UCSF
University of California
San Francisco

GIT Tutorial

Version control and shared development

Peder Larson, PhD
Associate Professor
UCSF Department of Radiology and Biomedical Imaging

April 30, 2018

What is git?

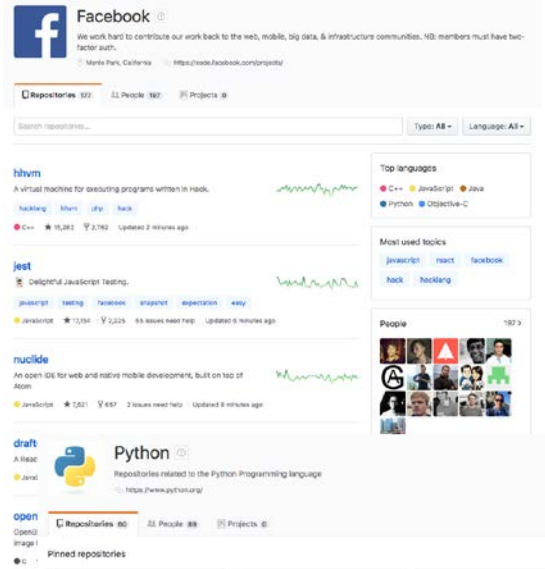
- Version control
  - Track changes you make to software or other documents
  - Go back to old versions, or look at changes
- Shared development platform
  - Designed to support projects with multiple developers
  - Create branches, merge and track individual user changes, report and assign issues
- Distributed
  - Every version of the repository, whether local or hosted (e.g. github, Radiology gitlab) is a full repository, so access to host required
  - Transition to and from local copy to hosted repository, between hosted sites, and between local copies

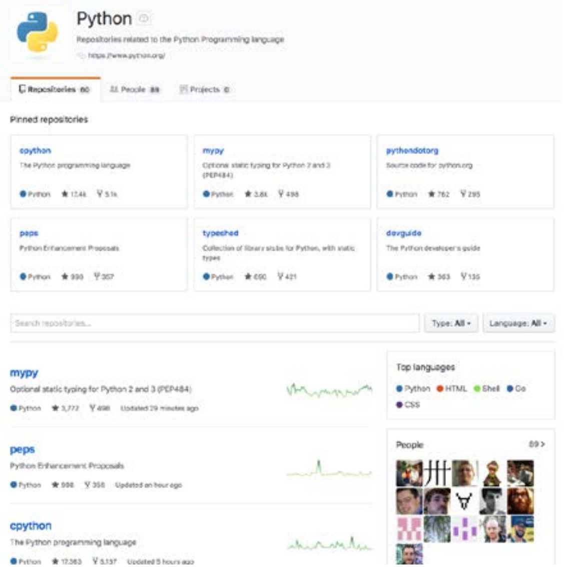
Why Git?

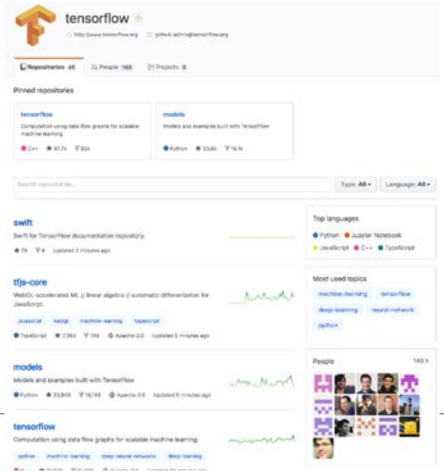
- Sharing and jointly developing code
- **Standard tool** for version control & software development, easy to work with others
- Open source and free
- Distributed – track changes on your local copies
- Jobs
  - Employers may consider your github/gitlab/bitbucket profile as part of your CV for tech jobs
  - Ask any graduate who works on software in industry – they must use version control
- Many Tools
  - e.g. github desktop client, probably many others
  - Web interfaces

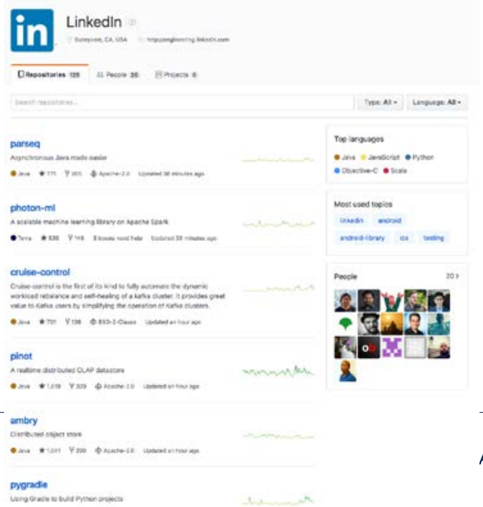
Prominent Git Software Groups

- TensorFlow (Google) https://github.com/tensorflow
- Python https://github.com/python
- Facebook https://github.com/facebook
- LinkedIn https://github.com/linkedin


Facebook
We work hard to contribute our work back to the web, mobile, big data, & infrastructure communities. NB: members must have two-factor auth.
Menlo Park, California https://code.facebook.com/projects/
Repositories 172 People 187 Projects 0
Type: All Language: All
hhvm
A virtual machine for executing programs written in Hack.
hacklang hhvm php hack
C++ ★ 15,283 ⑂ 2,762 Updated 2 minutes ago
jest
Delightful JavaScript Testing.
javascript testing facebook snapshot expectation easy
JavaScript ★ 17,154 ⑂ 2,225 55 issues need help Updated 6 minutes ago
nuclide
An open IDE for web and native mobile development, built on top of Atom
JavaScript ★ 7,821 ⑂ 657 2 issues need help Updated 8 minutes ago
Top languages
C++ JavaScript Java Python Objective-C
Most used topics
javascript react facebook hack hacklang
People 187


Python
Repositories related to the Python Programming language
https://www.python.org/
Repositories 60 People 89 Projects 0
Pinned repositories
cpython
The Python programming language
Python ★ 17.4k ⑂ 5.1k
mypy
Optional static typing for Python 2 and 3 (PEP484)
Python ★ 3.8k ⑂ 498
pythondotorg
Source code for python.org
Python ★ 762 ⑂ 295
peps
Python Enhancement Proposals
Python ★ 998 ⑂ 357
typeshed
Collection of library stubs for Python, with static types
Python ★ 690 ⑂ 421
devguide
The Python developer's guide
Python ★ 363 ⑂ 135
Type: All Language: All
mypy
Optional static typing for Python 2 and 3 (PEP484)
Python ★ 3,772 ⑂ 498 Updated 29 minutes ago
peps
Python Enhancement Proposals
Python ★ 998 ⑂ 356 Updated an hour ago
cpython
The Python programming language
Python ★ 17,363 ⑂ 5,137 Updated 5 hours ago
Top languages
Python HTML Shell Go CSS
People 89


tensorflow
http://www.tensorflow.org github-admin@tensorflow.org
Repositories 48 People 148 Projects 0
Pinned repositories
tensorflow
Computation using data flow graphs for scalable machine learning
C++ ★ 97.7k ⑂ 62k
models
Models and examples built with TensorFlow
Python ★ 33.8k ⑂ 19.1k
Type: All Language: All
swift
Swift for TensorFlow documentation repository.
★ 79 ⑂ 4 Updated 3 minutes ago
tfjs-core
WebGL-accelerated ML // linear algebra // automatic differentiation for JavaScript.
javascript webgl machine-learning typescript
TypeScript ★ 7,363 ⑂ 746 Apache-2.0 Updated 5 minutes ago
models
Models and examples built with TensorFlow
Python ★ 33,849 ⑂ 19,149 Apache-2.0 Updated 8 minutes ago
tensorflow
Computation using data flow graphs for scalable machine learning
python machine-learning deep-neural-networks deep-learning
Top languages
Python Jupyter Notebook JavaScript C++ TypeScript
Most used topics
machine-learning tensorflow deep-learning neural-network python
People 148


LinkedIn
Sunnyvale, CA, USA http://engineering.linkedin.com
Repositories 128 People 20 Projects 0
Type: All Language: All
parseq
Asynchronous Java made easier
Java ★ 771 ⑂ 203 Apache-2.0 Updated 38 minutes ago
photon-ml
A scalable machine learning library on Apache Spark
Terra ★ 535 ⑂ 148 3 issues need help Updated 38 minutes ago
cruise-control
Cruise-control is the first of its kind to fully automate the dynamic workload rebalance and self-healing of a kafka cluster. It provides great value to Kafka users by simplifying the operation of Kafka clusters.
Java ★ 721 ⑂ 138 BSD-2-Clause Updated an hour ago
pinot
A realtime distributed OLAP datastore
Java ★ 1,319 ⑂ 309 Apache-2.0 Updated an hour ago
ambry
Distributed object store
Java ★ 1,041 ⑂ 200 Apache-2.0 Updated an hour ago
pygradle
Using Gradle to build Python projects
Top languages
Java JavaScript Python Objective-C Scala
Most used topics
linkedin android android-library ios testing
People 20

Selected Imaging Repositories/Groups

- SIVIC - https://github.com/SIVICLab/sivic
- ANTS – Advanced Normalization Tools https://github.com/ANTsX/ANTs
- AFNI – **A**nalysis of **F**unctional **N**euro**I**mages https://github.com/afni/afni
- ISMRMRD – ISMRM raw data format
- BART – Berkeley advanced reconstruction toolbox https://github.com/mrirecon/bart
- Many more!

Git Servers and Structure

- https://git.radiology.ucsf.edu/
  - Radiology git server
  - Behind firewall, only accessible by Radiology users, no public repos
  - Sensitive projects (data, IP)
  - Easily explored by others in Radiology
- github.com
  - Most widely used service
  - Public and private repositories ("Repos")
  - Request academic account for unlimited free private repos for 2 years – education.github.com
- Individual accounts
- Groups – for groups with multiple shared projects, e.g. lab/research group, specific project/grant, organization

My Personal Git Ecosystem

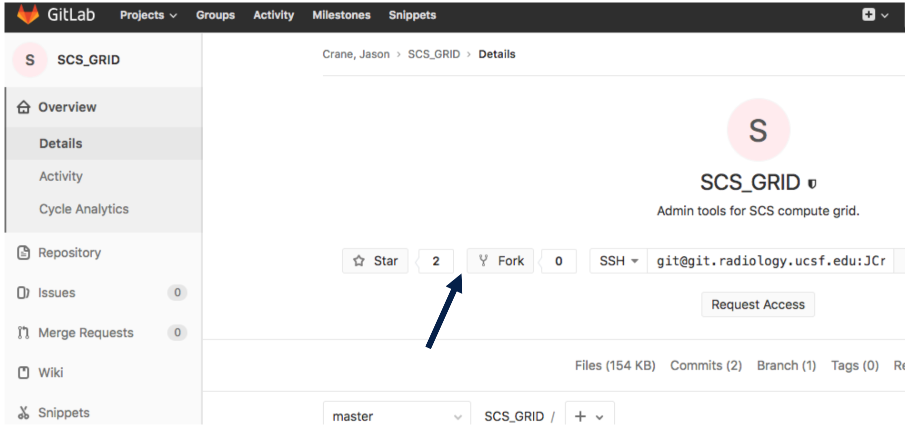
- Radiology git
  - plarson - personal account, for my own projects or initial development
    - matlab
    - EPSI processing
  - EPIC-MRI – group account for GE MRI EPIC programming projects (so far I'm the only user ☺)
    - 3dradial
    - prose_prostate
    - fidcsi_c13
    - 3dute
    - cones

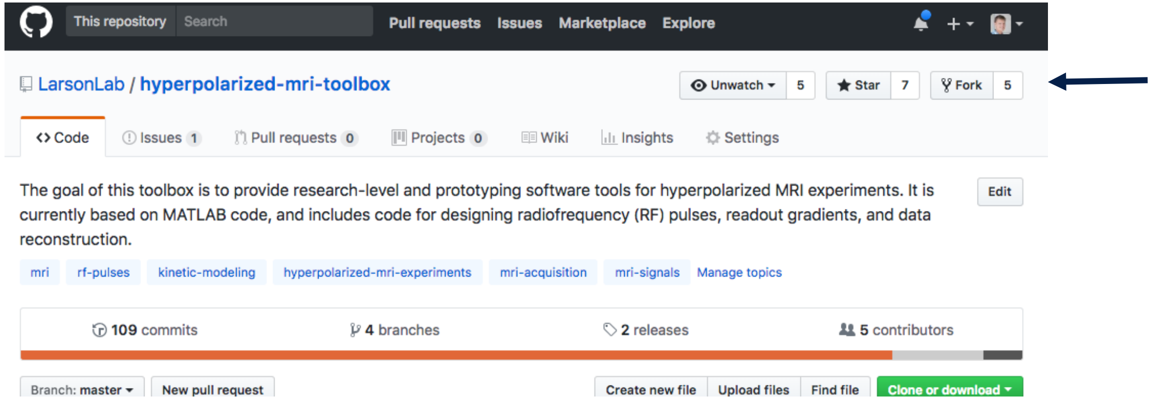
Incredibly valuable for these projects!

- GitHub
  - agentmess – personal account, personal projects, papers, playing around with code
  - LarsonLab – group account for shared projects and sustained projects
    - hyperpolarized-mri-toolbox
    - mripy (Python tools for MRI, including neural networks, originally from Peng Cao)
    - Spectral-Spatial-RF-Pulse-Design
    - MRI-education-resources
  - UCSF-EPIC-MRI
    - For sharing EPIC software with others
    - All private repositories (GE proprietary information)

GitHub/GitLab features

- Star – any interesting code
- Watch – be notified of repo changes
- Fork – make your own copy to use and modify





The goal of this toolbox is to provide research-level and prototyping software tools for hyperpolarized MRI experiments. It is currently based on MATLAB code, and includes code for designing radiofrequency (RF) pulses, readout gradients, and data reconstruction.

Remember to …

- Commit every day!
- Practice, and don’t worry about mistakes since there’s a history of all your changes

Initialization

1. Login – git.radiology.ucsf.edu or github.com
2. Create New Repository/Project
3. Clone ‘git clone <address>’
   - Copy address from web
   - Clone to multiple places (laptop, SCS network)
4. Add files or import in existing directory
5. Check file status ‘git status’, should show Untracked Files
6. Add these files to git repository with ‘git add’
7. Check file status ‘git status’, should show Changes to Be Committed
8. Commit ’git commit –m “<commit message>” ‘
9. Push changes to remote repository (e.g. github, radiology git) ‘git push’
10. Check web!

Daily Workflow

1. Pull changes from remote repo (in case others have added edits) ‘git pull’
   - If you are just a user (not developer) of repo/project, then this is all you need
2. Modify files
3. Check status ‘git status’
4. Add changes ‘git add’
5. Confirm status ‘git status’
6. Commit ‘git commit’
7. Push changes ‘git push’ (don’t need to do for every commit, but at least every day is best)

- Quick commit – ‘git commit –a –m <message>” stages all changes to be commited and then commits

Advanced Workflows - ignoring files

- .gitignore – this is a file within your git repository that can choose to ignore certain files. For example, large data files, temporary files, executables.
    - .gitignore templates at https://github.com/github/gitignore
    - Copy into main directory
    - In GitHub Desktop app, right-click to add files to .gitignore
- Want to store large files? Use “git-lfs” (large file storage) git-lfs.github.com

Advanced Workflows - Branching

- “Branch”
  - separate version of repository to work on
  - Create a branch to fix a bug, add a new feature, or play around without disrupting the “master” branch (default branch when you start a project
  - Easy to explore and visualize via web interfaces
  - First branch created is the “master”

Advanced Workflows – Creating and Editing New Branch

Via Web interface

Then switch to branch in repository

1. List all branches ‘git branch –a’
2. Checkout new branch ‘git checkout <branchname>’
3. Confirm you are now working on new branch ‘git branch’

Or Command Line

1. Check current branch ‘git branch -a’ (lists branches, * indicated curent branch_
2. Switch to starting branch if needed, e.g. ‘git checkout master’ to create branch from master
3. Create new branch ‘git branch <branchname>’
4. Checkout new branch ‘git checkout <branchname>’
5. Confirm you are now working on new branch ‘git branch’

Advanced Workflows – Creating and Editing New Branch

- Normally, a repository is a single directory and you can switch between branches (‘git branch’ to view branches, ‘git checkout <branchname>’ to switch)
- This can cause problems if you (like me) forget to check what branch you are working on
- Alternative
  - Keep one repository as a ‘master’
    - git clone git@git.radiology.ucsf.edu:PLarson/git-tutorial-test.git git-tutorial-test_master
  - Clone another version of repo for development(more like SVN)
    - git clone git@git.radiology.ucsf.edu:PLarson/git-tutorial-test.git git-tutorial-test_branch
    - cd git-tutorial-test_branch
    - git checkout <branchname>

Advanced Workflows – Merging Branches

- Create a Merge/pull request
  - When you want to put branches together, merge the changes
  - E.g. you have added feature in a feature branch, merge back into the master branch
  - I find this easiest via web interfaces
- When you push changes ‘git push’, message:
  - remote: To create a merge request for branch2, visit:
  - remote: https://git.radiology.ucsf.edu/PLarson/git-tutorial-test/merge_requests/new?merge_request%5Bsource_branch%5D=branch2
- Review and submit merge request
- Confirm request and submit (last steps are so you can review the changes carefully before continuing)

Fixing conflicts

Can arise when remote repo is out of sync with local copy, or during merging of branches

1. Find conflicting files ‘git status’
2. Edit with your favorite editor
    - Conflicting lines marked with <<<<, ====, >>>>
    - Choose appropriate changes, remove lines with <<<<, ====, >>>>
3. Mark resolution with ‘git add’
4. Commit
5. Push

Advanced Workflows - Other

- Multiple Remote Repos (e.g. github vs radiology git)
  - Can sync local copy with both
  - Move repository to other location

1. Create empty repository
2. Add as a remote ‘git remote add github https://github.com/agentmess/git-tutorial-test.git’ (can change “github” to be description of another remote repository, use ”origin” if you want to make this the new default repository)
3. Push to new remote ‘git push --all github’

Advanced Workflows - Other

- Tags/releases
  - When you’ve got a stable product
  - Allows others to easily find stable version or version that will work for them
- Issues
  - Keep track of bugs to fix or features to add

Other features for citing and sharing (GitHub based)

- Zenodo for citing code
  - Get DOI for citing your code!
  - https://zenodo.org/account/settings/github/
- Webpages – e.g. Radiology retreat, BART https://mrirecon.github.io/bart/
  - https://pages.github.com/
  - Setup at username.github.io
- MATLAB File Exchange – automatically post your MATLAB code from github here

More resources

- Google your error message
- https://intrarad.ucsf.edu/twiki/bin/view/Sysadmin/GitLab
- https://git.radiology.ucsf.edu/EPIC-MRI/BestPractices
- Getting Started guides on github and gitlab

Remember to …

- Commit every day!
- Practice, and don’t worry about mistakes since there’s a history of all your changes

- git on it
- git your roll on
- git ‘er done
- everybody git together