BIOL425 Comp Mol Bio

Part 1. Git & Unix

Professor Weigang Qiu

Hunter College of CUNY

Spring 2023

Version Control with Git (Chapter 2)

- I will use it to share files (e.g., slides), as an alternative to Blackboard
- Later (hopefully), students will be able to upload files
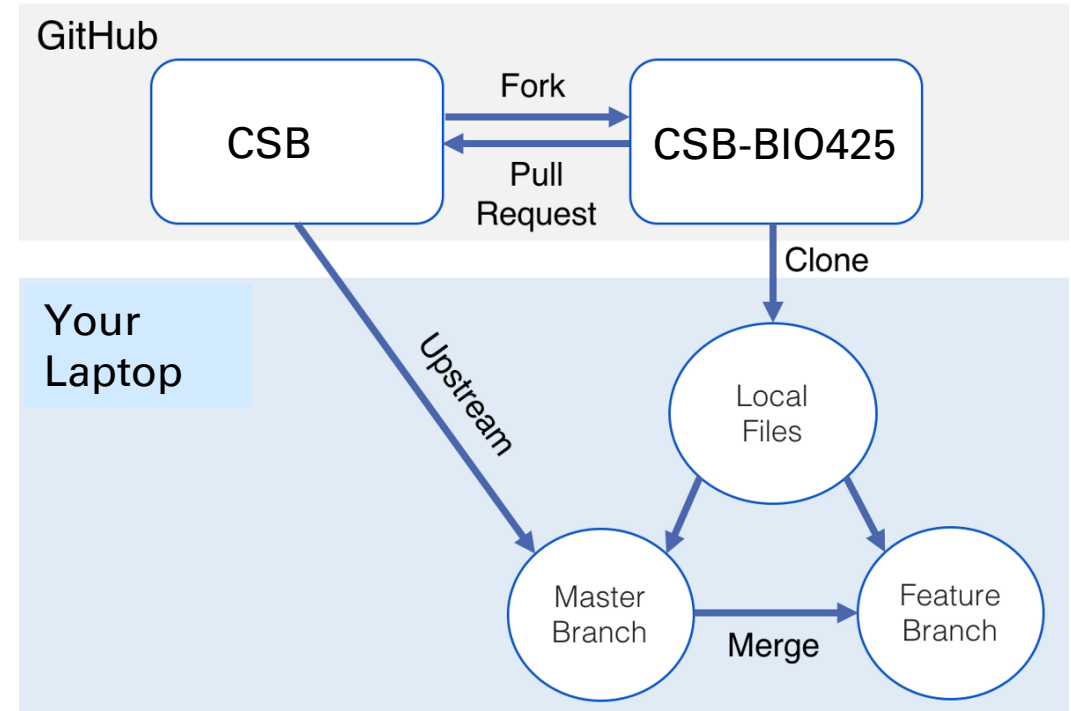
1. Download course repository

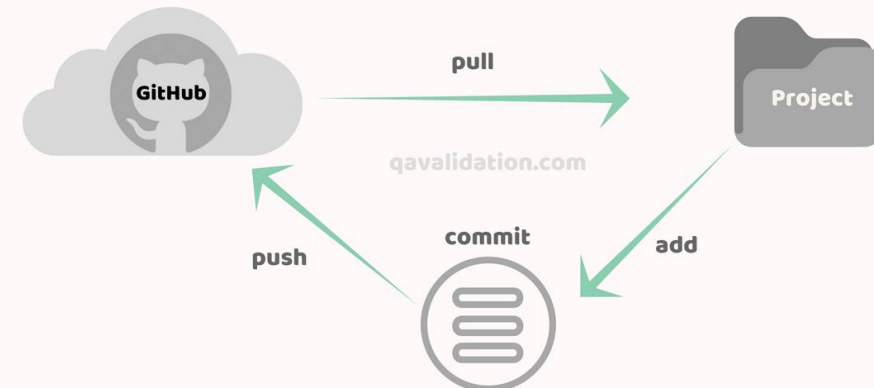   git clone https://github.com/weigangq/CSB-BIOL425.git

2. Pull the latest versions

   git pull

3. Upload/Update file (ignore; not yet working; we will work on it later)

   git add <filename> # add a new file

   git commit -m "message"

   git push

GitHub

CSB — Fork → CSB-BIO425

CSB ← Pull Request — CSB-BIO425

Clone

Your Laptop

Upstream

Local Files

Master Branch — Merge → Feature Branch

Git PUSH PULL

GitHub — pull → Project

push   commit   add

qavalidation.com

UNIX Basics (Chapter 1)

```
1. Directory listing & shorthand
   ls -lrth       # long, reverse, timestamp, human-readable
   pwd            # present working directory
   ~              # home directory
   .              # current directory
   ..             # parent directory
2. Keyboard shorthand (for command editing)
   Ctrl-a         # go to the beginning
   Ctrl-e         # go to the end
   Ctrl-l         # clear the screen
   Ctrl-u         # clear text before the cursor
   Ctrl-k         # clear text after the cursor
   Ctrl-c         # kill the (stalled) command
3. Directory navigation
   cd ~/CSB-BIOL425/python/data     # absolute path
   Cd ../../unix/data               # relative path
   cd -                             # toggle 2 directories
4. Use auto-completion & command history
   Tab                       # NEVER type out a full filename
   Arrow keys                # EDIT & NEVER retype a command
   ls bad\ file\ name.txt    # NEVER use spaces in filenames
```

3

Exercise 1

Intermezzo 1.1

(a) Go to your home directory.

(b) Navigate to the sandbox directory within the CSB/unix directory.

(c) Use a relative path to go to the data directory within the python directory.

(d) Use an absolute path to go to the sandbox directory within python.

(e) Return to the data directory within the python directory.

UNIX Basics (Cont'd)

```
1. Copy files and directories
   cp ~/CSB-BIOL425/unix/data/Buzzard2015_about.txt ~/CSB-BIOL425/unix/sandbox/      # use absolute path
   cd ~/CSB-BIOL425/unix/sandbox/
   cp ../data/Buzzard2015_about.txt .                                # use relative path
   cp ../data/Buzzard2015_about.txt ./Buzzard2015_about2.txt         # copy & rename
   cp -r ../data .                                                   # recursive copy
2. Move or rename a file
   mv Buzzard2015_about2.txt ../data/                           # move file to a different directory
   mv ../data/Buzzard2015_about2.txt ../data/Buzzard2015_about_new.txt      # move and rename
3. Remove file or directory
   touch new_file.txt           # create an empty file (and update timestamp)
   rm -i new_file.txt           # Proceed with caution; EXTREMELY destructive
   mkdir -p d1/d2/d3            # make nested directories
   rm -r d1                    # recursively remove a directory (and its sub-directories)
4. View & filter text files
   cd  ~/CSB-BIOL425/unix/data
   less Marra2014_data.fasta # spacebar to page down; b to page up; Q to quit
   cat *.txt                   # concatenate all ".txt" files
   wc *.txt                    # word count (all ".txt" files)
   head Gesquiere2011_data.csv      # show top lines
   tail -n 2 Gesquiere2011_data.csv  # show tail two lines
   sort Gesquiere2011_data.csv      # sort lines in a file (alphabetically)
   sort -n Gesquiere2011_data.csv   # sort lines numerically
```

Exercise 2

Intermezzo 1.2

To familiarize yourself with these basic Unix commands, try the following:

(a) Go to the data directory within CSB/unix.

(b) How many lines are in file Marra2014_data.fasta?

(c) Create the empty file toremove.txt in the CSB/unix/sandbox directory without leaving the current directory.

(d) List the contents of the directory unix/sandbox.

(e) Remove the file toremove.txt.

UNIX Advanced: "cut"

```
1. Redirect output (save output to file)
   cd ~/CSB-BIOL425/unix/sandbox
   echo "My first line" > test.txt                    # redirect echo output to a new file
   echo "My second line" >> test.txt                  # append a second line
   cat test.txt                                        # show file content
   ls -lrt ../data/Saavedra2013 > filelist.txt         # list files and save the list to a file
   cat filelist.txt                                    # show file
   ls  ../data/Saavedra2013 | wc -l                    # use pipe (|) to count # of files
2. Select columns using "cut"
   cd ~/CSB-BIOL425/unix/data
   head Pacifici2013_data.csv                          # show top 10 lines
   head Pacifici2013_data.csv | cut -d ";" -f 1        # select 1st field, delimited by ";"
   head Pacifici2013_data.csv | cut -d ";" -f 1-4      # select columns 1-4
   cut -d ";" -f 2 Pacifici2013_data.csv | tail -n +2  # select 2nd column, skip header (1st line)
   cut -d ";" -f 2 Pacifici2013_data.csv | tail -n +2 | sort | uniq       # show unique lines
```

Intermezzo 1.3

(a) If we order all species names (fifth column) of `Pacifici2013_data.csv` in alphabetical order, which is the first species? Which is the last?

(b) How many families are represented in the database?

UNIX Advanced: "tr" & "sed"

1. **Character substitutions with "tr"**
   ```
   echo "ACtGGcAaTT" | tr 'actg' 'ACTG'      # lower to upper cases
   echo "ACtGGcAaTT" | tr 'a-z' 'A-Z'        # same as above
   echo "aaacttGGcaa" | tr -d 'a'            # delete all 'a'
   echo "aaacttGGcaa" | tr -s 'a'            # squeeze consecutive 'a'
   ```
2. **Exercise: build a single command for the following tasks**
   ```
   cd ~/CSB-BIOL425/unix/sandbox
   1. Remove header from the file "../data/Pacifici2013_data.csv"
   2. Select columns 2-6 (Order, Family, Genus, Scientific_name, AdultBodyMass_g)
   3. Substitute ':' with a Tab ("\t")
   4. Sort by body mass, larger values first
   5. Save to a file "BodyM.tsv"
   ```
3. **String substitution with "sed"**
   ```
   cd ~/CSB-BIOL425/unix/data
   cat Bb-filelist.txt | sed "s/Borreliella/Borrelia/"    # substitute a word
   cat Bb-filelist.txt | sed "s/Borreliella/Borrelia/g"   # substitute globally
   cat Bb-filelist.txt | sed "s/ NCBI //"                 # remove a string
   ```
4. **Use wildcards to process multiple files**
   ```
   cd ~/CSB-BIOL425/unix/data/miRNA
   wc -l *.fasta              # count # lines for all ".fasta" files
   head -n 2 pp*             # show top two lines for all files starts with "pp"
   file *.???               # find file types for files with 3-letter extensions
   Exercise: find file type for all FASTA files; remove the string "_miR" from each line
   ```

UNIX Advanced: "grep"

```
cd ~/CSB-BIOL425/unix/sandbox
grep "Vombatidae" BodyM.tsv                          # filter lines containing a term
grep -c "Vombatidae" BodyM.tsv                       # count # of lines containing term
grep -w "Bos" BodyM.tsv                              # match only a full word
grep -i "Bos" BodyM.tsv                              # case-insensitive match
grep -B 2 -A 2 "Gorilla gorilla" BodyM.tsv           # include two lines before and after
grep -n "Gorilla gorilla" BodyM.tsv                  # show line number of the match
grep Gorilla BodyM.tsv | grep -v gorilla             # show lines without a match (reverse)
grep -w "Gorilla\|Pan" BodyM.tsv                     # "\|" to match any of multiple strings
```

Intermezzo 1.4

(a) Navigate to CSB/unix/sandbox. Without navigating to a different location, find a CSV file that contains Dalziel in its file name and is located within the CSB directory. Copy this file to the Unix sandbox.

(b) Print the first few lines on the screen to check the structure of the data. List all unique cities in column loc (omit the header). How often does each city occur in the data set?

(c) The fourth column reports cases of measles. What is the maximum number of cases reported for Washington, DC?

(d) What is the maximum number of reported measles cases in the entire data set? Where did this occur?

UNIX Advanced: "for" loops & BASH scripting

1. Loop through files, strings, and numbers
```
   cd ~/CSB-BIOL425/unix/data/miRNA                       # microRNA data
   for file in *.fasta; do head –n 2 $file; done     # show top 2 lines for each fasta file
   # find three miRNA across fasta files and save each to a new fasta file:
   for miR in miR-208a miR-564 miR-3170; do grep $miR –A1 *.fasta > $miR.fasta; done
   # increment an index:
   for i in {1..10}; do echo $i; done                    # increment by 1
   for i in {1..10..2}; do echo $i; done                 # increment by 2
```
2. BASH scripting
```
   1) Download, install & start an editor (vi, emacs, gedit, NotePad++)
   2) cd ~/CSB-BIOL425/unix/sandbox
   3) Paste the following & save as "extract_body_mass.bash": tail -n +2
      ../data/Pacifici2013_data.csv | cut -d ";" -f 2-6 | tr ";" "\t" | sort -r -n -k 6 >
      BodyM.tsv
   4) bash extract_body_mass.bash               # Run script
   5) Add the bash path to the beginning & save: #!/usr/bin/env bash
   6) chmod +x extract_body_mass.bash           # change permission to make executable
   7) ./extract_body_mass.bash                  # Run script again
   8) Make input and output filenames as arguments; Add comments (see next slide)
   9) ./extract_body_mass.bash ../data/Pacifici2013_data.csv BodyM.tsv
```

```bash
File Edit Options Buffers Tools Sh-Script Help
#!/usr/bin/env bash

#########################
# function of script:
# take a CSV file delimited by ";" (first argument)
# remove the header
# make tab separated
# sort according to the 6th (numeric) column
# in descending order
# redirect to a file (second argument)

# Author: Weigang Qiu
# Date: Jan 30, 2023
#########################

# input file name as the 1st argument
in_file=$1

# output filename as the 2nd argument
out_file=$2

# remove the header
tail -n +2 $1 > $1.tmp1

# extract columns
cut -d ";" -f 2-6 $1.tmp1 > $1.tmp2

# make tab separated
tr ";" "\t" < $1.tmp2 > $1.tmp3

# sort and redirect to output
sort -r -n -k 6 $1.tmp3 > $2

# remove temporary, intermediate files
rm $1.tmp*

exit;
```

```
-UU-:----F1  extract_body_mass.bash   All L36   Git:master  (Shell
```

Add the bash interpreter path as the first line

Add comments as documentation

Make input and output as arguments (Do not hard-code filenames within a script)

```
# save; make it executable
chmod +x extract_body_mass.bash

# Run with arguments:
./extract_body_mass.bash
../data/Pacifici2013_data.csv
BodyM.tsv
```

11

SUMMARY

- Run "git pull" to get the latest files from the course repository

- Unix command line interface (CLI):
  - Highly efficient (faster than Python)
  - Faster than graphic user interface (GUI, point-and-click)
  - No need for programming for most of the text-wrangling

- Next week: Quiz #1, based on the following 4 exercises
  - 1.10.1 Next generation sequencing data
  - 1.10.2 Hormone levels in Baboons
  - 1.10.3 Plant-pollinator networks
  - 1.10.4. Data explorer