

## Quiz Section Week 9

comp bio tools

2019-05-30

### practical computing

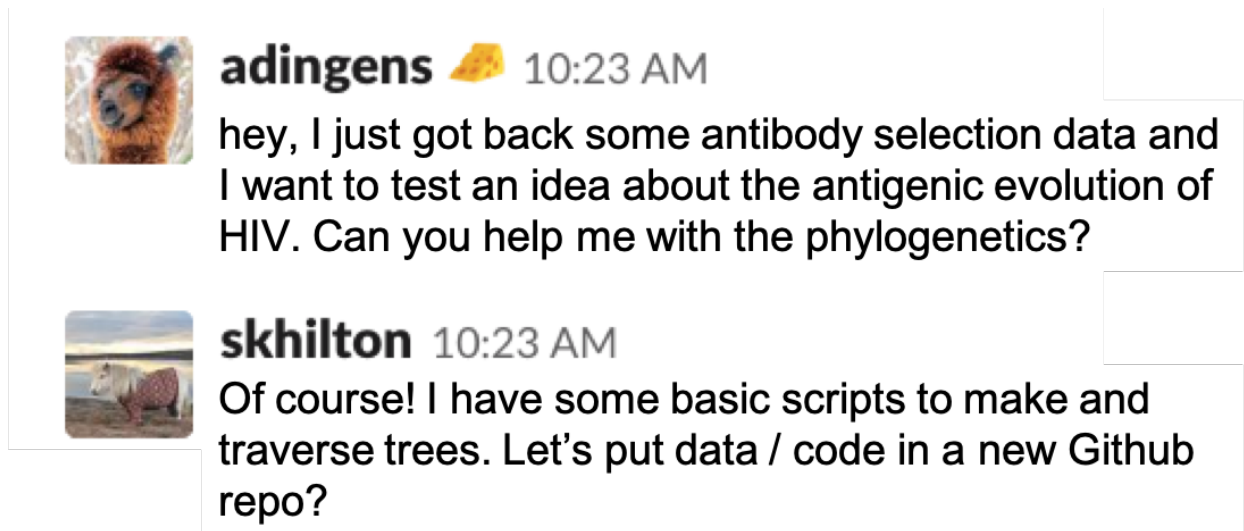


Figure 1:

### version control / git

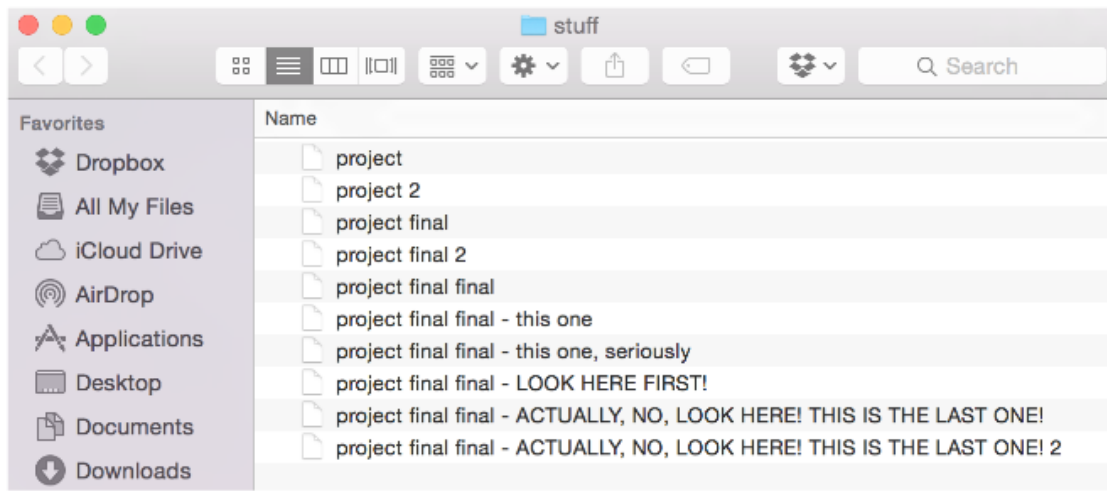
#### why version control?

#### why version control?

- facilitate collaboration
- restore previous versions
- backup your work
- track your code as you write it

### git vs. Github

- **Git** is the version control software - it's the software keeping track of the changes/versions



DOGHOUSEDIARIES

Figure 2:

```

68 69     def cmd_exists(cmd):
69 70         return any(
70 71             os.access(os.path.join(path, cmd), os.X_OK)
✱ @@ -75,6 +76,10 @@ def cmd_exists(cmd):
75 76         logging.info('ERROR bedtools not available')
76 77         sys.exit()
77 78
79 +     if not cmd_exists("samtools"):
80 +         logging.info('ERROR samtools not available')
81 +         sys.exit()
82 +
78 83     # Make sorted bed file from all bams
79 84     qcf = open(qc_info, 'a')
80 85     if len(args.bamlist) == 1:
✱ @@ -85,15 +90,15 @@ def cmd_exists(cmd):
85 90         args.force_overwrite_all:
86 91         args.force_overwrite_all = True
87 92         logging.info('Merge started.')
88 -         subprocess.check_call('samtools merge %s.merge.bam %s' % (OUTPUT_PREFIX,
93 +         subprocess.check_call('module load samtools/latest; samtools merge %s.merge.bam %s' % (OUTPUT_PREFIX,

```

Figure 3:

- Github is a website that allows you to store the version information off your computer
- Github users have IDs (ex. skhilton)
- Github accounts are free

## Github website vocab

- users
- repos

clone, push, pull

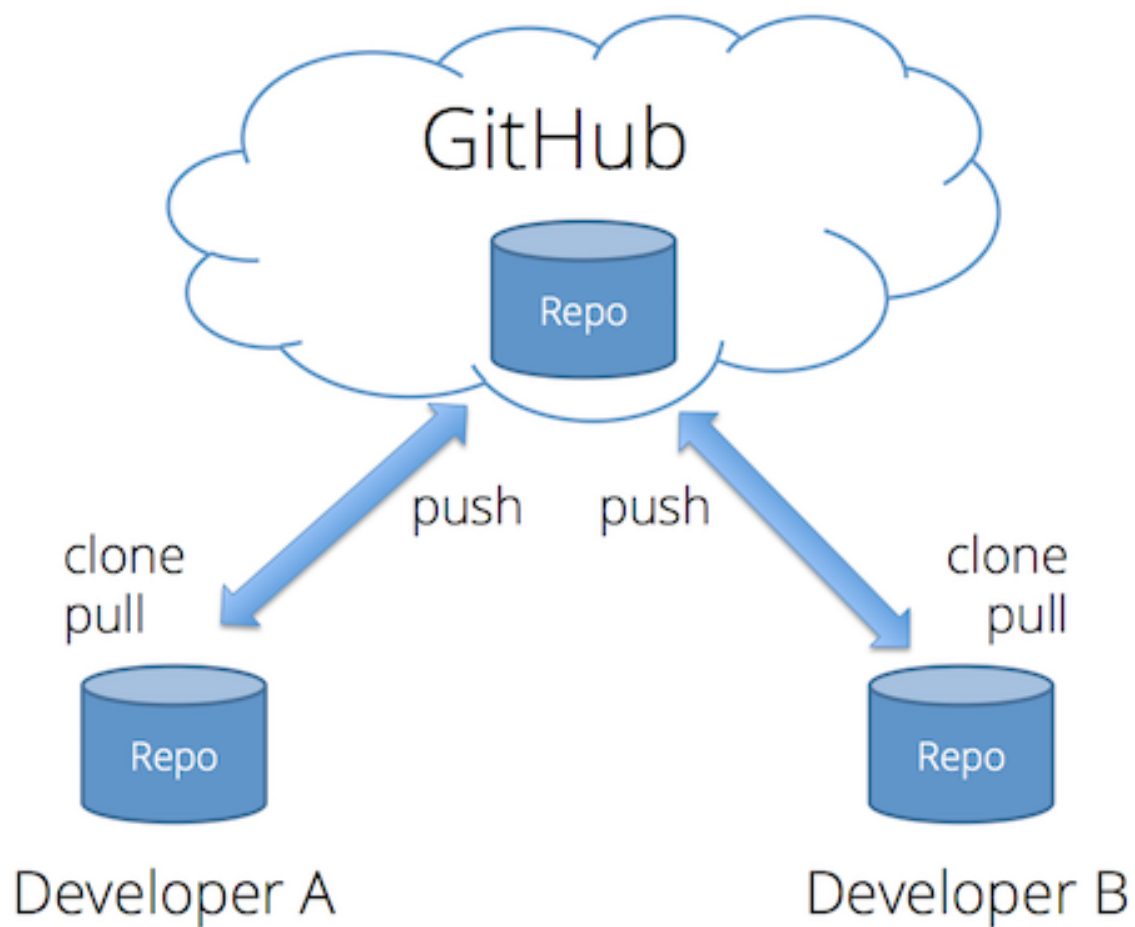


Figure 4:

**vocab**  
remote, local  
**source**

## how to clone a repo

```
git clone <repo address>
```

## common sequence of commands

adding local files to remote repo

```
git status
```

```
git add <file name>
```

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

```
git push
```

pulling files from remote repo to my local repo

```
git pull
```

## Want to test Github out?

- create an account `github.com`
- create a repo called `homework9`
- feel free to email me with questions if you need help cloning/adding/pushing

## a (forever?) learning curve

## package managers

## what is a package manager?

tool that allows you to install and manage additional libraries not part of a standard distribution

- what are some standard libraries we have used this quarter?

## what is the python package manager?

`pip!`

## how to use pip

```
pip --version
```

```
pip install <package name>
```

```
pip install --upgrade <package name>
```



Figure 5:

what are some modules you may want to install right now?

- `pandas`
- `numpy`
- `scipy`
- `BioPython`
- `scikit-learn`

redo of hw6 with BioPython

[github link](#)

environments

what is anaconda / conda / miniconda?

`anaconda`

cross platform package and environment manager

...

“it’s like `pip` for non-python things and `pip` for python things and more”

what’s the difference between anaconda and pip?

This highlights a key difference between conda and pip. Pip installs Python packages whereas conda installs packages which may contain software written in any language... Another key difference between the two tools is that conda has the ability to create isolated environments that can contain different versions of Python and/or the packages installed in them.

what is an environment?

- environments allow you to specify *exact* versions of software and specify dependencies
- environments are helpful when you are developing software or starting a new project because you get a “clean slate”

summary

Programming as a computational biologist

- your code should be available to others ([Github](#))
- your code should be understandable to others (documentation!)

- your code should be runnable by others (this is very hard!)
- be conscious of when you should implement something yourself (more control) and when you should use preexisting packages (don't reinvent the wheel)

**call for questions for the review session**