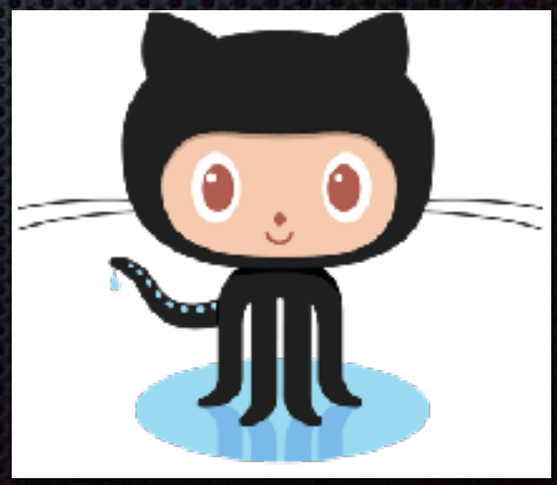
Version Control & Basic Unix Commands

What is GitHub and why do we use it?

- “code hosting platform for version control and collaboration”
- Version control keeps track of your changes and why you made them
- Keeps your code in one place (easy to keep updated when working on multiple computers)
- Allows you to easily collaborate and share your work with others for feedback

Installing GitHub

- https://github.com/ **(where most text from this tutorial is taken)
- https://desktop.github.com/



Important Terms

- **Git**- version control GitHub is built on top of
- **GitHub**- company name of the software that help you interact with Git repositories
- **GitHub.com**- website that you log into to view repositories online
- **GitHub Desktop**- application that helps you synchronize the local code on your computer with GitHub.com

Important Terms

- ***repository***- contain folders and files, images, videos, spreadsheets, and data sets and is used to organize a single project
  - We will have a “Kalan Lab” Repository for general lab scripts. The test repository today is for practice.
  - Can be ***private*** (only you or your collaborators can see code in it, but requires academic or paid subscription) or ***public*** (anyone can see it but you choose who is allowed to make changes to it)
  - Stored in two places
    - locally on your computer- you can work on it without internet connection
    - remotely on GitHub.com

Important Terms

- ***commit***- a saved change that is associated with a ***commit message*** denoting why the change was made
- once commit changes to your local repository, you want to ***push*** these changes to the remote repository
- before you start working, you will want to ***pull*** changes that have been made to make sure you are working with the most up to date copy
- if you made a bad change and want to go back, you can ***revert*** a commit

Important Terms

- ***branching***- a way to work on more than one version of a repository at a time
  - *Generally we are all working on the main branch
- ***pull request***- proposing changes and requesting that someone review and pull in your contribution and merge them into their branch

Important Terms

- ***Cloning*** a repository creates a new local copy of a repository on your computer
- A ***fork*** is also a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project

Kalan Lab Repository

- **bin**- basic lab scripts for things like running QIIME, calculating fasta or fastq read lengths, etc.
- **mapping_files**
  - FreezerTracking
  - metadata- sample information associated with studies
  - run_maps- sequencing run information
- **reference**
- **scripts**- contains folder for each lab member to store project specific scripts

Basic Unix Commands

- **man *command*** - gives you the manual with information on what the command does and different flags you can use
- **echo** - write arguments to standard output
- **ls** - lists your files
- **ls -hl** - lists your files in human readable long format
- **ls -a** - lists all files
- **du -h** tells you disk usage

Basic Unix Commands

- **pwd** - print working directory (i.e. tells you where you are)
- **mkdir *dirname***- makes a new directory
- **cd *dirname*** - change directory
- **rmdir *dirname***- removes a new directory ** be very careful- this permanently deletes files (rather than sending to trash can)

Basic Unix Commands

- **mv *filename1 filename2***- moves a file (i.e. renames a file or moves it to a different directory)
- **cp *filename1 filename2***- copies a file
- **rm *filename*** - removes a file ** be very careful- this permanently deletes files (rather than sending to trash can)
- adding a **-r** flag to each command means to apply the commands recursively (i.e. to the listed directory and all contents within the directory)

Commonly used text editors

- **emacs**
- **vim**
- **nano**

- nano *filename.txt*
- control+o = write file
- control+x= exit

Using GitHub

- Make a new temporary file under the scripts folder in your username directory
- Commit this change
- Alternatively, you can use the GUI…

cd ~/Kalan_lab/scripts/lkalan/

git pull

git status*

mkdir test

cd test

nano test.txt

git add test.txt

git commit -m “adding a test file”

git status*

git push

*optional

Other Fun Unix Commands

- **wc *filename*** - tells you how many lines, words, and characters are in a file
  - if working with a single-line fasta file, where the first line is the header and the second line is the sequence, you can use **wc -l** to count the number of lines in the file and divide it by 2 to get your sequence count (for fastq files, you divide by 4)
- **grep *string filename(s)*** - searches for a string within a file
  - if working with fasta file, grep for the header carrot “>” and use the -c flag to count the number of sequences in your file

Other Fun Unix Commands

- **chmod *options filename*** - lets you change the read, write, and execute permissions of a file
  - ex. chmod u=rwx,g=rx,o=r myfile.txt
- **whoami** - returns your username

Executing scripts

- A script is a list of commands in a a single file
- The first line in the script contains the “shebang" (#!) that tells the system which interpreter to use
  - So, for a bash script, the shebang would be “#!/bin/sh”, whereas if you wrote the script in perl, it would be “#!/bin/perl”
- Let’s write and execute a test bash script…

Copying files

- **scp** (secure copy)
- **scp *source destination***

Alternative: use FileZilla (https://filezilla-project.org/)