

Introduction to Git and Github

- 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 . .

Distributing a Shiny app

- 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.”

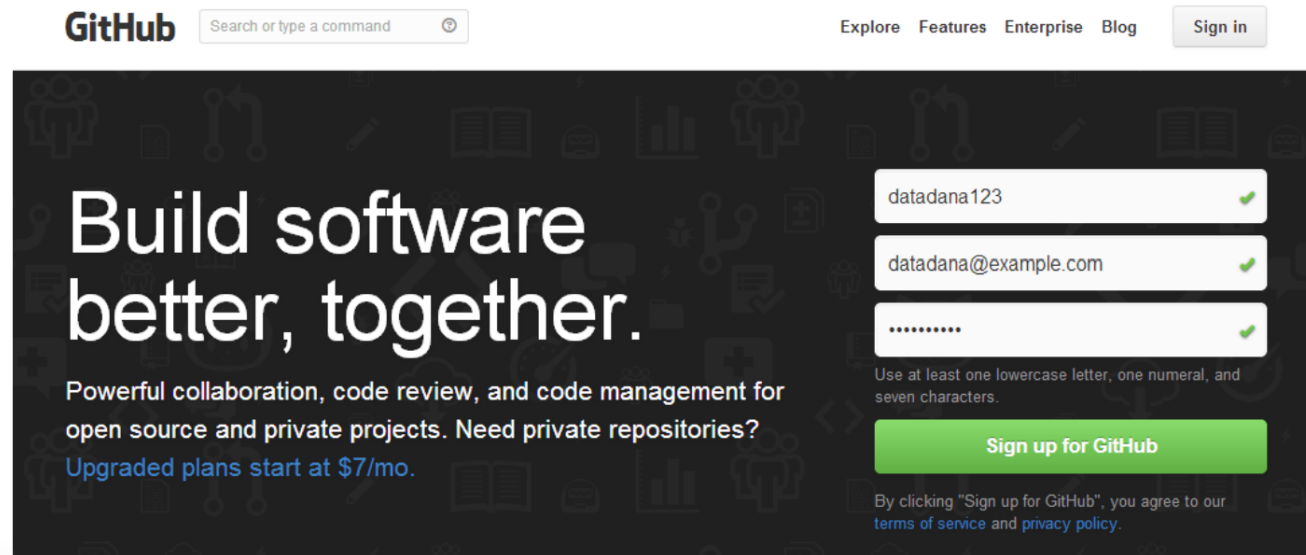
<http://en.wikipedia.org/wiki/GitHub>

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"

A screenshot of the GitHub homepage with the sign-up form filled out. The header includes the GitHub logo, a search bar, and navigation links. The main content area has a dark background with the slogan 'Build software better, together.' and a description of GitHub's features. The sign-up form on the right has three input fields: a username 'datadana123', an email 'datadana@example.com', and a password represented by dots. Each field has a green checkmark on the right. Below the password field is a note about password requirements. A green 'Sign up for GitHub' button is at the bottom of the form, followed by a line of text about agreeing to terms of service and privacy policy.

GitHub Search or type a command

Explore Features Enterprise Blog Sign in

Build software better, together.

Powerful collaboration, code review, and code management for open source and private projects. Need private repositories? Upgraded plans start at \$7/mo.

datadana123 ✓

datadana@example.com ✓

..... ✓

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

Sign up for GitHub


By clicking "Sign up for GitHub", you agree to our [terms of service](#) and [privacy policy](#).


Set Up a GitHub Account


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

Welcome to GitHub

You've taken your first step into a larger world, @datadana123.

 Completed
Set up a personal account

 Step 2:
Choose your plan

 Step 3:
Go to your dashboard

Choose your personal plan

Plan	Cost	Private repos	
Large	\$50/month	50	<button>Choose</button>
Medium	\$22/month	20	<button>Choose</button>
Small	\$12/month	10	<button>Choose</button>
Micro	\$7/month	5	<button>Choose</button>
Free	\$0/month	0	<button>Choose</button>

Each plan includes:

Unlimited collaborators

Unlimited public repositories

- ✓ Free setup
- ✓ SSL Protection
- ✓ Email support
- ✓ Wikis, Issues, Pages, & more

Don't worry, you can cancel or upgrade at any time.

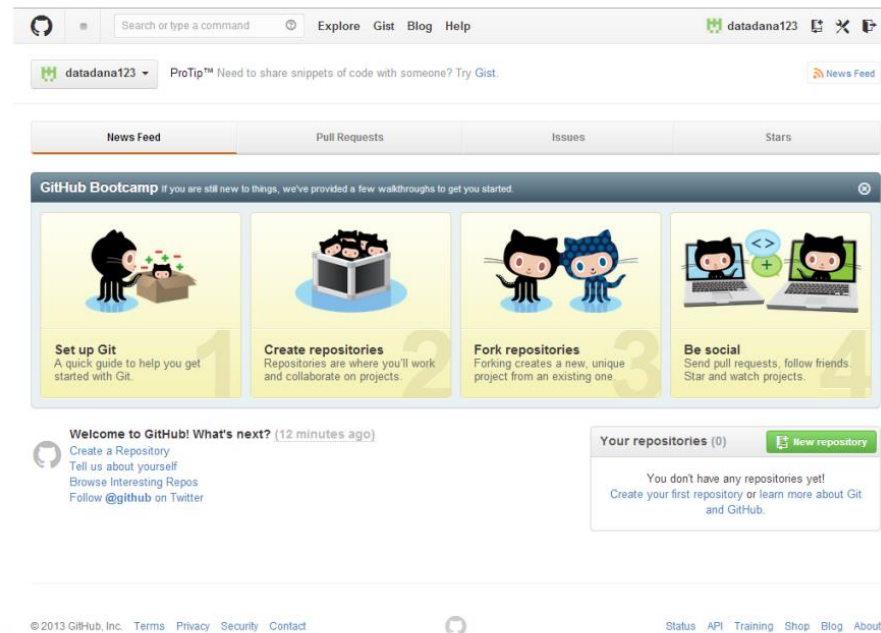
☐ **Help me set up an organization next**
Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees.
[Learn more about organizations.](#)

[Finish sign up](#)

Navigating GitHub

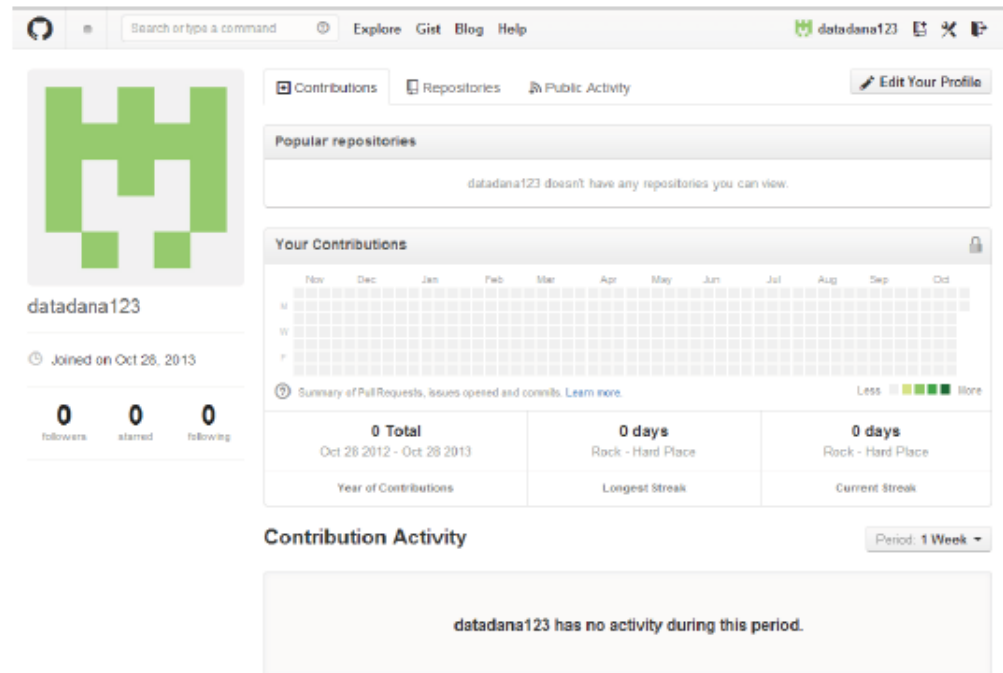
- After signing up, you will find yourself on this page, which has several helpful resources for learning more about Git and GitHub
- Try clicking on your username in the upper righthand corner of the screen to view your GitHub profile

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

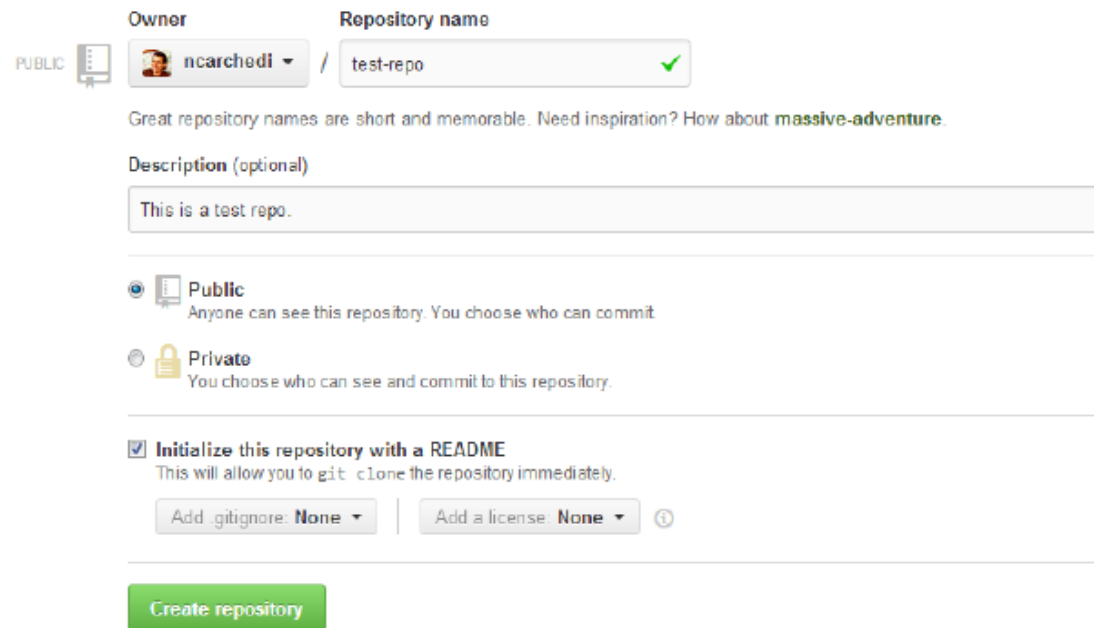
- Either go to your profile page (<https://github.com/yourUserNameHere/>) and click on "Create a new repo" in the upper righthand corner of the page

...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



The screenshot shows the GitHub 'Create new repository' form. At the top, there's a 'PUBLIC' label with a lock icon. The 'Owner' field shows 'ncarchedi' with a dropdown arrow. The 'Repository name' field contains 'test-repo' with a green checkmark. Below this, a hint says 'Great repository names are short and memorable. Need inspiration? How about massive-adventure.' The 'Description (optional)' field contains 'This is a test repo.' Below the description, there are two radio button options: 'Public' (selected) with the text 'Anyone can see this repository. You choose who can commit.' and 'Private' with the text 'You choose who can see and commit to this repository.' At the bottom, there's a checked checkbox for 'Initialize this repository with a README' with the text 'This will allow you to git clone the repository immediately.' Below this, there are two dropdown menus: 'Add .gitignore: None' and 'Add a license: None', followed by a help icon. At the very bottom is a green 'Create repository' button.

PUBLIC

Owner: ncarchedi / Repository name: test-repo ✓

Great repository names are short and memorable. Need inspiration? How about [massive-adventure](#).

Description (optional): This is a test repo.

☒ 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 allow you to git clone the repository immediately.

Add .gitignore: None | Add a license: None ⓘ

Create repository



Start a Repository From Scratch



- Congratulations! You've created a GitHub repository.


PUBLIC  **ncarchedi** / **test-repo** Unwatch 1 Star 0 Fork 0


This is a test repo. — Edit

1 commit 1 branch 0 releases 1 contributor

 branch: master test-repo / 

Initial commit
 ncarchedi authored in a few seconds latest commit bceaf8fc7d 

 README.md Initial commit in a few seconds


 README.md


test-repo

This is a test repo.

Code

- Issues 0
- Pull Requests 0
- Wiki
- Pulse
- Graphs
- Network
- Settings

HTTPS clone URL
 

You can clone with HTTPS, SSH, or Subversion. 

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

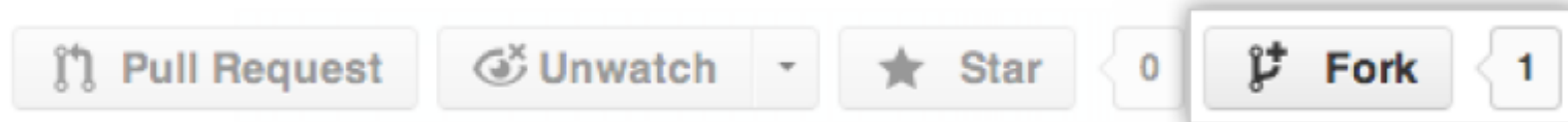
Nick@NICK-PC ~/test-repo
$ git init
Initialized empty Git repository in c:/Users/Nick/test-repo/.git/

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 repository 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 repository 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?

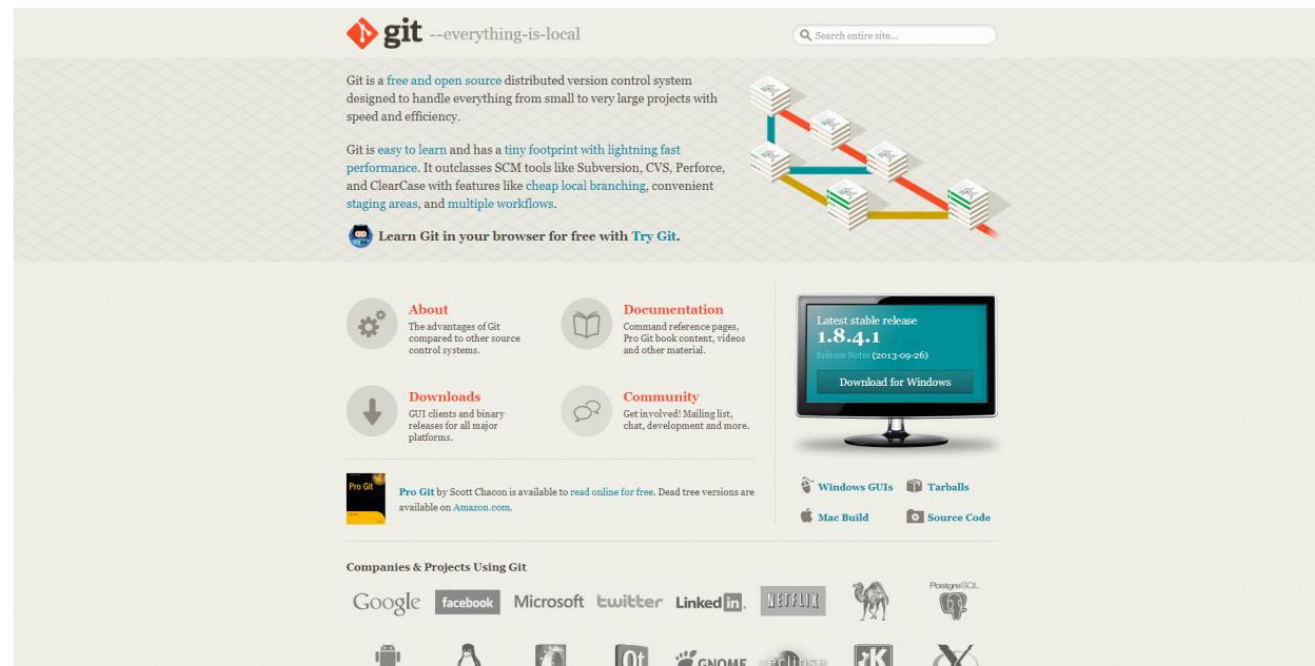
- Created by the same people who developed Linux
- The most popular implementation of version control today
- Everything is stored in local repositories on your computer
- Operated from the command line

<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



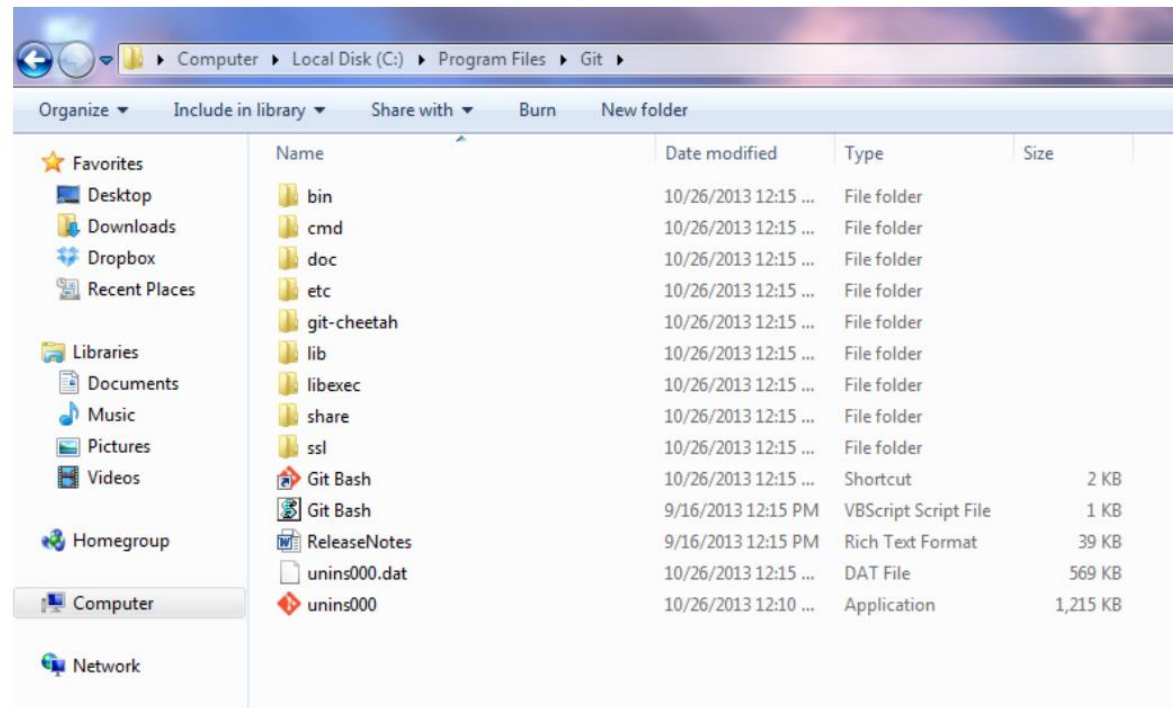
Install Git

- 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

```
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 ~
$
```

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
```

```
Nick@NICK-PC ~
$ git config --global user.name "John Doe"

Nick@NICK-PC ~
$ git config --global user.email "john@gmail.com"

Nick@NICK-PC ~
$ git config --list
core.symlinks=false
core.autocrlf=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
pack.packsizelimit=2g
help.format=html
http.sslcainfo=/bin/curl-ca-bundle.crt
sendemail.smtpserver=/bin/msmtp.exe
diff.astextplain.textconv=astextplain
rebase.autosquash=true
user.name=John Doe
user.email=john@gmail.com

Nick@NICK-PC ~
$ _
```

What's Next?

- Go ahead and close Git Bash with following command:

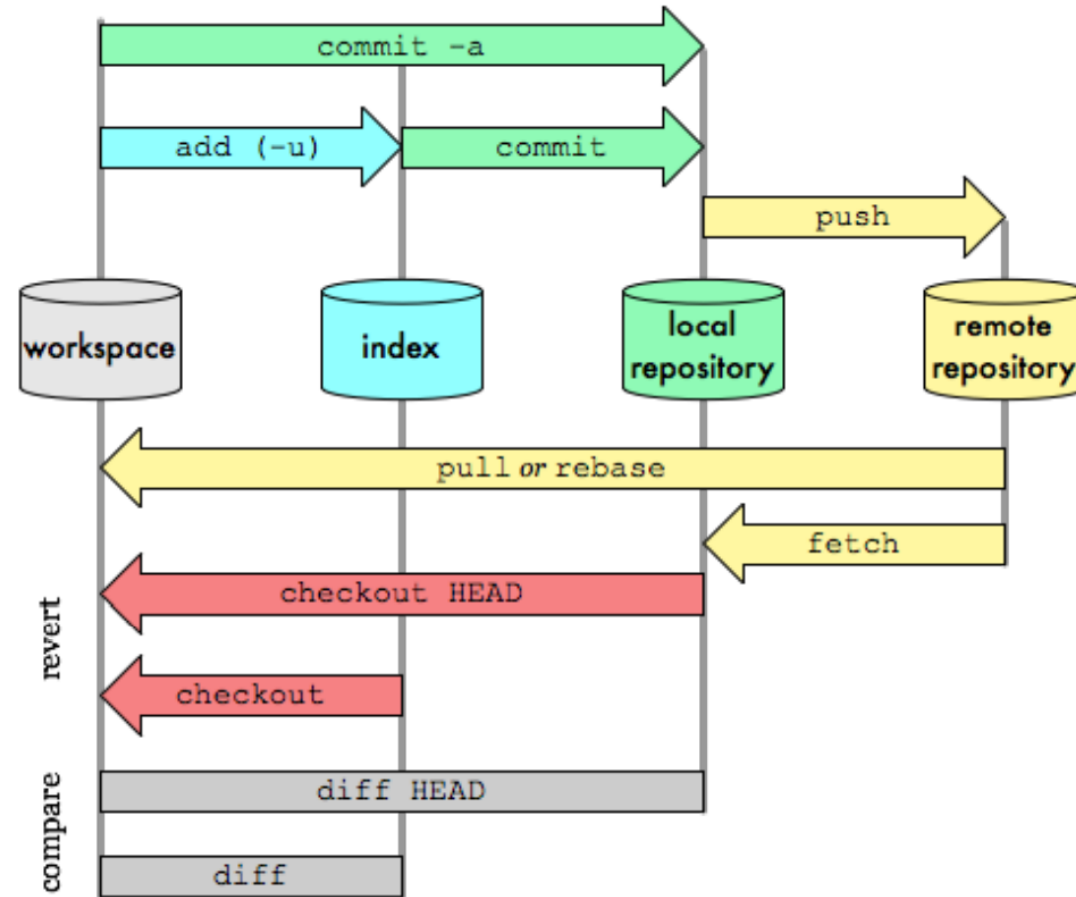
```
$ exit
```

- 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

Git Data Transport Commands

<http://osteele.com>



<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

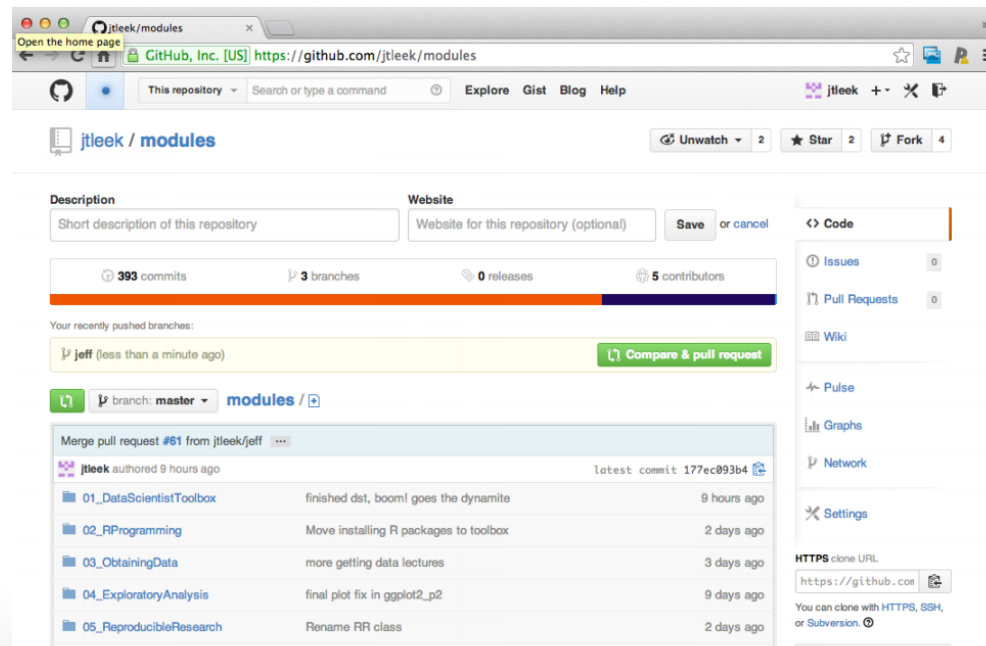
- You have saved local commits you would like to update on the remote (Github)
- You type the command
 - `git push`

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 separately
- 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


```
128 TCLTK_LIBS      flags needed for linking against the Tcl and Tk libraries
129 "
130 if test "${R_OSTYPE}" = "windows"; then  usage="${usage}
131 Windows only:
132 COMPILED_BY      name and version of compiler used to build R"; fi
133 usage="${usage}
134
135 Report bugs at <https://bugs.R-project.org>."
136 ## <NOTE>
137 ## The variables are basically the precious configure variables (with
138 ## the R_* and MAIN_* ones removed), plus FLIBS and BLAS_LIBS.
139 ## One could use
140 ##   precious_configure_vars='~/src/R/configure --help \
141 ##   | sed -n '/^Some influential/,/^[\ ]/p' \
142 ##   | sed '/^[^ ]/d' \
143 ##   | sed 's/^ //' \
144 ##   | cut -f1 -d ' ' \
145 ##   | grep -v '^MAIN_' \
146 ##   | grep -v '^R_' \
147 ##   | sort \
148 ##   | uniq'
149 ## to obtain the configure vars and hence create most of the above usage
150 ## info as well as the list of accepted variables below automatically.
151 ## </NOTE>
152 if test $# = 0; then  echo "${usage}";  exit 1; fi
153 git status
154 cd Topological_score_TopS/
155 git init
156 git remote add origin https://github.com/WashburnLab/Topological-score-TopS-.git
157 ls
158 git push
159 git push -u origin master
160 git remote -v
161 ls
162 git push origin master
163 git commit -m "First commit"
164 git add .
165 git commit -m "First commit"
166 git remote add origin https://github.com/WashburnLab/Topological-score-TopS-.git
167 git push origin master
168 history









ms@DD172TG51M2 MINGW64 /h/Topological_score_TopS (master)
$ 7~^C

ms@DD172TG51M2 MINGW64 /h/Topological_score_TopS (master)
$ |
```

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
```


 master ▾  1 branch  0 tags

 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

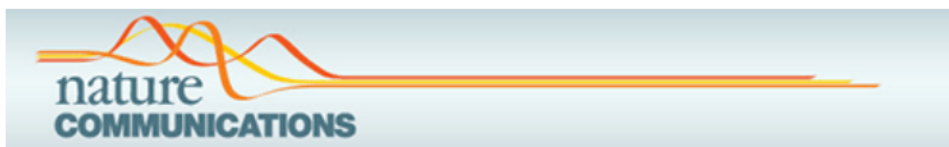
No packages published



Try out [PMC Labs](#) and tell us what you think. [Learn More.](#)



Journal List > Nat Commun > PMC6408525



[Nat Commun.](#) 2019; 10: 1118.

PMCID: PMC6408525

Published online 2019 Mar 8. doi: [10.1038/s41467-019-09123-y](https://doi.org/10.1038/s41467-019-09123-y)

PMID: [30850613](https://pubmed.ncbi.nlm.nih.gov/30850613/)

Topological scoring of protein interaction networks

[Mihaela E. Sardiū](#),¹ [Joshua M. Gilmore](#),^{1,3} [Brad D. Groppē](#),^{1,4} [Arnob Dutta](#),^{1,5} [Laurence Florens](#),¹ and [Michael P. Washburn](#)^{1,2}

► [Author information](#) ► [Article notes](#) ► [Copyright and License information](#) [Disclaimer](#)

This article has been [cited by](#) other articles in PMC.

Associated Data

► [Supplementary Materials](#)

Formats:

[Article](#) | [PubReader](#) | [ePub \(beta\)](#) | [PDF \(1.7M\)](#) | [Cite](#)

Share

[Facebook](#) [Twitter](#) [Google+](#)

Save items

☆ Add to Favorites

Similar articles in PubMed

Conserved abundance and topological features in chromatin-remodeling protein interaction networks. [EMBO Rep. 2015]

MCL-CAw: a refinement of MCL for detecting yeast complexes from weighted PPI networks by incorpor [BMC Bioinformatics. 2010]

Predicting physical interactions between protein complexes. [Mol Cell Proteomics. 2013]

Identifying protein complexes and functional modules--from static PPI networks to dynamic PPI networks. [Brief Bioinform. 2014]

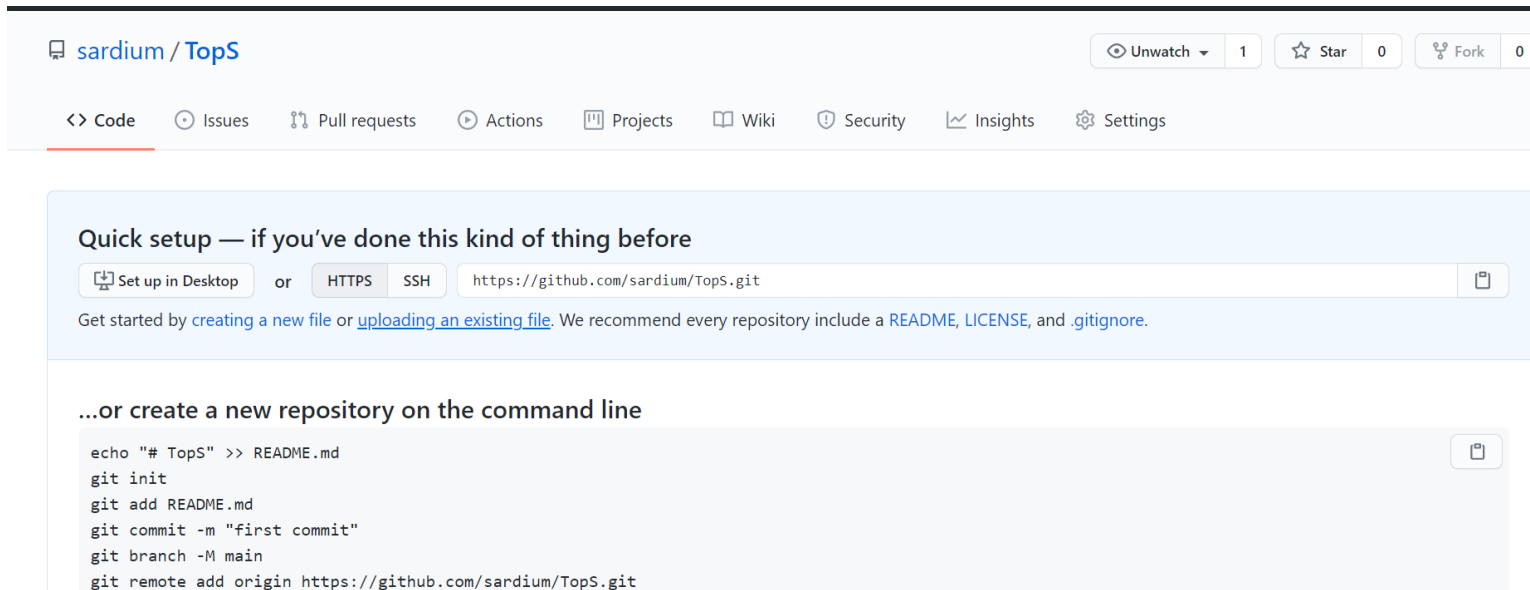
A survey of computational methods for protein complex prediction from protein interaction networks. [J. Bioinform. Comput. Biol. 2013]

- 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.



Bioinformatics

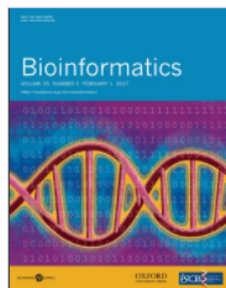


Issues Advance articles Submit ▼ Purchase Alerts About ▼

All Bioinformatics ▼



Advanced
Search



Volume 33, Issue 3
1 February 2017

Article Contents

Abstract

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** ▼



PDF

Split View

Cite

Permissions

Share ▼

Abstract

Google Fiber

2 Gig internet
has arrived.

The fastest internet
in your area is here.

2000 Mbps

Google Fiber
2 Gig internet

Advertisement

START: Shiny Transcriptome Analysis Resource Tool

[Getting Started](#)[Input Data](#)[Filter Data](#)[Group Plots](#)[Analysis Plots](#)[Gene Expression Boxplots](#)[Heatmaps](#)[Instructions](#)[News](#)[Terms & Conditions](#)

Getting Started with START

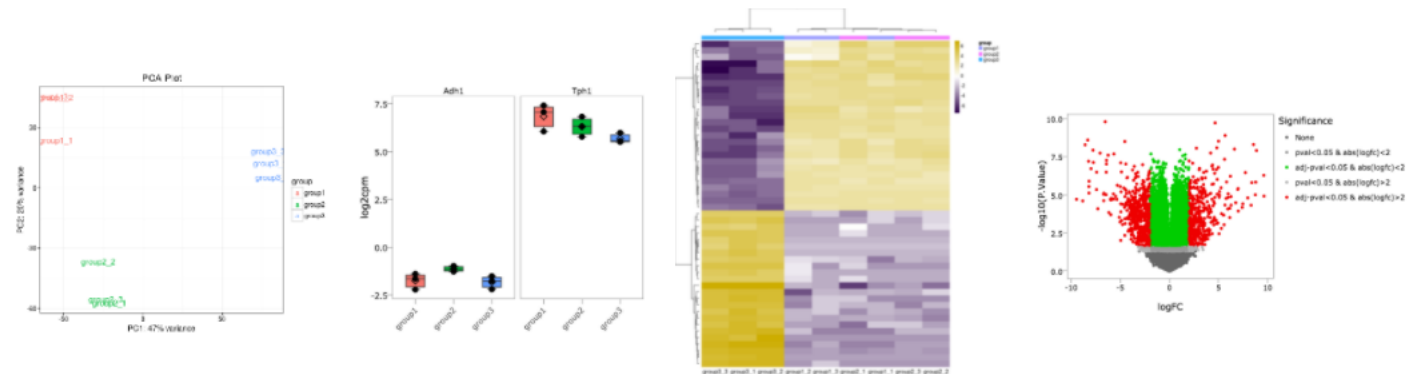
[Features](#)[Data Formats](#)[Save Data for Future Upload](#)[More Help](#)

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: kcv.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 ▾

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 ▾

About

The START App: R Shiny Transcriptome Analysis Resource Tool

 kcv.shinyapps.io/start/

visualization

bioinformatics

rna-seq

r


shiny

transcriptome

shiny-apps


genome-biology








 Readme

 GPL-3.0 License

Releases 4



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