Version Control and Reproducible Research with GitHub

Tad Dallas

December 2013

What is GitHub?

"Dropbox on crack"
Code sharing, publishing and development service for collaborative projects

Why use it?

- Version control
- Open collaboration with other scientists
- Creepily watch what other people are working on!

Under the hood

- Git is the version control language that GitHub is the GUI for
- Created by Linus Torvalds, a central developer of the Linux OS
- Command-line, but really straight-forward

A couple quick definitions

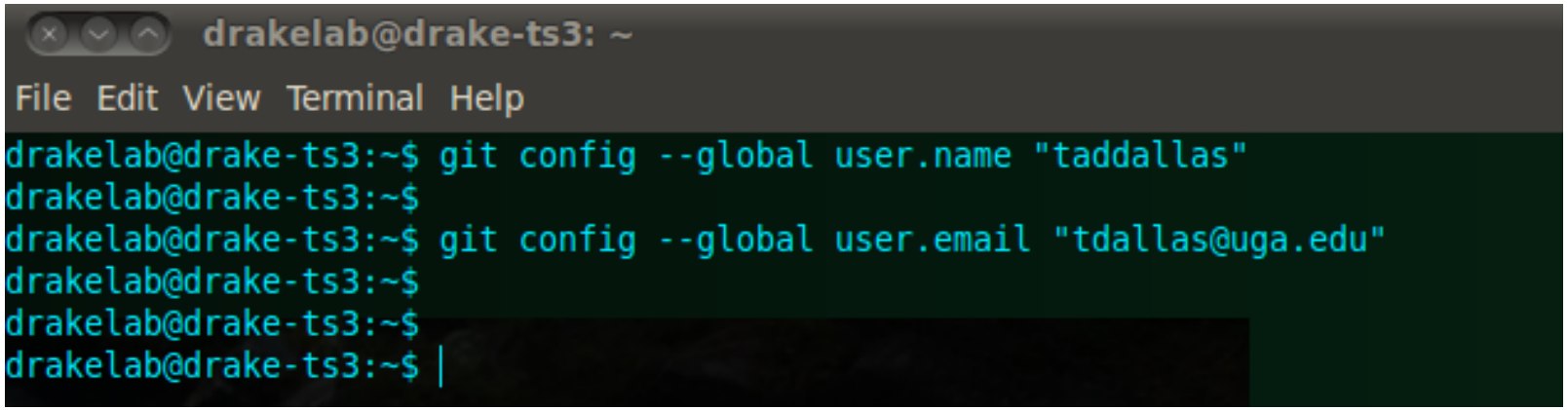
- **Repository**: Storage space where your projects reside
- **Commit**: Takes 'snapshot' of your repository, so you can log a new change, or revert to a previous state (Common command)
- **Branch**: Think of a folder within a repository, but cooler. More on this later
- **Fork**: What it sounds like. You're taking someone's project and making a copy of it for your own use (either to collaborate and to merge later or to use as a template for a different project)
- **Push**: The act of updating your project files (you will "push" your **commits**)
- **Pull**: Gets commits from a repository to your machine
- **Fetch**: A better version of **pull** that doesn't merge **commits**

How to begin

1. Set up an account (go to https://github.com/) and download Git (http://git-scm.com/downloads)
2. Open a terminal window

How to begin

1. Set up an account (go to https://github.com/) and download Git (http://git-scm.com/downloads)
2. Open a terminal window



Okay. Now we have Git on our machines and GitHub accounts.

The GitHub framework

Getting your files on GitHub

- Do work in a local directory
- Create a Git repo in this local directory
- **Add** and **commit** your files ('put stones in the catapult')
- **Push** your files to Github ('catapult those stones')

Make your directory

```
drakelab@drake-ts3:~$ cd /media/MYLIFE/githubTutorial
drakelab@drake-ts3:/media/MYLIFE/githubTutorial$ mkdir githubTutorial
drakelab@drake-ts3:/media/MYLIFE/githubTutorial$ cd githubTutorial
drakelab@drake-ts3:/media/MYLIFE/githubTutorial/githubTutorial$ git init
Initialized empty Git repository in /media/MYLIFE/githubTutorial/githubTutorial/
.git/
drakelab@drake-ts3:/media/MYLIFE/githubTutorial/githubTutorial$ touch README
drakelab@drake-ts3:/media/MYLIFE/githubTutorial/githubTutorial$
```

Commit your README file

```
drakelab@drake-ts3:/media/MYLIFE/githubTutorial/githubTutorial$ git add README
drakelab@drake-ts3:/media/MYLIFE/githubTutorial/githubTutorial$ git commit -m "F
irst commit"
[master (root-commit) 0784b55] First commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README
drakelab@drake-ts3:/media/MYLIFE/githubTutorial/githubTutorial$
```

Push it!

```
drakelab@drake-ts3:/media/MYLIFE/githubTutorial/githubTutorial$ git remote add o
rigin https://github.com/taddallas/githubTutorial.git
drakelab@drake-ts3:/media/MYLIFE/githubTutorial/githubTutorial$
drakelab@drake-ts3:/media/MYLIFE/githubTutorial/githubTutorial$ git push origin
master
Username for 'https://github.com': taddallas
Password for 'https://taddallas@github.com':
```

General framework for edits thereafter

1. Edit your files locally
2. $ git add *files in your local directory*
3. $ git commit -m "message about this commit"
4. $ git push origin master

This skips the *git remote add origin 'your repo'* step, as you do this once per repo.
Also, steps 2 and 3 can be combined using $ git commit -a -m 'message about commit'

How to collaborate using GitHub

Up to this point, it's been a solitary experience of **making** and **pushing**

Methods of collaboration

Two ways:

- **'Fork and Pull' model** : better
- **'Shared Repository' model** : easier

Pulling files from GitHub

- cd to local repository
- $ git remote -v outputs the .git repos you can push/pull to/from. Use $ git remote add 'http://github.com/name/project.git' if necessary
- $ git pull . fetches and merges files

Forking a repo *from* GitHub

This repository | Search or type a command | Explore | Gist | Blog | Help | taddallas

PUBLIC taddallas / metacom

Unwatch 1 | Star 0 | Fork 0

R package for the analysis of metacommunity structure — Edit

29 commits | 1 branch | 0 releases | 1 contributor

branch: master | metacom /

Update README.md ···

taddallas authored 18 hours ago | latest commit 01216c949d

R	update to allow ordination based on abundance data	19 hours ago
data	second commit with updated names	22 days ago
man	third commit	22 days ago
DESCRIPTION	first commit	22 days ago
NAMESPACE	first commit	22 days ago
README.md	Update README.md	18 hours ago
metacom_1.3.tar.gz	update to allow ordination based on abundance data	19 hours ago

Code
Issues 0
Pull Requests 0
Wiki
Pulse
Graphs
Network
Settings

HTTPS clone URL
https://github.com

You can clone with HTTPS, SSH,

Forking from command-line

```
$ git clone git://github.com/somename/someproject.git someproject
```

#This initializes a new local directory on your machine in a folder called 'someproject'

Some useful commands

Check status :
$ git status

See your remote locations you can pull from (and the master you can push to) :
$ git remote -v

View commit history :
$ git log

Revert to previous version since last commit:
$ git checkout – *filename*

Questions?

Questions?

Let's now look at the user interface of GitHub and play around a bit