Introduction to Git and Github

Distributing a Shiny app

- The quickest way is to send (or put on github or gist or dropbox or whatever) someone the app directory and they can then call runApp

- You could create an R package and create a wrapper that calls runApp. Of course, these solutions only work if the user knows R

- Another option is to run a shiny server · ·

  - Requires setting up a (Shiny server)[http://www.rstudio.com/shiny/server/]
  - Probably easiest if you use one of the virtual machines where they already have Shiny servers running well (for example, on AWS)
  - Setting up a Shiny server is beyond the scope of this class as it involves some amount of linux server administration
  - Groups are creating a Shiny hosting services that will presumably eventually be a fee for service or freemium service
  - BTW, don't put system calls in your code (this is one of the first things many of us do for fun, but it introduces security concerns)

What is GitHub?

> “GitHub is a web-based hosting service for software development projects that use the Git revision control system.”

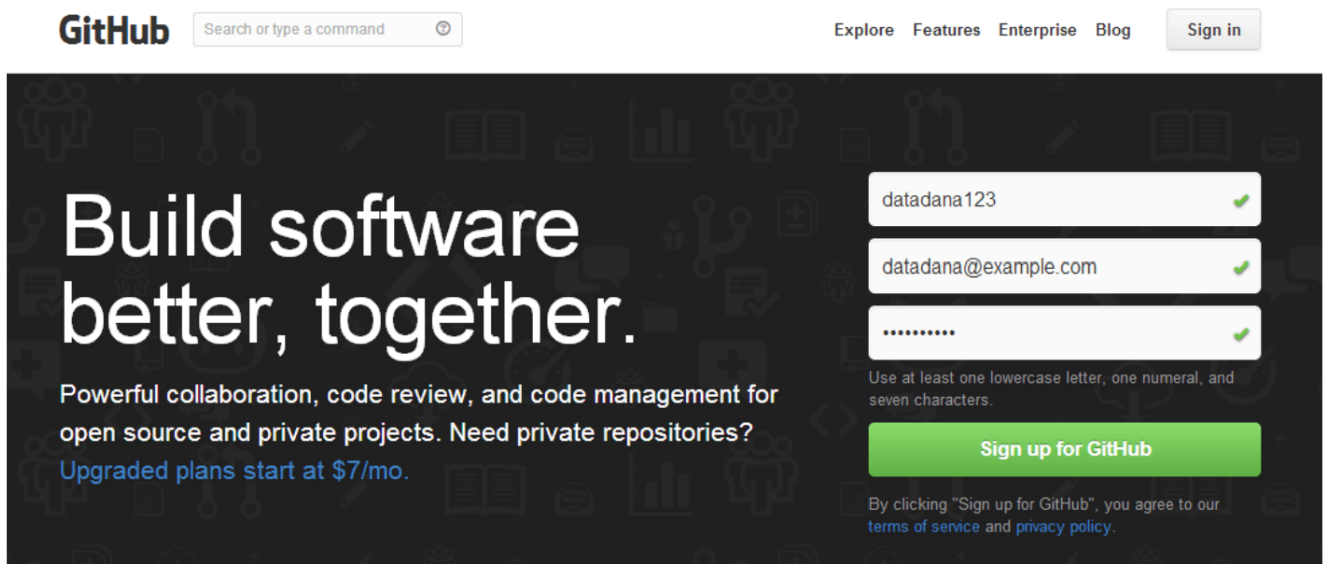
http://en.wikipedia.org/wiki/GitHub

What is GitHub?

- Allows users to "push" and "pull" their local repositories to and from remote repositories on the web
- Provides users with a homepage that displays their public repositories
- Users' repositories are backed up on the GitHub server in case something happens to the local copies
- Social aspect allows users to follow one another and share projects

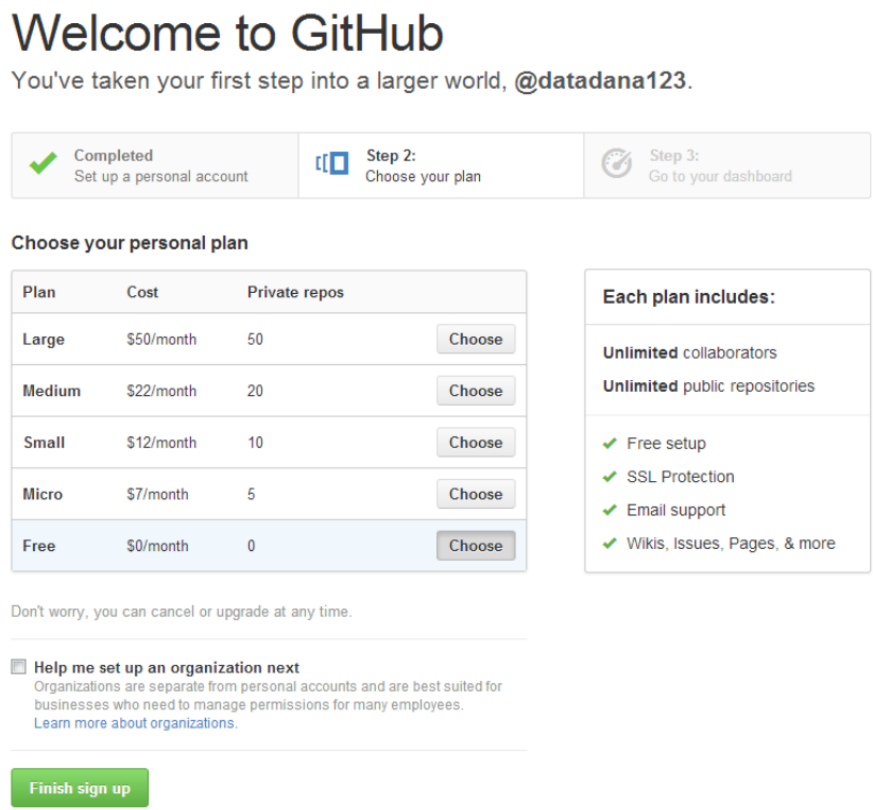
Set Up a GitHub Account

- Go to the GitHub homepage at https://github.com/
- Enter a username, email, and password and click "Sign up for GitHub"



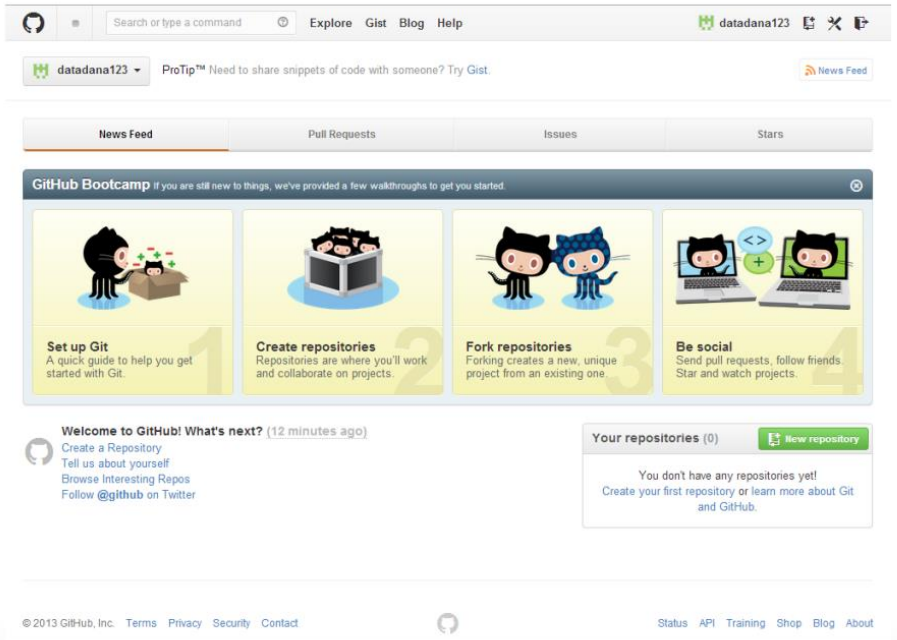
Set Up a GitHub Account

- On the next screen, select the free plan and click "Finish sign up"



Navigating GitHub

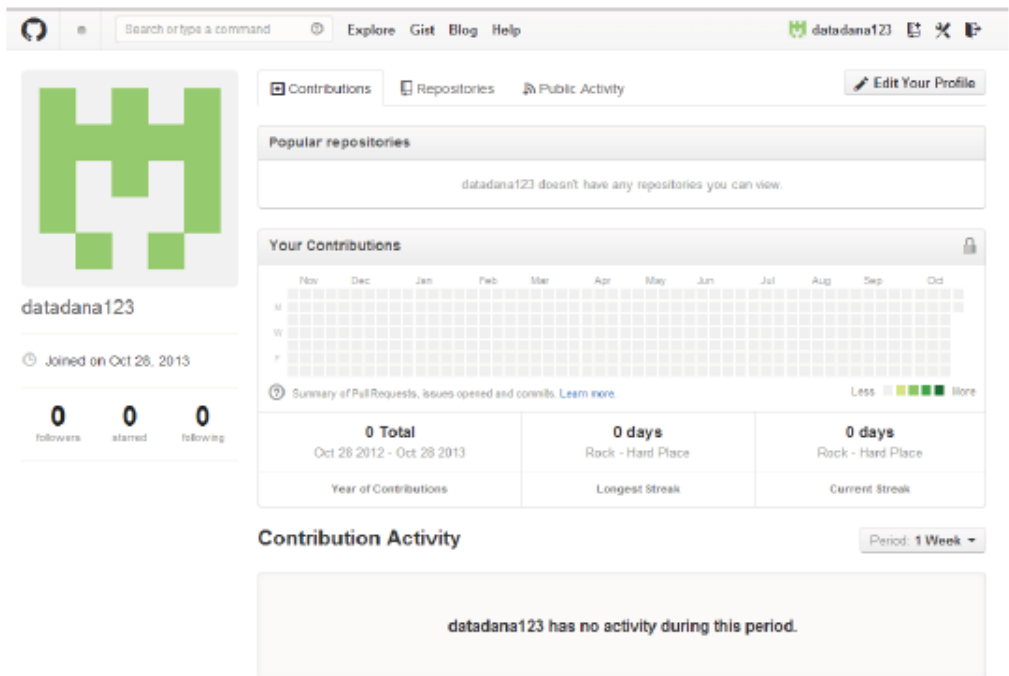
- After signing up, you will find yourself on this page, which has several helpful resources for learning more about Git and GitHub
- Try clicking on your username in the upper righthand corner of the screen to view your GitHub profile



Create repository and add your codes and a README.doc !!!!!

Your GitHub Profile

- Your profile is where all of your activity on GitHub is displayed
- Allows you to show other people who you are and what you are working on
- As you work on more and more projects, your profile becomes a portfolio of your work



Your GitHub Profile

- Finally, if you click on "Edit Your Profile" in the top righthand portion of the screen you can add some basic information about yourself to your profile
- This is totally optional, but if you do good work, you ought to take some credit for it!

Creating a GitHub Repository

Recap: Git vs. GitHub

- You don't need GitHub to use Git
- Git = Local (on your computer); GitHub = Remote (on the web)
- GitHub allows you to:
  1. Share your repositories with others
  2. Access other users' repositories
  3. Store remote copies of your repositories (on GitHub's server) in case something happens to your local copies (on your computer)

Creating a GitHub Repository

- Two methods of creating a GitHub repository:
  1. Start a repository from scratch
  2. "Fork" another user's repository
- We'll start with the first method
- *NOTE: A repository is often referred to as a "repo"*

Start a Repository From Scratch

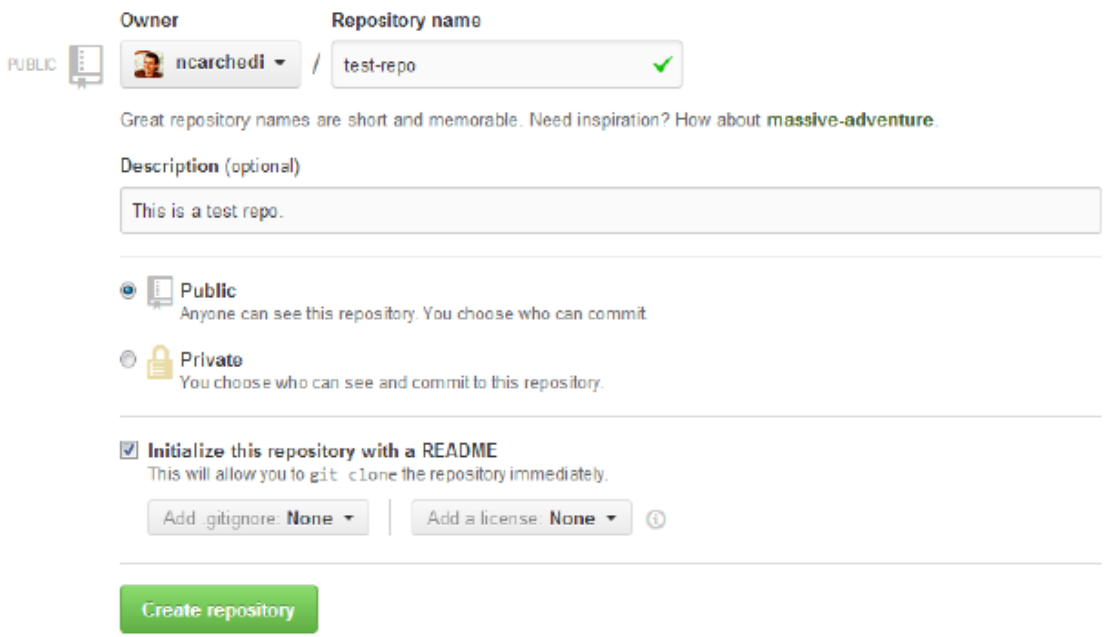
- Either go to your profile page (https://github.com/yourUserNameHere/) and click on "Create a new repo" in the upper righthand corner of the page

...OR...

- Go directly to https://github.com/new (you'll need to log into your GitHub account if you haven't already done so)

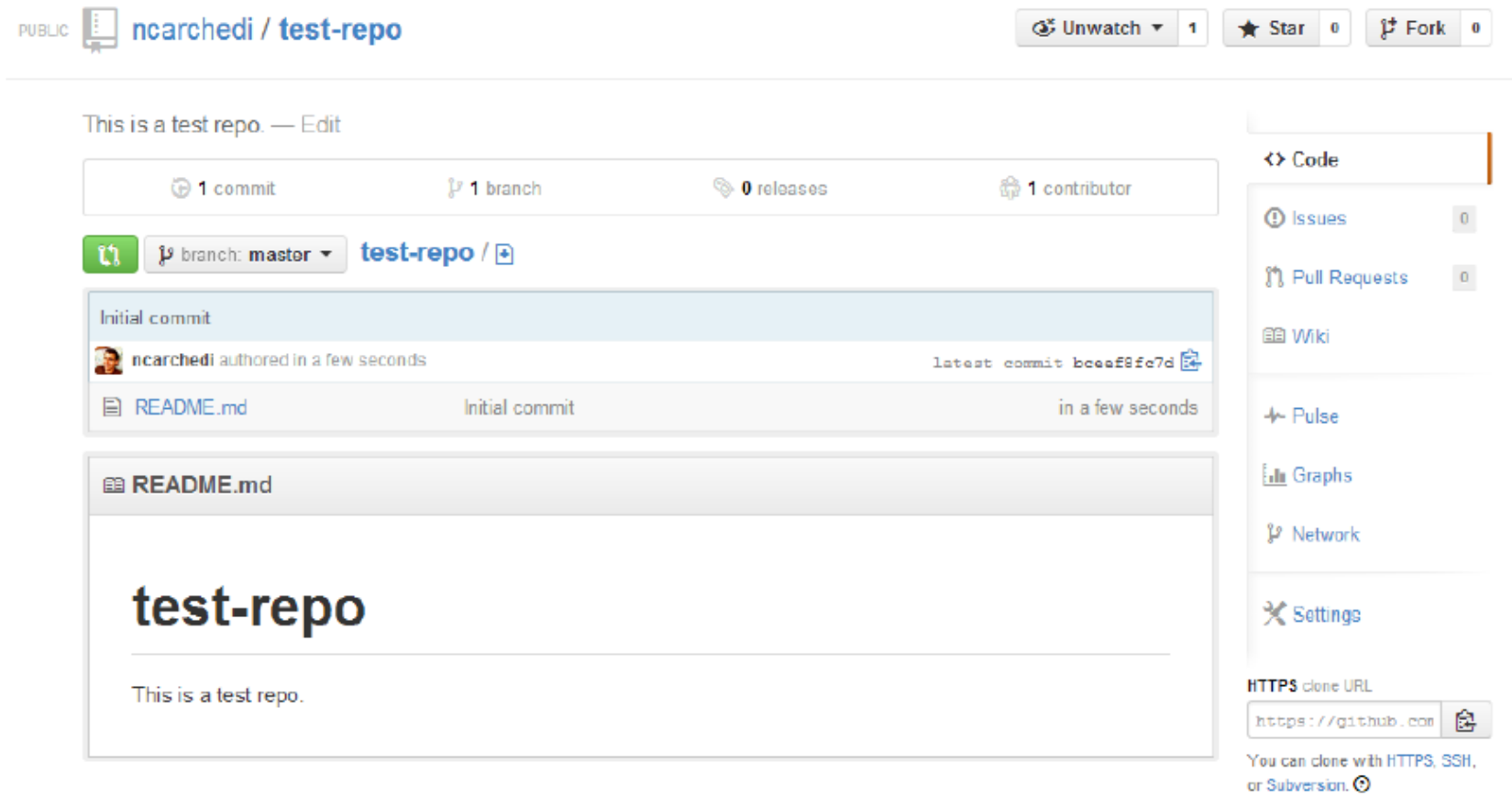
Start a Repository From Scratch

- Create a name for your repo and type a brief description of it
- Select "Public" (Private repos require a paid [or education] account)
- Check the box next to "Initialize this repository with a README"
- Click the "Create repository" button



Start a Repository From Scratch

- Congratulations! You've created a GitHub repository.



Creating a Local Copy

- Now you need to create a copy of this repo on your computer so that you can make changes to it
- Open Git Bash
- Create a directory on your computer where you will store your copy of the repo:

```
$ mkdir ~/test-repo
```

- Navigate to this new directory using the following command:

```
$ cd ~/test-repo
```

Creating a Local Copy

- Initialize a local Git repository in this directory

```
$ git init
```

- Point your local repository at the remote repository you just created on the GitHub server

```
$ git remote add origin https://github.com/yourUserNameHere/test-repo.git
```

Creating a Local Copy

- Here's what this process looks like in action:

```
Welcome to Git (version 1.8.4-preview20130916)


Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

Nick@NICK-PC ~
$ mkdir ~/test-repo

Nick@NICK-PC ~
$ cd ~/test-repo

Nick@NICK-PC ~/test-repo
$ git init
Initialized empty Git repository in c:/Users/Nick/test-repo/.git/

Nick@NICK-PC ~/test-repo (master)
$ git remote add origin https://github.com/ncarchedi/test-repo.git

Nick@NICK-PC ~/test-repo (master)
$ _
```

Fork a Another User's Repository

- The second method of creating a respository is to make a copy of someone else's
- This process is called "forking" and is an important aspect of open-source software development
- Begin by navigating to the desired repository on the GitHub website and click the "Fork" button shown below

https://help.github.com/articles/fork-a-repo

Clone the Repo

- You now have a copy of the desired respository on your GitHub account
- Need to make a local copy of the repo on your computer
- This process is called "cloning" and can be done using the following command:

```
$ git clone https://github.com/yourUserNameHere/repoNameHere.git
```

- *NOTE: This will clone the repository into your current directory.*

What Else?

- If you make changes to your local copy of the repo, you'll probably want to push your changes to GitHub at some point
- You also may be interested in staying current with any changes made to the original repository from which you forked your copy

  - https://help.github.com/articles/fork-a-repo
  - http://git-scm.com/book/en/Git-Basics-Getting-a-Git-Repository

What is Git?

> “Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.”

http://git-scm.com/

What is Git?

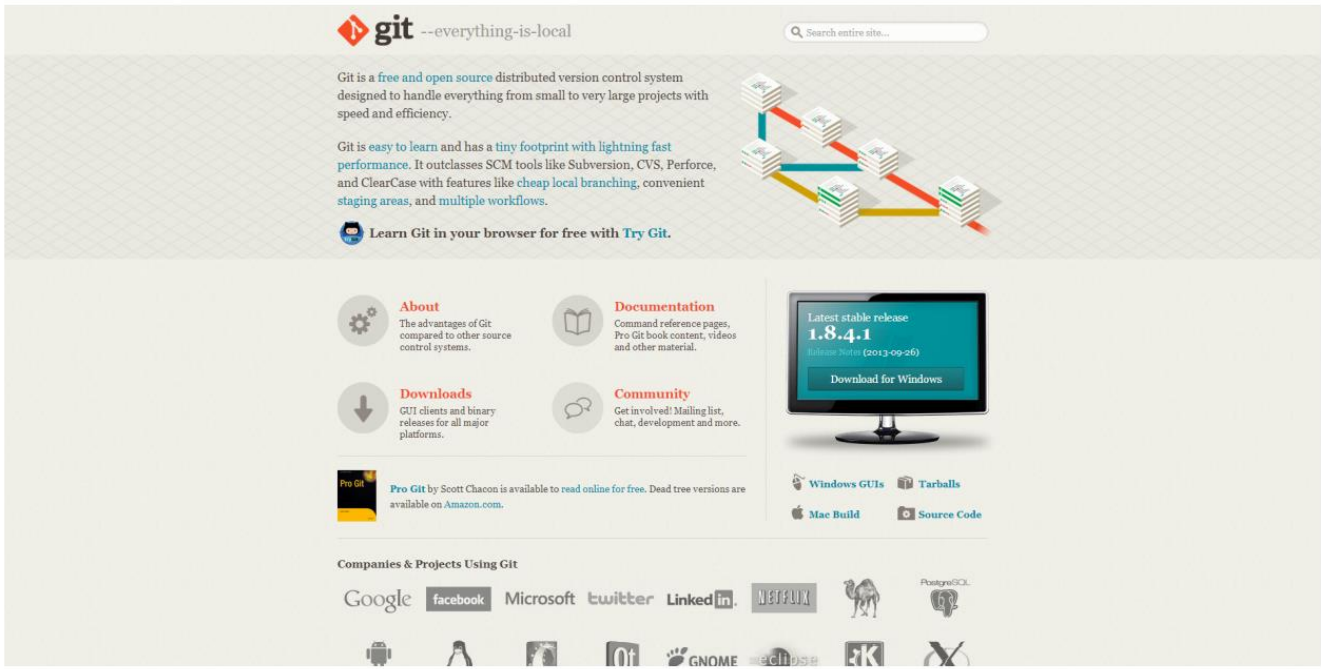
- Created by the same people who developed Linux
- The most popular implementation of version control today
- Everything is stored in local repositories on your computer
- Operated from the command line

http://git-scm.com/book/en/Getting-Started-A-Short-History-of-Git

Download Git

- Go to the following website and click on the download link for your operating system (Mac, Windows, Linux, etc):

http://git-scm.com/downloads



Install Git

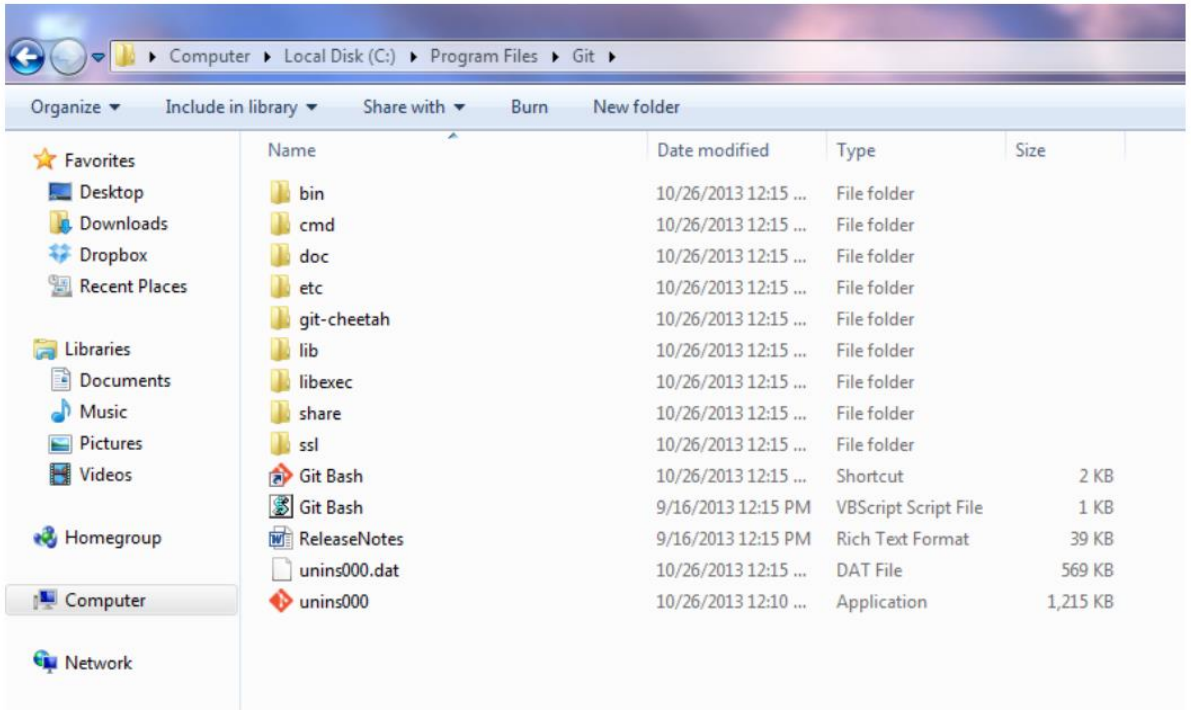
- Once the file is done downloading, open it up to begin the Git installation

Install Git

- Unless you really know what you are doing, just go with the default options at each step of the installation
- Once the install is complete, hit the "Finish" button (you may want to uncheck the box next to "Review ReleaseNotes.rtf")

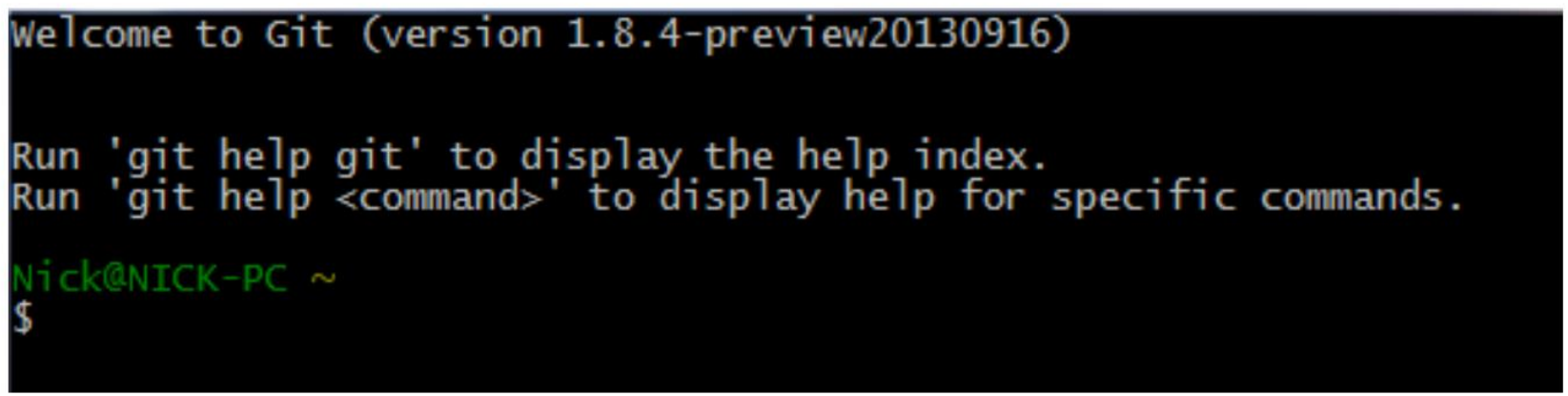
Open Git Bash

- Find a program called Git Bash, which is the command line environment for interacting with Git
- It should be located in the directory into which Git was installed (or, for Windows users, in the Start Menu)



Open Git Bash

- Once Git Bash opens, you'll see a short welcome message followed by the name of your computer and a dollar sign on the next line
- The dollar sign means that it's your turn to type a command



Configure Username and Email

- Each commit to a Git repository will be "tagged" with the username of the person who made the commit
- Enter the following commands in Git Bash, one at a time, to set your username and email:

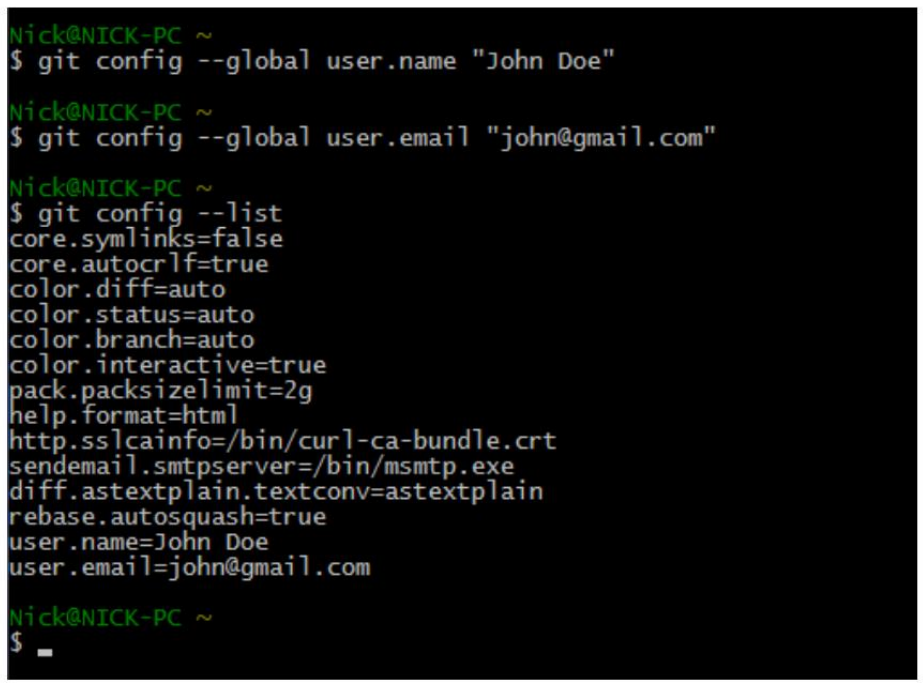
```
$ git config --global user.name "Your Name Here"
$ git config --global user.email "your_email@example.com"
```

- You'll only have to do this once, but you can always change these down the road using the same commands

Configure Username and Email

- Now type the following to confirm your changes (they may be listed toward the bottom):

```
$ git config --list
```



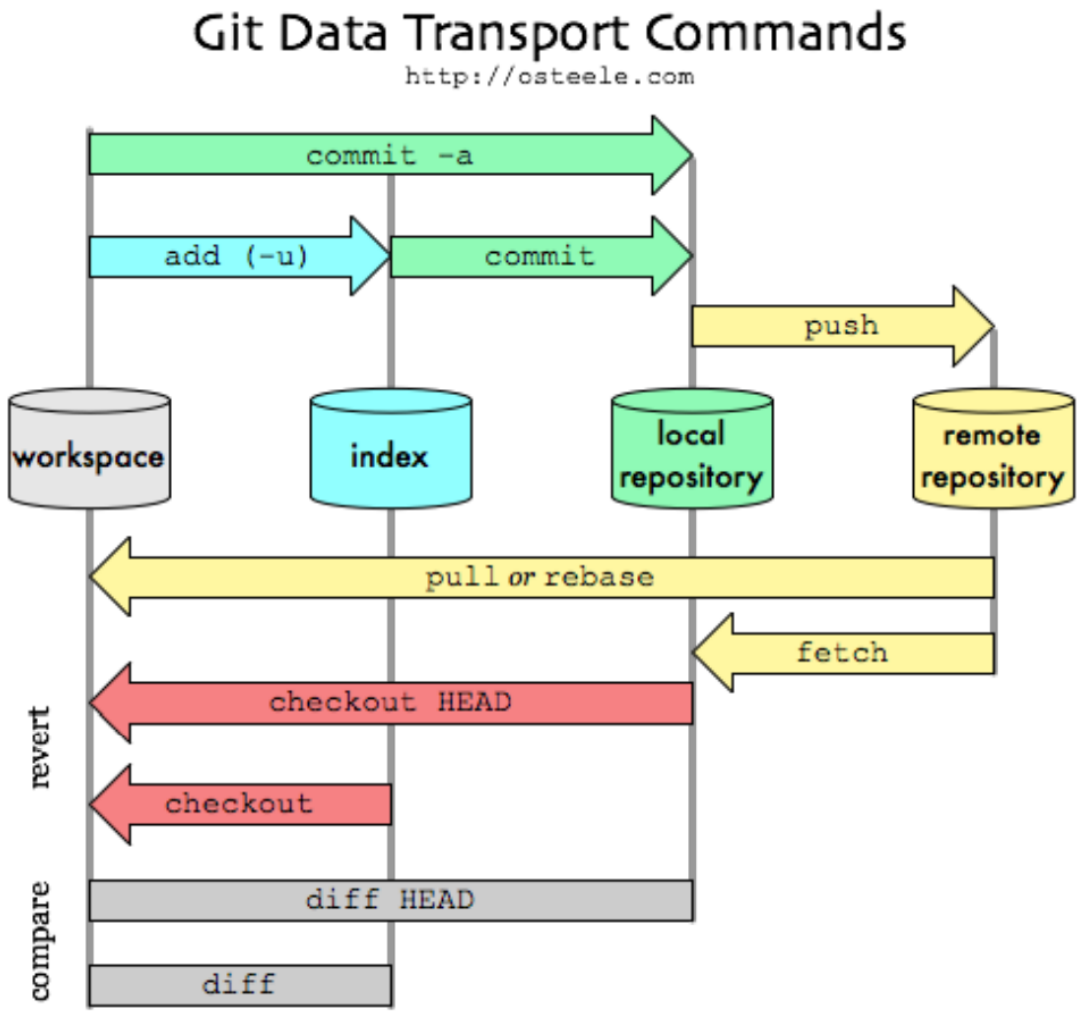
What's Next?

- Go ahead and close Git Bash with following command:

```
$ exit
```

- Now that Git is set up on your computer, we're ready to move on to GitHub, which is a web-based platform that lets you do some pretty cool stuff
- Once GitHub is up and running, we'll show you how to start using these tools to your benefit

Pushing and pulling



http://gitready.com/beginner/2009/01/21/pushing-and-pulling.html

Adding

- Suppose you add new files to a local repository under version control
- You need to let Git know that they need to be tracked
  - `git add .` adds all new files
  - `git add -u` updates tracking for files that changed names or were deleted
  - `git add -A` does both of the previous
- You should do this before committing

Committing

- You have changes you want to commit to be saved as an intermediate version
- You type the command
  - `git commit -m "message"` where message is a useful description of what you did
- This only updates your local repo, not the remote repo on Github

Pushing

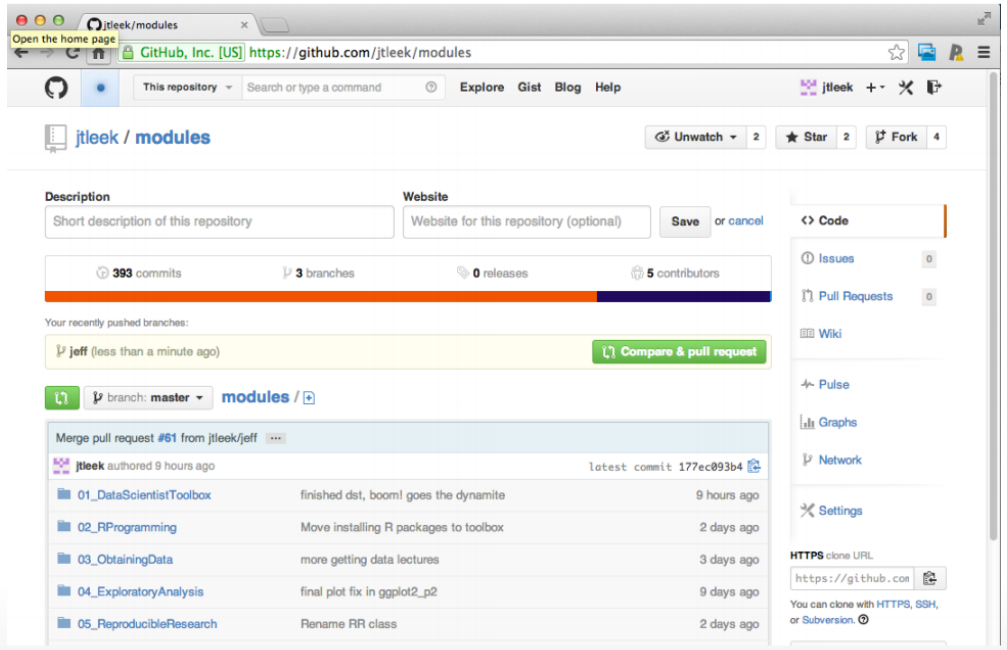
- You have saved local commits you would like to update on the remote (Github)
- You type the command
  - `git push`

Branches

- Sometimes you are working on a project with a version being used by many people
- You may not want to edit that version
- So you can create a branch with the command
  - `git checkout -b branchname`
- To see what branch you are on type:
  - `git branch`
- To switch back to the master branch type
  - `git checkout master`

Pull requests

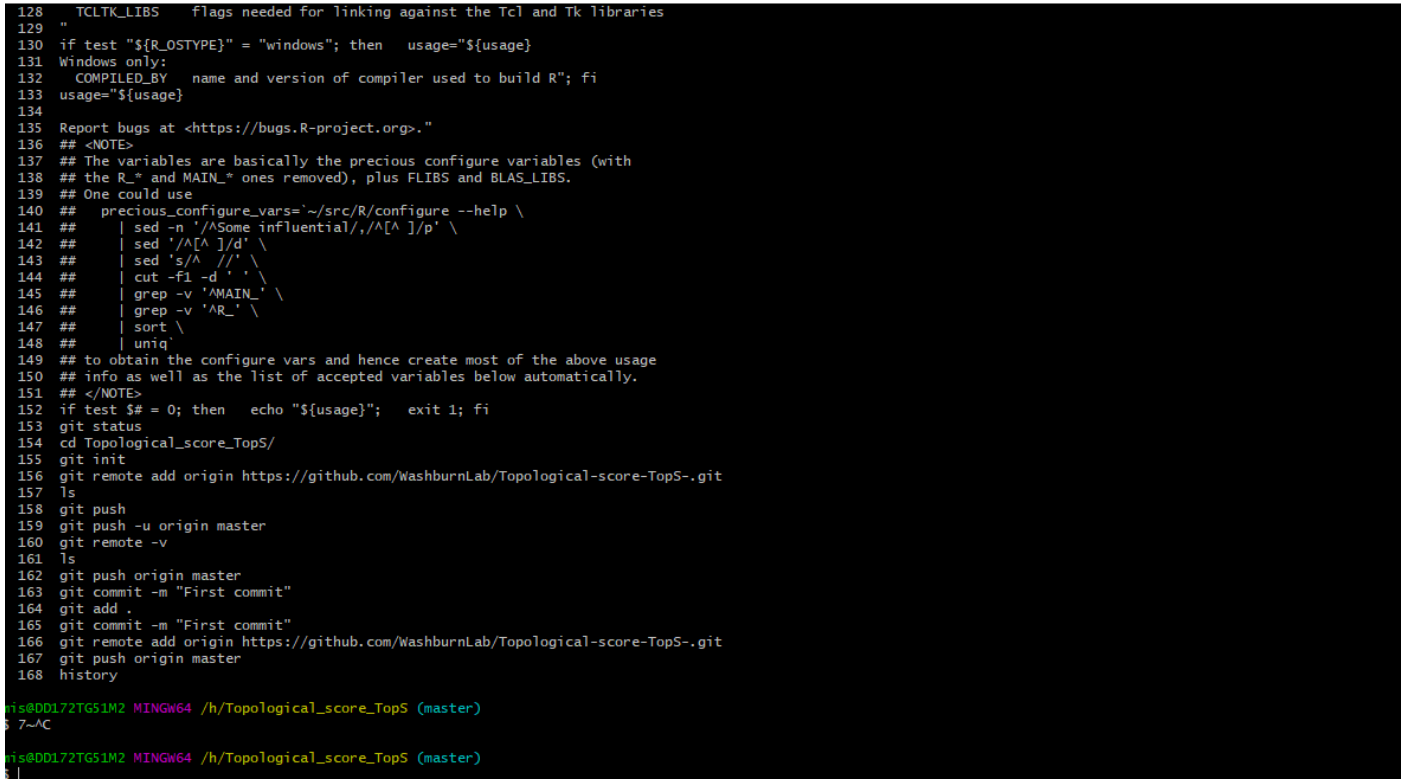
- If you fork someone's repo or have multiple branches you will both be working seperately
- Sometimes you want to merge in your changes into the other branch/repo
- To do so you need to send a pull request.
- This is a feature of Github.



Your git

```
msardiu@BIL-5CG1086G5Q MINGW64 ~
$ cd mihaela/


msardiu@BIL-5CG1086G5Q MINGW64 /g/mihaela
$

msardiu@BIL-5CG1086G5Q MINGW64 /g/mihaela
$ ls
ANS_HBC-talk_3-30-2021.pptx                                camera.csv
Converted/                                                 data_colins.csv
Healthy_OV_intersection_plasma_FORSTAT_forqspec.txt         test.csv
Healthy_OV_intersection_plasma_FORSTAT_forqspec.txt_qprot_fdr  test.txt
ResearchStrategy/                                          test1.txt
TEACHING/

msardiu@BIL-5CG1086G5Q MINGW64 /g/mihaela
$ ls
ANS_HBC-talk_3-30-2021.pptx                                ResearchStrategy/  test.csv
Converted/                                                 TEACHING/          test.txt
Healthy_OV_intersection_plasma_FORSTAT_forqspec.txt         camera.csv         test1.txt
Healthy_OV_intersection_plasma_FORSTAT_forqspec.txt_qprot_fdr  data_colins.csv

msardiu@BIL-5CG1086G5Q MINGW64 /g/mihaela
$
```

Push your files to github



In your home directory

```
git commit -m "First commit"
git add .
git commit -m "First commit"
git remote add origin https://github.com/WashburnLab/Topological-score-TopS-.git
git push origin master
```

github.com/mihaelas/Topological_score-TopS-

Update

Why GitHub? Team Enterprise Explore Marketplace Pricing

Search / Sign in Sign up

mihaelas / Topological_score-TopS-

Notifications Star 0 Fork 0

Code Issues Pull requests Actions Projects Wiki Security Insights

master 1 branch 0 tags

Go to file Code

mihaelas Add files via upload f4cf63b on Feb 3, 2020 2 commits

README.md	Initial commit	17 months ago
README_document.docx	Add files via upload	17 months ago
code_score.c	Add files via upload	17 months ago
input_file.csv	Add files via upload	17 months ago
output_file.txt	Add files via upload	17 months ago
server.R	Add files via upload	17 months ago

https://github.com/mihaelas/Topological_score-TopS-/security

About

Creating topological score for the analysis of large data

Readme

Releases

No releases published

Packages

No packages published

nature COMMUNICATIONS

Nat Commun. 2019; 10: 1118.
Published online 2019 Mar 8. doi: 10.1038/s41467-019-09123-y

PMCID: PMC6408525
PMID: 30850613

Topological scoring of protein interaction networks

Mihaela E. Sardiu,[1] Joshua M. Gilmore,[1,3] Brad D. Groppe,[1,4] Arnob Dutta,[1,5] Laurence Florens,[1] and Michael P. Washburn[1,2]

▸ Author information ▸ Article notes ▸ Copyright and License information Disclaimer

Associated Data

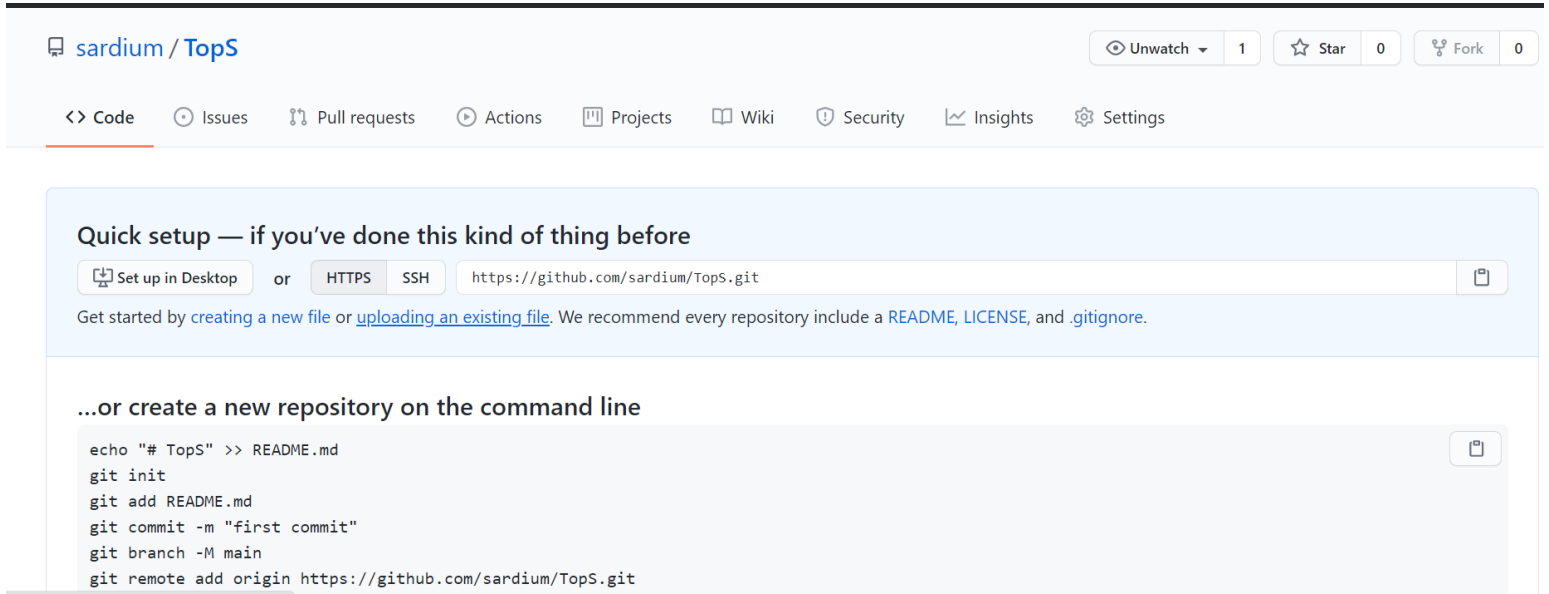
▸ Supplementary Materials

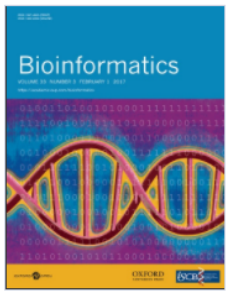
- You can **upload** and commit an existing **file** to a **GitHub** repository.

…

Tips:

- On **GitHub**, navigate to the main page of the repository.
- Above the list of **files**, using the Add **file** drop-down, click **Upload files**.
- Drag and drop the **file** or folder you'd like to **upload** to your repository onto the **file** tree.





Volume 33, Issue 3
1 February 2017

Article Contents

The START App: a web-based RNAseq analysis and visualization resource FREE

Jonathan W Nelson, Jiri Sklenar, Anthony P Barnes, Jessica Minnier

Bioinformatics, Volume 33, Issue 3, 1 February 2017, Pages 447–449,
https://doi.org/10.1093/bioinformatics/btw624
Published: 06 October 2016 **Article history** ▾

Abstract

Summary:

START: Shiny Transcriptome Analysis Resource Tool

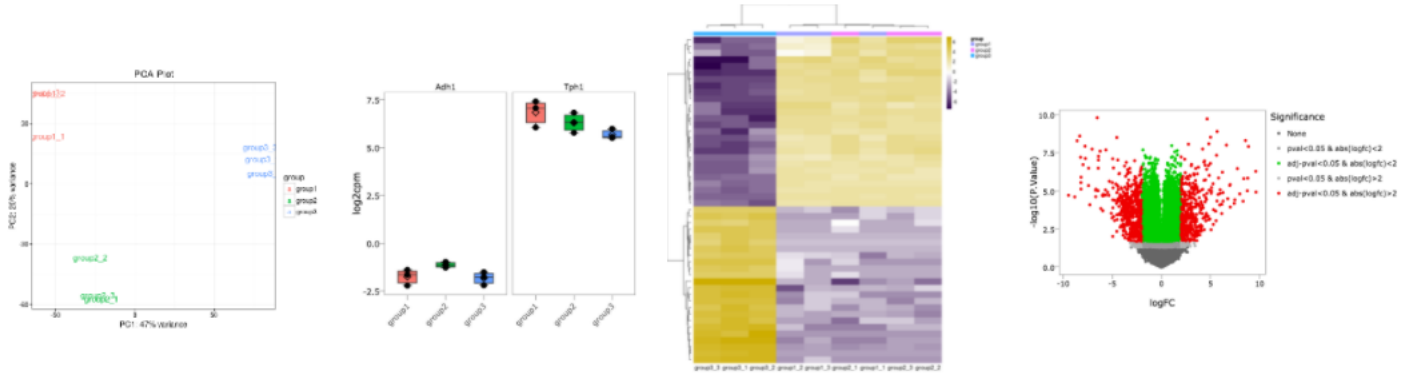
Getting Started with START

The START app allows users to visualize RNA-seq data starting with count data.

- *Explore* the app's features with the example data set pre-loaded by clicking on the tabs above.
- *Upload* your data in the "Input Data" tab.

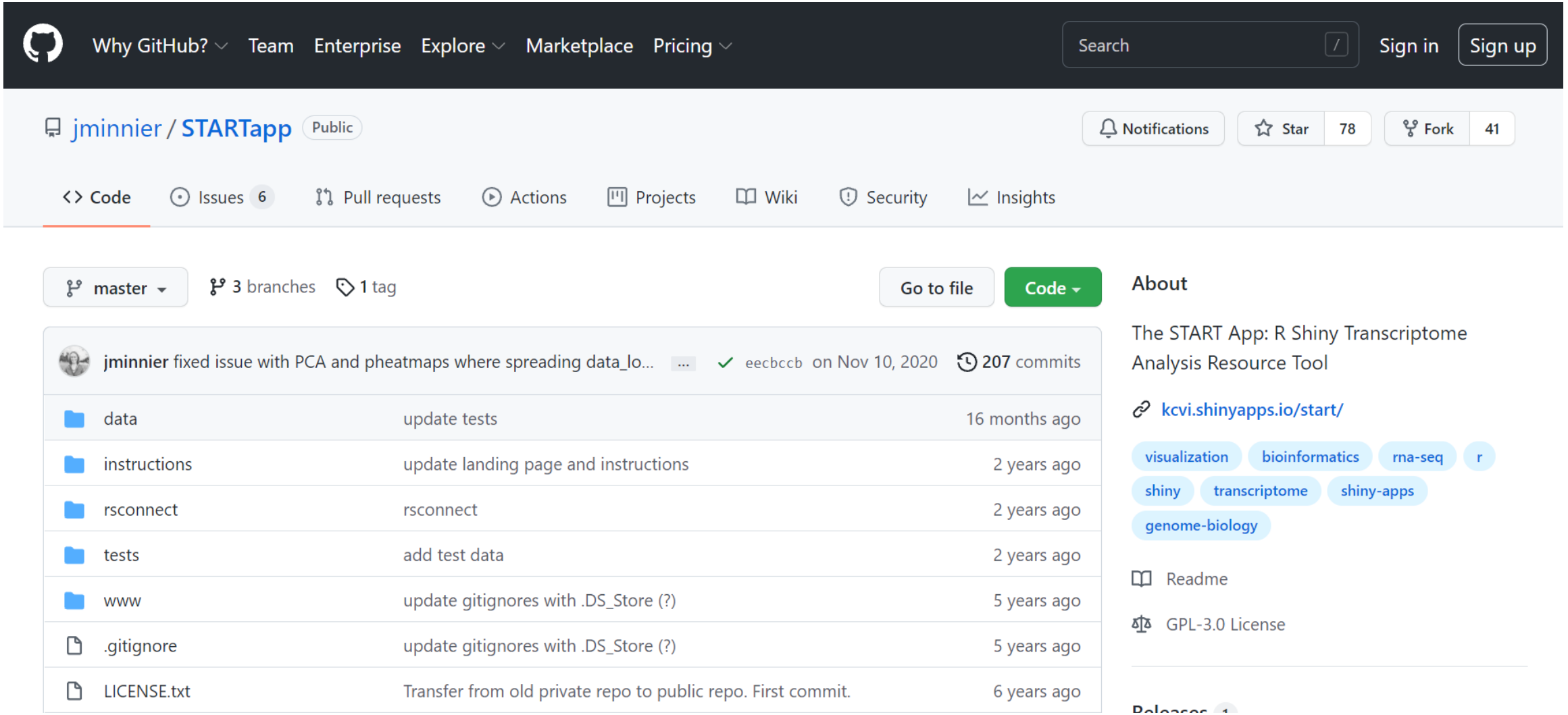
Note, this is version 1.1.1, published on 12/9/19. If you need the previous version, it is hosted here: kcvi.shinyapps.io/STARTapp_2018/

Features



Visualize your data:

- clustering (PCA plots, heatmaps)
- group comparisons (scatterplots, volcano plots)
- gene-level boxplots of expression values


Why GitHub?
Team
Enterprise
Explore
Marketplace
Pricing
Search
Sign in
Sign up
jminnier / STARTapp
Public
Notifications
Star 78
Fork 41
Code
Issues 6
Pull requests
Actions
Projects
Wiki
Security
Insights
master
3 branches
1 tag
Go to file
Code
jminnier fixed issue with PCA and pheatmaps where spreading data_lo...
eecbccb on Nov 10, 2020
207 commits
data update tests 16 months ago
instructions update landing page and instructions 2 years ago
rsconnect rsconnect 2 years ago
tests add test data 2 years ago
www update gitignores with .DS_Store (?) 5 years ago
.gitignore update gitignores with .DS_Store (?) 5 years ago
LICENSE.txt Transfer from old private repo to public repo. First commit. 6 years ago
About
The START App: R Shiny Transcriptome Analysis Resource Tool
kcvi.shinyapps.io/start/
visualization
bioinformatics
rna-seq
r
shiny
transcriptome
shiny-apps
genome-biology
Readme
GPL-3.0 License