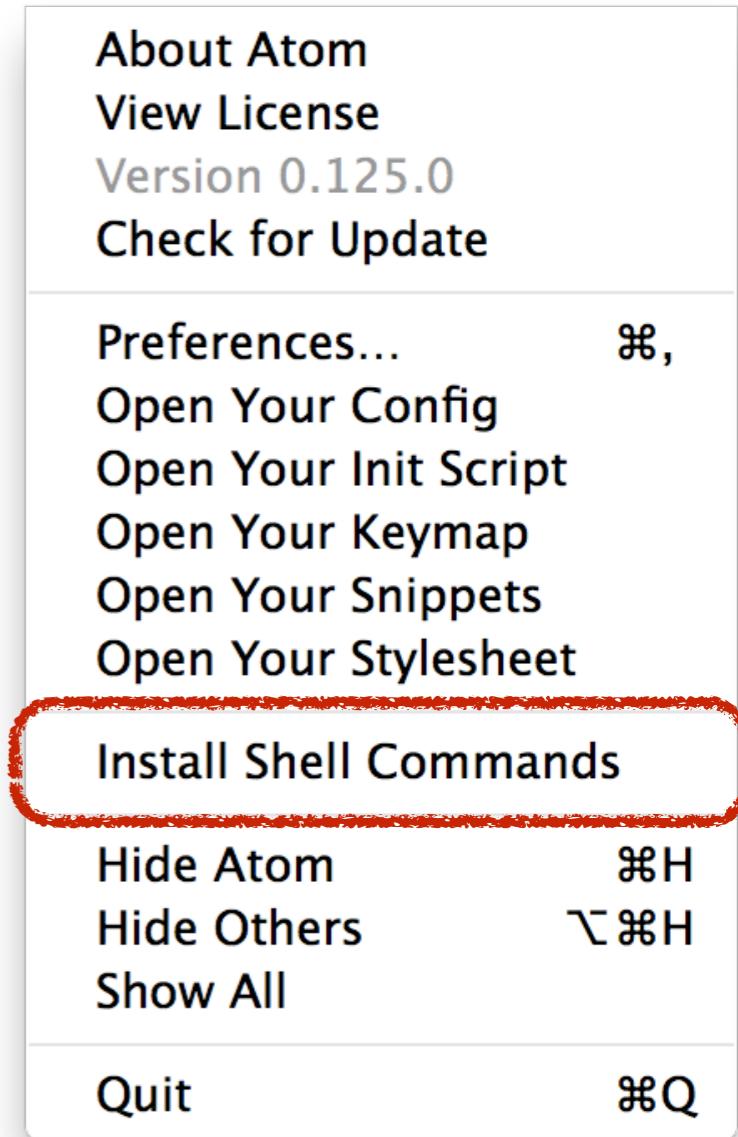
Editing your bash profile

1. Install the atom shell commands.

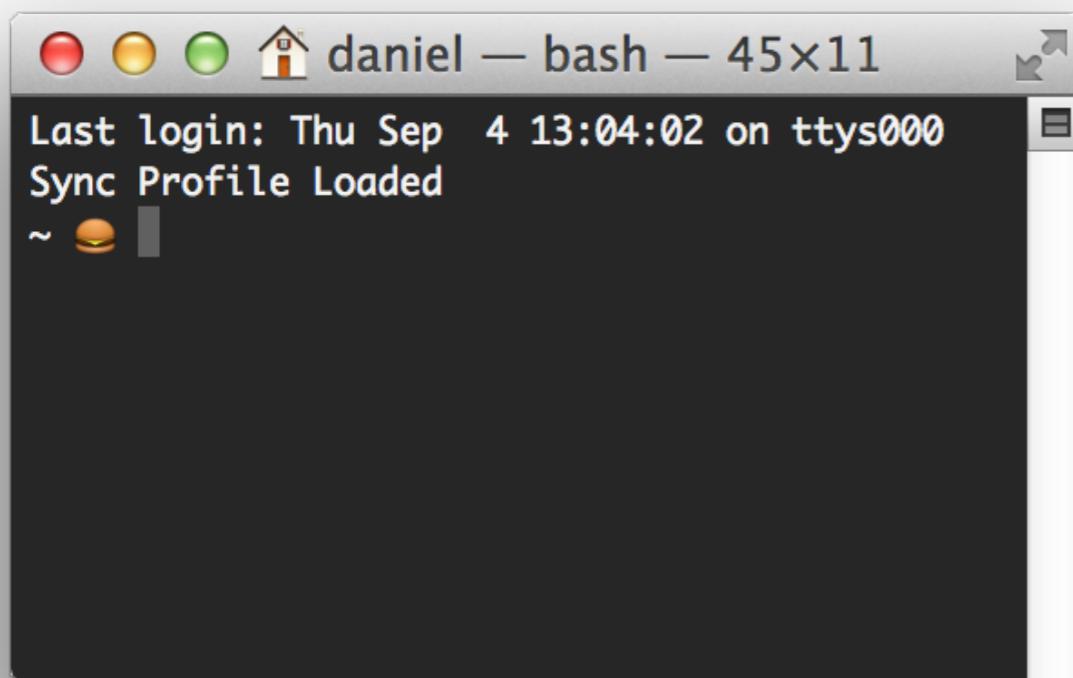
[Atom > Install Shell Commands](#)

2. Type:

`atom .bash_profile`



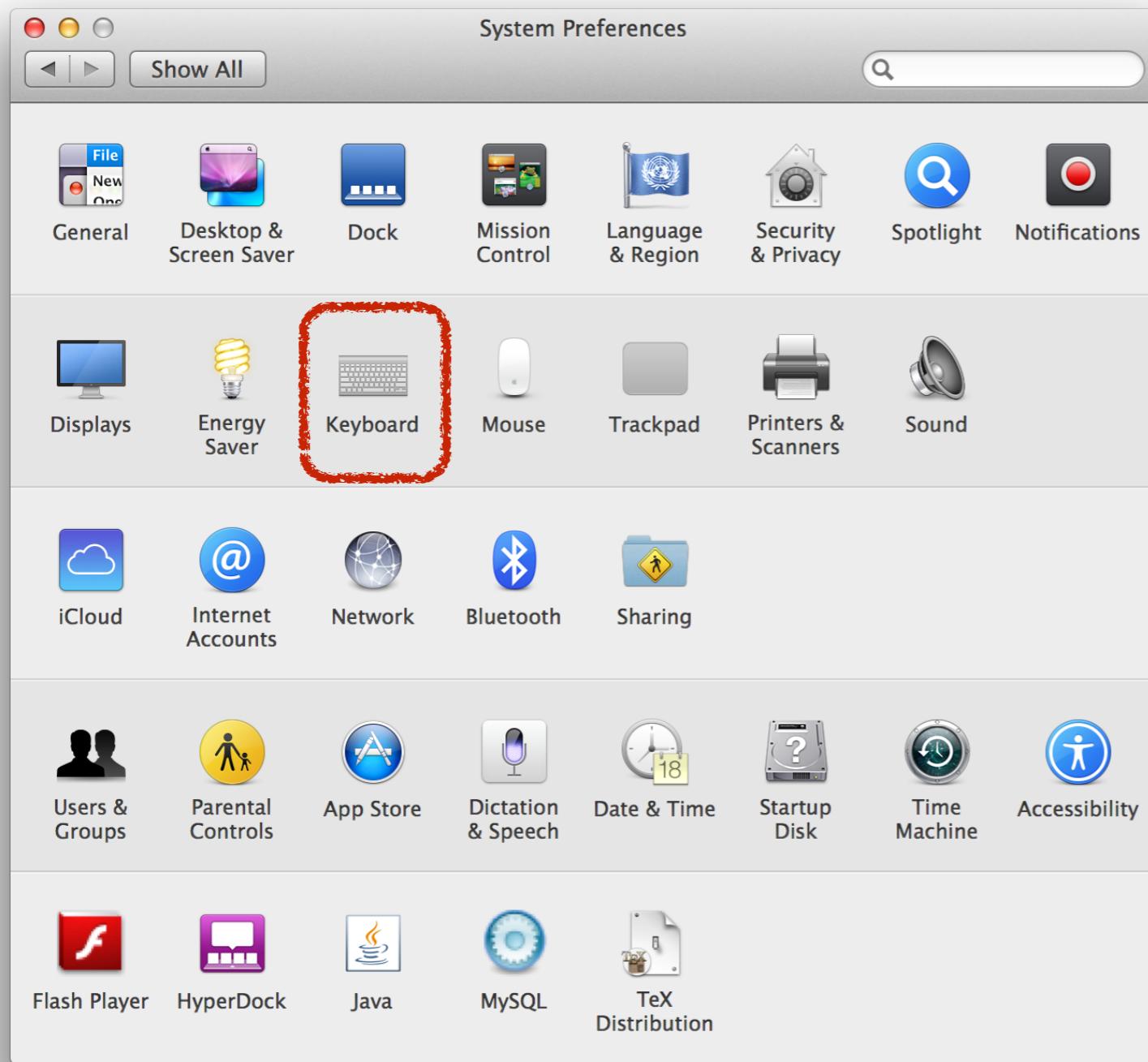
Personalize your Terminal



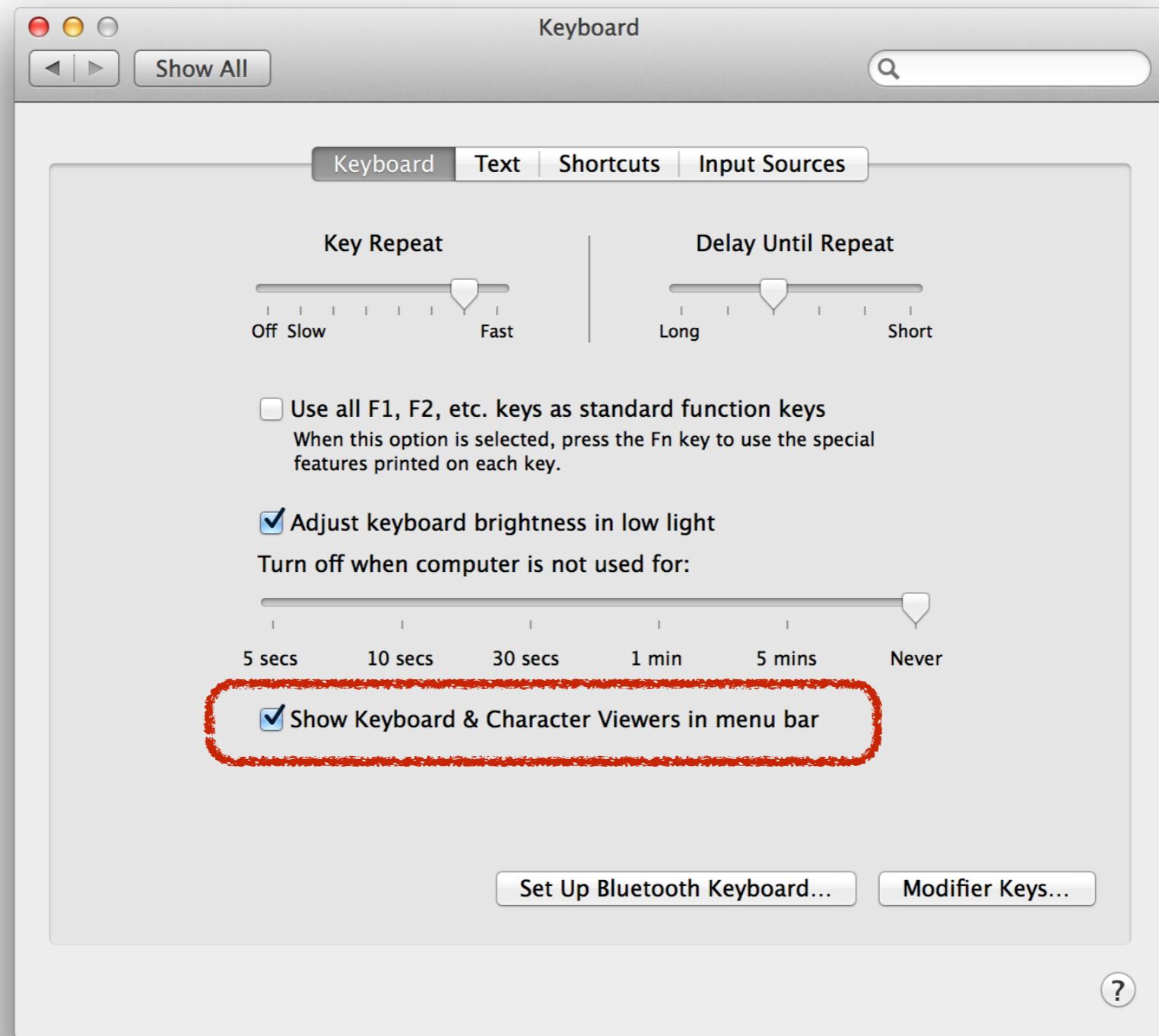
Add an icon to see command
lines



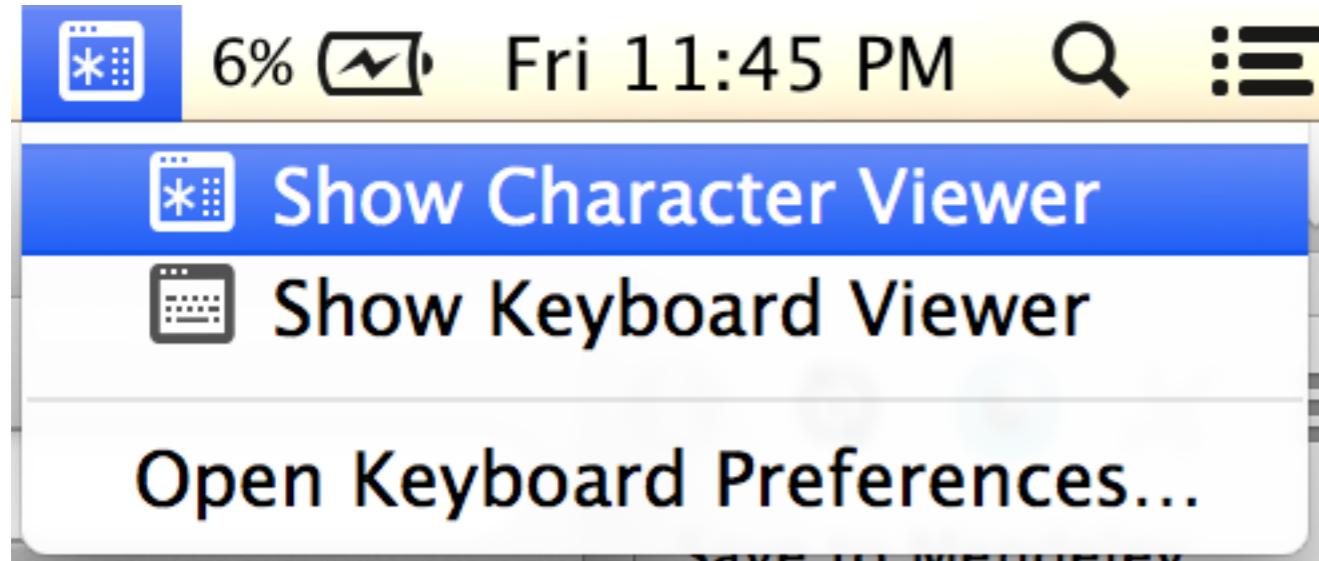
Open System Settings



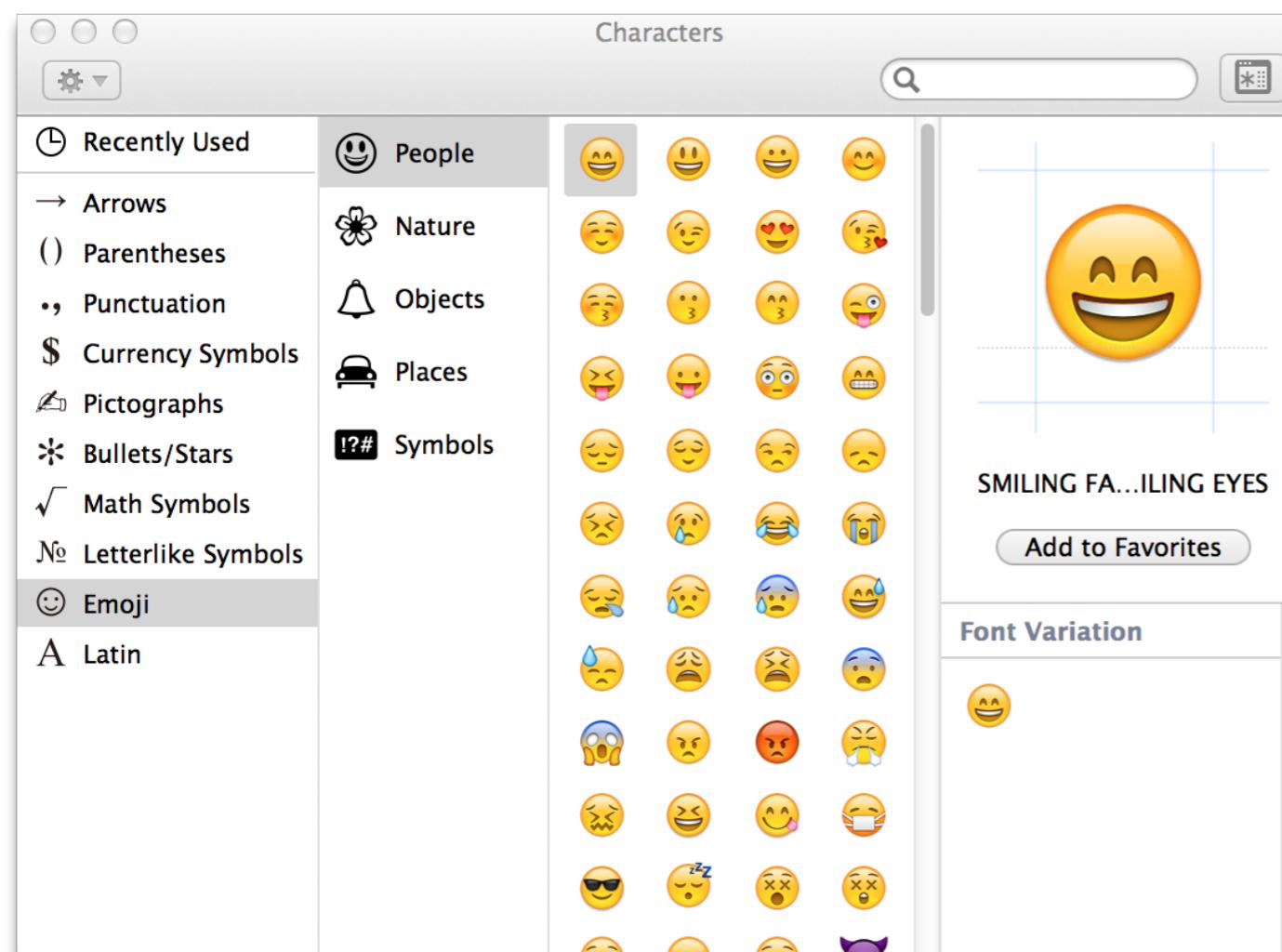
Click "Keyboard"



Check “Show Keyboard & Character Viewers in menu bar”

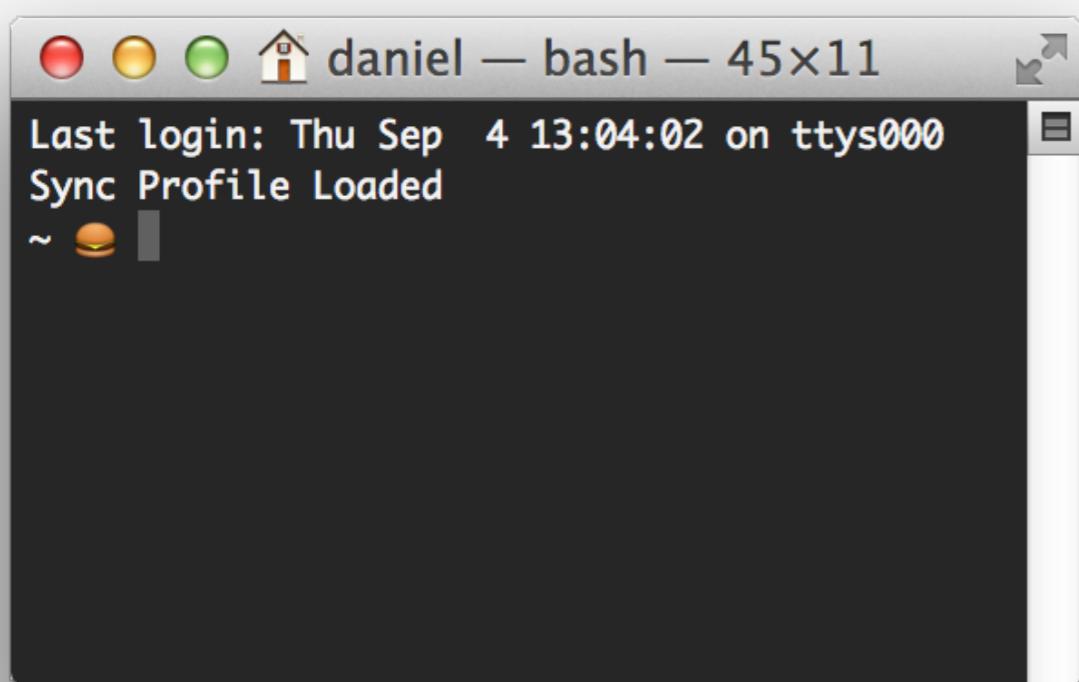


Open the character viewer.



Pick a fun icon!

Personalize your Terminal



Put your cursor here
and double click on the icon

```
export PS1="\w "
```