Version Control with Git

Kaylea Nelson

research.computing@yale.edu

Yale Center for Research Computing

Version Control with Git

- What is Version Control and Git?
- Putting Your Code into Git
- Connecting Your Repository to Bitbucket
- Utilizing Your Repository's History
- Collaboration: Merging and Conflicts

"FINAL".doc









FINAL_rev.2.doc







FINAL_rev.6.COMMENTS.doc

FINAL_rev.8.comments5. CORRECTIONS.doc



WWW. PHDCOMICS. COM

What is Version Control?

"The whole idea behind any version control system is to store "safe" copies of a project so that you never have to worry about irreparably breaking your code base."

- Bitbucket.org

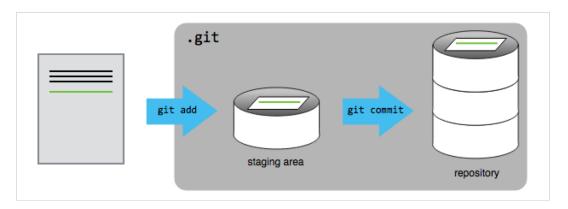
- Easy and powerful way to track changes to your work
- Useful for both writing (if using e.g. LaTeX) and code
- Backups of your work
- General coding safety net

What is Git? How does it work?

Git tracks changes to a file (or set of files) through a series of snapshots called "commits" or "revisions".



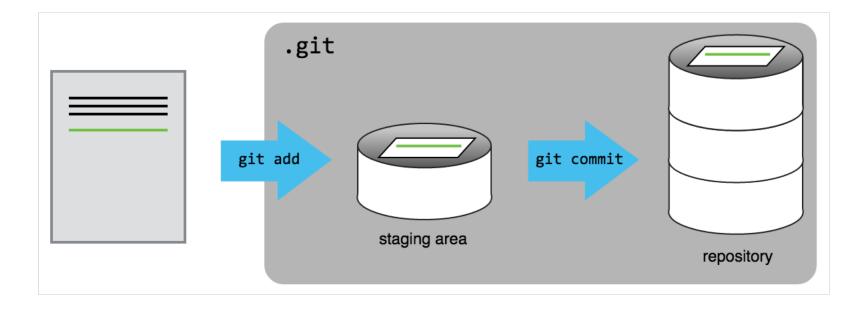
These snapshots are stored in a "repository" which contains a history of all the changes to the files.



How is Git useful to me?

- "Why isn't it working all of a sudden?"
- Cleaner file system (no more "code, codev2, codev3_test, codev3_test1" directories)
- Record of your edits (and thought process!)
- Check for bugs in inconsistent results
- Unlimited and powerful "undo"
- Collaboration!

Putting Your Code into Git



Configure Git

• Global configurations for Git

\$ git config --global user.name "Your Name"
\$ git config --global user.email "your.email@yale.edu"

Setup Repository

• Initialize repository

\$ git init

This create a .git directory in your directory that contains all the version control information. DO NOT DELETE!!!

\$ ls -a			
•	• •	.git	

Add Existing Files to Repository

• The Git repository can be initialized before or after you create any files. To version control existing files, just add them to the repository.

\$ git add myplot.py

Check Status of Repository

```
$ git status
On branch master
Initial commit
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
   new file: myplot.py
Untracked files:
  (use "git add <file>..." to include in what will
be committed)
   myfigure.png
```

Make Initial Commit

 Now we want to permanently save the changes to repository, an action called "committing"

\$ git commit -m "initial commit"

Leaving Files Out of Repository

- You don't want to add every file to your repository. The good rule of thumb is to exclude files if they are a product of your code. Examples of files to exclude:
 - Image files
 - PDFs
 - Compiled code (including .o or .pyc files)
 - System files (e.g., .DS_Store)

Automate Exclusions

• To easily automate exclusions, create a .gitignore file.

```
$ cat .gitignaore
.DS_Store
*.png
*.pyc
$ git add .gitignore
$ git commit -m "added .gitignore"
```

 Now these files won't show up as "untracked" in the git status command and can't accidentally get added to the repository

Make Changes!

 Make changes to the file and then check on the repository

```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be
committed)
  (use "git checkout -- <file>..." to discard
changes in working directory)
  modified: myplot.py
no changes added to commit (use "git add" and/or
"git commit -a")
```

Make Changes!

• Add and commit your changes

\$ git add myplot.py \$ git commit -m "increased frequency" [master 21e2dd2] increased frequency 1 file changed, 1 insertion(+), 1 deletion(-)

Review Changes

 You can check to see what has been modified before adding files using git diff

```
$ git diff
diff --git a/myplot.py b/myplot.py
index 3c179cc..3eb9a45 100644
--- a/myplot.py
+++ b/myplot.py
@@ -2,7 +2,7 @@ import matplotlib.pyplot as plt
 import numpy as np
 t = np.arange(0.0, 2.0, 0.01)
-s = np.sin(2*np.pi*t)
+s = np.sin(4*np.pi*t)
 plt.plot(t, s)
 plt.xlabel('time (s)')
```

Review Changes

 You can check to see what has been modified before committing using git diff --staged

```
$ git diff --staged
diff --git a/myplot.py b/myplot.py
index 3c179cc..3eb9a45 100644
--- a/myplot.py
+++ b/myplot.py
@@ -2,7 +2,7 @@ import matplotlib.pyplot as plt
 import numpy as np
 t = np.arange(0.0, 2.0, 0.01)
-s = np.sin(2*np.pi*t)
+s = np.sin(4*np.pi*t)
 plt.plot(t, s)
 plt.xlabel('time (s)')
```

Writing a Good Commit Message

- The commit message should be a high level explanation of the change
 - Don't be too brief
 - Also, don't exactly quote the change
- Example:
 - Bad: "Changes"
 - Bad: "Changed line 178 in plot_bM_vs_t.py"
 - Better: "Change color of pressure line to red"
- Most important question: If you are looking at this message in 6 months, is it going to make sense and be useful?

Other Useful Commands

• Rename or move a file in the repository

\$ git mv <old_filename> <new_filename>

• Delete a file from the repository

\$ git rm <filename>

Connecting Your Repository to Github

- 1. Create repository on your online account
- Follow included instructions to get your local repository connected to your remote repository
- 3. Push committed changes to the remote repository

```
…
$ git commit -m "<message>"
$ git push
```

Create a New Repository on Github

Search or jump to	Pull requests	s Issues Marketplace Explore		<u>++</u> * 🔡*
	ycrc / ycrc.github.io Code Issues	Ill requests 0 🔲 Projects 0 💷 Wiki	Output 1 ★ Star 1 % Fork 0 Image: Insights Image: Settings	New repository Import repository New gist New organization
	documentation repo for the YCRC Manage topics	This repository New issue		
	T 73 commits	۶ 2 branches الم	4 1 environment 2 contributors	
	Branch: master • New pull request		Create new file Upload files Find File Clone or download -	
	brevans Deployed cc9d6a5 with Mk	Docs version: 1.0.4	Latest commit 3aebab2 4 days ago	
	assets	Deployed e6b64ca with MkDocs version: 1.0.4	2 months ago	
	clusters-at-yale	Deployed cc9d6a5 with MkDocs version: 1.0.4	4 days ago	
	🖬 data	Deployed 506039e with MkDocs version: 1.0.4	2 months ago	
	🖬 files	Deployed e6b64ca with MkDocs version: 1.0.4	2 months ago	
	img	Deployed e6b64ca with MkDocs version: 1.0.4	2 months ago	
	is js	Deployed 13e9b34 with MkDocs version: 1.0.4	2 months ago	
	ational-hpcs	Deployed 506039e with MkDocs version: 1.0.4	2 months ago	
	online-tutorials	Deployed 506039e with MkDocs version: 1.0.4	2 months ago	
	search	Deployed cc9d6a5 with MkDocs version: 1.0.4	4 days ago	
	stylesheets	Deployed 8f7d0f1 with MkDocs version: 1.0.4	2 months ago	

Create a New Repository on Github

Search or jump to	7 Pull requests Issues Marketplace Explore	♠ +• ‱ •
	Create a new repository A repository contains all project files, including the revision history.	
	Owner Repository name *	
	 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: None Add a license: None	

Follow the Instruction to Push

Search or jump to	Pull requests Issues Marketplace Explore	♠ +• <u>₩</u> •
	A kayleanelson / my_latest_work Private ● Unwatch → 1 ★ Star 0 Work Ø Unwatch → 1	
	Code ① Issues 0 ① Pull requests 0 巴 Projects 0 E Wiki 山 Insights ② Settings	
	Quick setup — if you've done this kind of thing before Image: Set up in Desktop or HTTPS SSH git@github.com:kayleanelson/my_latest_work.git Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.	
	<pre>or create a new repository on the command line echo "# my_latest_work" >> README.md</pre>	
	<pre>git init git add README.md git commit -m "first commit" git remote add origin git@github.com:kayleanelson/my_latest_work.git git push -u origin master</pre>	
	or push an existing repository from the command line	
	git remote add origin git@github.com:kayleanelson/my_latest_work.git git push -u origin master	
	or import code from another repository You can initialize this repository with code from a Subversion, Mercurial, or TFS project.	

Push Future Commits

 After the initial "push" to the remote repository, just remember to push any new commits and you will have easy remote backups of your work!

… \$ git commit -m "<message>" \$ git push

Remote Repository Hosts Options

- github.com unlimited free public and private repos for everyone
- Bitbucket.org unlimited repos [but be careful signing up with yale.edu email]
- git.yale.edu free fully featured accounts.
 Only available on Yale network or VPN and with Yale netid

Utilizing Your Repository's History

Search or jump to	Pull requests Issues Marketplace Explore	♠ + - ₩-
	<>Code ① Issues 0 ⑦ Pull requests 0 Ⅲ Projects 0 Ⅲ Wiki 山 Insights ♦ Settings	
	Branch: src -	
	Commits on Apr 4, 2019	
	touch-up on permissions pagecc9d6a5brevans committed 4 days ago<>	
	- Commits on Apr 2, 2019	
	removed binding recommendation, added tip for testing gpu codes Image: bsee3c5 Image: bsee3c5 Image: bsee3c5 Image: brevans committed 6 days ago Image: bsee3c5 Image: bsee3c5 Image: bsee3c5 Image: bsee3c5	
	fixed broken link Image: Big State S	
	- Commits on Mar 22, 2019	
	fix milgram dsq module name Image: bit state in the s	
	Commits on Mar 21, 2019	
	increase mpi partition node limit kayleanelson committed 18 days ago	
	- Commits on Mar 19, 2019	

Review History

You can see a history of all recent commits
 Detailed Log:

\$ git log
\$ git log -1

– Simplified Log

\$ git log --oneline
\$ git log --oneline --graph --decorate

Compare Revisions

 You can compare two revisions to see what changes were made with git diff

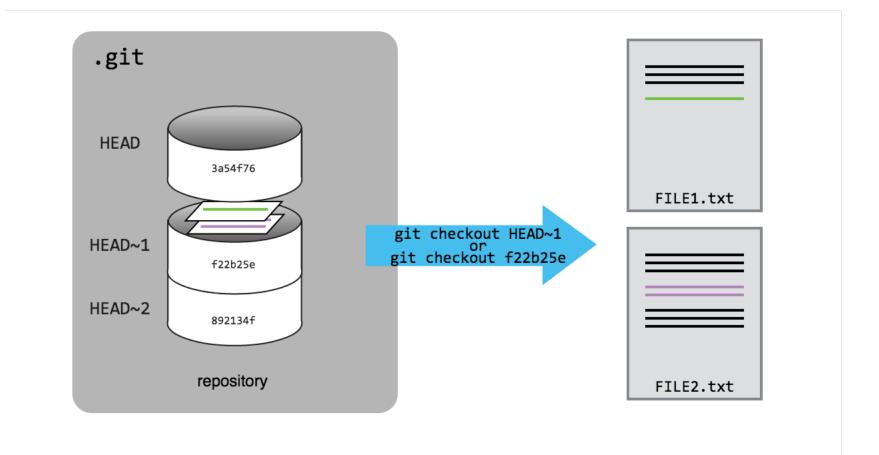
```
$ git diff HEAD 0c7aa71
diff --git a/myplot.py b/myplot.py
index 3c179cc..3eb9a45 100644
--- a/myplot.py
+++ b/myplot.py
@@ -2,7 +2,7 @@ import matplotlib.pyplot as plt
 import numpy as np
 t = np.arange(0.0, 2.0, 0.01)
-s = np.sin(2*np.pi*t)
+s = np.sin(4*np.pi*t)
 plt.plot(t, s)
 plt.xlabel('time (s)')
```

Review History

 The interfaces on Github and Bitbucket are also great for exploring the commit history and tracking changes

•••	M Inbox - kayleanelson@gmail.cc × 🧧 kayleanelson / my.latest.work ×	Kaylea	••	Minbox - kayleanelson@gmail.c: x 😨 kayleanelson / my_latest_work: x	Kaylea
\leftrightarrow \rightarrow	C 🛆 Atlassian, Inc. [US] https://bitbucket.org/kayleanelson/my_latest_work/commits/21e2dd251adadd86	10b475cc06daf 🛠 👰 🕭 🤫 🛐 🗄		C A Atlassian, Inc. [US] https://bitbucket.org/kayleanelson/my_latest_work/commits/all	🖈 👰 🕭 🦷 🔝 🗄
M gmail	Y osTicket 😰 cal G feedly interwebs iffe icrc ig+g icoding inew iwrf ito read	Cther Bookmarks		all YosTicket go cal G feedly ≧ interwebs ≧ life ≧ crc ≧ g+g ≧ coding ≧ new ≧ wrf ≧ to read	Conter Bookmarks
0	Kaylea Nelson / my_latest_work		=	Bitbucket Teams · Projects · Repositories · Snippets ·	Find a repository Q 🕐 🕌
	Commits			Kaylea Nelson / my_latest_work	
ш	Kaylea Nelson committed 21e2dd2 6 hours ago	Approve		Commits	
	increased frequency	24 🔛	ليب	↓ All branches -	Q. Find commits
\$		\$ 8c7aa71		Author Commit Message	Date Builds
\mathcal{V}		₽ master	¢	Kaylea Nelson 21e2dd2 increased frequency	8 hours ago
đ		View raw commit	V	 Waylea Nelson ec7aa71 added gitignore 	8 hours ago
0		Stop watching	đ		
4			0		
· ·	Comments (0)		4		
¢	What do you want to say?	4			
			\$	Blog · Support · Plans & pricing · Documentation · API · Site status · Version info · Ter	Prince and an
	Files changed (1)			JIRA Software · Confluence · Bamboo · SourceTree · HipChat	
	+1 -1 M mypoupy				
	B myplot.py MODIFIED Side-by-si	de diff View file Comment ***		XAtlassian	
	2 2 import numpy as np 3 3				
	4 4 t = np.arange(0.0, 2.0, 0.01) 5 -s = np.sin(4*np.pi*t)				
»	5 +s = np.sin(2*np.pi*t) 6 6		>>		

Checkout Previous Commits



Checkout Previous Commits

• After, you have identified the revision you need to revert to, "checkout" that revision

\$ git checkout <revision>

• Or just a specific file from that revision

\$ git checkout <revision> <filename>

Warning: If you checkout from an old revision, any uncommitted changes to the project will be lost.

Throwaway All New Changes

Revert your working directory to the last commit

\$ git reset --hard

Warning: Any uncommitted changes to the project will be lost.

Getting Your Code in a New Location

 If you have a remote repository, you can "clone" it to a new location to continue your work (e.g., copying code to the cluster, recovering your code to a new laptop)

\$ git clone
https://kayleanelson@github.com/kayleanelson/my_latest
_work.git

Collaboration

- Once your work is in a remote repository, it is very easy to being to collaborate with others
 - Git has a sophisticated system for managing multiple people editing the same code base through "merging"
- Usage Examples
 - Multiple collaborators on a code
 - LaTeX papers!

Sharing Your Repository

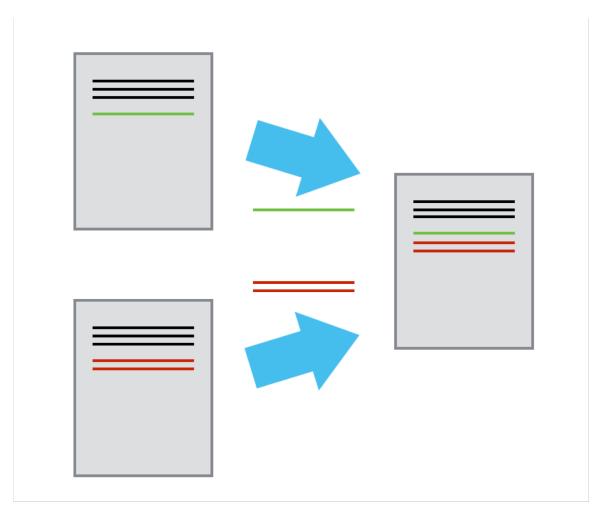
Search or jump to	Pull requests	Issues Marketplace Explore	ب +۰ ₩۰
é	A kayleanelson / my_latest_wo ⇔ Code ① Issues ◎ ハ Pul	rk Private I requests 0 III Projects 0 III Wiki III Insights ♀ Settings	
	Options Collaborators	Collaborators Push access to the repository	
-	Webhooks Notifications	This repository doesn't have any collaborators yet. Use the form below to add a collaborator.	
	Integrations & services	Search by username, full name or email address You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.	
	Deploy keys	Add collaborator	
	2019 GitHub, Inc. Terms Privacy Se	curity Status Help Contact GitHub Pricing API Training Blog About	
٣		curity Status Help Contact GitHub Pricing API Training Blog About	
https://github.com/kayleanelson/my_lates	st_work/wiki		

Basic Collaborative Workflow

- Pull down new commits
- Make your edits
- Add your modified files and commit
- Push commits to remote

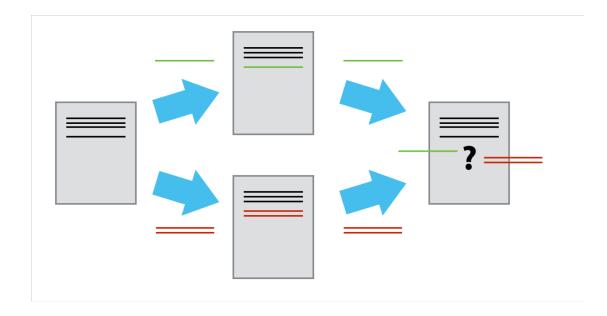
```
$ git pull
...
$ git add <files>
$ git commit -m <message>
$ git push
```

Merging



Conflicts

 Inevitably, you and your collaborator will commit overlapping changes to a file. This will create a "merge conflict".



Resolving Conflicts

• Pull in commits and Oops!

\$ git pull Auto-merging myplot.py CONFLICT (content): Merge conflict in myplot.py Automatic merge failed; fix conflicts and then commit the result.

Resolving Conflicts

• Git marks the conflicted line in the file

```
$ cat myplot.py
import matplotlib.pyplot as plt
import numpy as np
t = np.arange(0.0, 2.0, 0.01)
<<<<<< HEAD
s = np.sin(3*np.pi*t)
======
s = np.sin(4*np.pi*t)
>>>>> 7232b521f34cf3deed50f4d8aac6260616683ddf
```

• Manually merge the code in a text editor and commit the changes

Uncommitted Conflicts

- Git will also complain if you pull in changes to a file you have modified but not committed.
 You have two options.
 - Undo the changes to the file back to last committed revision by checking it out from the HEAD

\$ git checkout -- <filename>

 Commit your changes and then redo the pull (and potentially merge the changes, if applicable)

Questions?

To summarize, add 3 commands to your daily workflow for unlimited undo and online backups of your code!

\$ git add <files>
\$ git commit -m <message>
\$ git push



Even more information:

- Great in depth tutorials on all things git:
 - try.github.io
 - https://play.instruqt.com/public/topics/gettingstarted-with-git
- Software Carpentry (thanks for the images!)
 - <u>https://swcarpentry.github.io/git-novice/01-basics/</u>
- Contact us: research.computing@yale.edu