Enough to Git Started

2019 MedRec Hackathon at UConn Health

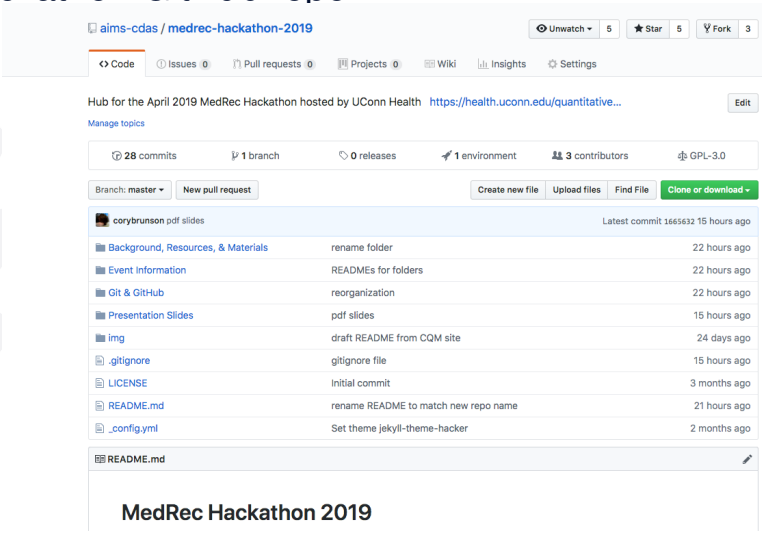
Jason Cory Brunson

Center for Quantitative Medicine, UConn Health

2019 April 13

UCONN HEALTH

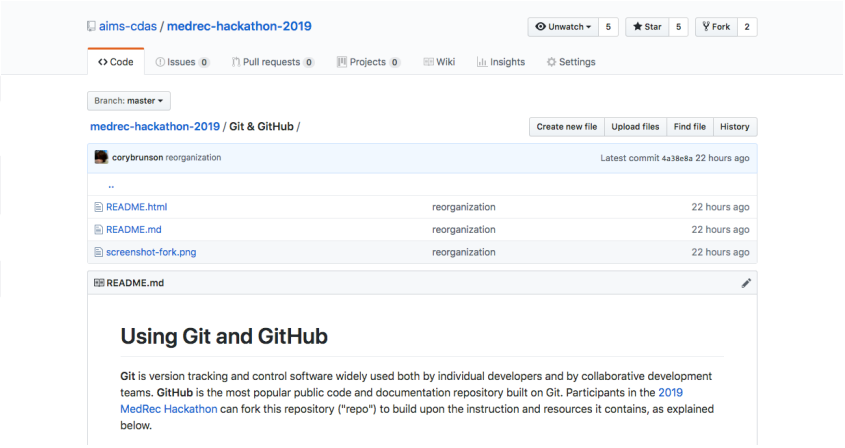
the MedRec Hackathon GitHub repo



https://github.com/aims-cdas/medrec-hackathon-2019

UCONN HEALTH

Introduction to Git



UCONN HEALTH

Installation

Git introduction:

https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control

Git installation (Mac, Windows, source):

https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

Documentation (and help)

1. Git documentation:

   ```
   git <command> --help
   ```

2. Git Book:

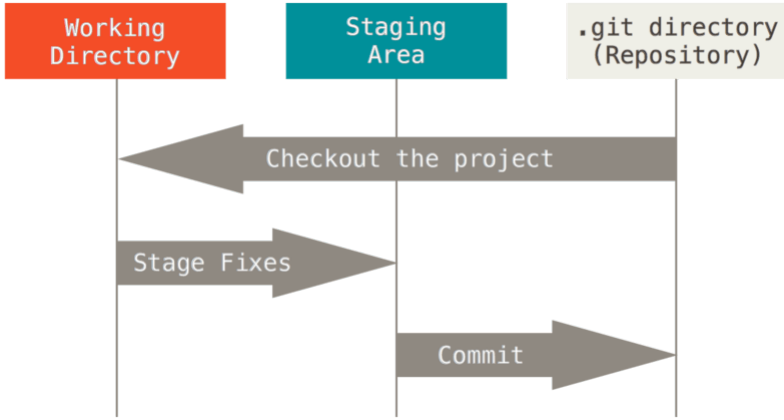
https://git-scm.com/book/en/v2/

3. StackOverflow:

https://stackoverflow.com/

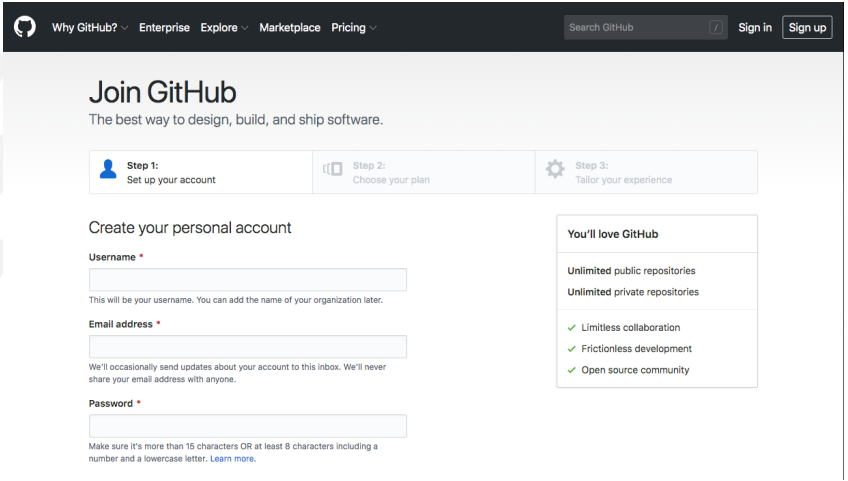
Note: Git is challenging to learn, but much more challenging to break.

Version control with Git



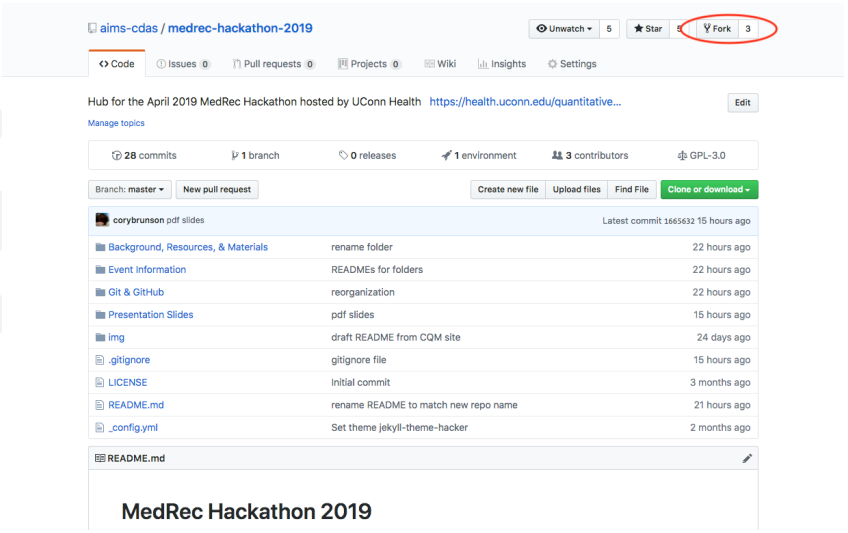
Create a GitHub account

https://github.com/join



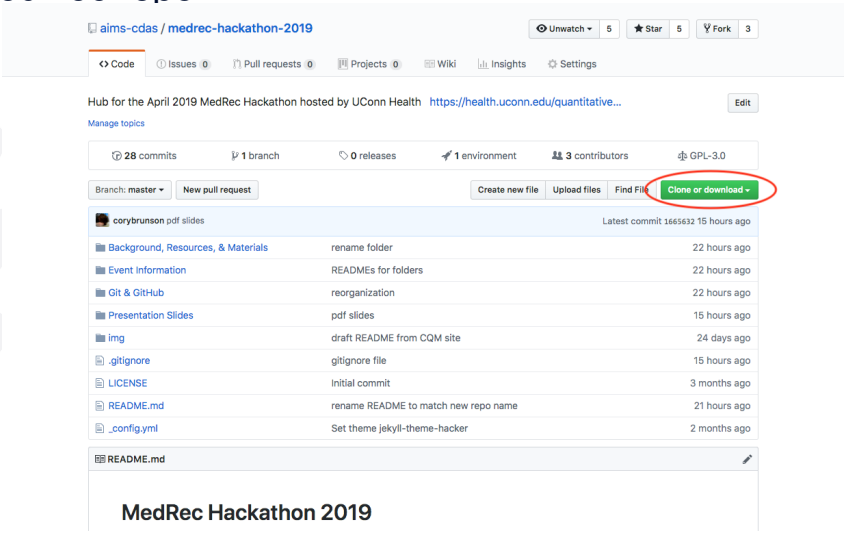
UCONN HEALTH

Fork the MedRec repo



UCONN HEALTH

Duplicate the MedRec repo



https://help.github.com/en/articles/duplicating-a-repository

UCONN HEALTH

Clone your forked copy or private repo

```
# clone repo from URL
git clone \
  https://github.com/<username>/medrec-hackathon-2019

# <navigate into local directory>

# list remotes linked to local directory
git remote
git remote --verbose
```

UCONN HEALTH

Have Git ignore certain files

Example `.gitignore` file:

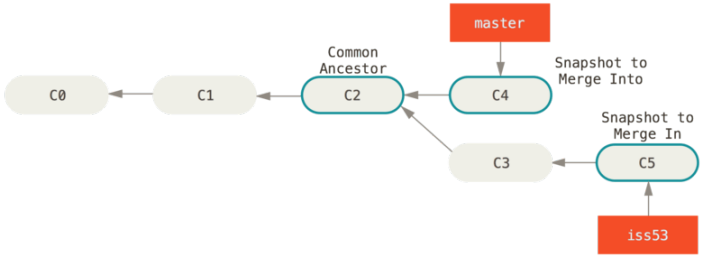
```
# Mac OS X metadata
.DS_Store

# R & RStudio metadata
.Rhistory
.Rapp.history
.RData
.Rproj.user

# experimentation folder
sandbox
```

Basic merging

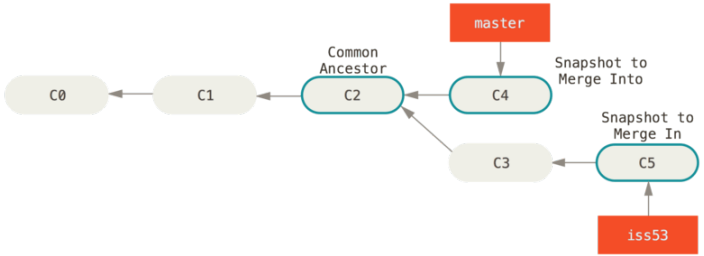
Setup:



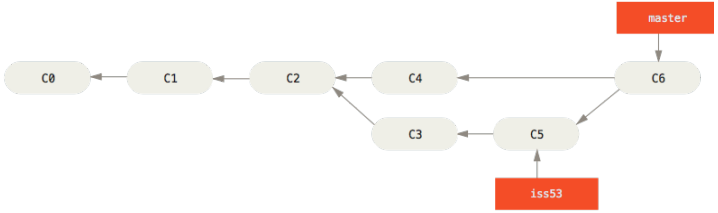
UCONN
HEALTH

Basic merging

Setup:



Merge:



UCONN HEALTH

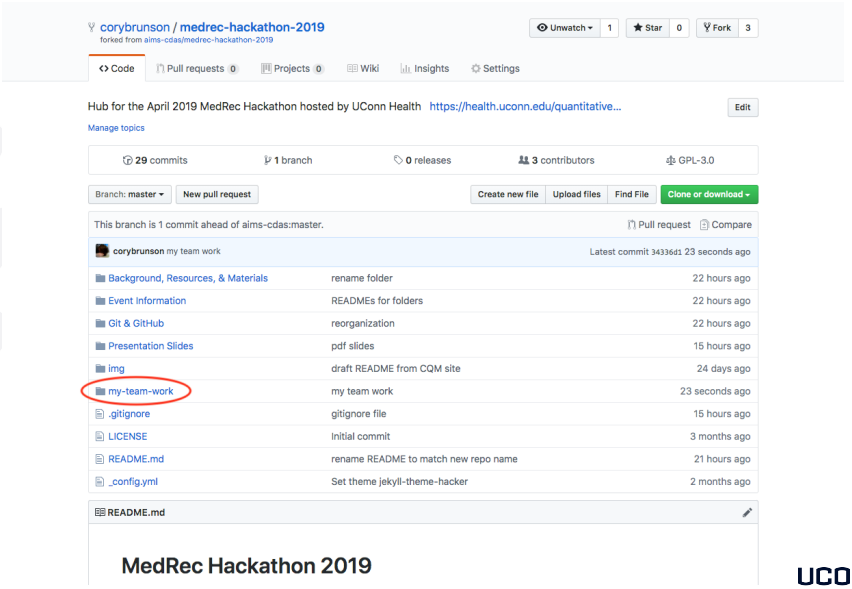
Work in a new branch

```
# view all branches
git branch

# create a new branch
git branch experimental

# switch to new branch
git checkout experimental
```

Work in project folders



UCONN HEALTH

Stage changes

```
# list all tracked changes (committed or not)
git status

# add changes to a file to the staging area
git add <filename>

# stage any changes to all tracked files
git add .

# stop tracking a file (without deleting it)
git rm --cached <filename>
```

Commit staged changes

```
# retrieve the last committed version of a file
# (erase uncommitted changes)
git checkout <filename>

# commit staged changes
git commit --message="this is what i  did"

# message option shortcut
git commit -m "this is what i did"
```

In general, **don't** change branches with uncommitted changes.

Merge into the master branch

```
# switch (back) to the master branch
git checkout master

# merge commits from the experimental branch
git merge experimental
```

Merge conflicts

```
<<<<<<< HEAD:<filename>
<content in current branch>
=======
<conflicting content in to-be-merged branch>
>>>>>>> experimental:<filename>

# open a merge tool to resolve merge conflicts
git mergetool
```

UCONN HEALTH

Fin

UCONN HEALTH