

Computational Genomics

Part 1. Git, Unix & BpWrapper



Professor Weigang Qiu
Hunter College of CUNY
Summer 2026

Kunming Institute of Zoology, CAS, China

OVERVIEW

- Run “git pull” to get the latest files from the course repository
- Unix command line interface (CLI):
 - **\$ command [--options] [arguments]**
 - High efficiency (faster than Python)
 - Faster than graphic user interface (GUI, point-and-click)
 - No need for programming for most of the text-wrangling
- BpWrapper tool kits: Hernandez et al (2018). BMC Bioinformatics
 - bioseq: manipulate FASTA sequences
 - bioaln: manipulate CLUSTALW alignments
 - biotree: manipulate NEWICK trees

Version Control with Git

1. Download course repository

```
git clone
```

```
https://gitee.com/huntercollege/comp-  
genomics-kiz.git
```

2. Pull the latest versions

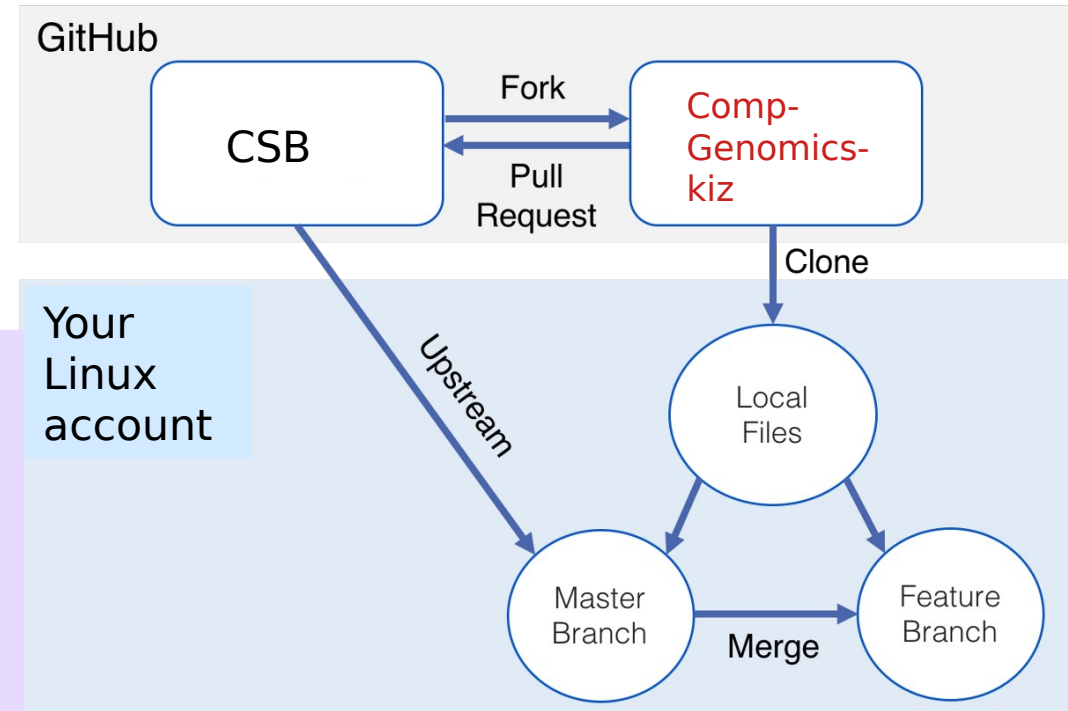
```
git pull
```

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

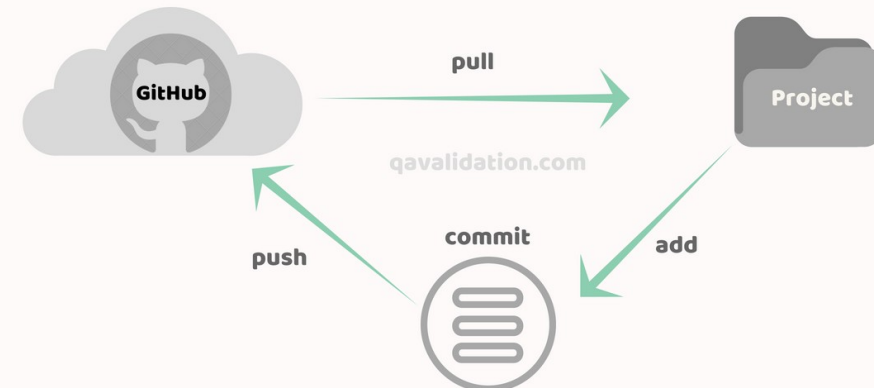
```
git add <filename> # add a new file
```

```
git commit -m "message"
```

```
git push
```



Git PUSH PULL



UNIX Basics

+

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 ~/comp-genomics-kiz/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
```

Exercise 1

Intermezzo 1.1

- (a) Go to your home directory. ~/comp-genomics-kiz/unix
 - (b) Navigate to the sandbox directory within the ~~CSD/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 ~/comp-genomics-kiz/unix/data/Buzzard2015_about.txt ~/comp-genomics-kiz/unix/sandbox/
                                                                    # use absolute path

cd ~/comp-genomics-kiz/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 ~/comp-genomics-kiz/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:

`~/comp-genomics-kiz/unix`

- (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 ~/comp-genomics-kiz/unix/sandbox
echo "My first line" > test.txt
echo "My second line" >> test.txt
cat test.txt
ls -lrt ../data/Saavedra2013 > filelist.txt
cat filelist.txt
ls ../data/Saavedra2013 | wc -l
```

redirect echo output to a new file
append a second line
show file content
list files and save to a file
show file
use pipe (|) to count # of files

2. Select columns using “cut”

```
cd ~/comp-genomics-kiz/unix/data
head Pacifici2013_data.csv
head Pacifici2013_data.csv | cut -d ";" -f 1
head Pacifici2013_data.csv | cut -d ";" -f 1-4
cut -d ";" -f 2 Pacifici2013_data.csv | tail -n +2
cut -d ";" -f 2 Pacifici2013_data.csv | tail -n +2 | sort | uniq
```

show top 10 lines
select 1st field, delimited by “;”
select columns 1-4
select 2nd column, skip header (1st line)
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 ~/comp-genomics-kiz/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 ~/comp-genomics-kiz/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 ~/comp-genomics-kiz/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: “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 ~/comp-genomics-kiz/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 ~/comp-genomics-kiz/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 ~/comp-genomics-kiz/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 ~/comp-genomics-kiz/unix/sandbox
grep "Vombatidae" BodyM.tsv           # filter lines containing a term
grep --color "Vombatidae" BodyM.tsv   # color the 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
```

```
find ~/comp-genomics-kiz -name "*Dalziel*"
```

Intermezzo 1.4

- 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.
- 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?
- The fourth column reports cases of measles. What is the maximum number of cases reported for Washington, DC?
- 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 ~/comp-genomics-kiz/unix/data/miRNA                # microRNA data
for file in *.fasta; do head -n 2 $file; done          # show top 2 lines for each 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 ~/comp-genomics-kiz/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** # input_file=\$1; out_file=\$2
- 9) ./extract_body_mass.bash ../data/Pacifici2013_data.csv BodyM.tsv

BpWrapper: **bioseq --man** (manipulate sequences)

1. Describe sequences

```
cd ~/comp-genomics-kiz/BpWrapper           # Lyme pathogen genome data
bioseq -l B_burgdorferi_B31_genome.fasta    # sequence ids and lengths
bioseq -n B_burgdorferi_B31_genome.fasta    # sequence counts
```

2. Pick or delete sequences

```
bioseq -p "order:3" B_burgdorferi_B31_genome.fasta # pick by order
bioseq -p "re:cp26" B_burgdorferi_B31_genome.fasta # pick by pattern
bioseq -p "id:chromosome" B_burgdorferi_B31_genome.fasta # pick by id
```

3. Transform sequences

```
bioseq -p "order:1" B_burgdorferi_B31_ORFs.fasta > b01.nuc # pick a seq & save
bioseq -r k01.nuc                                           # reverse & complement
bioseq -t1 k01.nuc                                           # translate (1st reading frame)
bioseq -t3 k01.nuc                                           # translate in 3 reading frames
bioseq -t6 k01.nuc                                           # translate in all 6 reading frames
```

4. Look up genetic code

```
bioseq --codon-table "AAA" # amino acid encoded by the codon
bioseq --codon-table "L"  # all codons of an amino acid
```

BpWrapper: **bioaln --man** (manipulate an alignment)

1. Describe an alignment

```
cd ~/comp-genomics-kiz/BpWrapper          # Lyme pathogen genome data
bioaln -l Borreliella_OspC.aln             # alignment lengths
bioaln -n Borreliella_OspC.aln             # count sequences in the alignment
bioaln -L Borreliella_OspC.aln             # list sequence ids
bioaln -a Borreliella_OspC.aln             # Calculate average sequence identity
```

2. Pick or delete sequences

```
bioaln -p "B.sinica_CMN3,B.afzelii_K78" Borreliella_OspC.aln # pick sequences (by ids)
bioaln -d "B.sinica_CMN3,B.afzelii_K78" Borreliella_OspC.aln # delete sequences (by ids)
```

3. Manipulate an alignment

```
bioaln -g Borreliella_OspC.aln             # remove gaps
bioaln -r "B.sinica_CMN3" Borreliella_OspC.aln # change the reference (1st) seq
bioaln -s "10,100" Borreliella_OspC.aln    # get a sub-alignment
```

4. Evolutionary analysis

```
bioaln -o "phylip" Borreliella_OspC.aln    # transform alignment format
bioaln --pair-diff Borreliella_OspC.aln    # get pairwise seq diffs
```

```
1) bioseq -t1 B_burgdorferi_ospC.nuc > B_burgdorferi_ospC.nuc # translate
2) muscle -align B_burgdorferi_ospC.pep -output B_burgdorferi_ospC.aln # align peps
3) bioaln --pep2dna B_burgdorferi_ospC.nuc -i'fasta' B_burgdorferi_ospC.aln >
   B_burgdorferi_ospC_cds.aln # Get a codon alignment
```

BpWrapper: **biotree --man** (manipulate a tree)

1. Describe a tree

```
cd ~/comp-genomics-kiz/BpWrapper          # Lyme pathogen genome data
biotree -l Borreliella_species_tree.dnd    # get tree length
biotree -u Borreliella_species_tree.dnd    # list OTUs
biotree -n Borreliella_species_tree.dnd    # show number of OTUs
biotree --dist-all Borreliella_species_tree.dnd # tree distance between all OTU pairs
```

2. Manipulate a tree

```
# delete taxa:
biotree -d'B.garnii,B.chilensis' Borreliella_species_tree.dnd Borreliella_OspC.aln

# get a tree for a subset of taxa
biotree -s'B.burgdorferi,B.afzelii,B.garnii,B.chilensis' Borreliella_species_tree.dnd

# Collapse weakly supported branches
biotree -D 0.9 Borreliella_species_tree.dnd

# reroot with an outgroup
biotree -r "otu:B.maritima" Borreliella_species_tree.dnd

# reroot on mid-point
biotree -m "otu:B.maritima" Borreliella_species_tree.dnd
```