Version Control and You

How to git started with git

Ross Markello

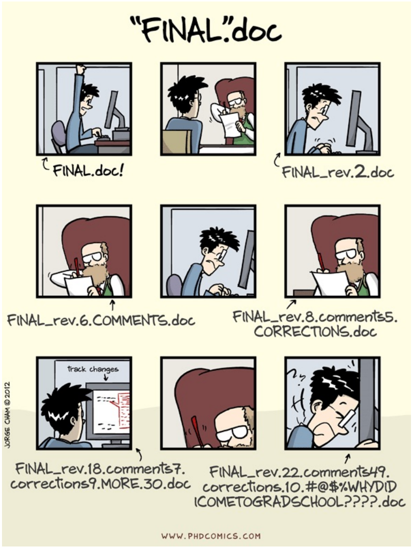
CU Open Science Group
Cornell University

March 29th, 2017

Overview

- What is version control?
    - git: version control for everyone
    - GitHub: remote version control (to stop e-mailing yourself things!)
- What are we doing today?
    - Hands-on workshop: installing git, creating a GitHub account, and gitting going with version control
    - Discussing resources available to learn more

Version Control

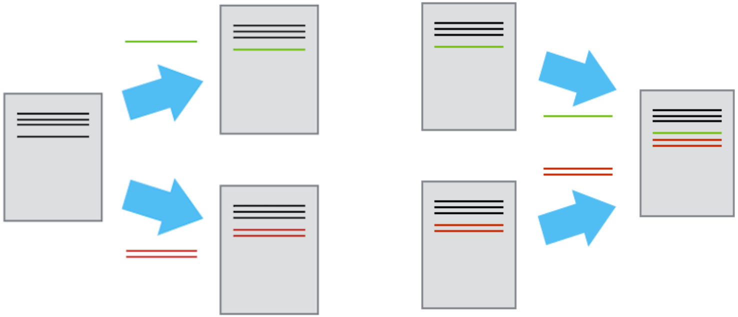


http://www.phdcomics.com/

- Version control is for people who are bad at planning — which, at times, is all of us!
- Creates a chronological log of all your changes
  - Without having multiple versions of the same file
  - And while keeping tracking of who did what
- Allows you to easily roll back to a previous version

Version Control (cont'd)

You can make changes to a document while someone else (say, a collaborator) does, and then seamlessly merge the changes back into one document.



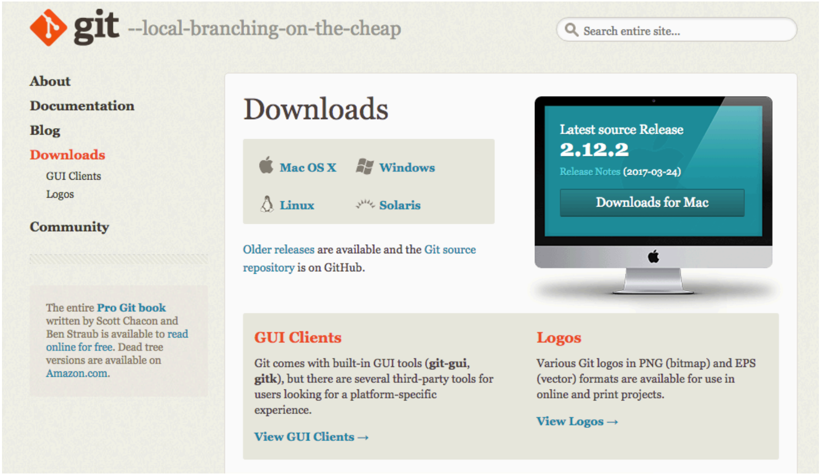
http://swcarpentry.github.io/git-novice/01-basics/

Git: the version control system for everyone

- One of the most popular version control systems to date
- Lightweight and super easy to learn — even, and especially, for those with no coding experience
  - But allows incredible control for those who want to dig deeper!
- An immense user base and thriving community means just about any problem you have is solvable with a quick Google search
  - Literally I google how to do something in git like, every day

Git-ting started! (last time, I swear)

Download git at https://git-scm.com/downloads.


git --local-branching-on-the-cheap

Search entire site...

About
Documentation
Blog
Downloads
GUI Clients
Logos
Community

The entire Pro Git book written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

Downloads

Mac OS X Windows
Linux Solaris

Older releases are available and the Git source repository is on GitHub.

Latest source Release
2.12.2
Release Notes (2017-03-24)
Downloads for Mac

GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

View GUI Clients →

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

View Logos →

The installation

Windows

- Keep "Use Git from the Windows Command Prompt" selected
- Keep "Checkout Windows-style, commit Unix-style line endings" selected
- Keep "Use Windows' default console window" selected

Mac OS X

- Just install it from the dmg as you would any other application — no need to select any special settings

Linux

- Get it from your distro's package manager (i.e., `apt-get` or `yum`)

Setting up git

Let's check and make sure that git was properly installed!

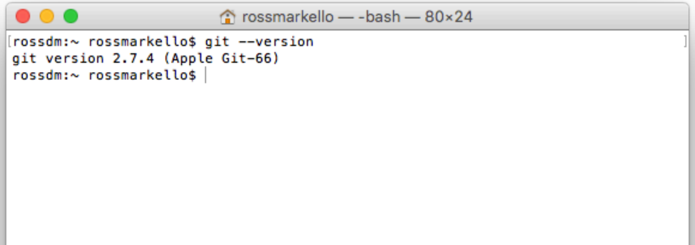
Windows

- Open up the Command Prompt (Click Start, type "cmd", and press `enter`)

Mac OS X/Linux

- Open up the terminal (Press `command-space`, type "Terminal", and press `enter`)

Now type `git --version` into the prompt. You should see something like:



```
rossdm:~ rossmarkello$ git --version
git version 2.7.4 (Apple Git-66)
rossdm:~ rossmarkello$
```

Setting up git, cont'd

In order to effectively use git, you need to tell it who you are! We can do this with the following series of commands:

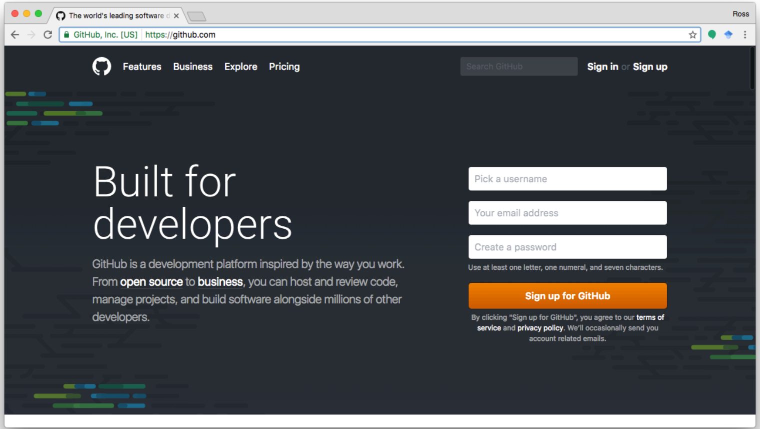
```
git config --global user.name "Ross Markello"
git config --global user.email "rdm222@cornell.edu"
```

Now, whenever we use git in the future, this information will be included in our changes.

GitHub

While version control and git seem great and all, what happens if we need to access things on different computers?

This is where GitHub comes into play!



Built for developers

GitHub is a development platform inspired by the way you work. From **open source** to **business**, you can host and review code, manage projects, and build software alongside millions of other developers.

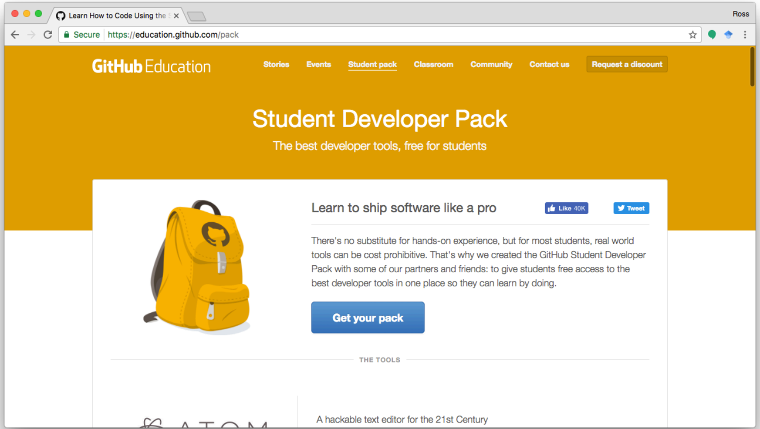
Use at least one letter, one numeral, and seven characters.

By clicking "Sign up for GitHub", you agree to our **terms of service** and **privacy policy**. We'll occasionally send you account related emails.

GitHub (cont'd)

GitHub is a service that integrates with git to allow you to remotely store your version controlled documents.

It's totally free to create an account, and as long as you're a student you can actually get some really awesome perks!



Student Developer Pack

The best developer tools, free for students

Learn to ship software like a pro

There's no substitute for hands-on experience, but for most students, real world tools can be cost prohibitive. That's why we created the GitHub Student Developer Pack with some of our partners and friends: to give students free access to the best developer tools in one place so they can learn by doing.

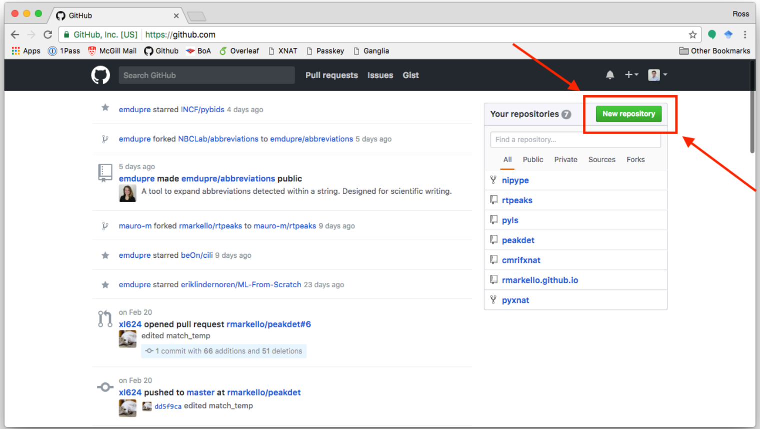
A hackable text editor for the 21st Century

GitHub: The basics

Let's create an account with GitHub at http://github.com.

Just pick a username (something reasonable; mine is `rmarkello`) and sign up with your e-mail (ideally an .edu e-mail!).

Then, we'll get started by creating a new "repository."



emdupre starred INCF/pybids 4 days ago

emdupre forked NBCLab/abbreviations to emdupre/abbreviations 5 days ago

5 days ago
emdupre made emdupre/abbreviations public
A tool to expand abbreviations detected within a string. Designed for scientific writing.

mauro-m forked rmarkello/rtpeaks to mauro-m/rtpeaks 9 days ago

emdupre starred beOn/cili 9 days ago

emdupre starred eriklindernoren/ML-From-Scratch 23 days ago

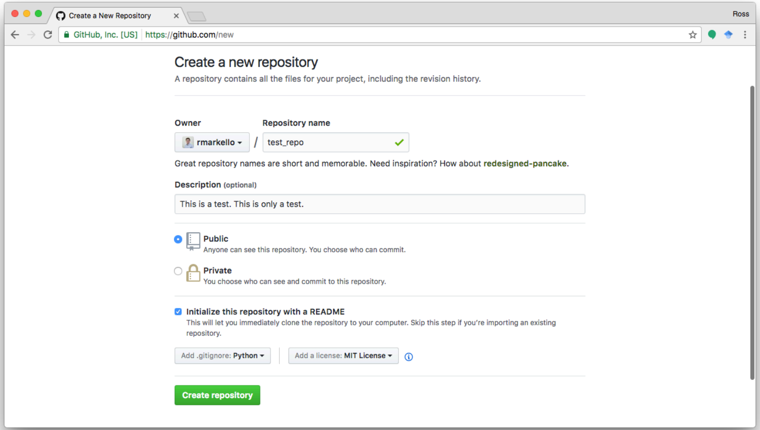
on Feb 20
xl624 opened pull request rmarkello/peakdet#6
edited match_temp
1 commit with 66 additions and 51 deletions

on Feb 20
xl624 pushed to master at rmarkello/peakdet
dd5f9ca edited match_temp

GitHub: Creating a repository

A repository (repo) is the git equivalent of a project.

You have to give it a name, usually a brief description, and then select relevant options (i.e., private or public, etc.).


Create a New Repository

Ross

GitHub, Inc. [US] https://github.com/new

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: rmarkello / Repository name: test_repo

Great repository names are short and memorable. Need inspiration? How about redesigned-pancake.

Description (optional): This is a test. This is only a test.

[✓ Public] Anyone can see this repository. You choose who can commit.

Private — You choose who can see and commit to this repository.

[✓] Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: Python | Add a license: MIT License

Create repository

GitHub: Setting up your repository

Options, options

Once you create the repo, GitHub will helpfully give you some information on how to start setting it up.

In the future, you can select the option that works best for you. For now, however, we don't yet have the software that we're going to use to work with this!

Structured madness

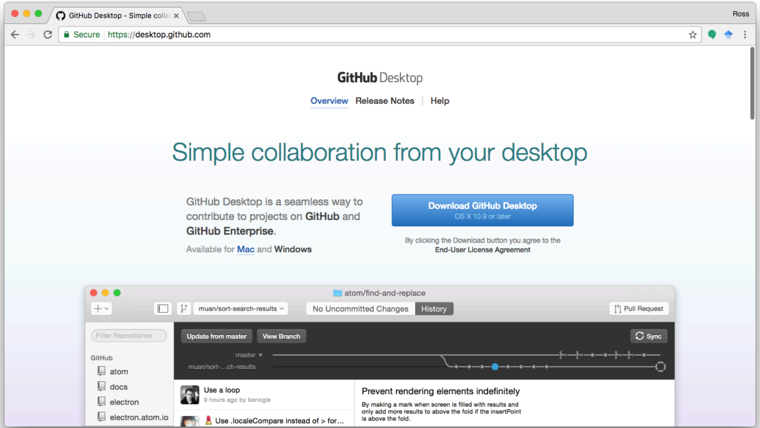
Repos are inherently unstructured; they can contain loose files, directories, and more. Whatever your file organization system is, a repo can handle it.

That being said, in the spirit of open science (i.e., reproducibility), it's probably good to think a bit about the outline for your project structure before getting started!

GitHub Desktop: A better (the best?) way to git

GitHub Desktop provides a (really freakin' nice) user interface to work with git and GitHub.

It handles all the git backend for you, so you never have to use the command line (unless you want to! — we'll get to that).



GitHub Desktop - Simple colla ×

Secure https://desktop.github.com

GitHub Desktop

Overview Release Notes | Help

Simple collaboration from your desktop

GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise.

Available for Mac and Windows

Download GitHub Desktop
OS X 10.9 or later

By clicking the Download button you agree to the End-User License Agreement

atom/find-and-replace

muan/sort-search-results | No Uncommitted Changes | History | Pull Request

Filter Repositories | Update from master | View Branch | Sync

GitHub: atom, docs, electron, electron.atom.io

master
muan/sort-...ch-results

Use a loop
9 hours ago by benogle

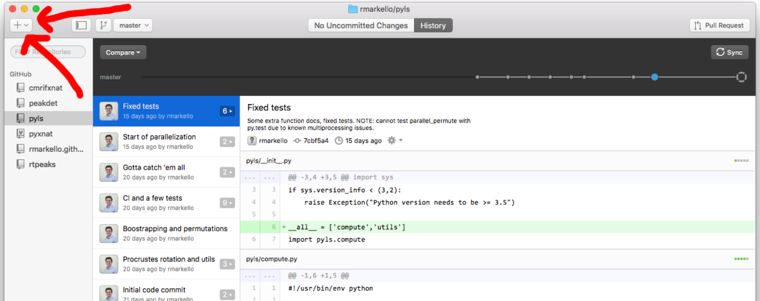
Use .localeCompare instead of > for...

Prevent rendering elements indefinitely

By making a mark when screen is filled with results and only add more results to above the fold if the insertPoint is above the fold.

GitHub Desktop: Cloning your repo

- After downloading and installing GitHub Desktop, open that bad boy/girl/non-gendered-application up!
- Log in with the username and password you set up just a few slides ago, and then we'll be able to get going.
- When you open it up, your GitHub Desktop will be pretty empty. But not for long! We're going to click that little '+' button in the top right corner and toggle over to "Clone."



Fixed tests 15 days ago by rmarkello	6
Start of parallelization 15 days ago by rmarkello	2
Gotta catch 'em all 20 days ago by rmarkello	2
CI and a few tests 20 days ago by rmarkello	9
Boostrapping and permutations 20 days ago by rmarkello	
Procrustes rotation and utils 20 days ago by rmarkello	3
Initial code commit 21 days ago by rmarkello	2

Fixed tests

Some extra function docs, fixed tests. NOTE: cannot test parallel_permute with py.test due to known multiprocessing issues.

rmarkello 7cbf5a4 15 days ago

pyls/__init__.py

```
@@ -3,4 +3,5 @@ import sys
 if sys.version_info < (3,2):
     raise Exception("Python version needs to be >= 3.5")

+__all__ = ['compute','utils']
 import pyls.compute
```

pyls/compute.py

```
@@ -1,6 +1,5 @@
 #!/usr/bin/env python
```

GitHub Desktop: Cloning your repo (cont'd)

- Select the test repo we just made and pick somewhere to save it. A new folder will be created there — this folder will be your repository, and everything related to that project must be stored in that folder.

Tidy git

To simplify things, many people recommend creating a single mega-folder that will house ALL your git repositories.

Each repo will get it's own folder within the mega-folder.

This makes it easy to know where your git repos are to separate them from all your other plain old (boring) folders!

Creating a file

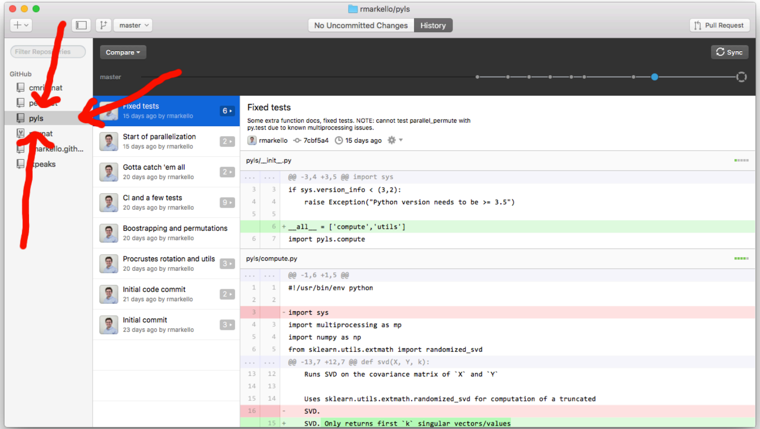
Great! Now, let's git down to business (for REAL the last pun, I swear).

For demonstration purposes, we're just going to create a text file, so crack open your favorite **plain** text editor (i.e., not Microsoft Word!) and get to typing.

Save the newly created document inside your repo directory!

GitHub Desktop: Committing & syncing (1)

Head back to GitHub Desktop and click on the repository name in the left-hand column:



Fixed tests — 15 days ago by rmarkello — 6
Start of parallelization — 15 days ago by rmarkello — 2
Gotta catch 'em all — 20 days ago by rmarkello — 2
CI and a few tests — 20 days ago by rmarkello — 9
Boostrapping and permutations — 20 days ago by rmarkello
Procrustes rotation and utils — 20 days ago by rmarkello — 3
Initial code commit — 21 days ago by rmarkello — 2
Initial commit — 23 days ago by rmarkello — 3

Fixed tests

Some extra function docs, fixed tests. NOTE: cannot test parallel_permute with py.test due to known multiprocessing issues.

rmarkello 7cbf5a4 15 days ago

pyls/__init__.py

```
@@ -3,4 +3,5 @@ import sys
 if sys.version_info < (3,2):
     raise Exception("Python version needs to be >= 3.5")

+__all__ = ['compute','utils']
 import pyls.compute
```

pyls/compute.py

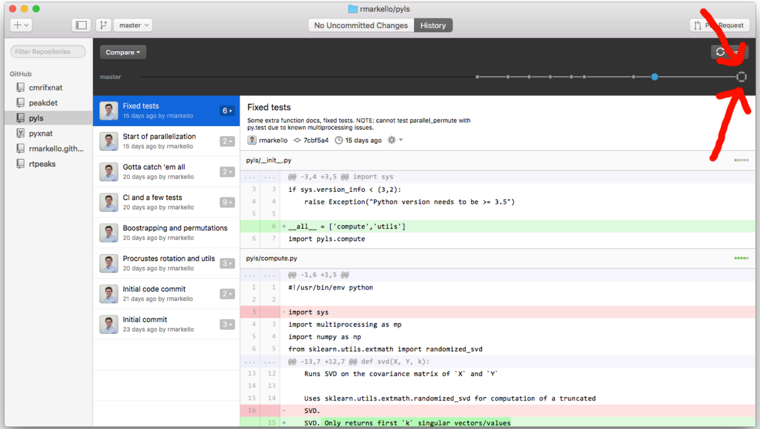
```
@@ -1,6 +1,5 @@
 #!/usr/bin/env python

-import sys
 import multiprocessing as mp
 import numpy as np
 from sklearn.utils.extmath import randomized_svd
@@ -13,7 +12,7 @@ def svd(X, Y, k):
     Runs SVD on the covariance matrix of `X` and `Y`

     Uses sklearn.utils.extmath.randomized_svd for computation of a truncated
-    SVD.
+    SVD. Only returns first `k` singular vectors/values
```

GitHub Desktop: Committing & syncing (2)

Next, click the little wheel over in the top-right corner:



Fixed tests
15 days ago by rmarkello
6

Start of parallelization
15 days ago by rmarkello
2

Gotta catch 'em all
20 days ago by rmarkello
2

CI and a few tests
20 days ago by rmarkello
9

Boostrapping and permutations
20 days ago by rmarkello

Procrustes rotation and utils
20 days ago by rmarkello
3

Initial code commit
21 days ago by rmarkello
2

Initial commit
23 days ago by rmarkello
3

Fixed tests

Some extra function docs, fixed tests. NOTE: cannot test parallel_permute with py.test due to known multiprocessing issues.

rmarkello 7cbf5a4 15 days ago

pyls/__init__.py

```
@@ -3,4 +3,5 @@ import sys
 if sys.version_info < (3,2):
     raise Exception("Python version needs to be >= 3.5")

+__all__ = ['compute','utils']
 import pyls.compute
```

pyls/compute.py

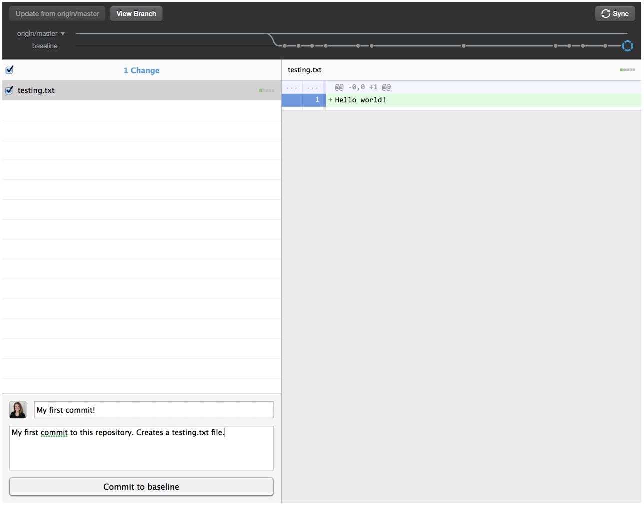
```
@@ -1,6 +1,5 @@
 #!/usr/bin/env python

-import sys
 import multiprocessing as mp
 import numpy as np
 from sklearn.utils.extmath import randomized_svd
@@ -13,7 +12,7 @@ def svd(X, Y, k):
     Runs SVD on the covariance matrix of `X` and `Y`

     Uses sklearn.utils.extmath.randomized_svd for computation of a truncated
-    SVD.
+    SVD. Only returns first `k` singular vectors/values
```

GitHub Desktop: Committing & syncing (3)

You should see a page that looks *something* like this (it won't be exactly the same, but it should be close!):



```
@@ -0,0 +1 @@
1 + Hello world!
```

My first commit!

My first commit to this repository. Creates a testing.txt file.

GitHub Desktop: Committing & syncing (4)

Commits

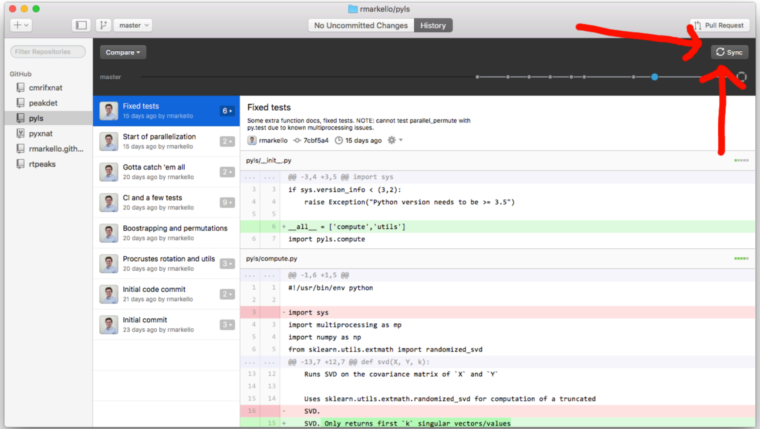
- A "commit" is how git makes a log of the current state of your files
- This state is stored with a "commit message," which should detail what's changed in your files since the last time you made a commit
- These messages should have a brief summary statement and a more detailed, in-depth message
    - There is NO DOWNSIDE to being very descriptive here

Commit files

- Not everything has to be included in a commit! You can select which files are included to create a *logical* history of your changes

GitHub Desktop: Committing & syncing (5)

Once you've made a commit, we need to sync it with the GitHub repository. To do so, simply click "Sync":



Fixed tests — 15 days ago by rmarkello — 6
Start of parallelization — 15 days ago by rmarkello — 2
Gotta catch 'em all — 20 days ago by rmarkello — 2
CI and a few tests — 20 days ago by rmarkello — 9
Boostrapping and permutations — 20 days ago by rmarkello
Procrustes rotation and utils — 20 days ago by rmarkello — 3
Initial code commit — 21 days ago by rmarkello — 2
Initial commit — 23 days ago by rmarkello — 3

Fixed tests

Some extra function docs, fixed tests. NOTE: cannot test parallel_permute with py.test due to known multiprocessing issues.

rmarkello 7cbf5a4 15 days ago

pyls/__init__.py

```
@@ -3,4 +3,5 @@ import sys
 if sys.version_info < (3,2):
     raise Exception("Python version needs to be >= 3.5")

+__all__ = ['compute','utils']
 import pyls.compute
```

pyls/compute.py

```
@@ -1,6 +1,5 @@
 #!/usr/bin/env python

-import sys
 import multiprocessing as mp
 import numpy as np
 from sklearn.utils.extmath import randomized_svd
@@ -13,7 +12,7 @@ def svd(X, Y, k):
     Runs SVD on the covariance matrix of `X` and `Y`

     Uses sklearn.utils.extmath.randomized_svd for computation of a truncated
-    SVD.
+    SVD. Only returns first `k` singular vectors/values
```

GitHub: Checking your remote

Now, we need to check and make sure that worked! Go to your GitHub repository and confirm that your file is there. Your repo will be located at:

http://github.com/your_username/your_reponame.

For example:

http://github.com/rmarkello/pyls

Pro status: command line git

Git is, of course, designed for the command line. Just because there are nice programs that handle a lot of the backend for us doesn't mean we HAVE to use them.

To that end, let's quickly take a look at what git looks like via the command line.

```
[MRI@fh-hd-aka47-l3 test_repo]$ git init
```

Further resources

The git manual — surprisingly accessible:

https://git-scm.com/book

A step-by-step walkthrough on command-line git usage:

https://try.github.io

Guides to understanding git and its integration with GitHub:

https://guides.github.com

Information about getting educational discounts for GitHub:

https://education.github.com

Check the slack channel for more (and join if you haven't!)