Class 16

Course Summary
Git & GitHub

Barry Grant

UC San Diego

http://thegrantlab.org

Today’s Menu

- Summary of major learning goals,

- Course discussion and feedback
(https://etherpad.wikimedia.org/p/bimm143_w22)

- Final exam

  ➡ Test structure, guidelines and rules

  ➡ Topics and example questions

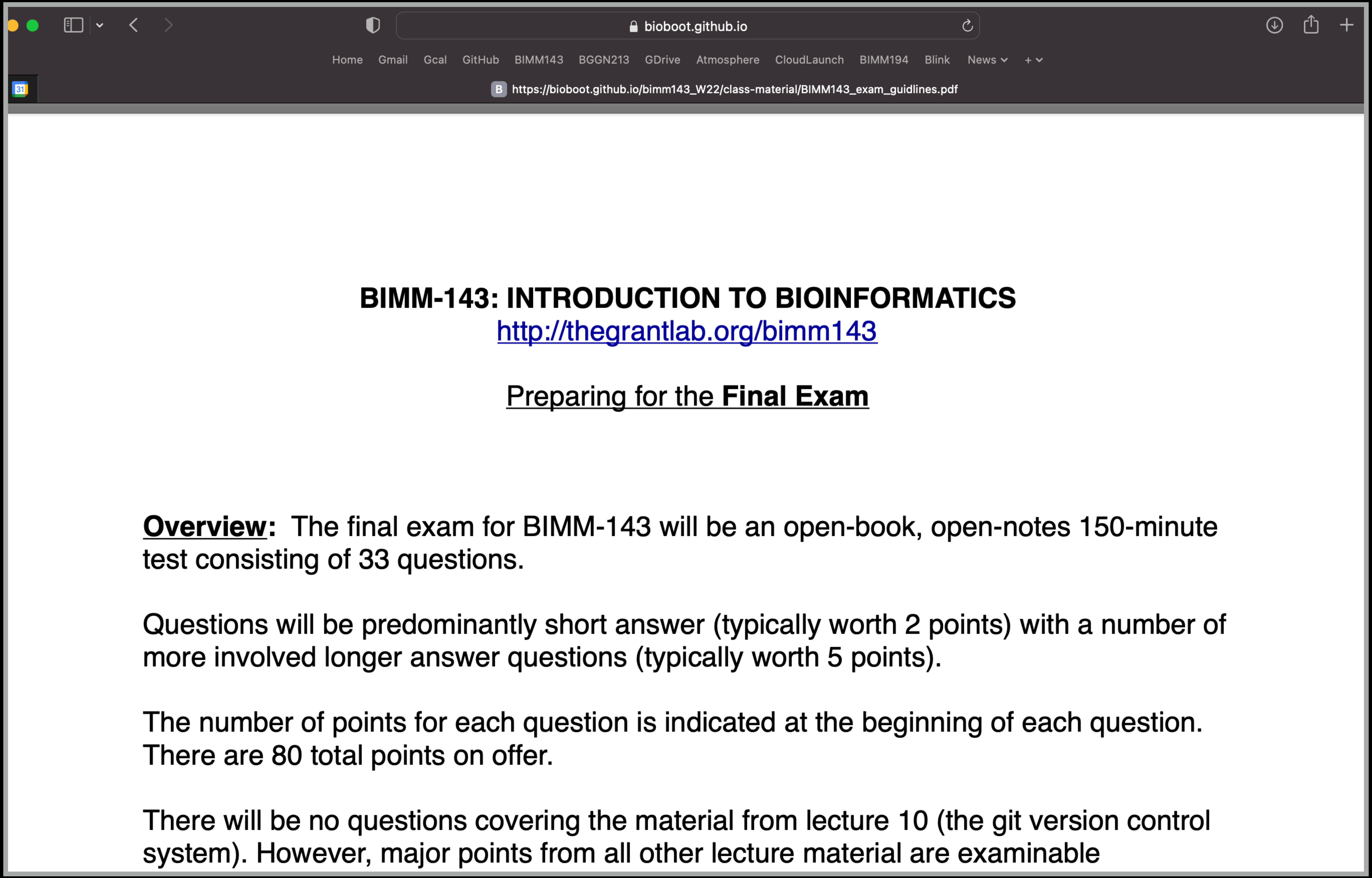
  ➡ Exam preparation, discussion and open study

- Introduction to Git & GitHub (+ website portfolios)

Today’s Menu

- Summary of major learning goals,
- Course discussion and feedback
(https://etherpad.wikimedia.org/p/bimm143_w22)
- Final exam
    - Test structure, guidelines and rules
    - Topics and example questions
    - Exam preparation, discussion and open study

- Introduction to Git & GitHub (+ website portfolios) https://bioboot.github.io/bimm143_W22/class-material/BIMM143_exam_guidlines.pdf



BIMM-143: INTRODUCTION TO BIOINFORMATICS

http://thegrantlab.org/bimm143

Preparing for the **Final Exam**

Overview: The final exam for BIMM-143 will be an open-book, open-notes 150-minute test consisting of 33 questions.

Questions will be predominantly short answer (typically worth 2 points) with a number of more involved longer answer questions (typically worth 5 points).

The number of points for each question is indicated at the beginning of each question. There are 80 total points on offer.

There will be no questions covering the material from lecture 10 (the git version control system). However, major points from all other lecture material are examinable

Q1. Did you enjoy this course? (Rank in relation to others you have experienced at UCSD)?

Q2. Should this course be offered again?

Q3. If so what changes would you recommend for this course? (e.g. more/less DataCamp & Projects)

Q4. Was the course effectively organized (lecture and lab material online vs handout or Canvas site)?

Q5. What advice would you give to another student who is considering taking this course?

Q6. Considering both the limitations and possibilities of the subject matter and the course, how would you rate the overall effectiveness of this course and instructor?

Q7. Do you agree or disagree - The course developed my abilities and skills for the subject?

Q8. On average, how many hours per week have you spent on this course, including attending classes, doing homework's and assignments?

Q9. Any other comments you would like to share?

EtherPad Link:
https://etherpad.wikimedia.org/p/bimm143_w22

Thank you very much!

Announcement!

Bonus:

Bioinformatics & Genomics in Industry

Live Stram Video

Enjoy a set of short open ended guest lectures from leading genomic scientists at **Illumina Inc.**, **Synthetic Genomics Inc.**, and the **La Jolla Institute for Allergy and Immunology**. Feel free to contact these scientists for networking and to have your questions about industry careers in Bioinformatics and Genomics answered.

Bonus:

Introduction to Git & GitHub

What is Git?

(1) An unpleasant or contemptible person. Often incompetent, annoying, senile, elderly or childish in character.

(2) A modern distributed version control system with an emphasis on speed and data integrity.

git

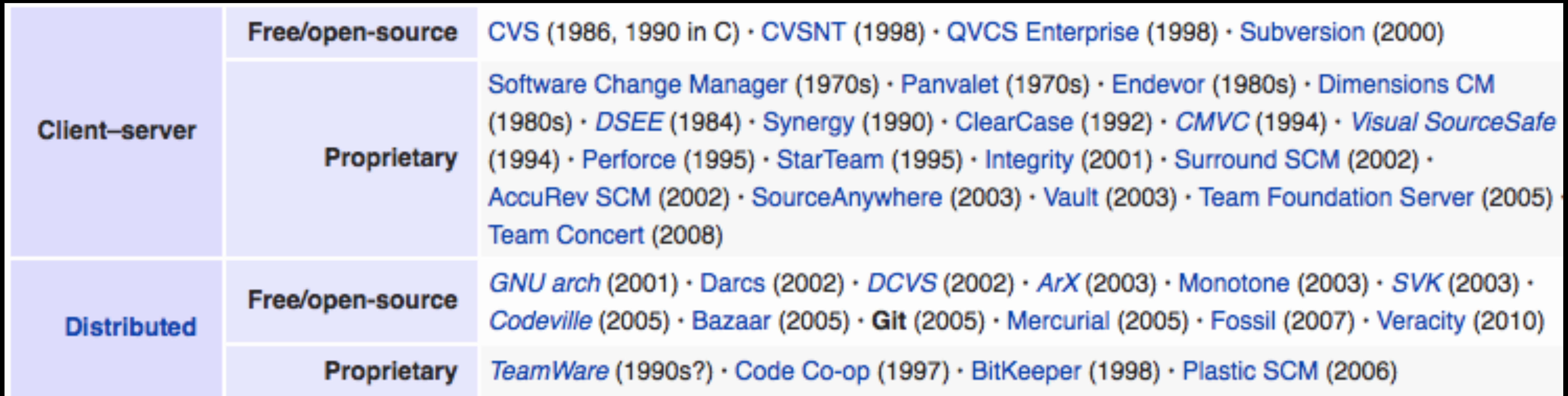
What is Git?

(1) An unpleasant or contemptible person. Often incompetent, annoying, senile, elderly or childish in character.

(2) A modern distributed version control system with an emphasis on speed and data integrity.

Version Control

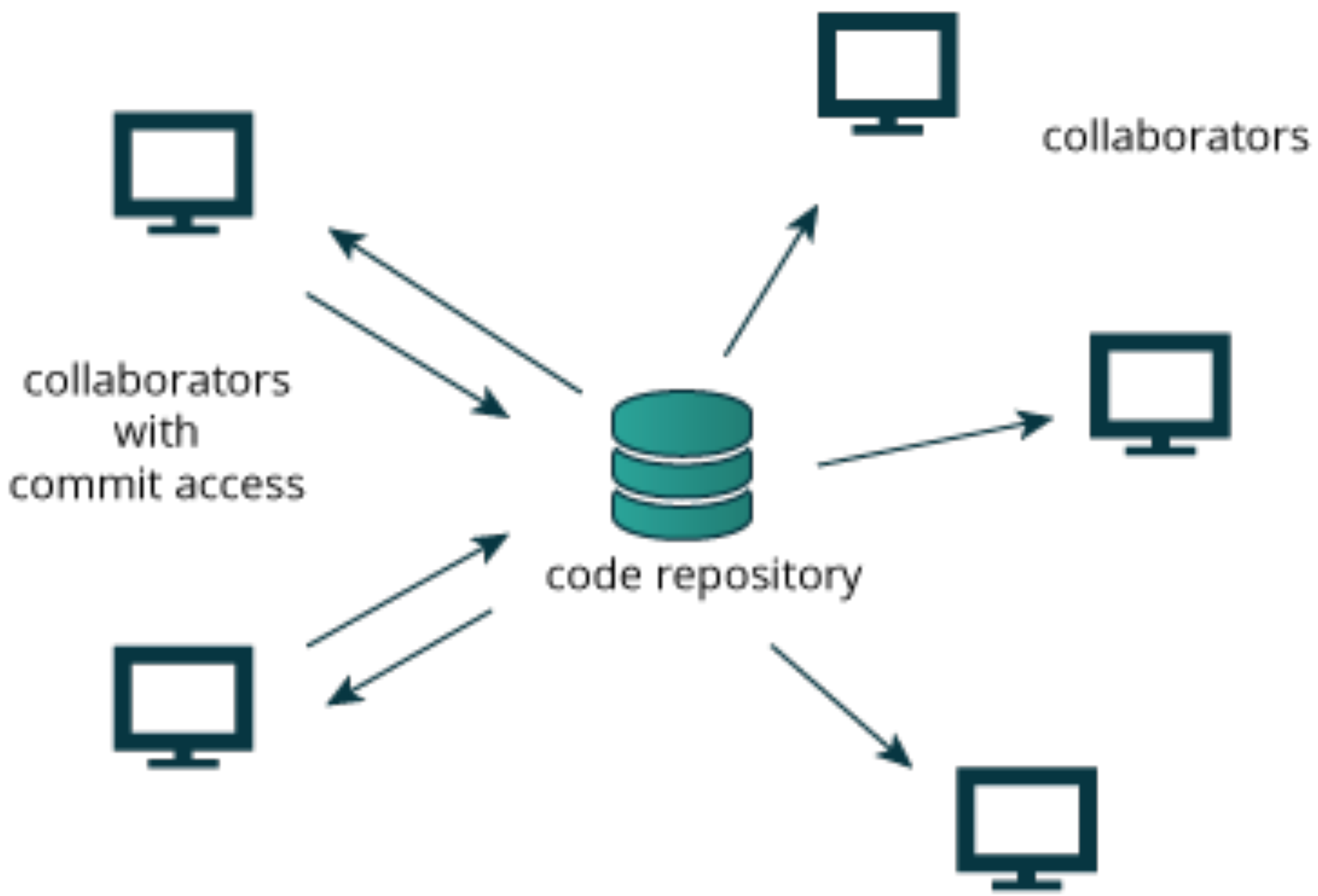
Version control systems (VCS) record changes to a file or set of files over time so that you can recall specific versions later.



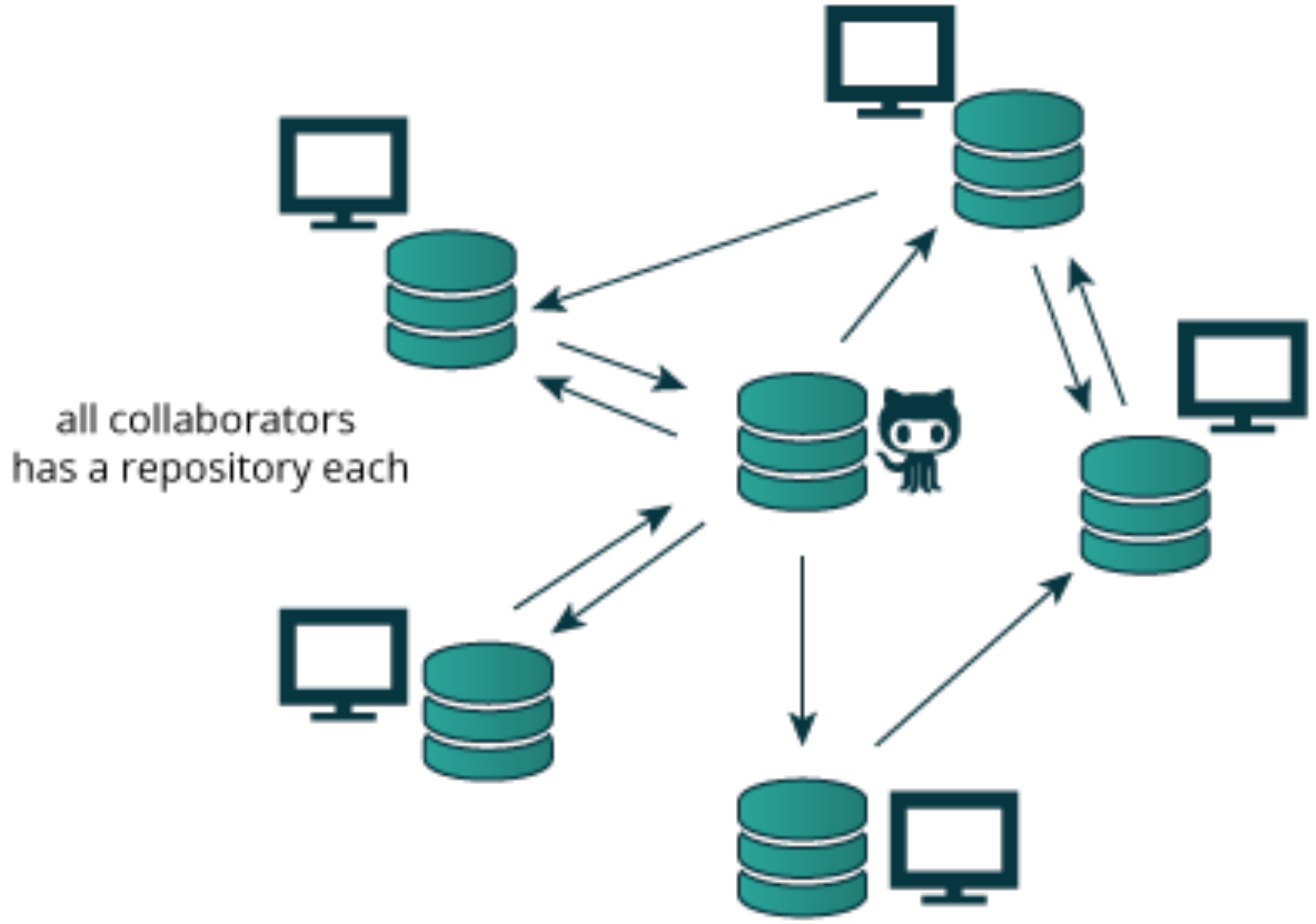
Client–server	Free/open-source	CVS (1986, 1990 in C) · CVSNT (1998) · QVCS Enterprise (1998) · Subversion (2000)
	Proprietary	Software Change Manager (1970s) · Panvalet (1970s) · Endevor (1980s) · Dimensions CM (1980s) · *DSEE* (1984) · Synergy (1990) · ClearCase (1992) · *CMVC* (1994) · *Visual SourceSafe* (1994) · Perforce (1995) · StarTeam (1995) · Integrity (2001) · Surround SCM (2002) · AccuRev SCM (2002) · SourceAnywhere (2003) · Vault (2003) · Team Foundation Server (2005) · Team Concert (2008)
Distributed	Free/open-source	*GNU arch* (2001) · Darcs (2002) · *DCVS* (2002) · *ArX* (2003) · Monotone (2003) · *SVK* (2003) · *Codeville* (2005) · Bazaar (2005) · **Git** (2005) · Mercurial (2005) · Fossil (2007) · Veracity (2010)
	Proprietary	*TeamWare* (1990s?) · Code Co-op (1997) · BitKeeper (1998) · Plastic SCM (2006)

There are many VCS available, see: https://en.wikipedia.org/wiki/Revision_control

Client-Server *vs* Distributed VCS

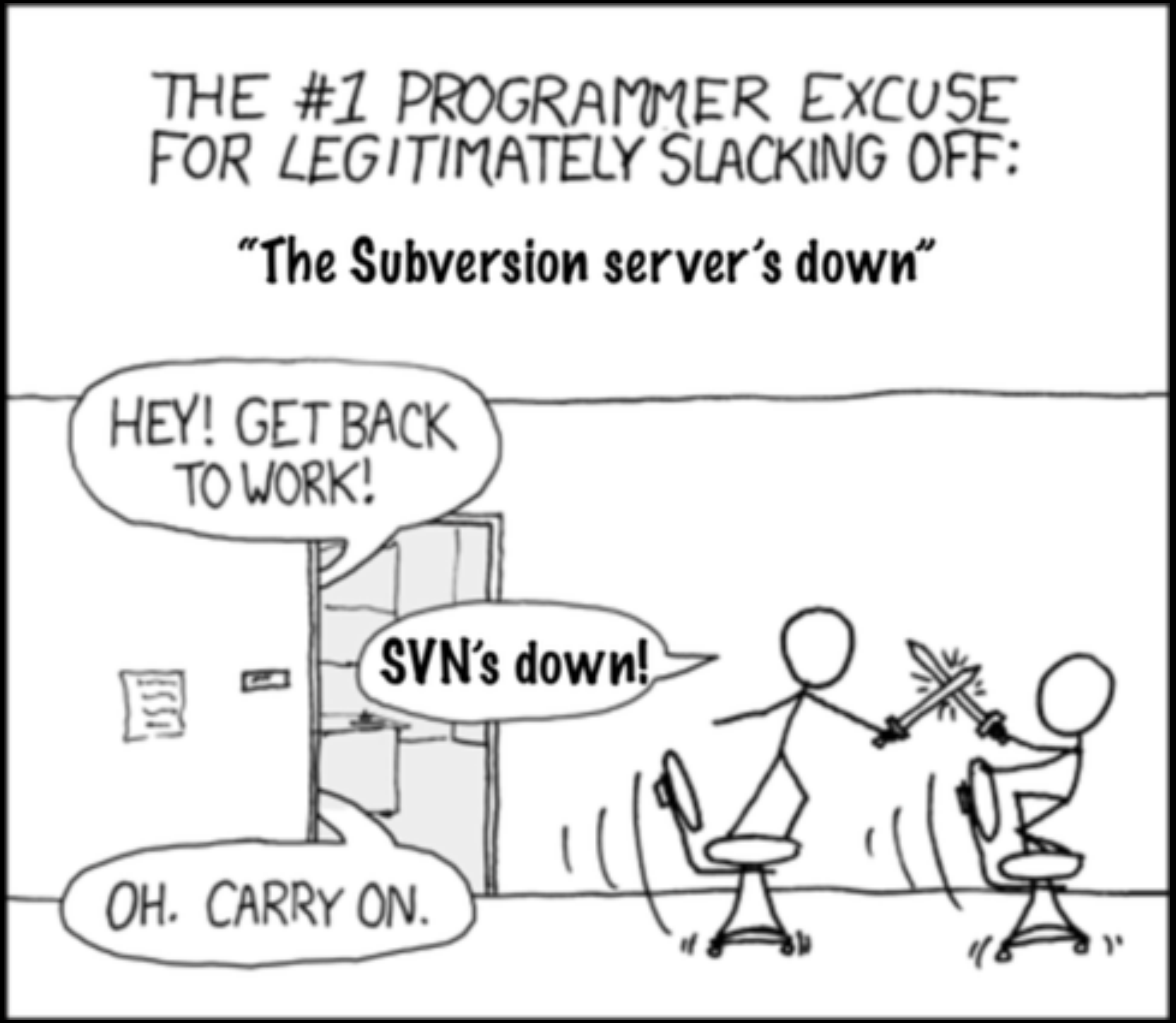


Client-server approach



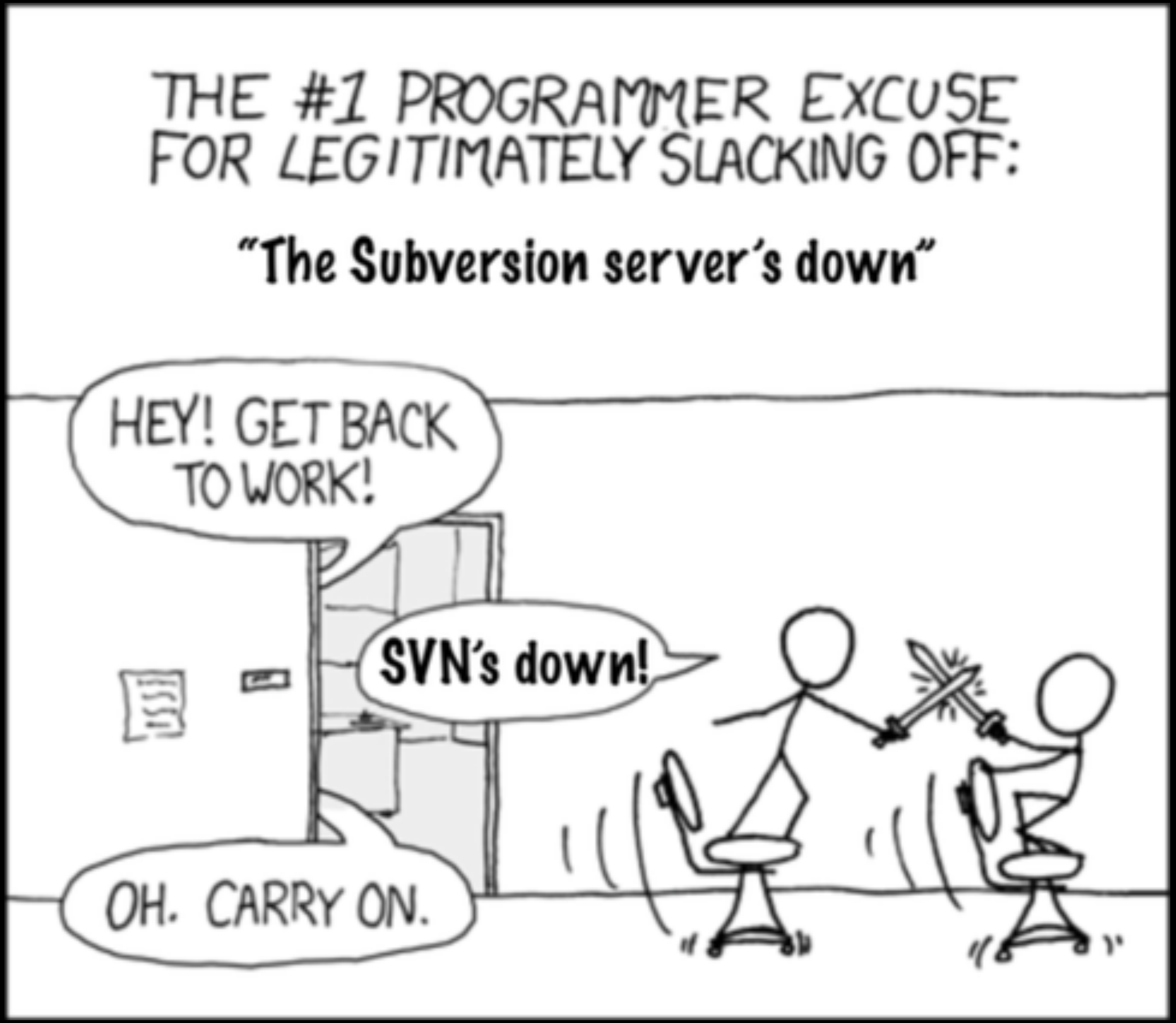
Distributed approach

Distributed version control systems (DCVS) allows multiple people to work on a given project without requiring them to share a common network.


THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:
"The Subversion server's down"
HEY! GET BACK TO WORK!
SVN's down!
OH. CARRY ON.

http://tinyurl.com/distributed-advantages

Git is now the most popular free VCS!


THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:
"The Subversion server's down"
HEY! GET BACK TO WORK!
SVN's down!
OH. CARRY ON.

Git offers:

- Speed
- Backups
- Off-line access
- Small footprint
- Simplicity*
- Social coding

http://tinyurl.com/distributed-advantages

Where did Git come from?

Written initially by Linus Torvalds to support Linux kernel and OS development.

Meant to be distributed, fast and more natural.

Capable of handling large projects.

Now the most popular free VCS!

Why use Git?

Q. Would you write your lab book in pencil, then erase and overwrite it every day with new content?

Q. Would you write your lab book in pencil, then erase and overwrite it every day with new content?

Version control is the lab notebook of the digital world: it’s what professionals use to keep track of what they’ve done and to collaborate with others.

Why use Git?

- Provides ‘snapshots’ of your project during development and provides a full record of project history.
- Allows you to easily reproduce and rollback to past versions of analysis and compare differences. (N.B. Helps fix software regression bugs!)
- Keeps track of changes to code you use from others such as fixed bugs & new features
- Provides a mechanism for sharing, updating and collaborating (like a social network)
- Helps keep your work and software organized and available

Obtaining Git

Note: You hopefully already have git installed!

To check open the "**Terminal**" tab in RStudio and type:

①

```
which git
```

②

```
git --version
```

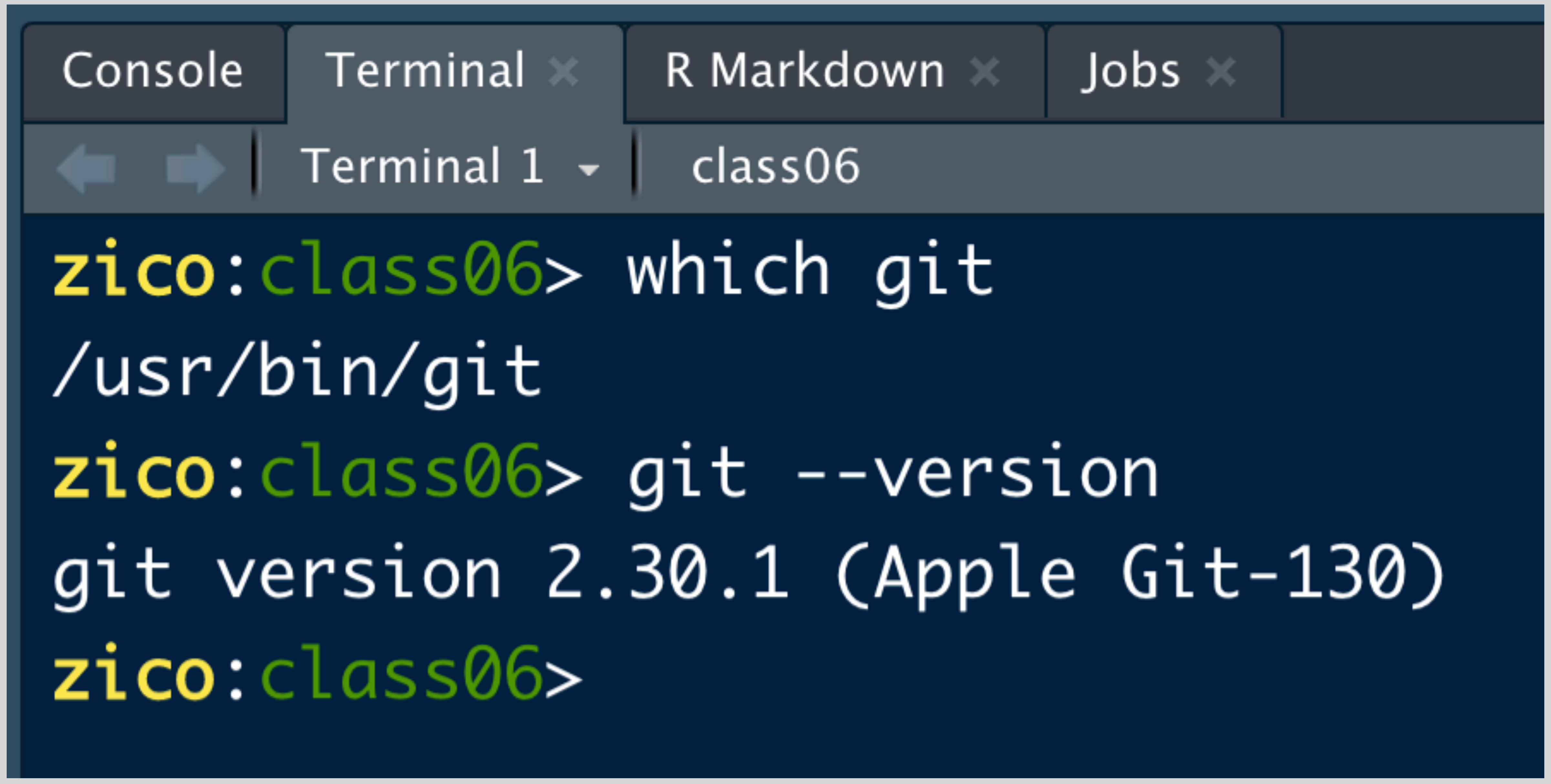
Obtaining Git

Note: You might already have git installed
To check open the "**Terminal**" tab in RStudio and type:

1 `which git`

2 `git --version`

Obtaining Git



```
zico:class06> which git
/usr/bin/git
zico:class06> git --version
git version 2.30.1 (Apple Git-130)
zico:class06>
```

Note: You might already have git installed
To check open the "**Terminal**" tab in RStudio and type:

```
1 which git
2 git --version
```

Obtaining Git

Windows Only (if you have problems)

If the "which git" command did not work, try:

```
where git
```

If this works see next slide. If not then you need to install GitBash, instructions here:

Class Computer Setup Page

Mac Only (if you have problems)

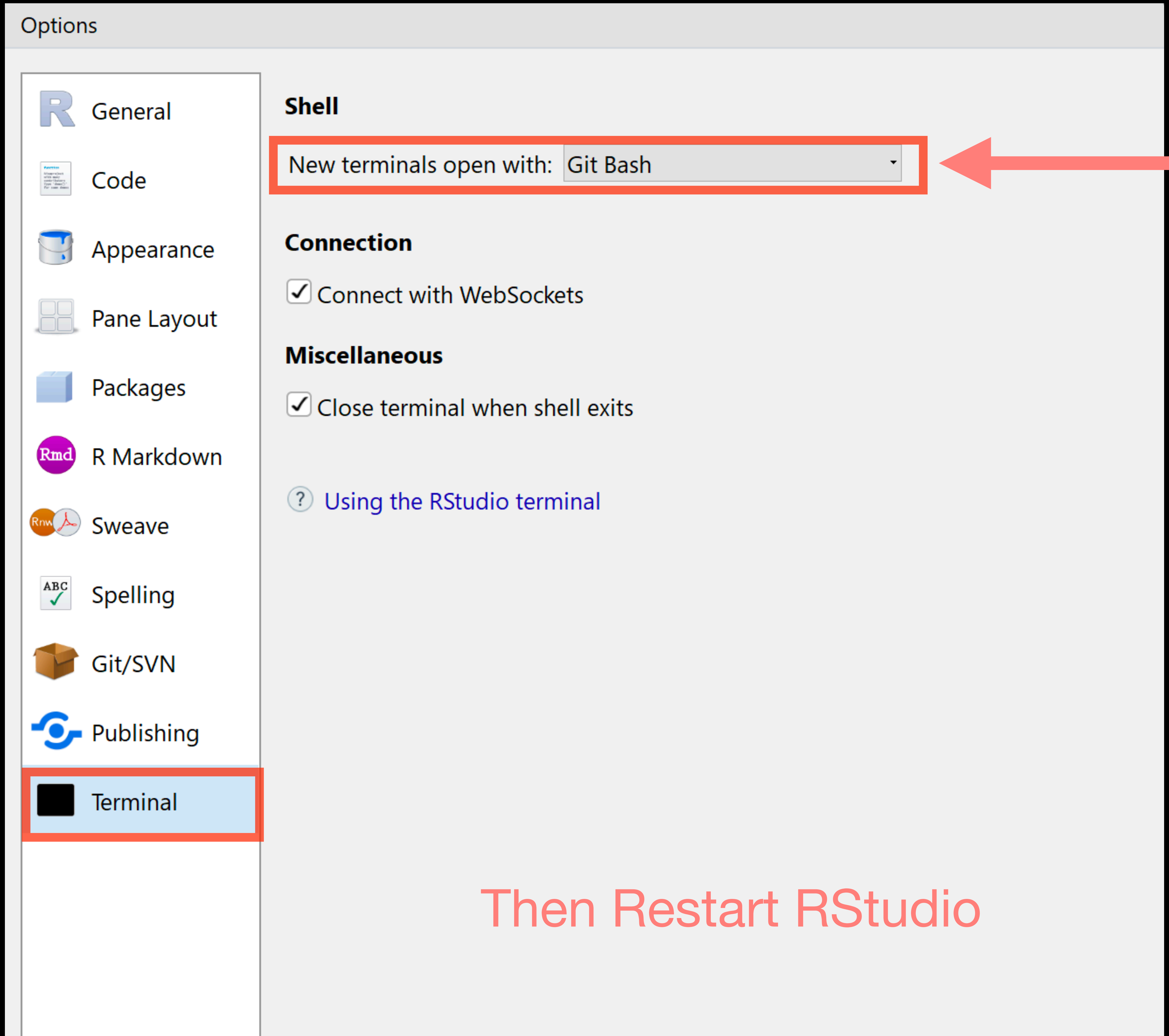
If the "which git" command did not work, you may need to install select developer tools.

In your Terminal type:

```
xcode-select --install
```

On a PC Only!

Go to: RStudio > Tools > Global Options > Terminal



Shell

New terminals open with: Git Bash

Connection

☑ Connect with WebSockets

Miscellaneous

☑ Close terminal when shell exits

? Using the RStudio terminal

Then Restart RStudio

Make sure Git Bash is selected!

Do it Yourself!

Note: You might already have git installed
To check open the "**Terminal**" tab in RStudio and type:

1
```
which git
```

2
```
git --version
```

Installing Git

Windows (if you have problems)
Follow the GitBash instructions here:
Class Computer Setup Page

Mac (if you have problems)
In the **Terminal** instal select developer tools

```
xcode-select --install
```

Configuring Git

Do it Yourself!

Configuring Git

(RStudio **Terminal** Tab)

(…or *RStudio* **>** *Tools* **>** *Shell*)

```
# First tell Git who you are
> git config --global user.name  “Barry Grant”
> git config --global user.email  “bjgrant@ucsd.edu”
```

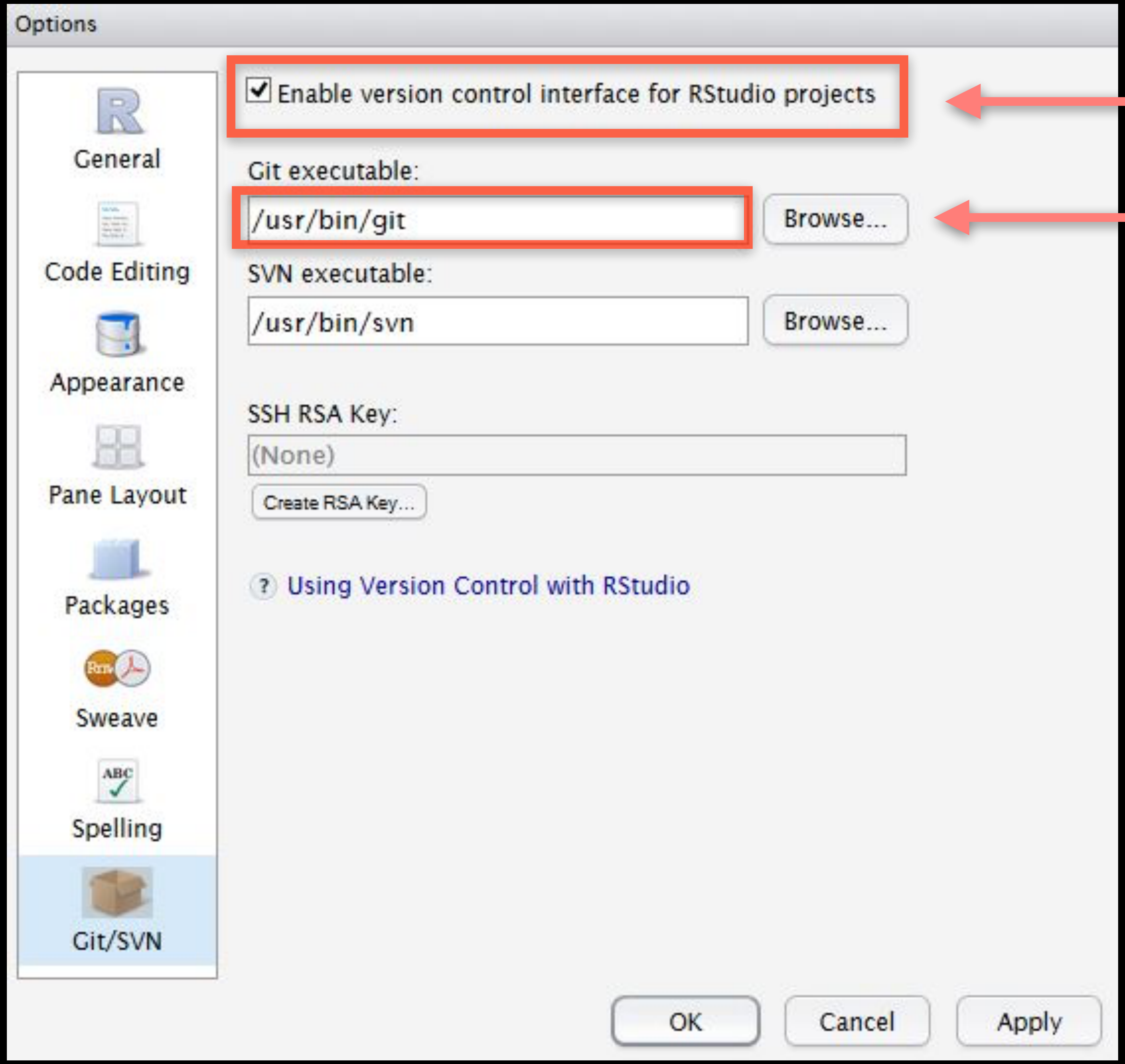
Configuring RStudio

Do it Yourself!

For Mac & Linux

(PC on next slide)

Go to: **RStudio > Tools > Global Options > Git/SVN**



☑ Enable version control interface for RStudio projects

Git executable: /usr/bin/git

SVN executable: /usr/bin/svn

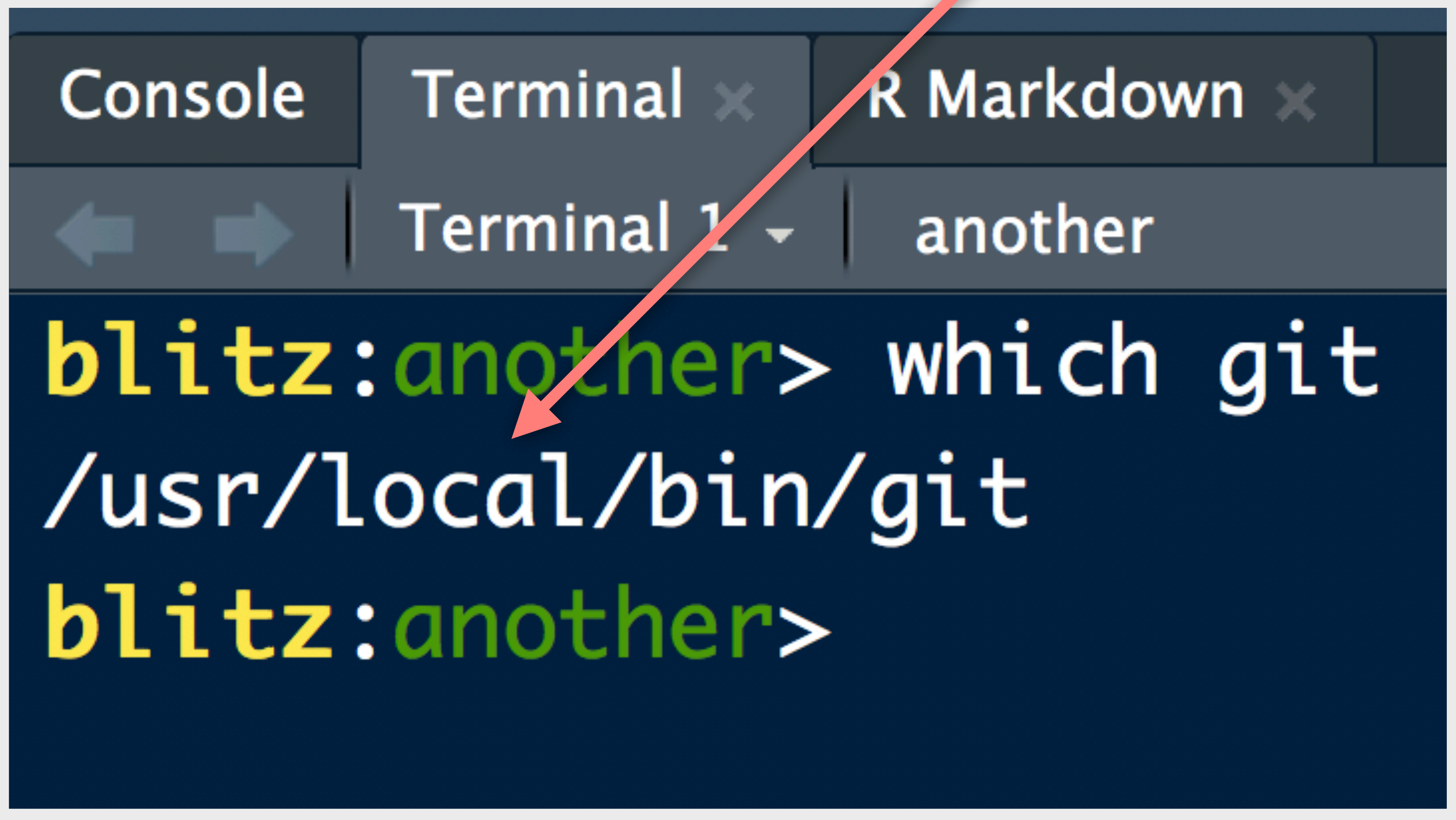
SSH RSA Key: (None)

Using Version Control with RStudio

1 *Make sure this is **ticked**!*

2 Make sure this is **correct**!

Check in your RStudio “Terminal” tab:

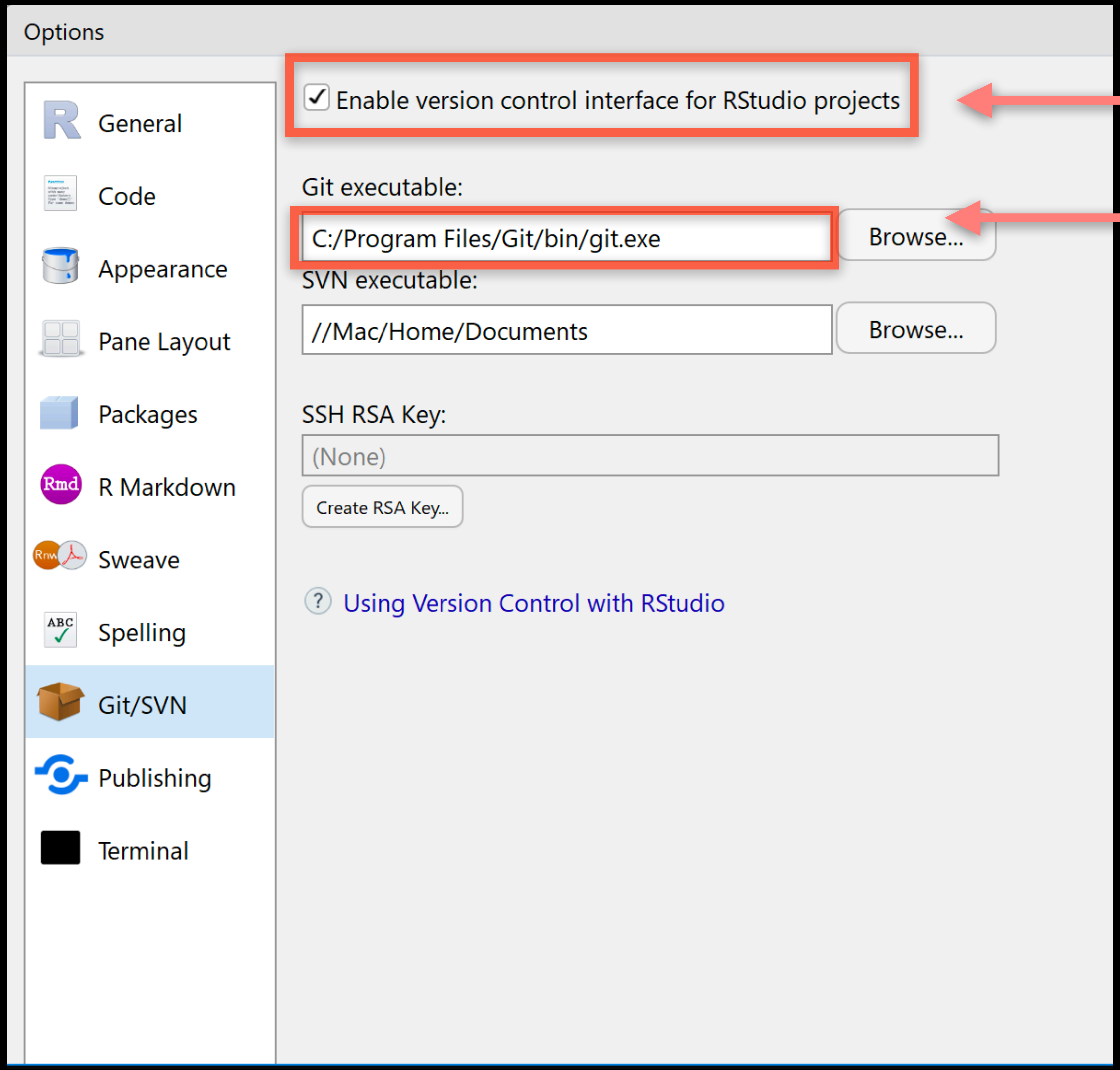


```
blitz:another> which git
/usr/local/bin/git
blitz:another>
```

Do it Yourself!

On a PC!

Go to: RStudio > Tools > Global Options > Git/SVN



☑ Enable version control interface for RStudio projects

Git executable: C:/Program Files/Git/bin/git.exe

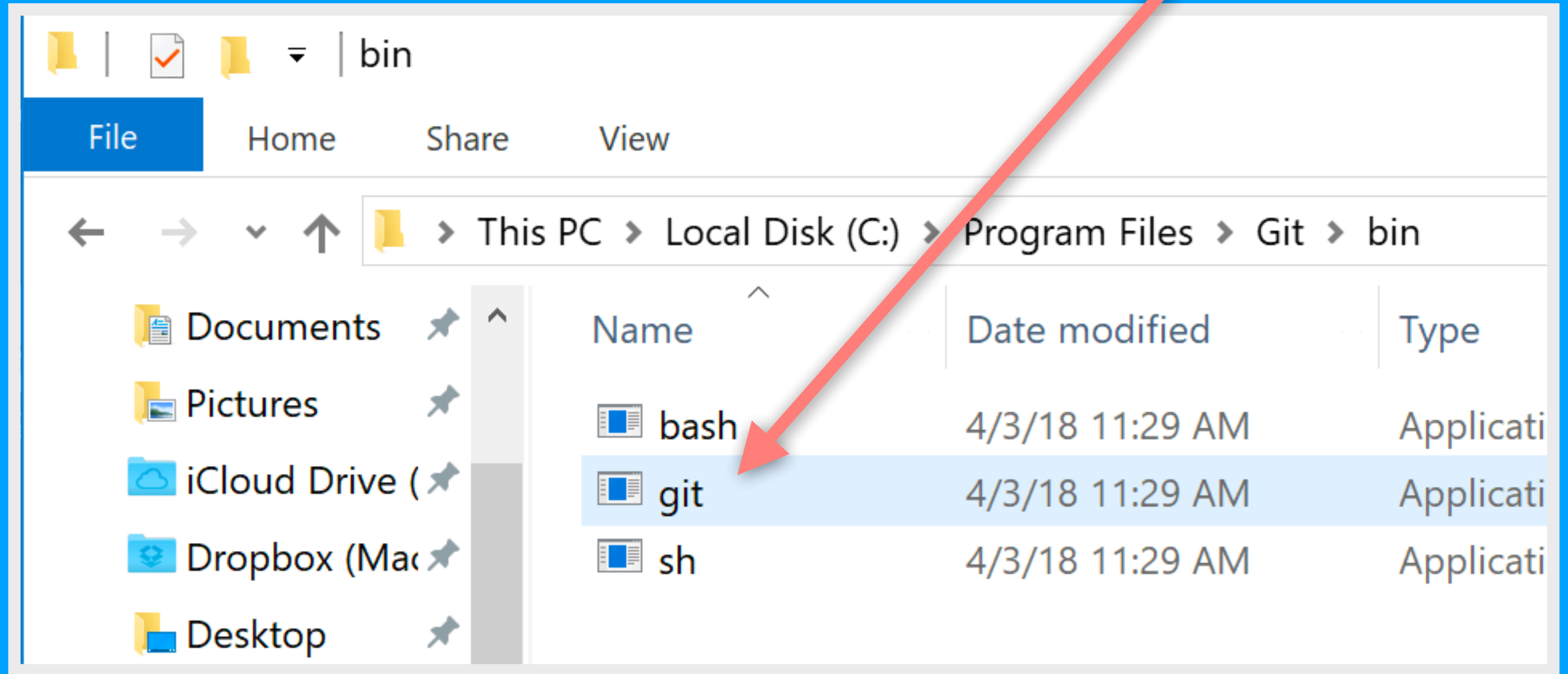
SVN executable: //Mac/Home/Documents

SSH RSA Key: (None)

1 *Make sure this is* ***ticked****!*

2 This is the PATH for **PC**!

Check in your Windows File Explorer:



Name	Date modified	Type
bash	4/3/18 11:29 AM	Applicati
git	4/3/18 11:29 AM	Applicati
sh	4/3/18 11:29 AM	Applicati

Restart RStudio!

Using Git

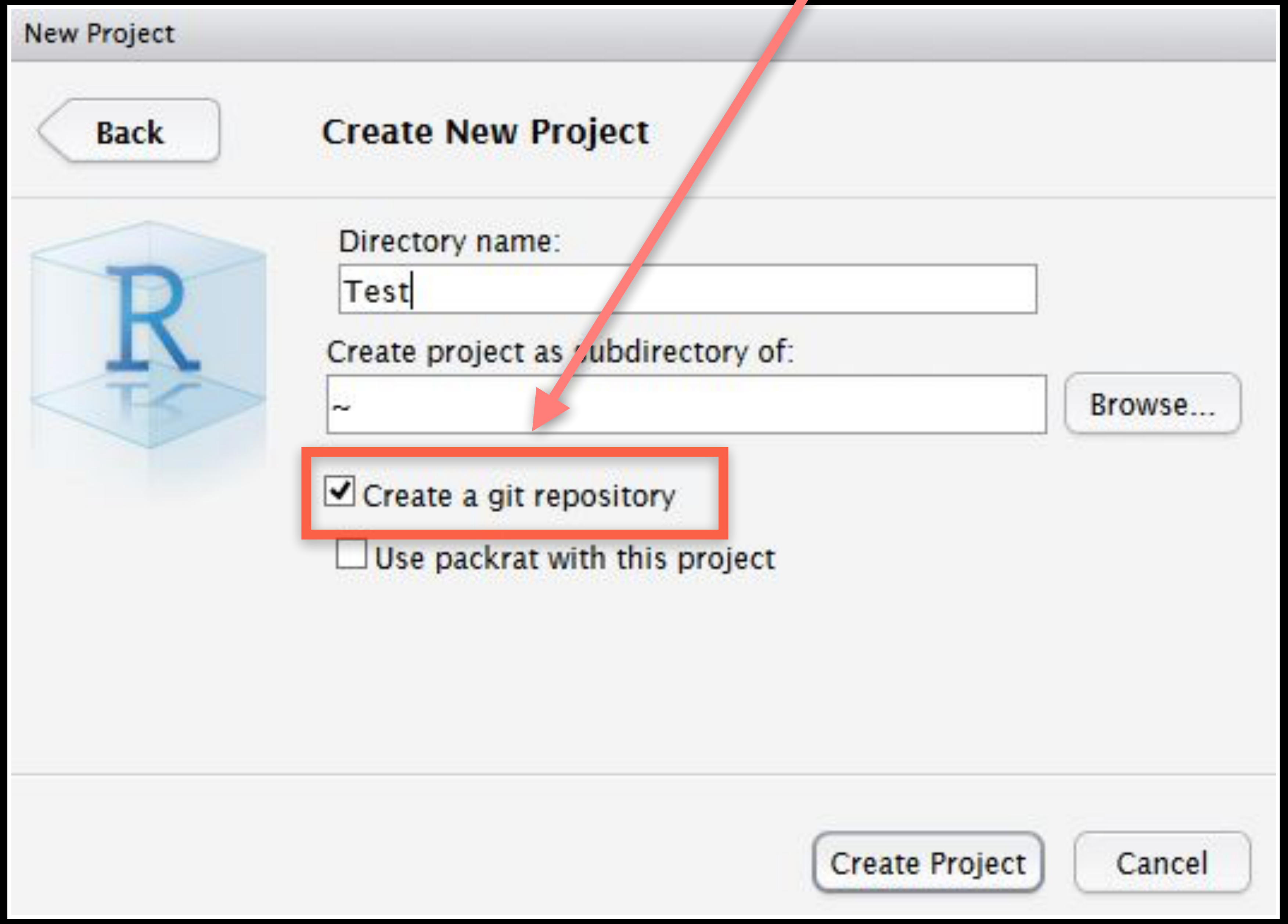
Using Git

1. Initiate a Git repository.
2. Edit content (i.e. change some files).
3. Store a ‘snapshot’ of the current file state.*

Create a new **Test** RStudio project

Do it Yourself!

1 New option to create a Git repository…



New Project

Back

Create New Project

Directory name:

Test

Create project as subdirectory of:

~

Browse...

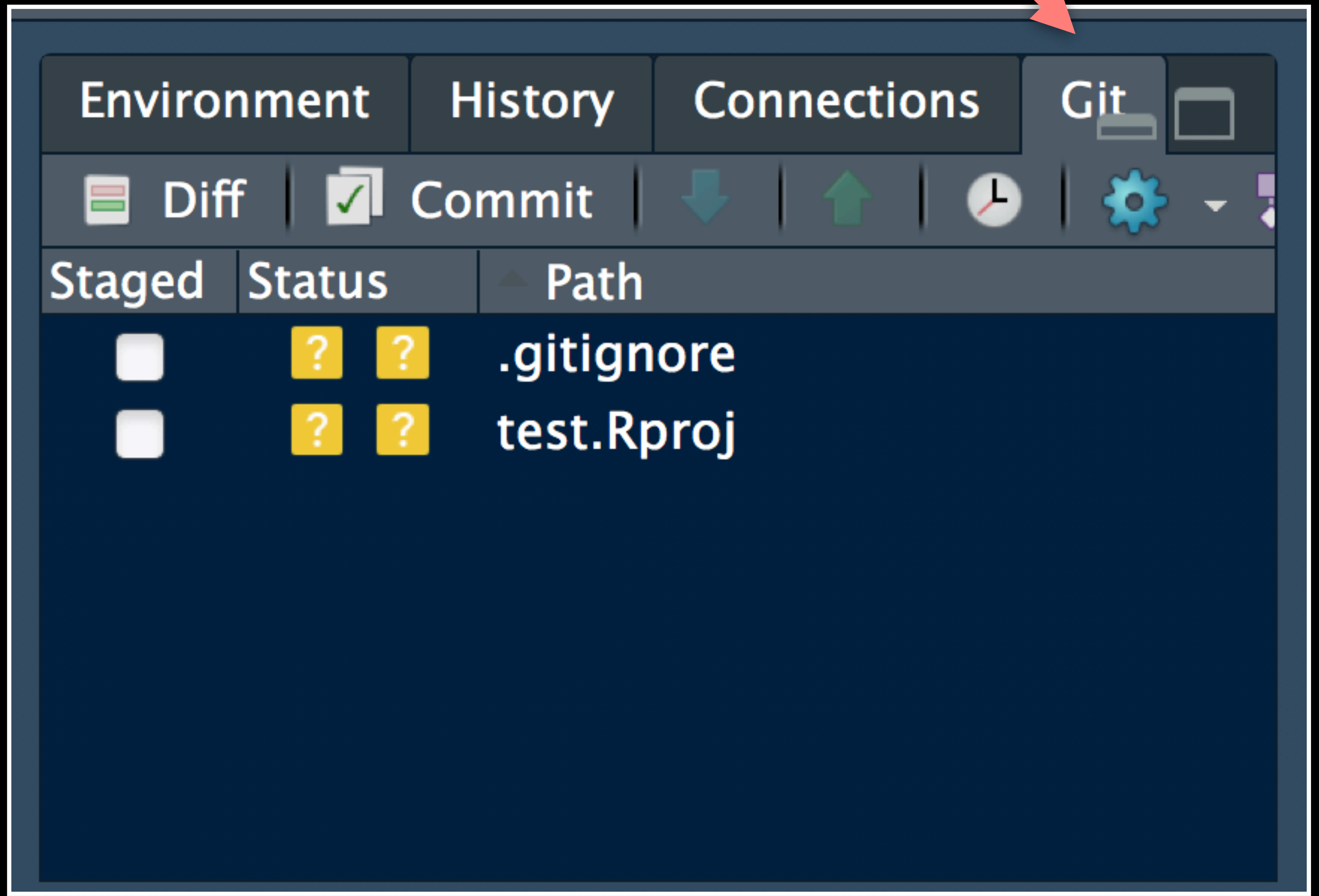
☑ Create a git repository

□ Use packrat with this project

Create Project

Cancel

2 New Git tab…



Staged	Status	Path
□	? ?	.gitignore
□	? ?	test.Rproj

Check if new Git options appear in RStudio?

Using Git in RStudio

1. **Initiate** a git repository for an RStudio Project

2. Do your work and edit content as normal

3. Store a ‘snapshot’ of the current file state
   (a) Periodically **add** important files to git “Staging Area”
   (b) **Commit** changes to your “git repository”

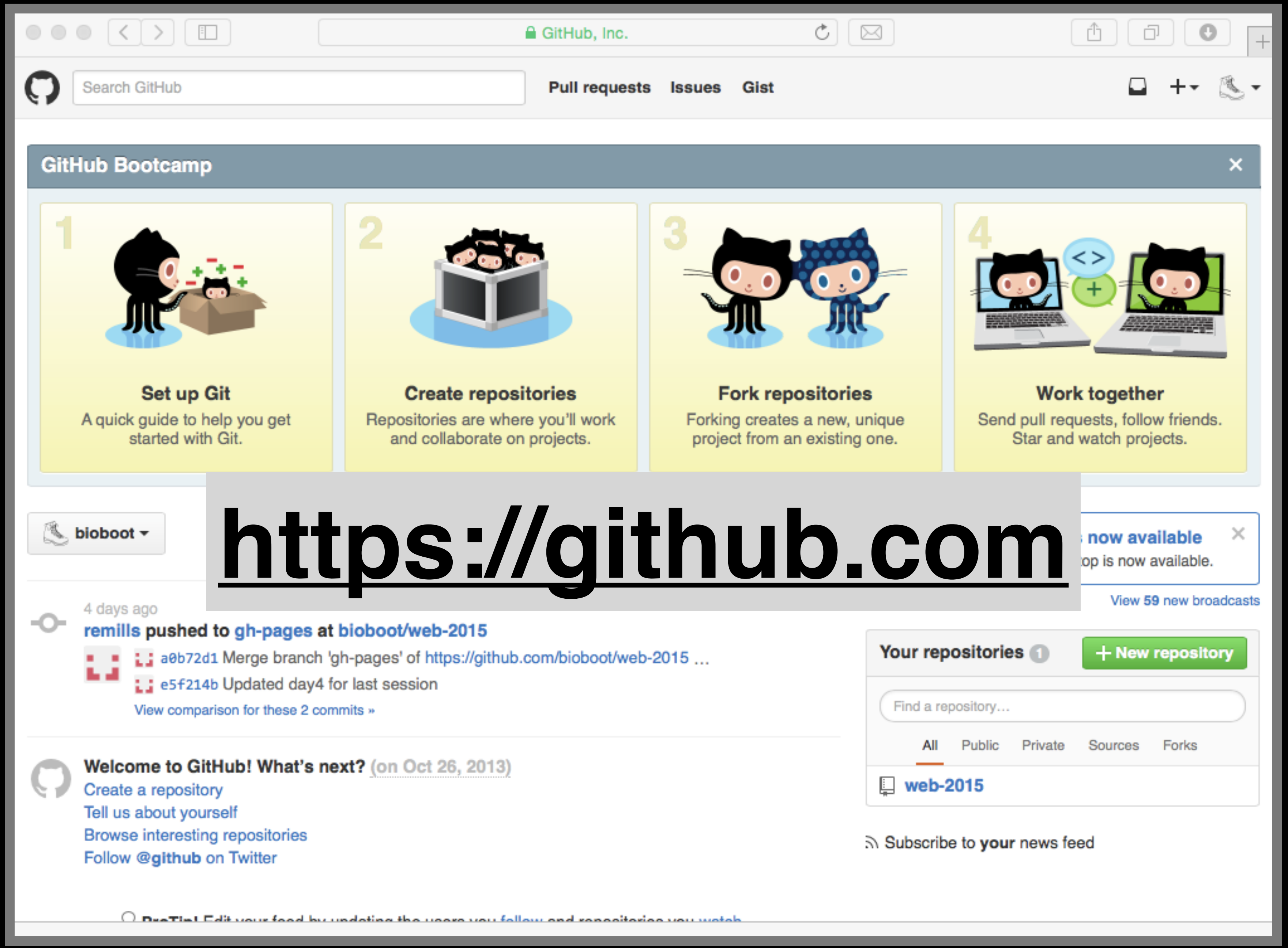
Rinse and repeat….

Demo:

Follow along!

GitHub & Bitbucket

GitHub and **Bitbucket** are two popular hosting services for Git repositories. These services allow you to share your projects and collaborate with others using both '**public**' and '**private**' repositories*.



https://github.com

GitHub Bootcamp

1 Set up Git
A quick guide to help you get started with Git.

2 Create repositories
Repositories are where you'll work and collaborate on projects.

3 Fork repositories
Forking creates a new, unique project from an existing one.

4 Work together
Send pull requests, follow friends. Star and watch projects.

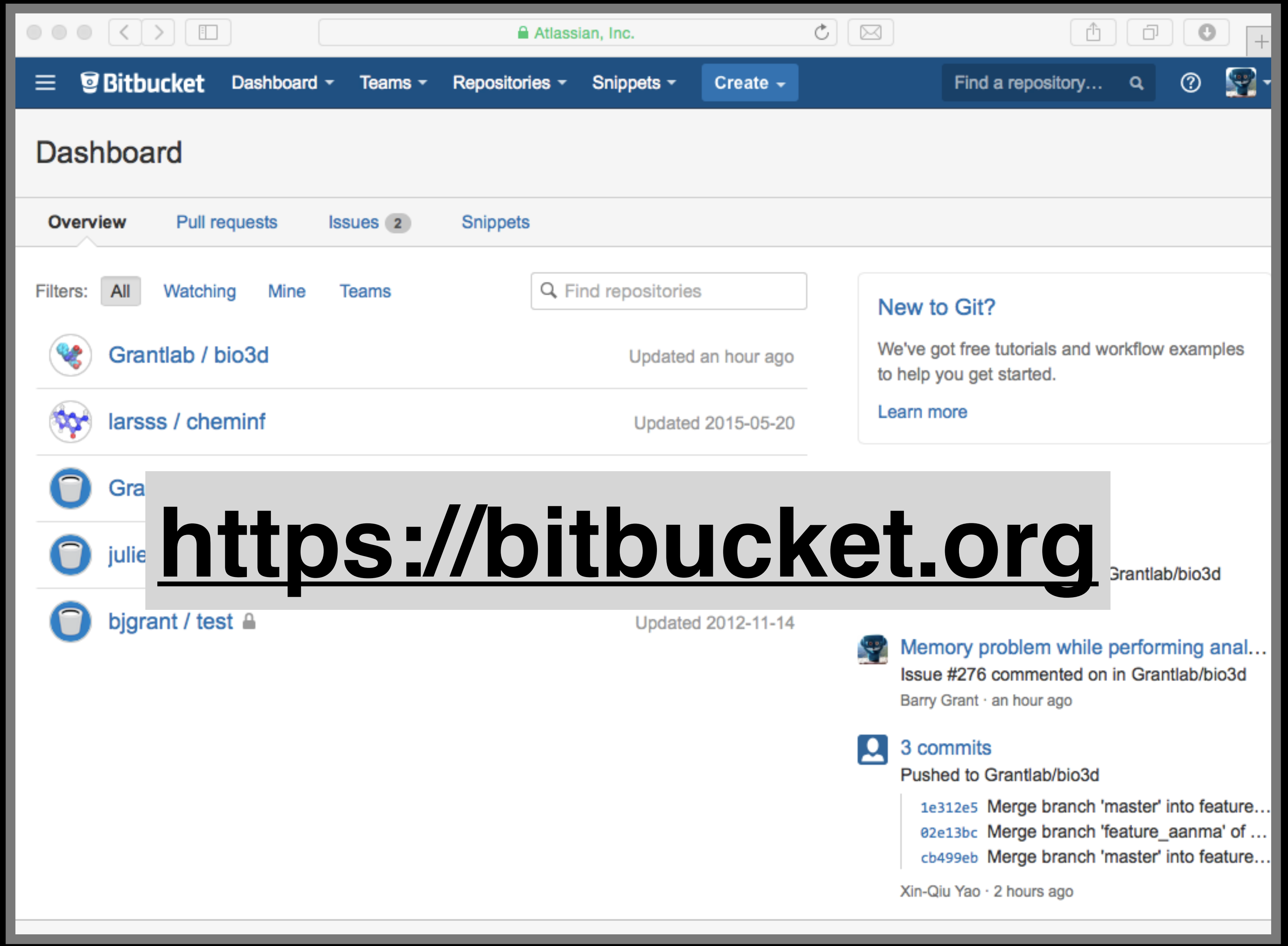
now available
op is now available.
View 59 new broadcasts

4 days ago
remills pushed to gh-pages at bioboot/web-2015
a0b72d1 Merge branch 'gh-pages' of https://github.com/bioboot/web-2015 …
e5f214b Updated day4 for last session
View comparison for these 2 commits »

Welcome to GitHub! What's next? (on Oct 26, 2013)
Create a repository
Tell us about yourself
Browse interesting repositories
Follow @github on Twitter

Your repositories 1
web-2015

Subscribe to your news feed



https://bitbucket.org

Dashboard

Filters: All Watching Mine Teams

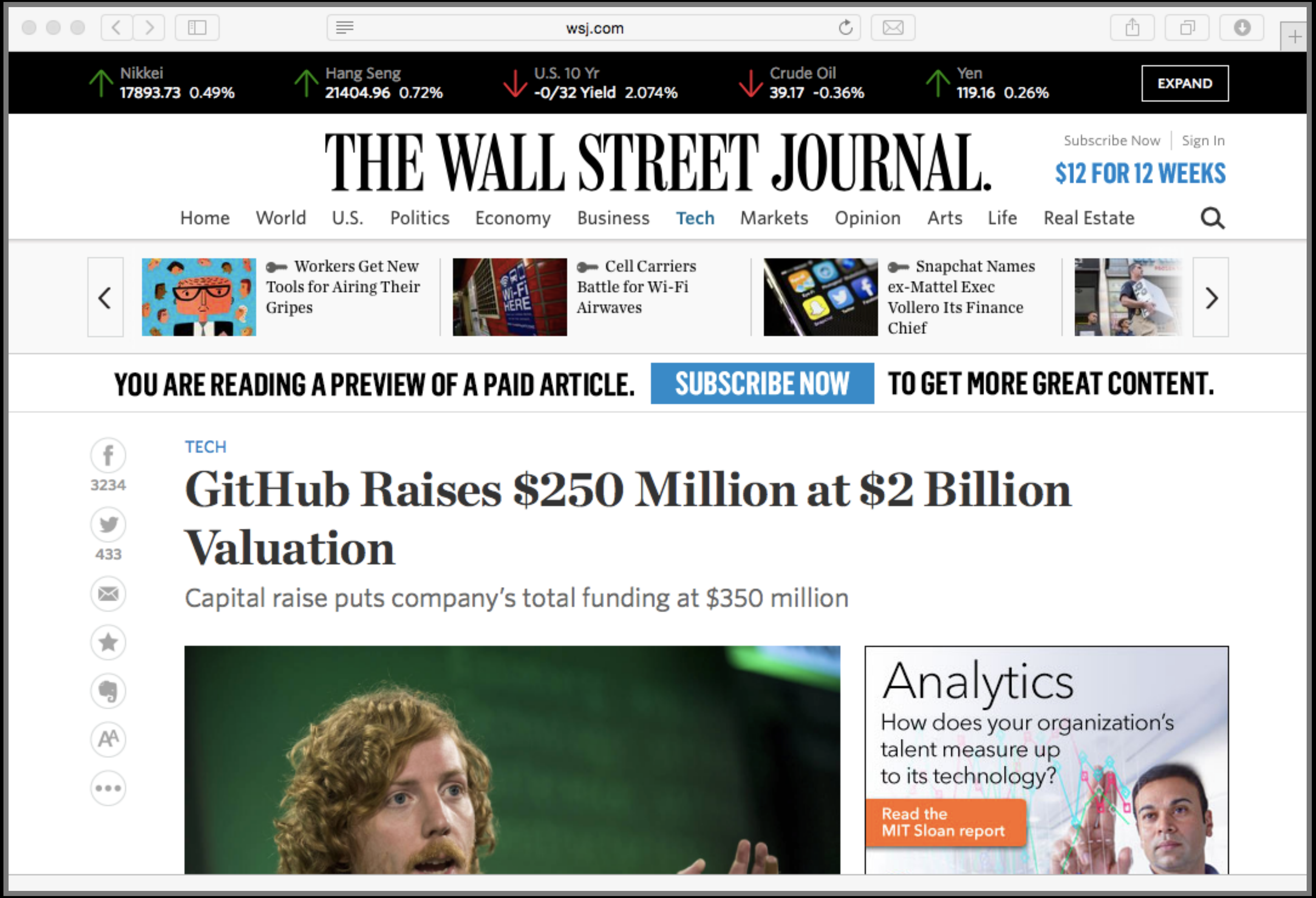
Grantlab / bio3d — Updated an hour ago
larsss / cheminf — Updated 2015-05-20
Gra
julie
bjgrant / test — Updated 2012-11-14

New to Git?
We've got free tutorials and workflow examples to help you get started.
Learn more

Grantlab/bio3d

Memory problem while performing anal…
Issue #276 commented on in Grantlab/bio3d
Barry Grant · an hour ago

3 commits
Pushed to Grantlab/bio3d
1e312e5 Merge branch 'master' into feature…
02e13bc Merge branch 'feature_aanma' of …
cb499eb Merge branch 'master' into feature…
Xin-Qiu Yao · 2 hours ago



Nikkei 17893.73 0.49%
Hang Seng 21404.96 0.72%
U.S. 10 Yr -0/32 Yield 2.074%
Crude Oil 39.17 -0.36%
Yen 119.16 0.26%
EXPAND

THE WALL STREET JOURNAL.

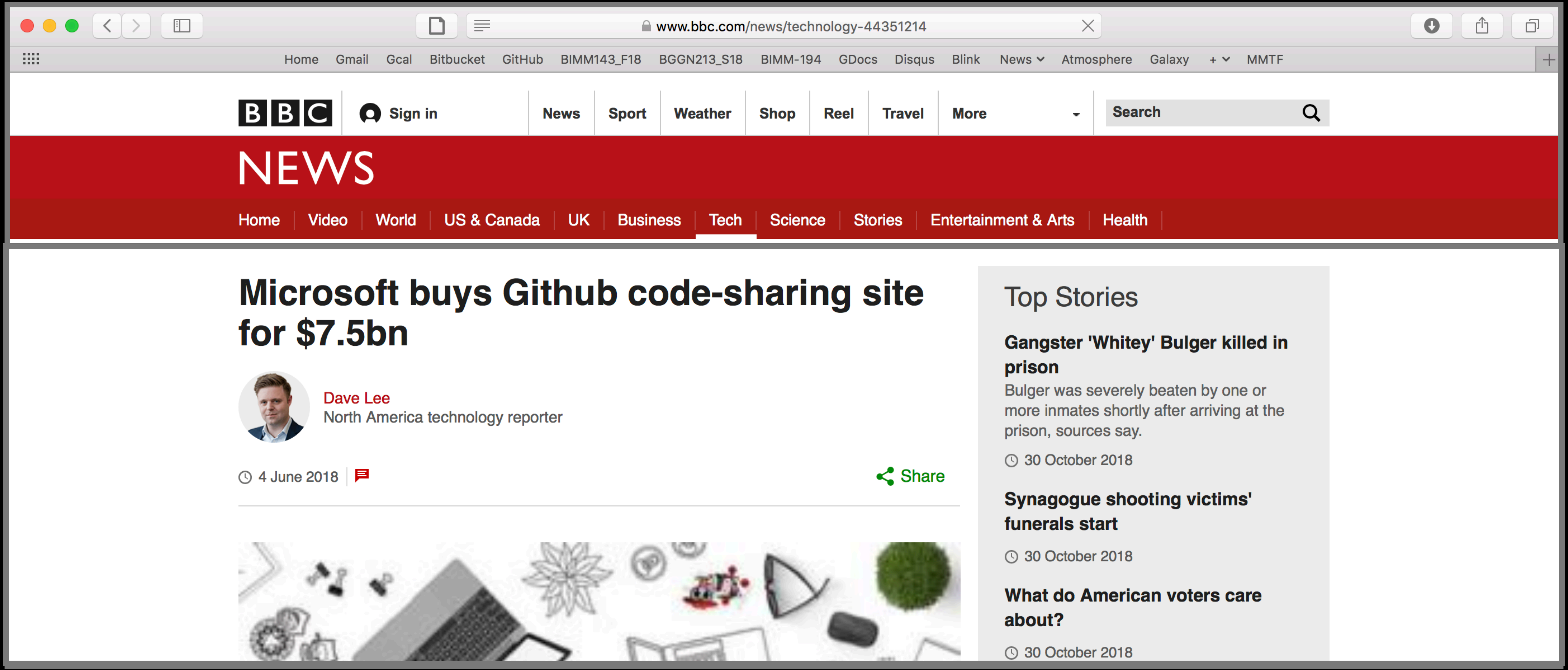
Home World U.S. Politics Economy Business Tech Markets Opinion Arts Life Real Estate

Workers Get New Tools for Airing Their Gripes
Cell Carriers Battle for Wi-Fi Airwaves
Snapchat Names ex-Mattel Exec Vollero Its Finance Chief

TECH

GitHub Raises $250 Million at $2 Billion Valuation

Capital raise puts company's total funding at $350 million



Microsoft buys Github code-sharing site for $7.5bn

Dave Lee
North America technology reporter

4 June 2018

Top Stories

Gangster 'Whitey' Bulger killed in prison

Bulger was severely beaten by one or more inmates shortly after arriving at the prison, sources say.

30 October 2018

Synagogue shooting victims' funerals start

30 October 2018

What do American voters care about?

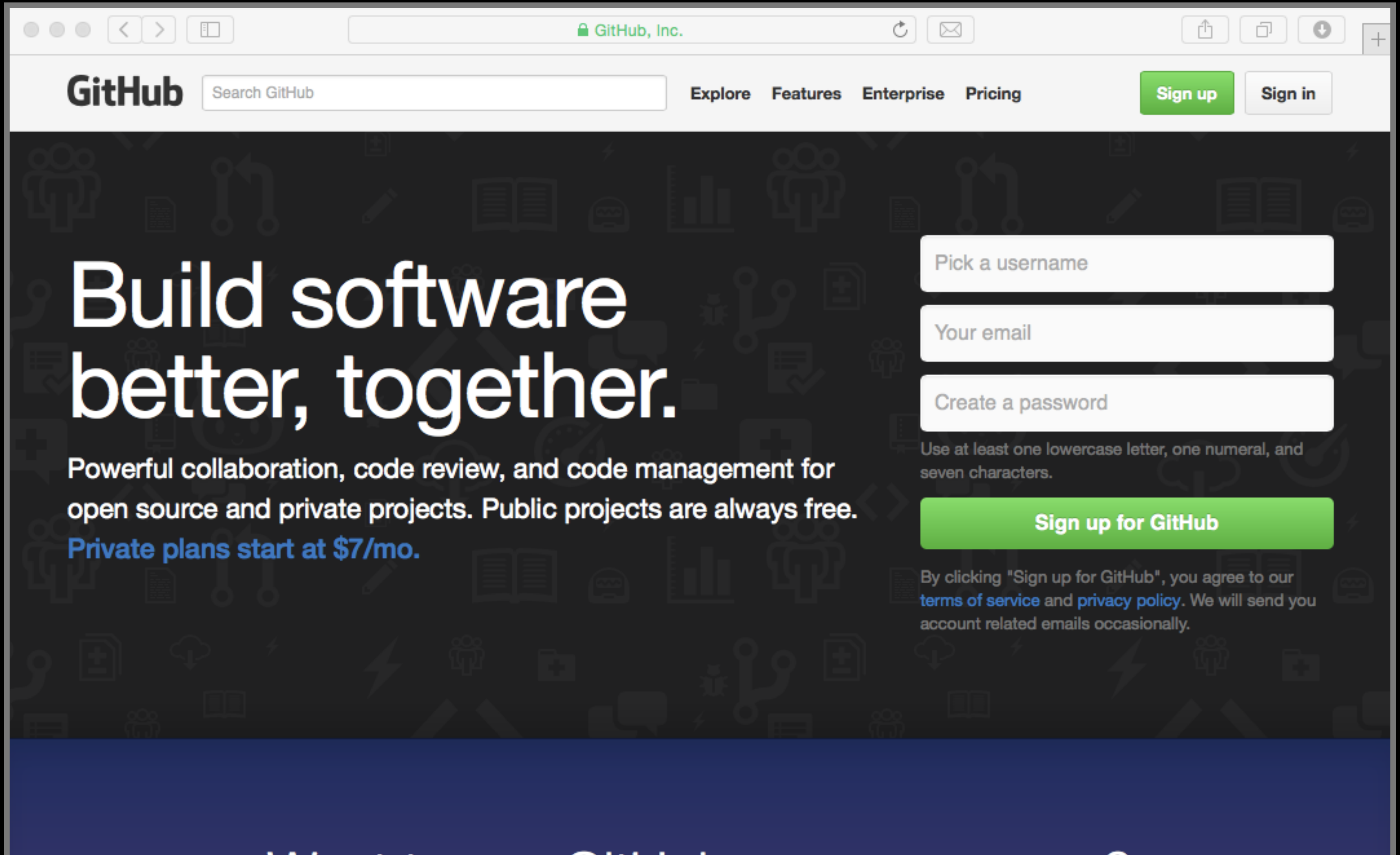
30 October 2018

What is the big deal?

- At the simplest level GitHub and Bitbucket offer **backup** of your projects history and a centralized mechanism for **sharing** with others by putting **your Git repo online**.
    - GitHub in particular is often referred to as the “nerds FaceBook and LinkedIn combined”.
- At their core both services **offer a new paradigm for open collaborative project development**, particularly for software.
    - In essence they allow anybody to contribute to any public project and get acknowledgment.

First sign up for a GitHub account

https://github.com



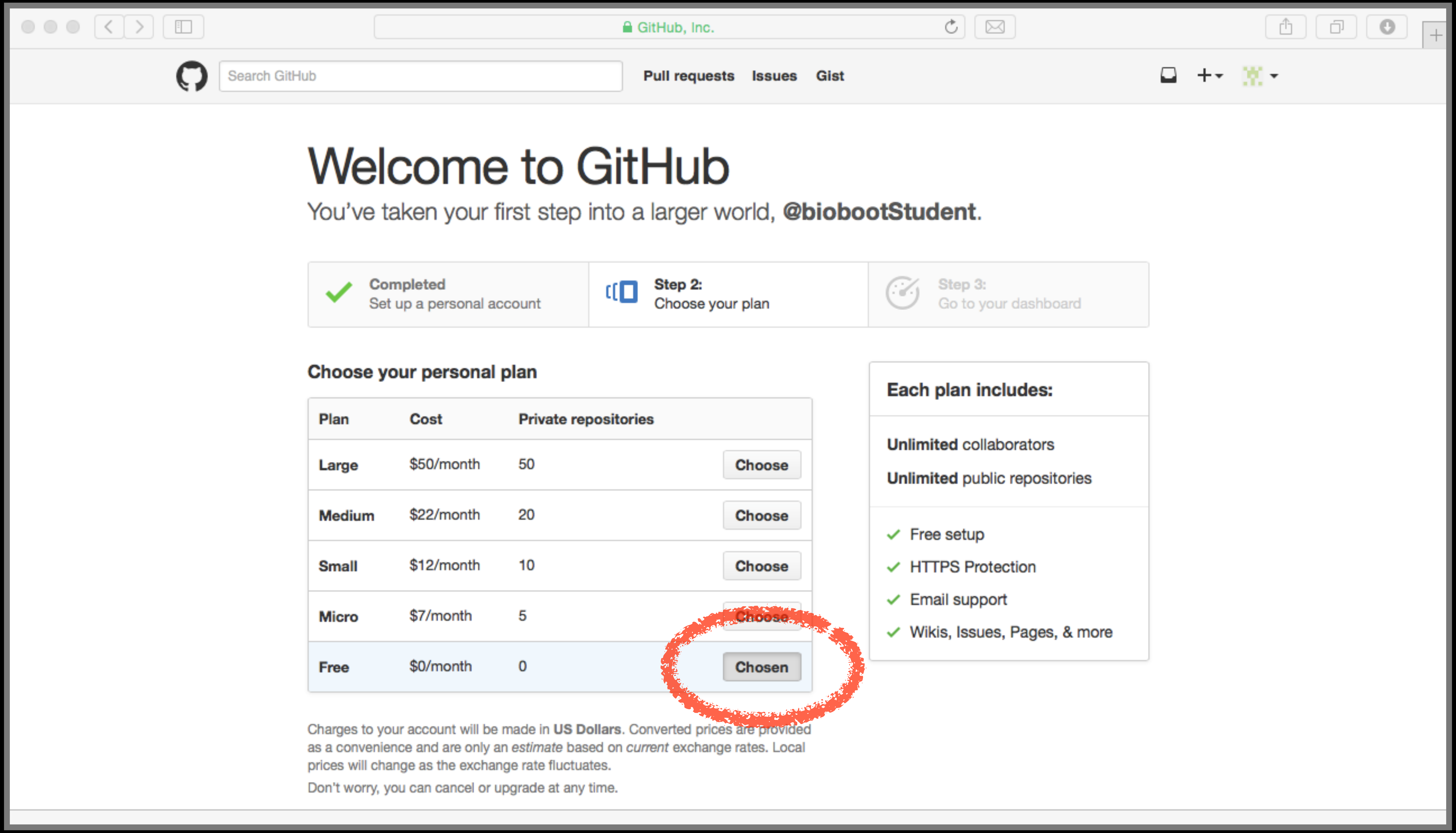
Build software better, together.

Powerful collaboration, code review, and code management for open source and private projects. Public projects are always free. Private plans start at $7/mo.

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

By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy. We will send you account related emails occasionally.

Pick the FREE plan!



Welcome to GitHub

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

✓ Completed
Set up a personal account

Step 2:
Choose your plan

Step 3:
Go to your dashboard

Choose your personal plan

Plan	Cost	Private repositories	
Large	$50/month	50	Choose
Medium	$22/month	20	Choose
Small	$12/month	10	Choose
Micro	$7/month	5	Choose
Free	$0/month	0	[✓ Chosen]

Charges to your account will be made in **US Dollars**. Converted prices are provided as a convenience and are only an *estimate* based on *current* exchange rates. Local prices will change as the exchange rate fluctuates.

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

Each plan includes:

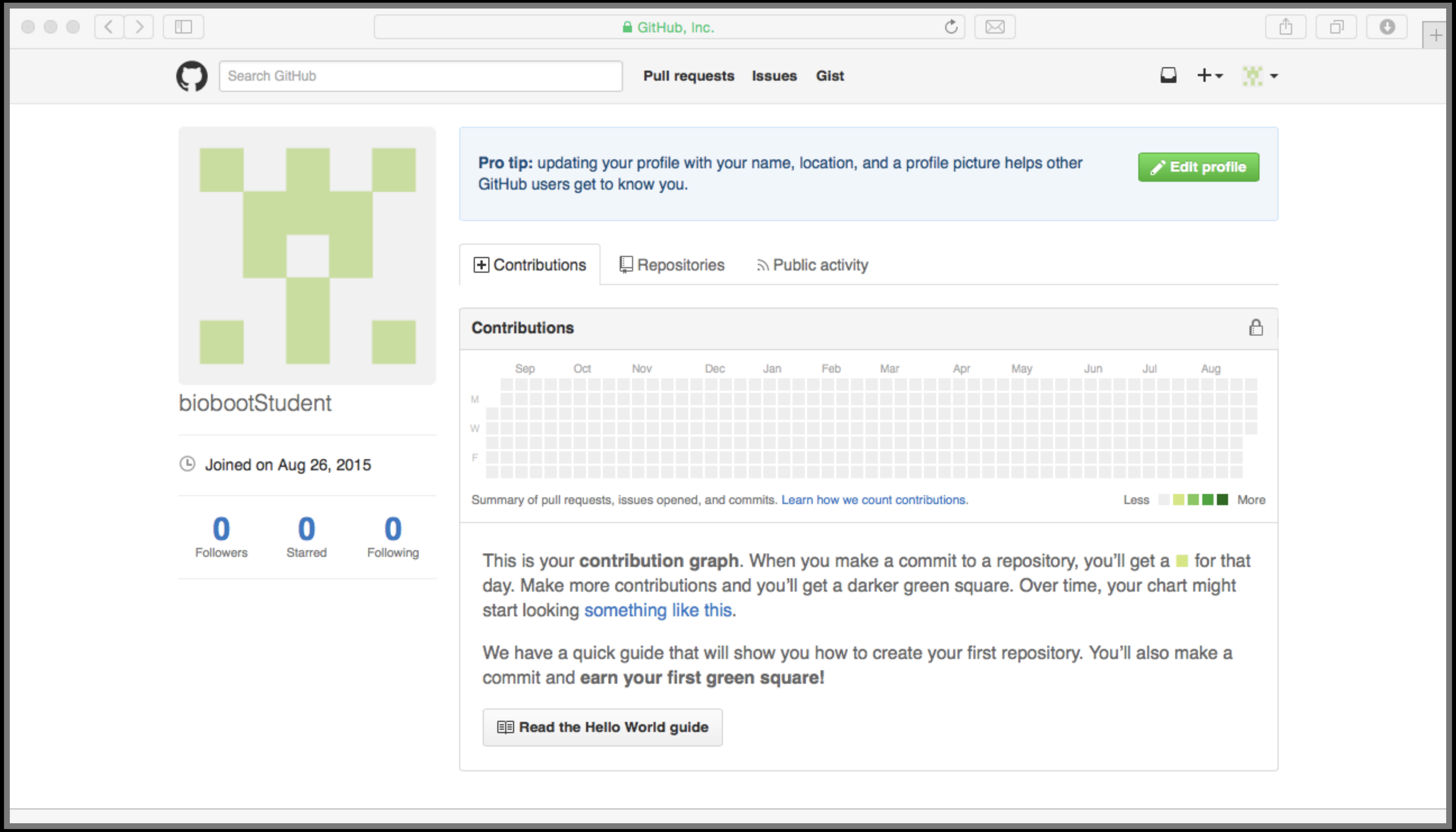
Unlimited collaborators

Unlimited public repositories

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

Your GitHub homepage

Check your email for verification request



Pro tip: updating your profile with your name, location, and a profile picture helps other GitHub users get to know you.

biobootStudent

Joined on Aug 26, 2015

0 Followers 0 Starred 0 Following

Contributions

Summary of pull requests, issues opened, and commits. Learn how we count contributions.

This is your **contribution graph**. When you make a commit to a repository, you'll get a 🟩 for that day. Make more contributions and you'll get a darker green square. Over time, your chart might start looking something like this.

We have a quick guide that will show you how to create your first repository. You'll also make a commit and **earn your first green square!**

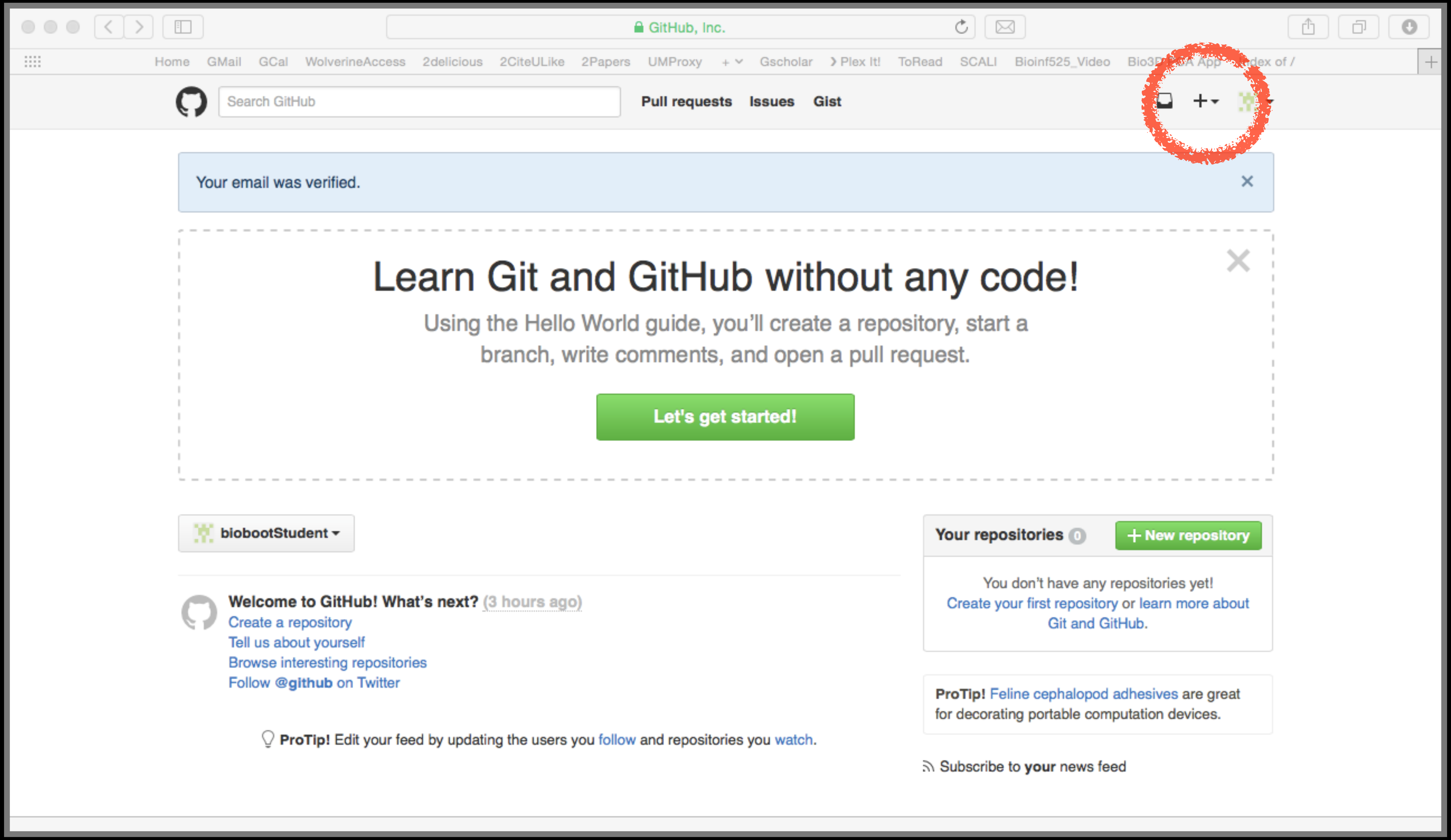
Connecting RStudio to GitHub

Create a **Personal Access Token** (PAT) on GitHub

See section 4 of lab worksheet

Skip the hello-world tutorial

https://guides.github.com/activities/hello-world/



Your email was verified.

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Let's get started!

biobootStudent

Welcome to GitHub! What's next? (3 hours ago)

- Create a repository
- Tell us about yourself
- Browse interesting repositories
- Follow @github on Twitter

ProTip! Edit your feed by updating the users you follow and repositories you watch.

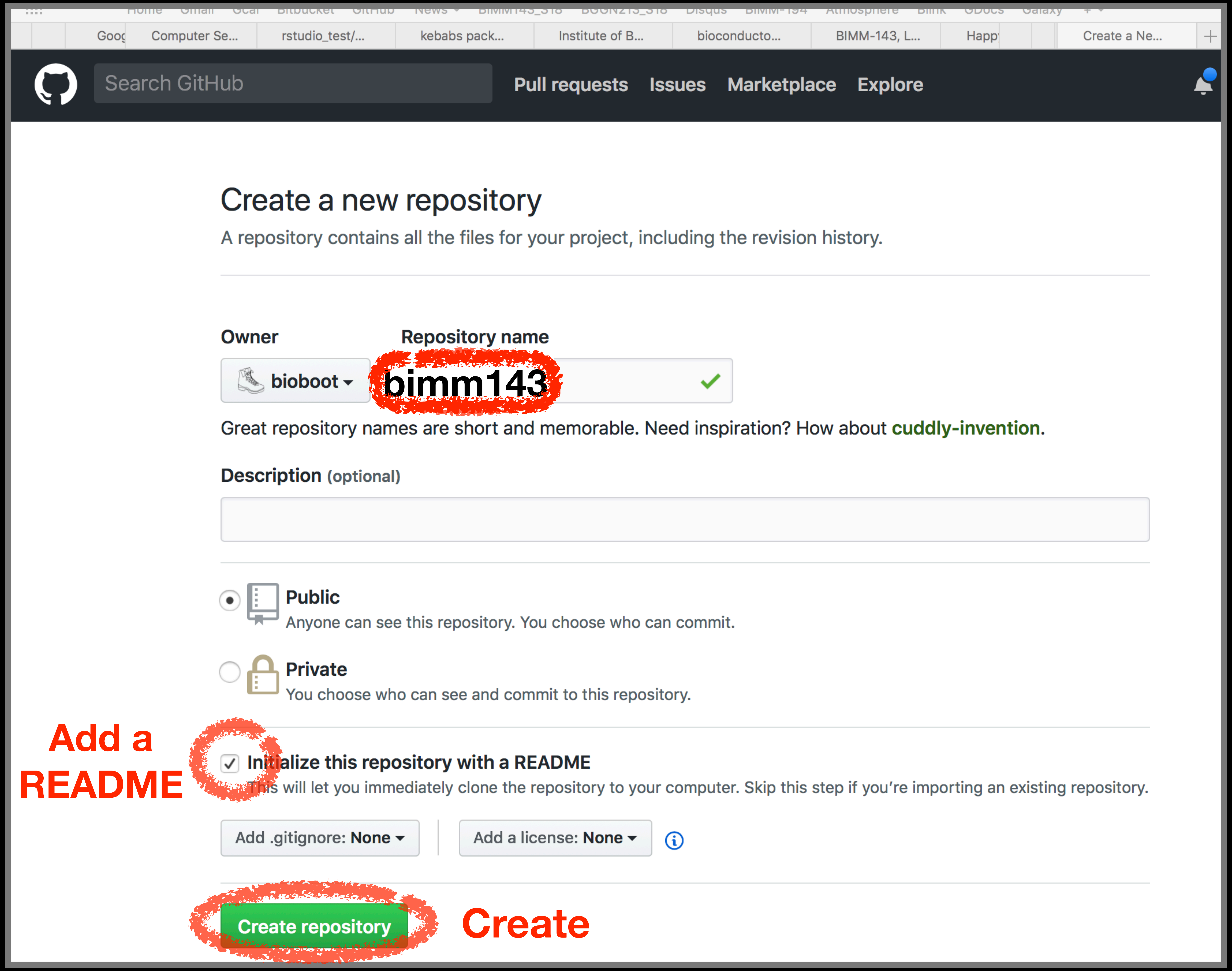
Your repositories 0 + New repository

You don't have any repositories yet! Create your first repository or learn more about Git and GitHub.

ProTip! Feline cephalopod adhesives are great for decorating portable computation devices.

Subscribe to **your** news feed

Name your repo bimm143



Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: bioboot

Repository name: bimm143 ✓

Great repository names are short and memorable. Need inspiration? How about cuddly-invention.

Description (optional)

● 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.

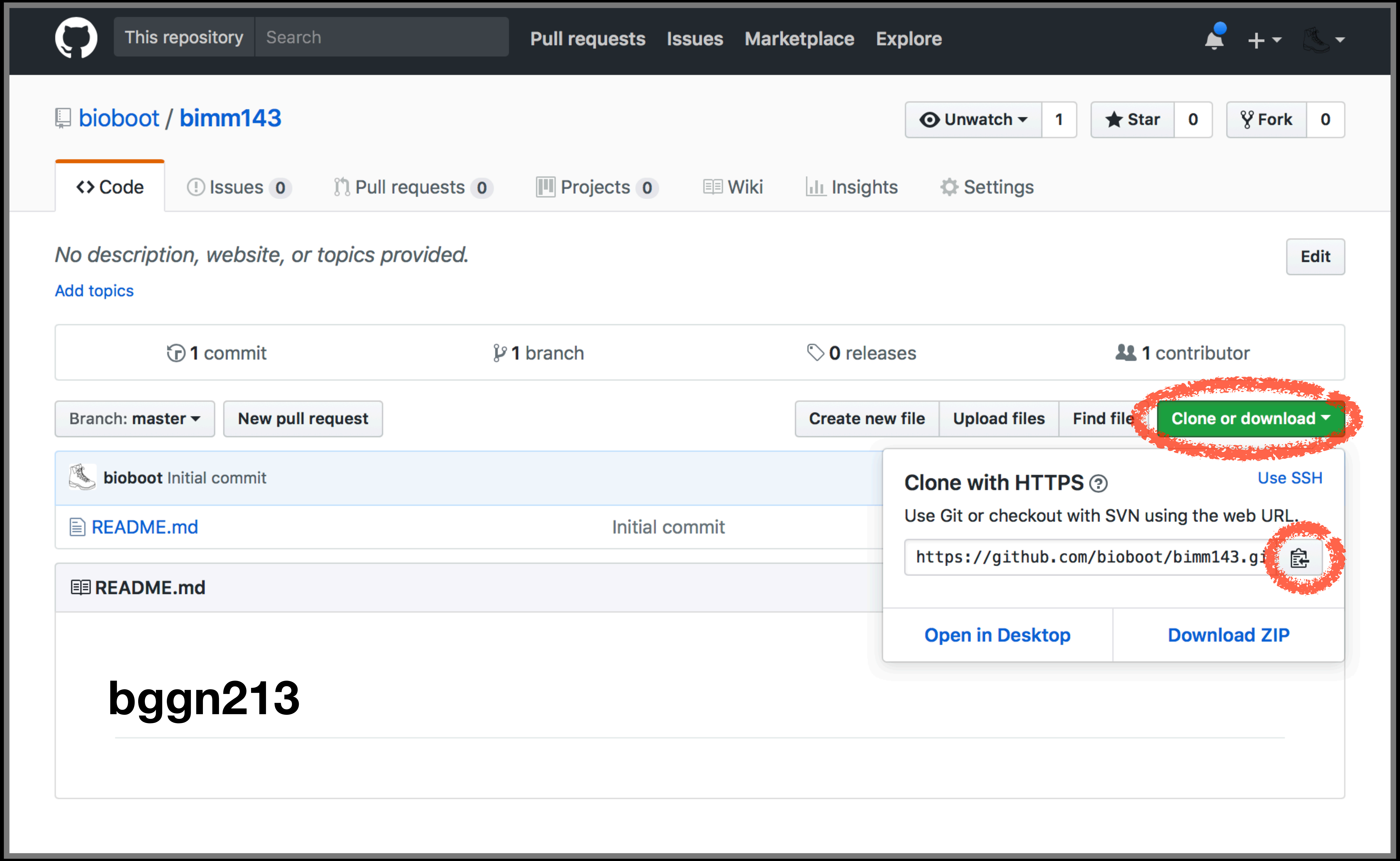
Add .gitignore: None

Add a license: None

Add a README

Create

Copy the “Clone” HTTPS link



No description, website, or topics provided.

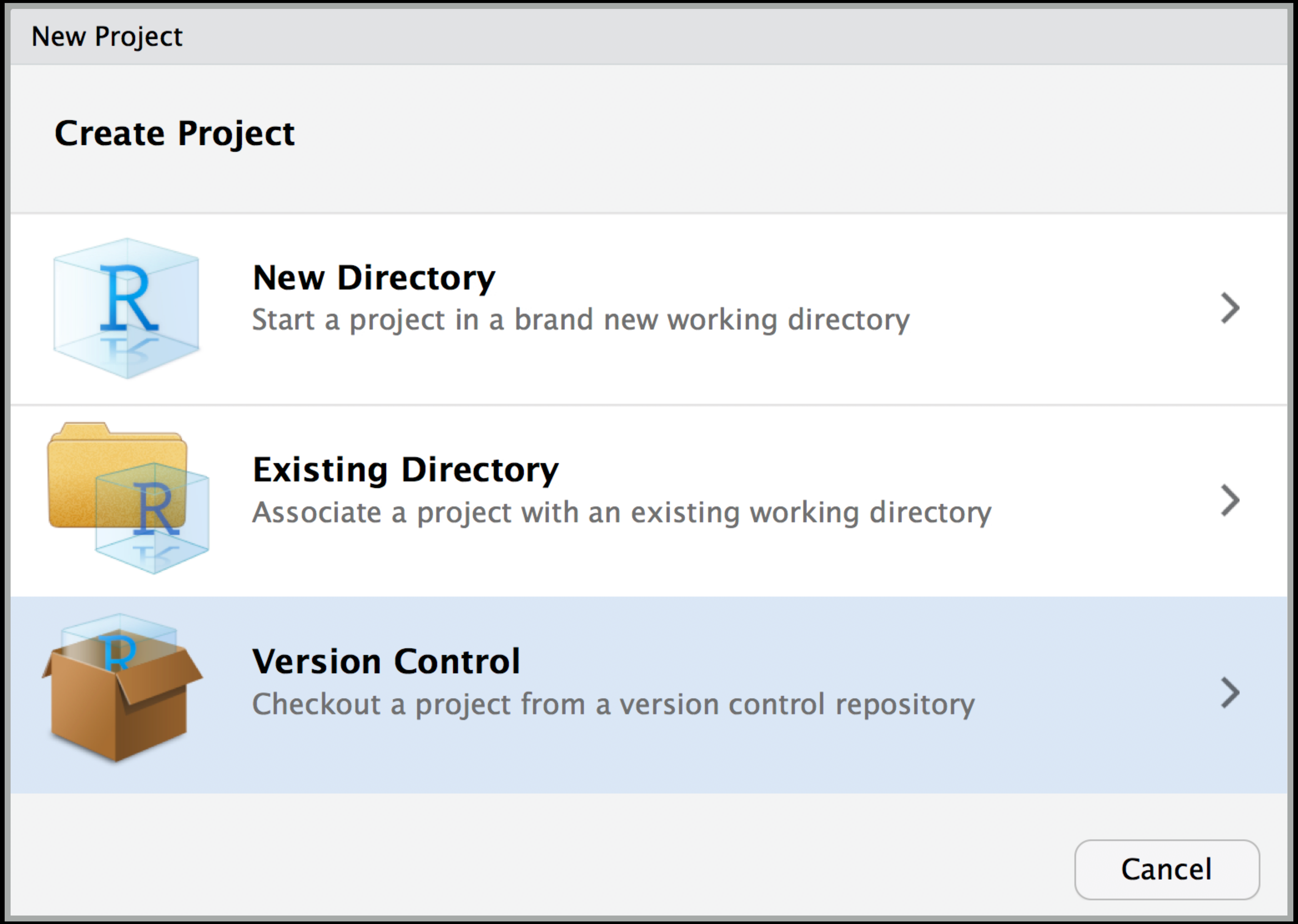
Clone with HTTPS

Use Git or checkout with SVN using the web URL.

```
https://github.com/bioboot/bimm143.gi
```

bggn213

RStudio > New Project > Version Control



New Project

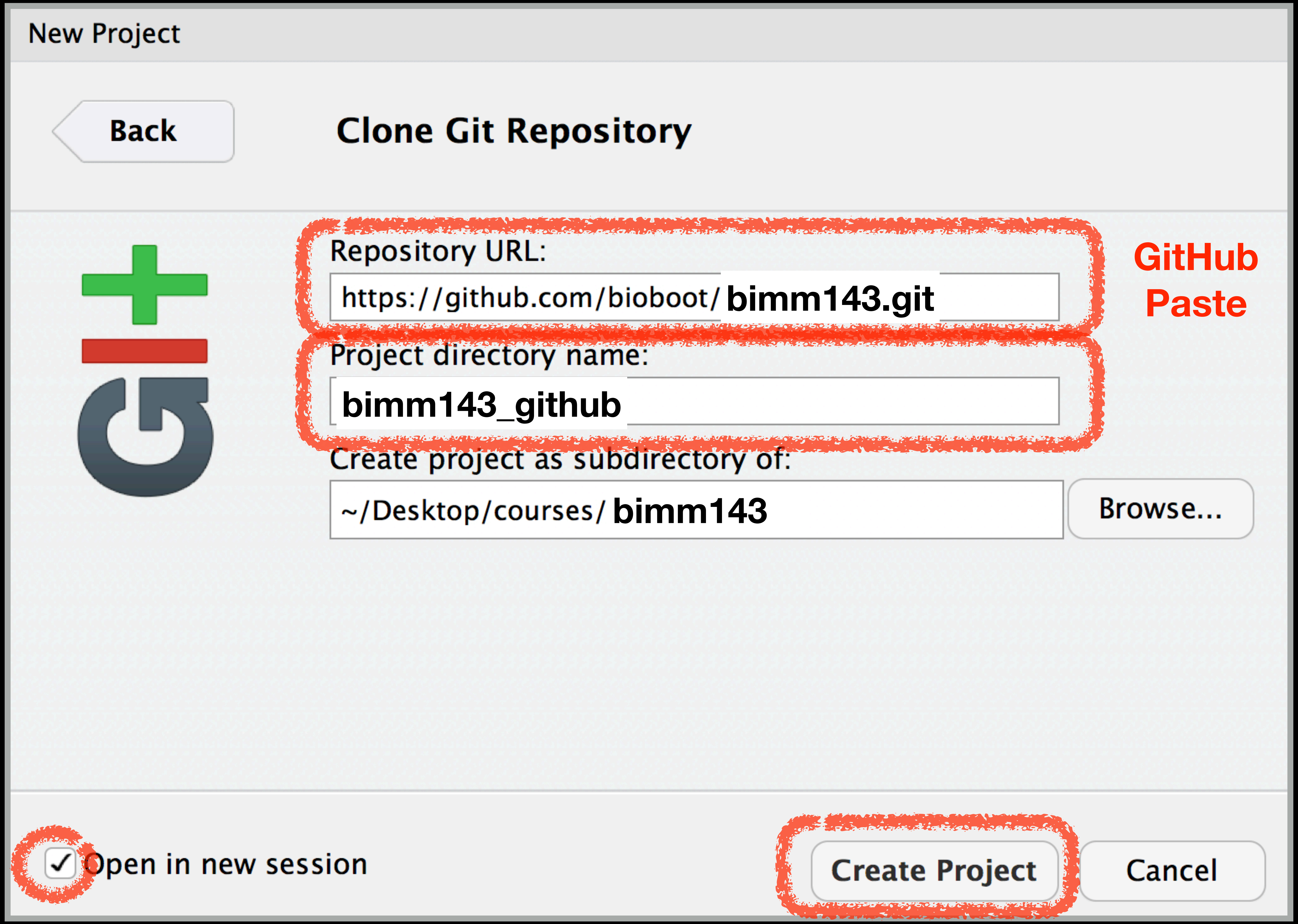
Create Project

New Directory
Start a project in a brand new working directory

Existing Directory
Associate a project with an existing working directory

Version Control
Checkout a project from a version control repository

RStudio > New Project > Version Control



New Project

Clone Git Repository

Repository URL: https://github.com/bioboot/bimm143.git

Project directory name: bimm143_github

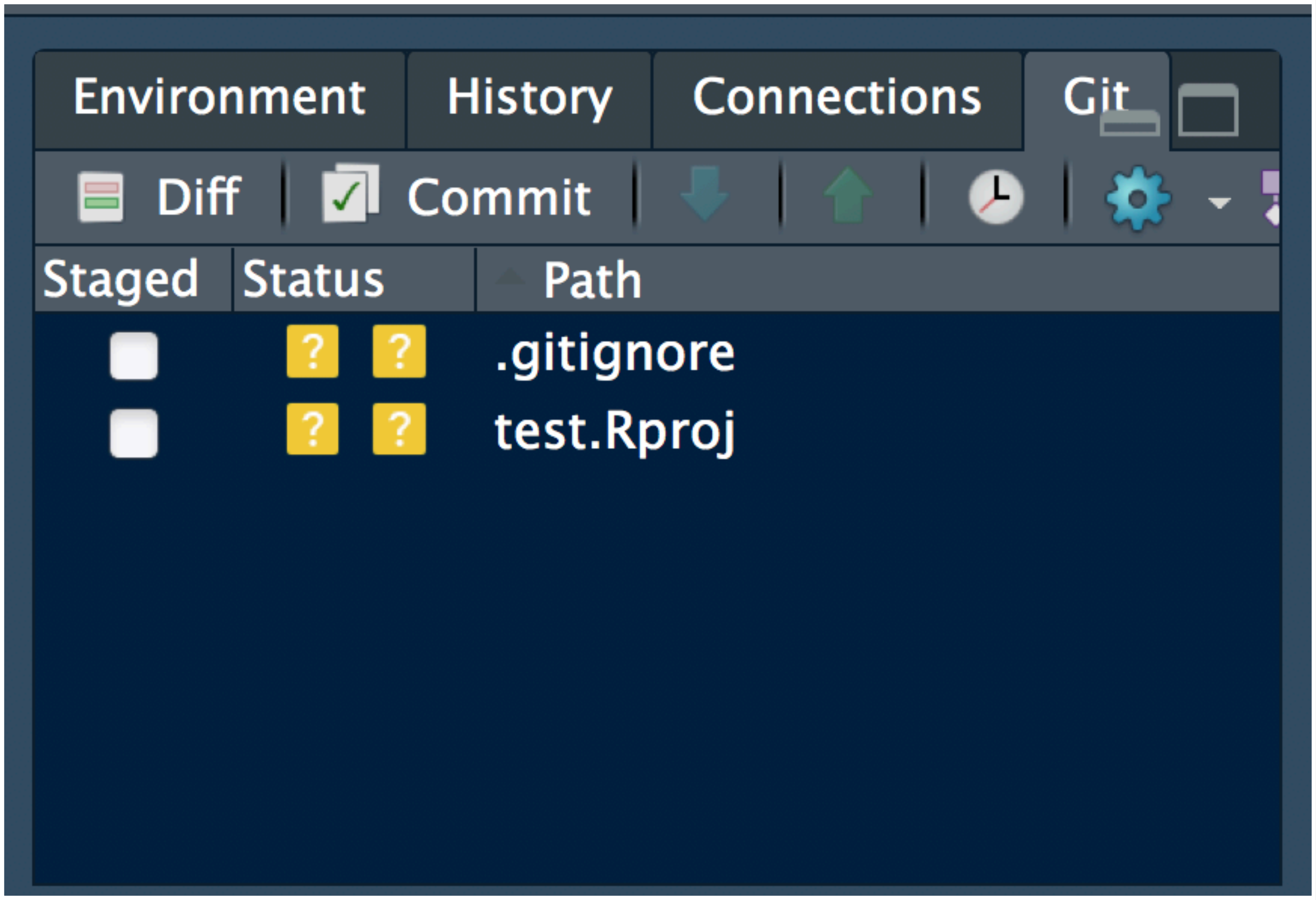
Create project as subdirectory of: ~/Desktop/courses/bimm143

☑ Open in new session

GitHub Paste

Demo of *editing, adding committing* and *pushing*

Check if new Git tab Appears in RStudio?

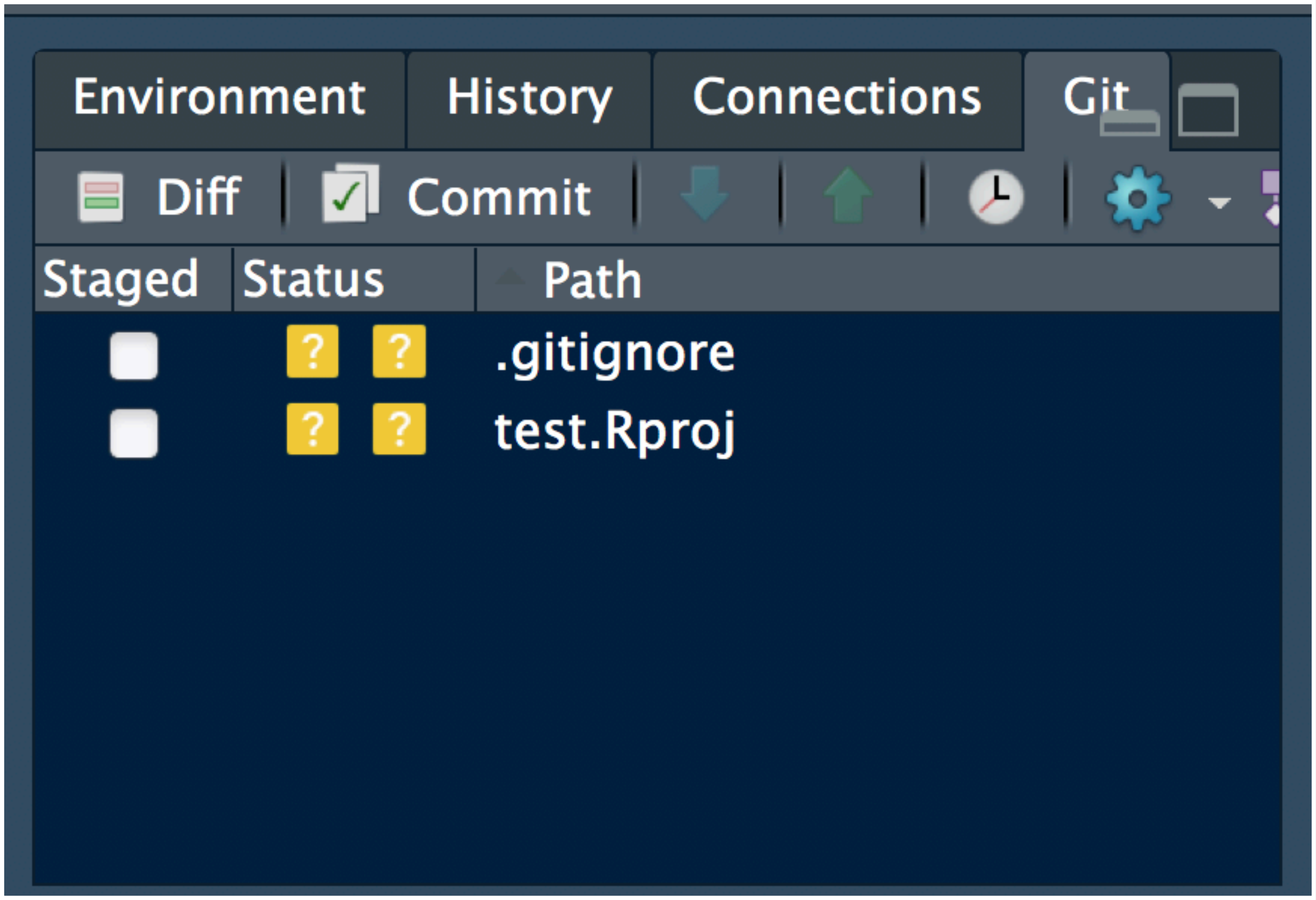


Staged	Status	Path
□	? ?	.gitignore
□	? ?	test.Rproj

Now experiment editing the README.md file in RStudio and adding, committing and pushing changes to GitHub via this tab

Demo of *editing*, *adding* *committing* and *pushing*

Check if new Git tab Appears in RStudio?



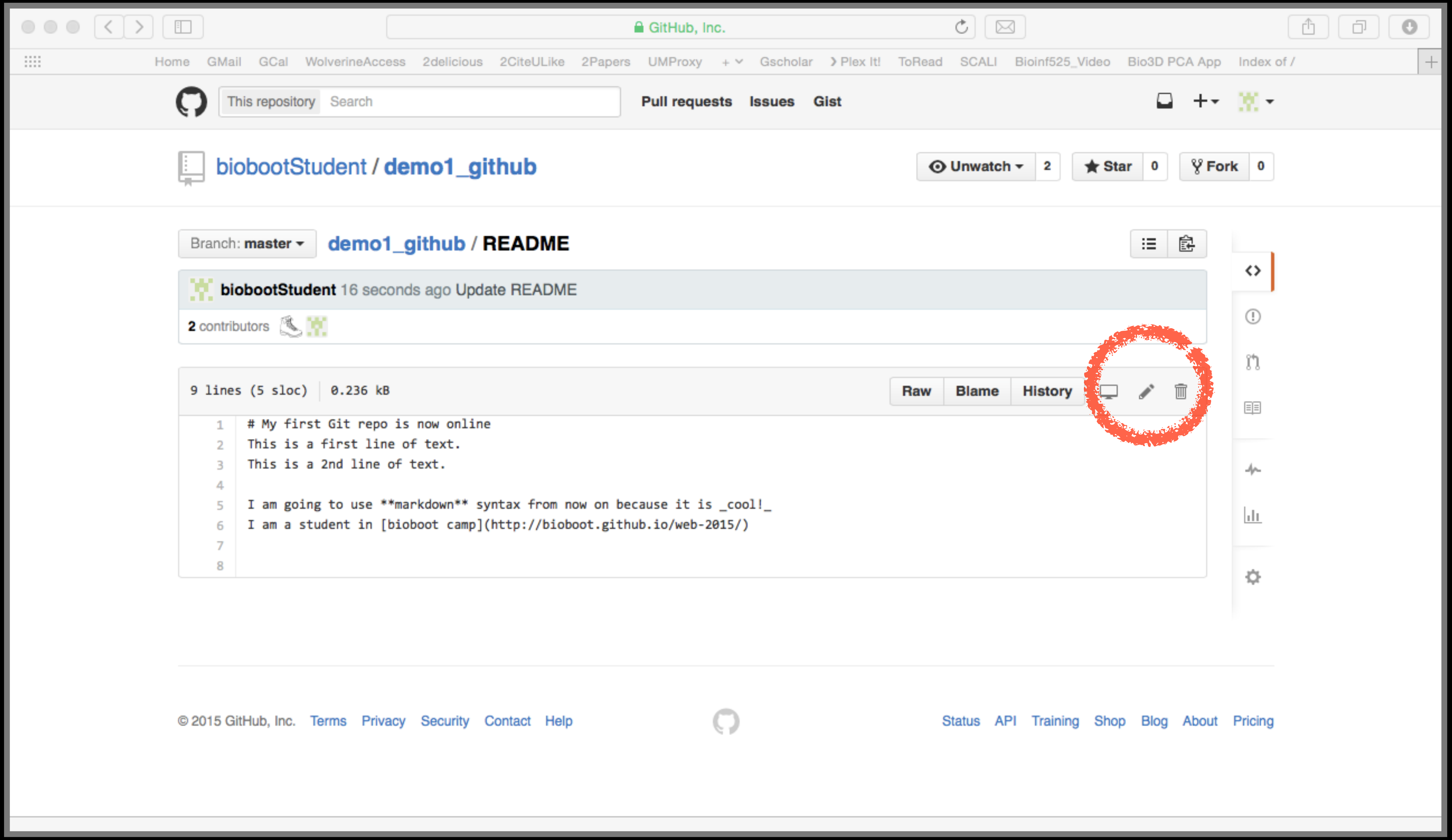
Staged	Status	Path
□	? ?	.gitignore
□	? ?	test.Rproj

Now experiment editing the README.md file in RStudio and adding, committing and pushing changes to GitHub via this tab

When you are ready copy your different class directories/projects to this new GitHub tracked folder

Side-note: How to edit online

Specifically lets add some Markdown content



GitHub, Inc.
Home GMail GCal WolverineAccess 2delicious 2CiteULike 2Papers UMProxy Gscholar Plex It! ToRead SCALI Bioinf525_Video Bio3D PCA App Index of /
This repository Search
Pull requests Issues Gist
Unwatch 2
Star 0
Fork 0
Branch: master
Raw
Blame
History
© 2015 GitHub, Inc. Terms Privacy Security Contact Help
Status API Training Shop Blog About Pricing

bioboostStudent / demo1_github

demo1_github / README

bioboostStudent 16 seconds ago Update README

2 contributors

9 lines (5 sloc) | 0.236 kB

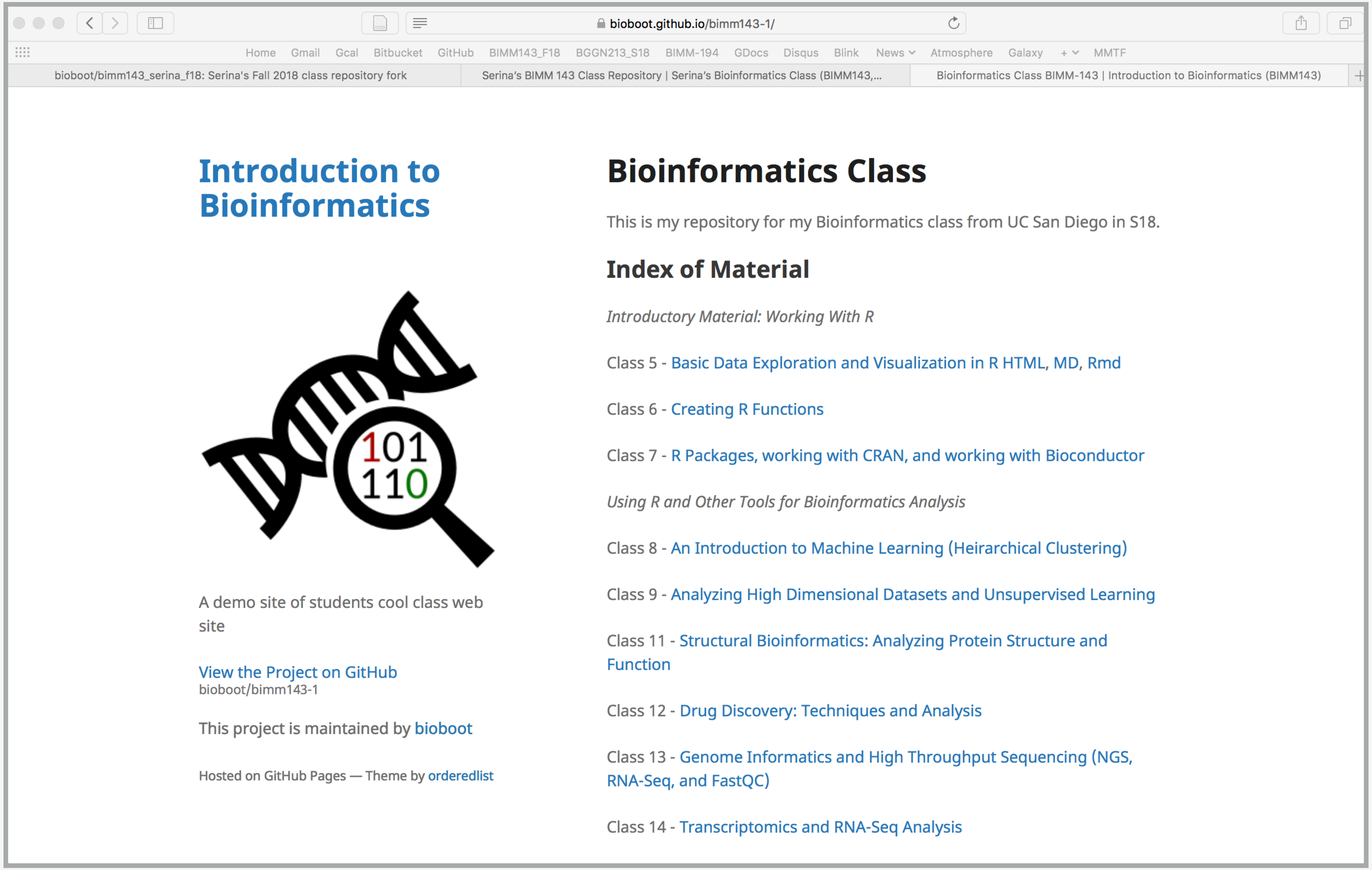
```
# My first Git repo is now online
This is a first line of text.
This is a 2nd line of text.

I am going to use **markdown** syntax from now on because it is _cool!_
I am a student in [bioboot camp](http://bioboot.github.io/web-2015/)


```

Summary

- Git is a popular ‘distributed’ version control system that is lightweight and free
- GitHub and BitBucket are popular hosting services for git repositories that have changed the way people contribute to open source projects
- Introduced basic git and GitHub usage within RStudio and encouraged you to adopt these ‘best practices’ for your future projects.

Bonus:

GitHub Spit & Polish


bioboot.github.io/bimm143-1/
Home
Gmail
Gcal
Bitbucket
GitHub
BIMM143_F18
BGGN213_S18
BIMM-194
GDocs
Disqus
Blink
News
Atmosphere
Galaxy
MMTF
bioboot/bimm143_serina_f18: Serina's Fall 2018 class repository fork
Serina's BIMM 143 Class Repository | Serina's Bioinformatics Class (BIMM143,...
Bioinformatics Class BIMM-143 | Introduction to Bioinformatics (BIMM143)
101
110

Introduction to Bioinformatics

A demo site of students cool class web site

View the Project on GitHub
bioboot/bimm143-1

This project is maintained by bioboot

Hosted on GitHub Pages — Theme by orderedlist

Bioinformatics Class

This is my repository for my Bioinformatics class from UC San Diego in S18.

Index of Material

Introductory Material: Working With R

Class 5 - Basic Data Exploration and Visualization in R HTML, MD, Rmd

Class 6 - Creating R Functions

Class 7 - R Packages, working with CRAN, and working with Bioconductor

Using R and Other Tools for Bioinformatics Analysis

Class 8 - An Introduction to Machine Learning (Heirarchical Clustering)

Class 9 - Analyzing High Dimensional Datasets and Unsupervised Learning

Class 11 - Structural Bioinformatics: Analyzing Protein Structure and Function

Class 12 - Drug Discovery: Techniques and Analysis

Class 13 - Genome Informatics and High Throughput Sequencing (NGS, RNA-Seq, and FastQC)

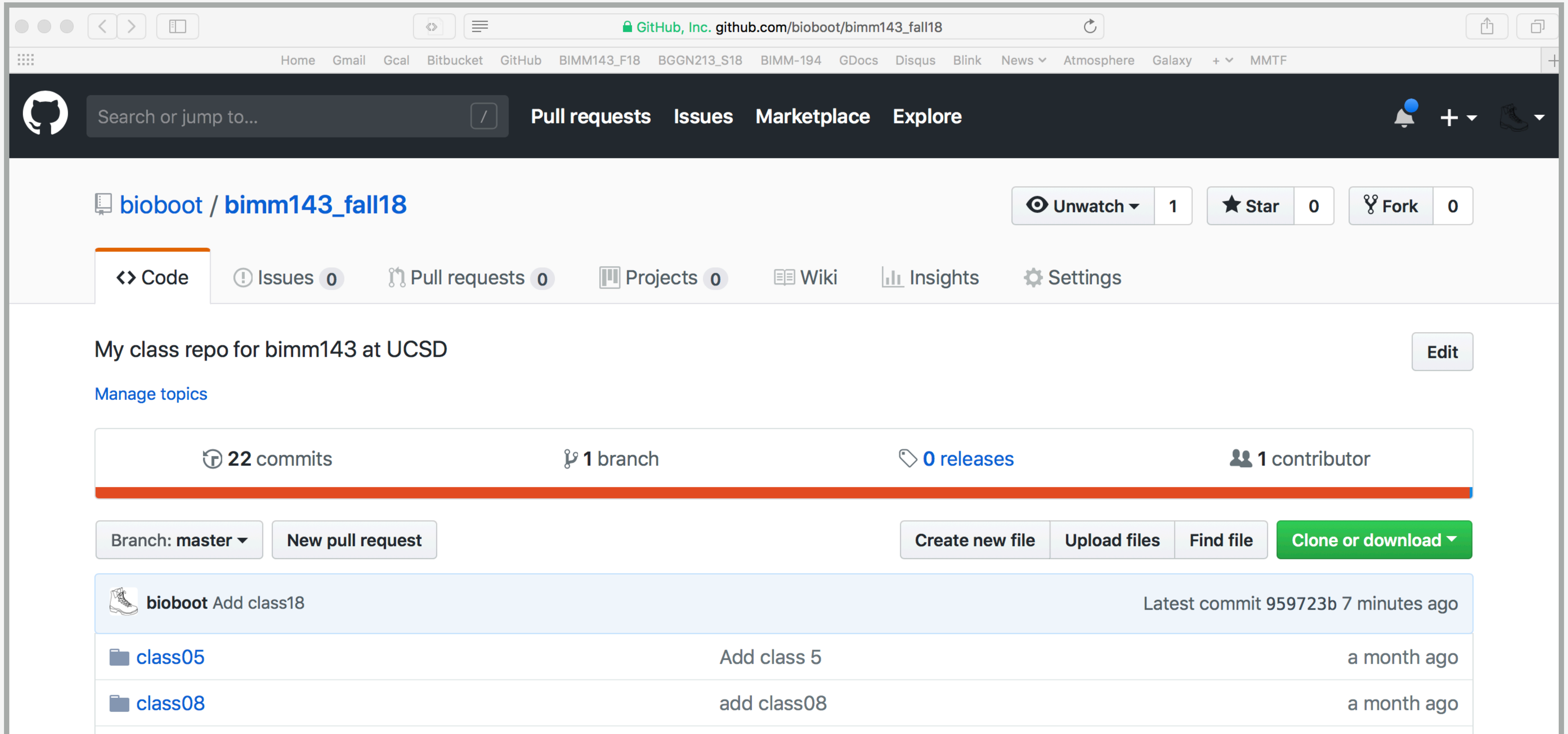
Class 14 - Transcriptomics and RNA-Seq Analysis

In your web browser navigate to your GitHub class repository < https://github.com/ >

Side-note:

To find the link to your GitHub repository from RStudio, open one of your past class projects and in the terminal type:

```
git remote -v
```

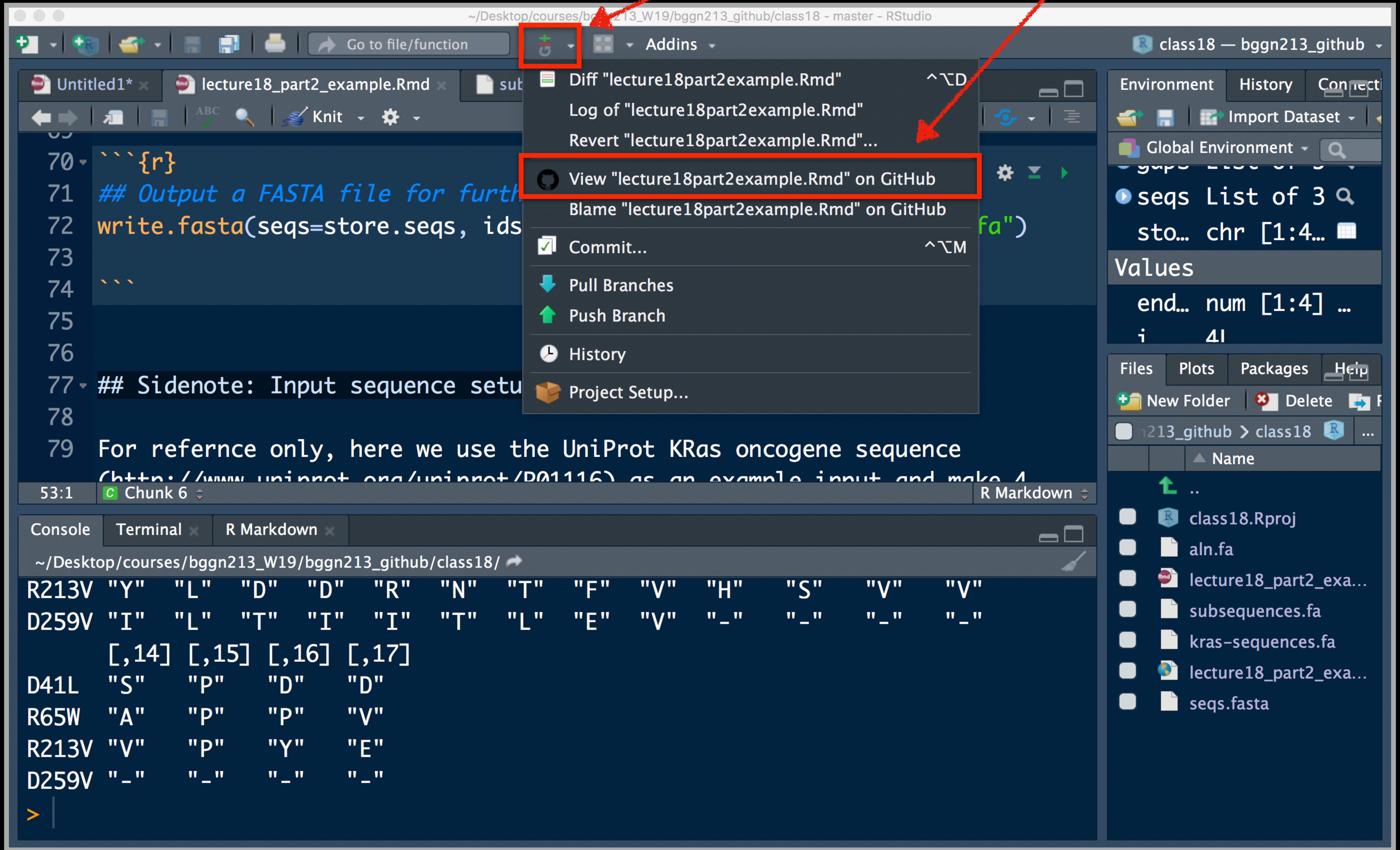

GitHub, Inc. github.com/bioboot/bimm143_fall18
Home Gmail Gcal Bitbucket GitHub BIMM143_F18 BGGN213_S18 BIMM-194 GDocs Disqus Blink News Atmosphere Galaxy MMTF
Search or jump to...
Pull requests Issues Marketplace Explore
bioboot / bimm143_fall18
Unwatch 1 Star 0 Fork 0
Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings
Edit
Manage topics
22 commits 1 branch 0 releases 1 contributor
Branch: master New pull request
Create new file Upload files Find file Clone or download

My class repo for bimm143 at UCSD

bioboot Add class18 Latest commit 959723b 7 minutes ago

class05	Add class 5	a month ago
class08	add class08	a month ago

Or for a given GitHub tracked file click GIT icon and "VIEW on GITHUB"



~~~
```{r}
## Output a FASTA file for furth
write.fasta(seqs=store.seqs, ids                                   fa")

```


## Sidenote: Input sequence setu

For refernce only, here we use the UniProt KRas oncogene sequence
(http://www.uniprot.org/uniprot/P01116) as an example input and make 4
~~~

```
seqs  List of 3
sto…  chr [1:4…
end…  num [1:4] …
i     4L
```

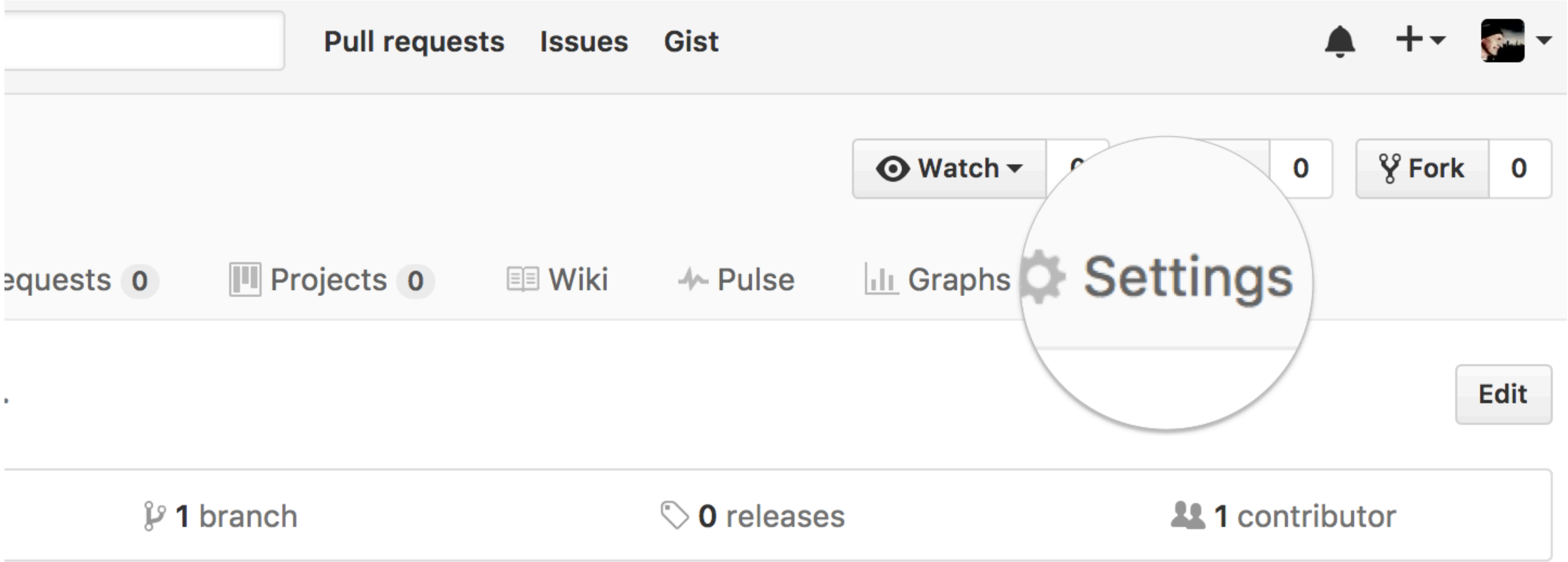
```
~/Desktop/courses/bggn213_W19/bggn213_github/class18/
R213V "Y"   "L"   "D"   "D"   "R"   "N"   "T"   "F"   "V"   "H"   "S"   "V"   "V"
D259V "I"   "L"   "T"   "I"   "I"   "T"   "L"   "E"   "V"   "-"   "-"   "-"   "-"
      [,14] [,15] [,16] [,17]
D41L  "S"   "P"   "D"   "D"
R65W  "A"   "P"   "P"   "V"
R213V "V"   "P"   "Y"   "E"
D259V "-"   "-"   "-"   "-"
>
```

1

Repository Settings

Head over to your GitHub class repository (where you are “pushing” all your class work)

Click on the Settings tab.

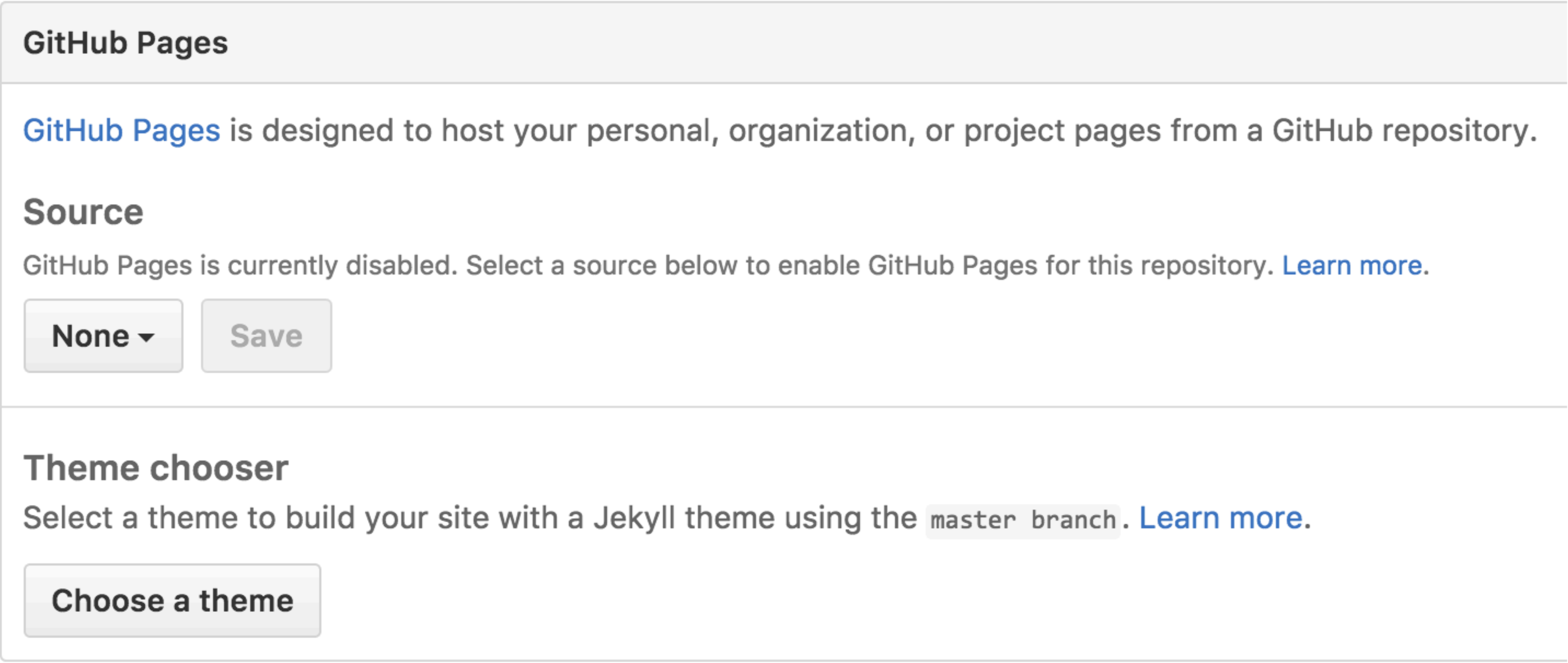


2

Theme chooser

Scroll down to the **GitHub Pages** section. ~~Press **Choose a theme**.~~

And set the Source to “main branch"



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Source

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. Learn more.

Theme chooser

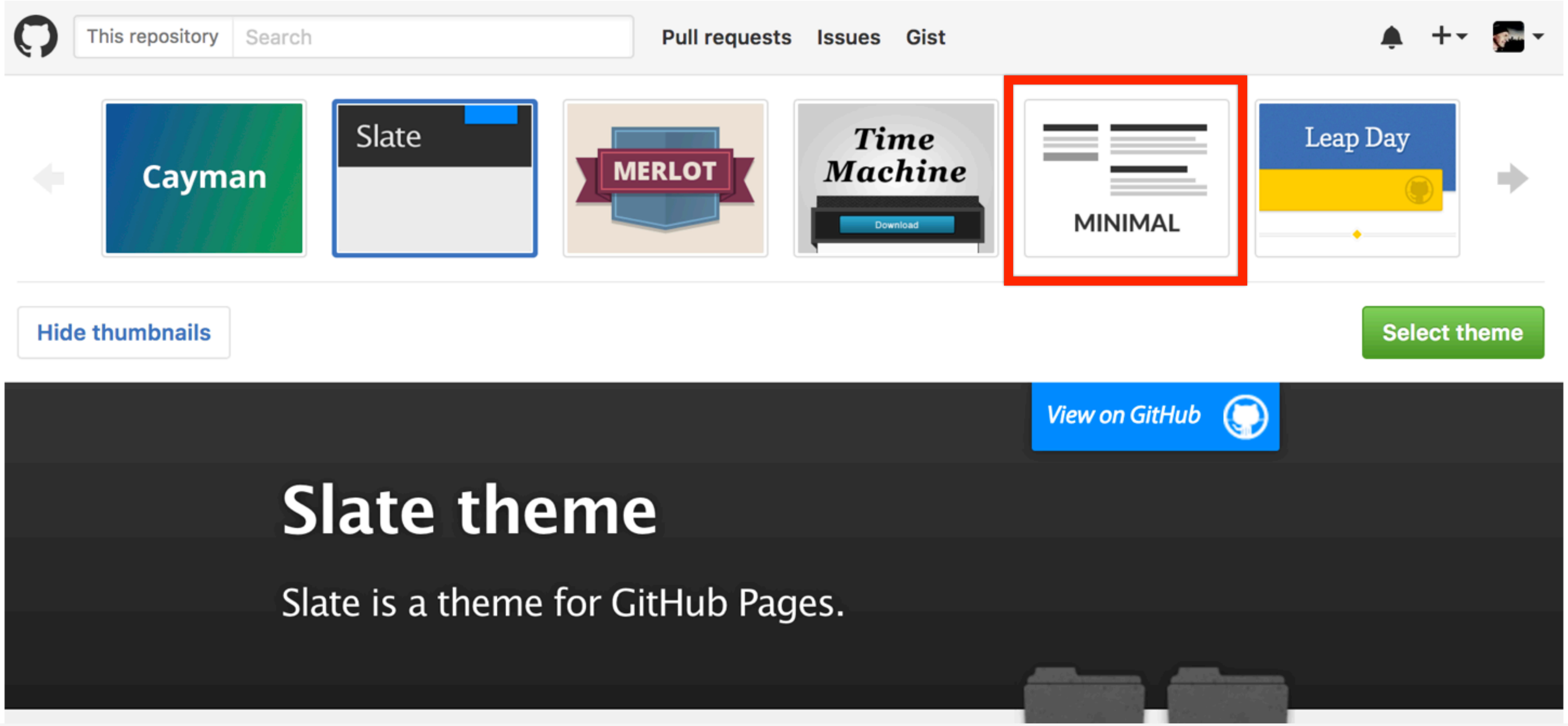
Select a theme to build your site with a Jekyll theme using the `master branch`. Learn more.

3

Pick a theme

Choose one of the themes from the carousel at the top.

When you're done, click **Select theme** on the right.



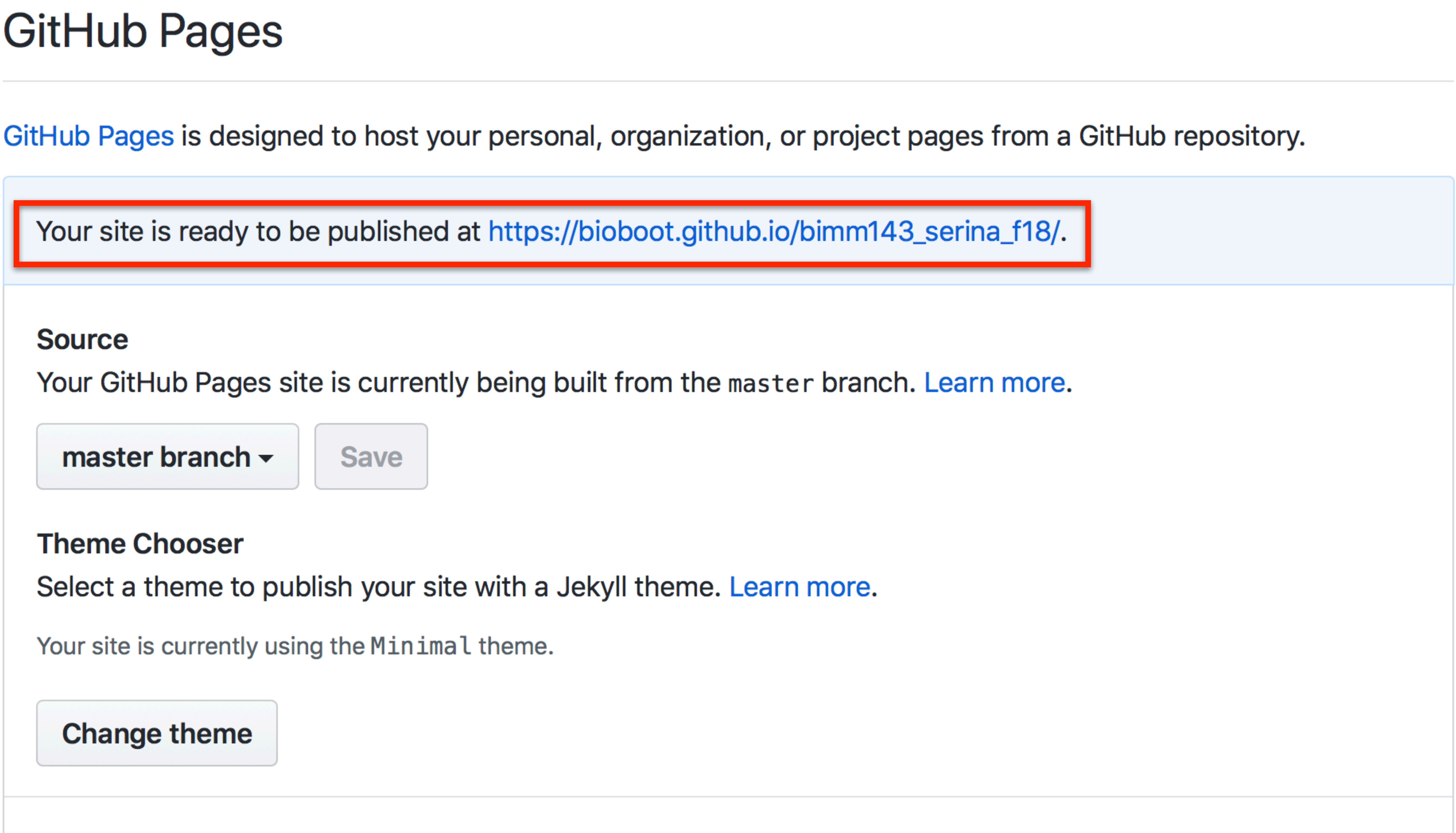
Slate theme

Slate is a theme for GitHub Pages.

4

Side-note:

Scroll down again to the GitHub Pages section to find the link to your new website. Open this link in a New Tab of your browser:



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is ready to be published at https://bioboot.github.io/bimm143_serina_f18/.

Source

Your GitHub Pages site is currently being built from the `master` branch. Learn more.

Theme Chooser

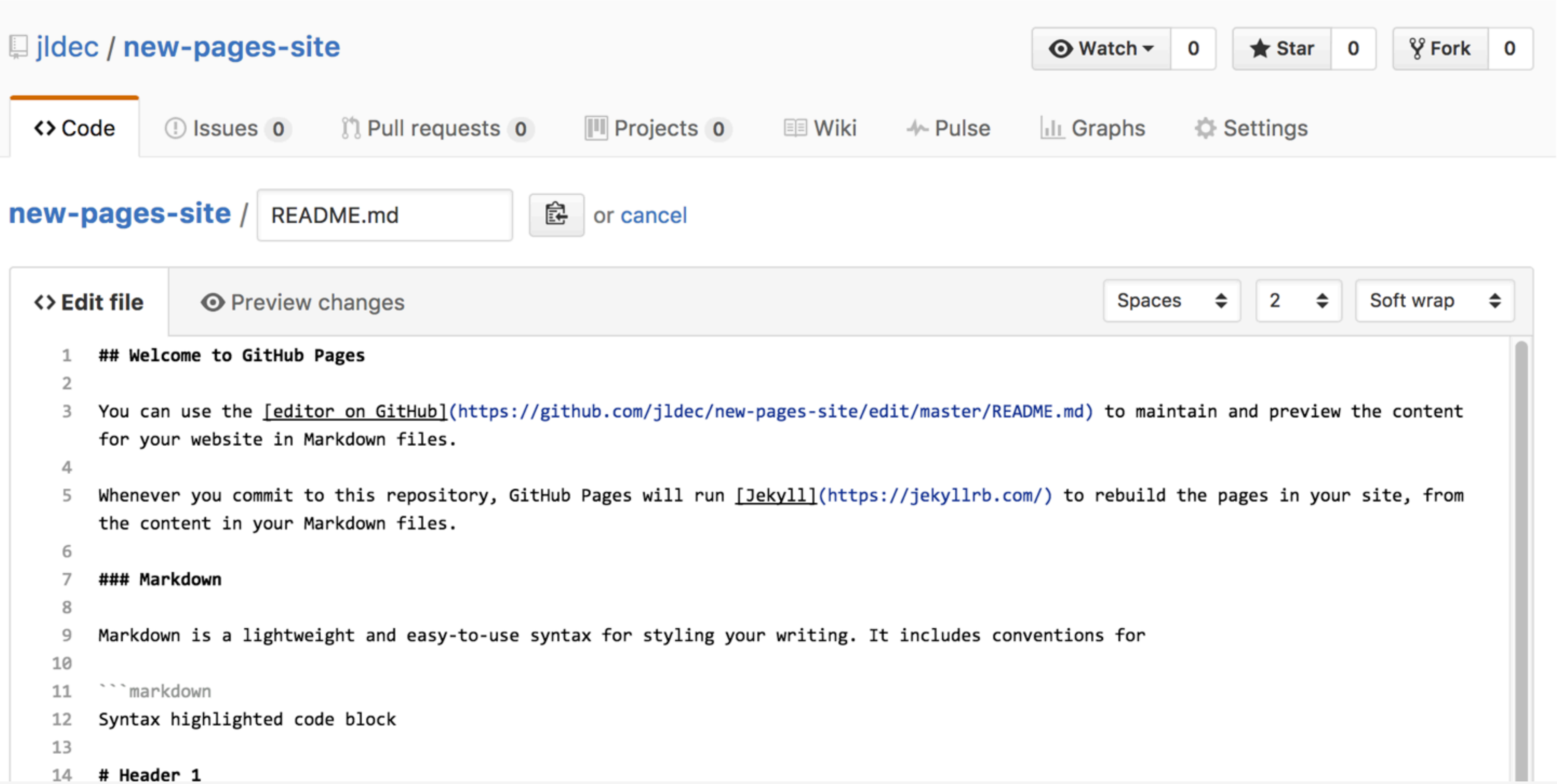
Select a theme to publish your site with a Jekyll theme. Learn more.

Your site is currently using the `Minimal` theme.

4

Edit content

Back on the repository main page use the GitHub online editor to add content. In particular, add links to each classes .MD file

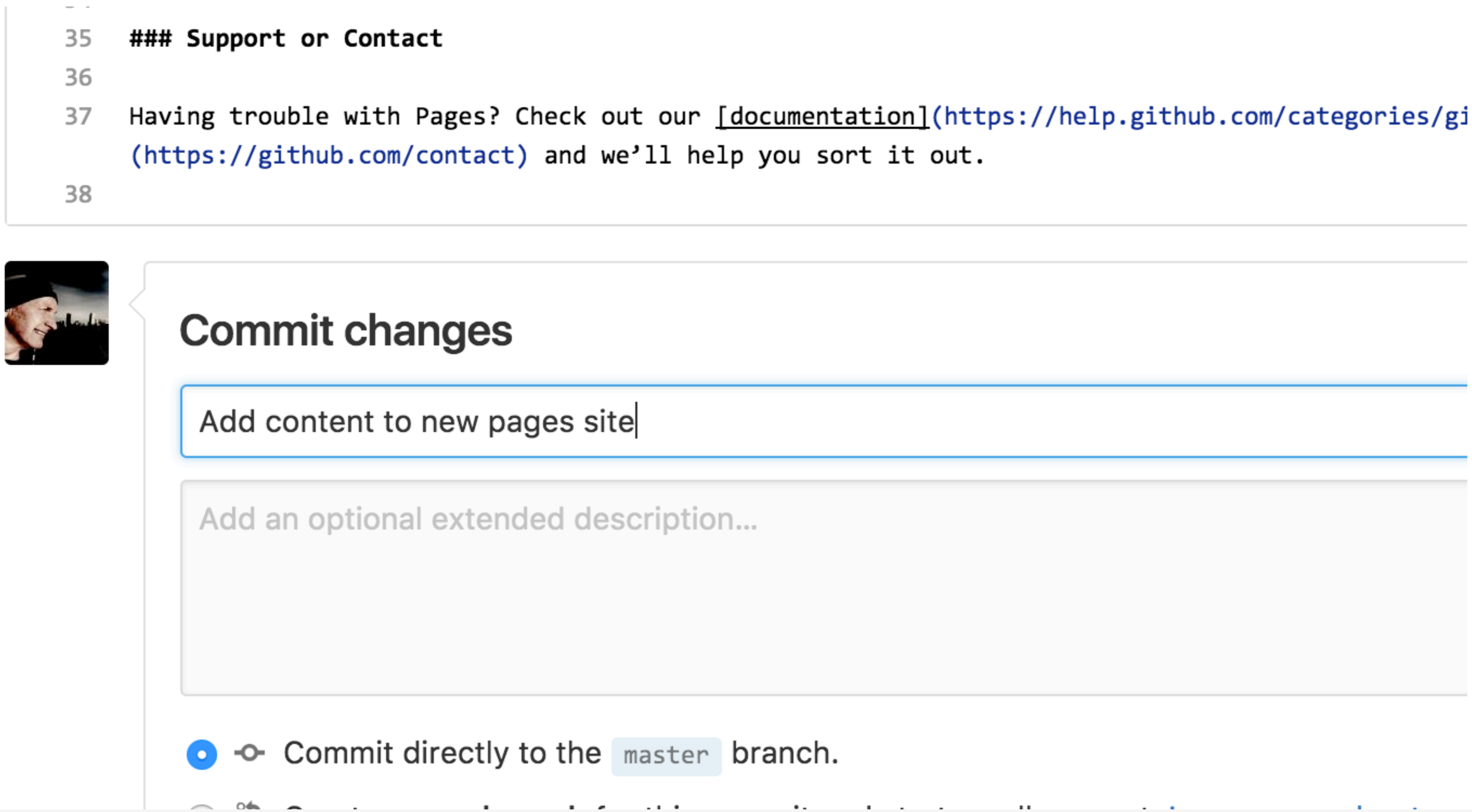


~~~
## Welcome to GitHub Pages

You can use the [editor on GitHub](https://github.com/jldec/new-pages-site/edit/master/README.md) to maintain and preview the content for your website in Markdown files.

Whenever you commit to this repository, GitHub Pages will run [Jekyll](https://jekyllrb.com/) to rebuild the pages in your site, from the content in your Markdown files.

### Markdown

Markdown is a lightweight and easy-to-use syntax for styling your writing. It includes conventions for

```markdown
Syntax highlighted code block

# Header 1
~~~

5

Commit

Enter a commit comment and click on **Commit changes** below the editor.



```
### Support or Contact

Having trouble with Pages? Check out our [documentation](https://help.github.com/categories/gi
(https://github.com/contact) and we’ll help you sort it out.

```

Commit changes

Add content to new pages site

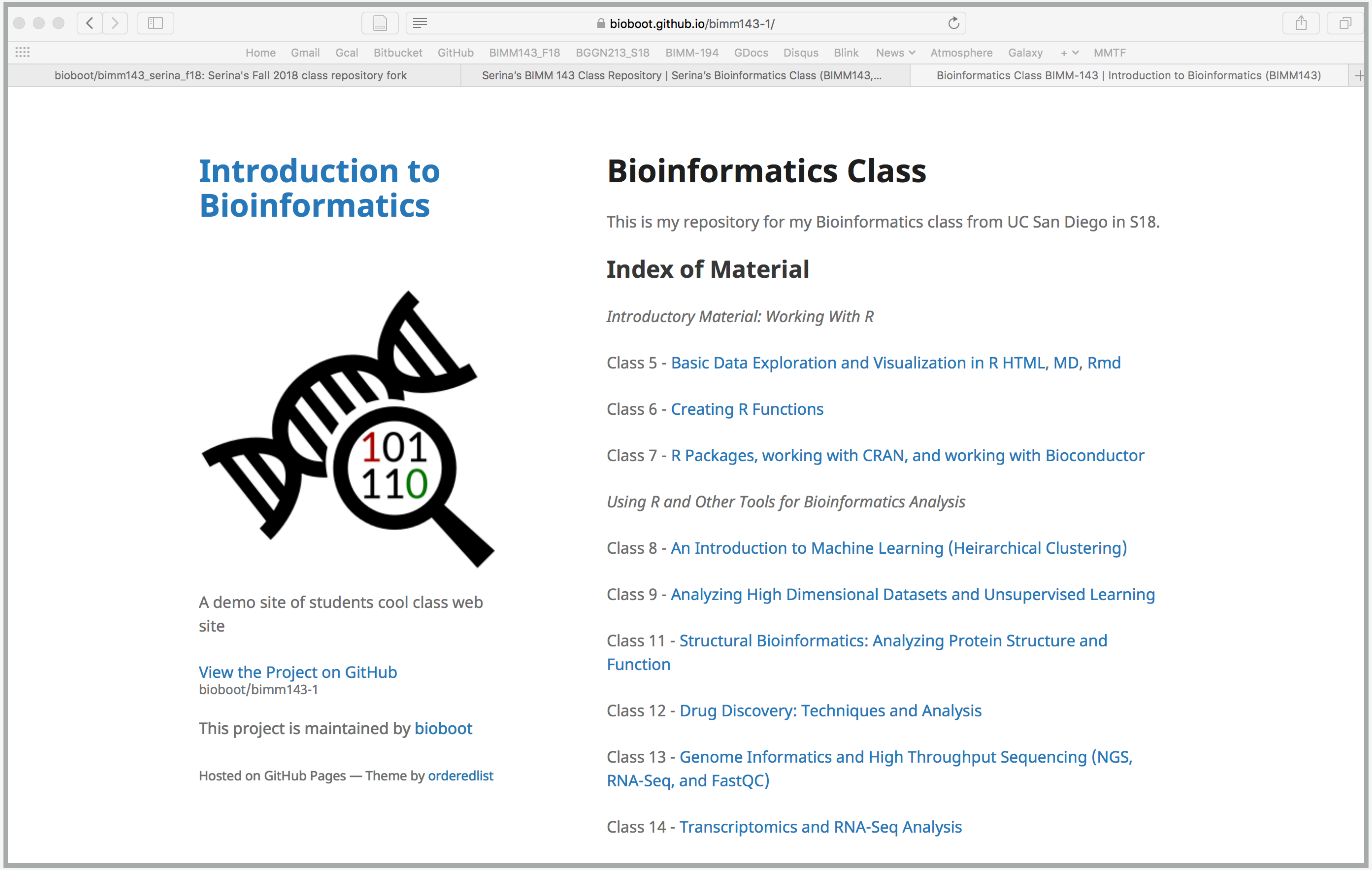
Add an optional extended description…

● Commit directly to the `master` branch.

6

…and you're done!

Fire up a browser and go to **http://*username*.github.io/*repository***.


bioboot.github.io/bimm143-1/
Home
Gmail
Gcal
Bitbucket
GitHub
BIMM143_F18
BGGN213_S18
BIMM-194
GDocs
Disqus
Blink
News
Atmosphere
Galaxy
MMTF
bioboot/bimm143_serina_f18: Serina's Fall 2018 class repository fork
Serina's BIMM 143 Class Repository | Serina's Bioinformatics Class (BIMM143,...
Bioinformatics Class BIMM-143 | Introduction to Bioinformatics (BIMM143)
101
110

Introduction to Bioinformatics

A demo site of students cool class web site

View the Project on GitHub
bioboot/bimm143-1

This project is maintained by bioboot

Hosted on GitHub Pages — Theme by orderedlist

Bioinformatics Class

This is my repository for my Bioinformatics class from UC San Diego in S18.

Index of Material

Introductory Material: Working With R

Class 5 - Basic Data Exploration and Visualization in R HTML, MD, Rmd

Class 6 - Creating R Functions

Class 7 - R Packages, working with CRAN, and working with Bioconductor

Using R and Other Tools for Bioinformatics Analysis

Class 8 - An Introduction to Machine Learning (Heirarchical Clustering)

Class 9 - Analyzing High Dimensional Datasets and Unsupervised Learning

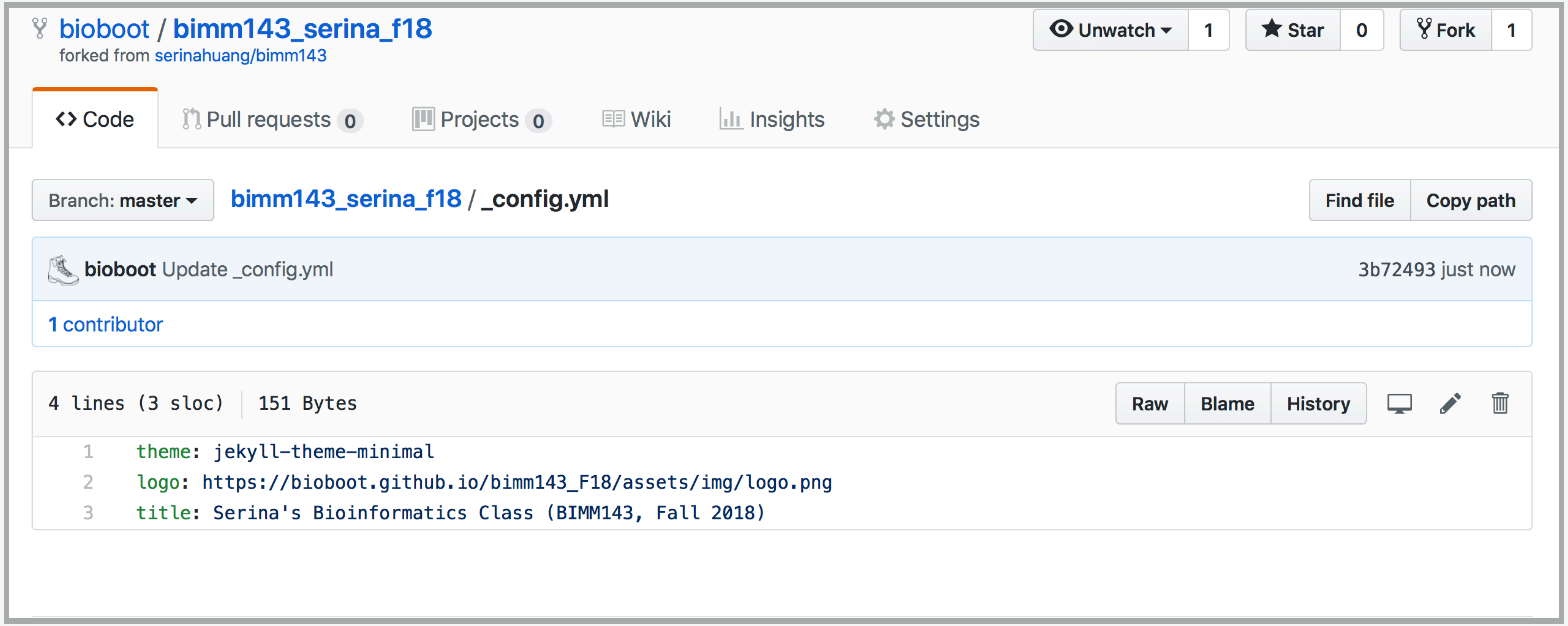
Class 11 - Structural Bioinformatics: Analyzing Protein Structure and Function

Class 12 - Drug Discovery: Techniques and Analysis

Class 13 - Genome Informatics and High Throughput Sequencing (NGS, RNA-Seq, and FastQC)

Class 14 - Transcriptomics and RNA-Seq Analysis

Here I: (1) forked Serina's Repo, (2) Chose the "minimal" theme, (3) Edited _config.yml (adding logo and title)



bioboot / bimm143_serina_f18

forked from serinahuang/bimm143

bimm143_serina_f18 / _config.yml

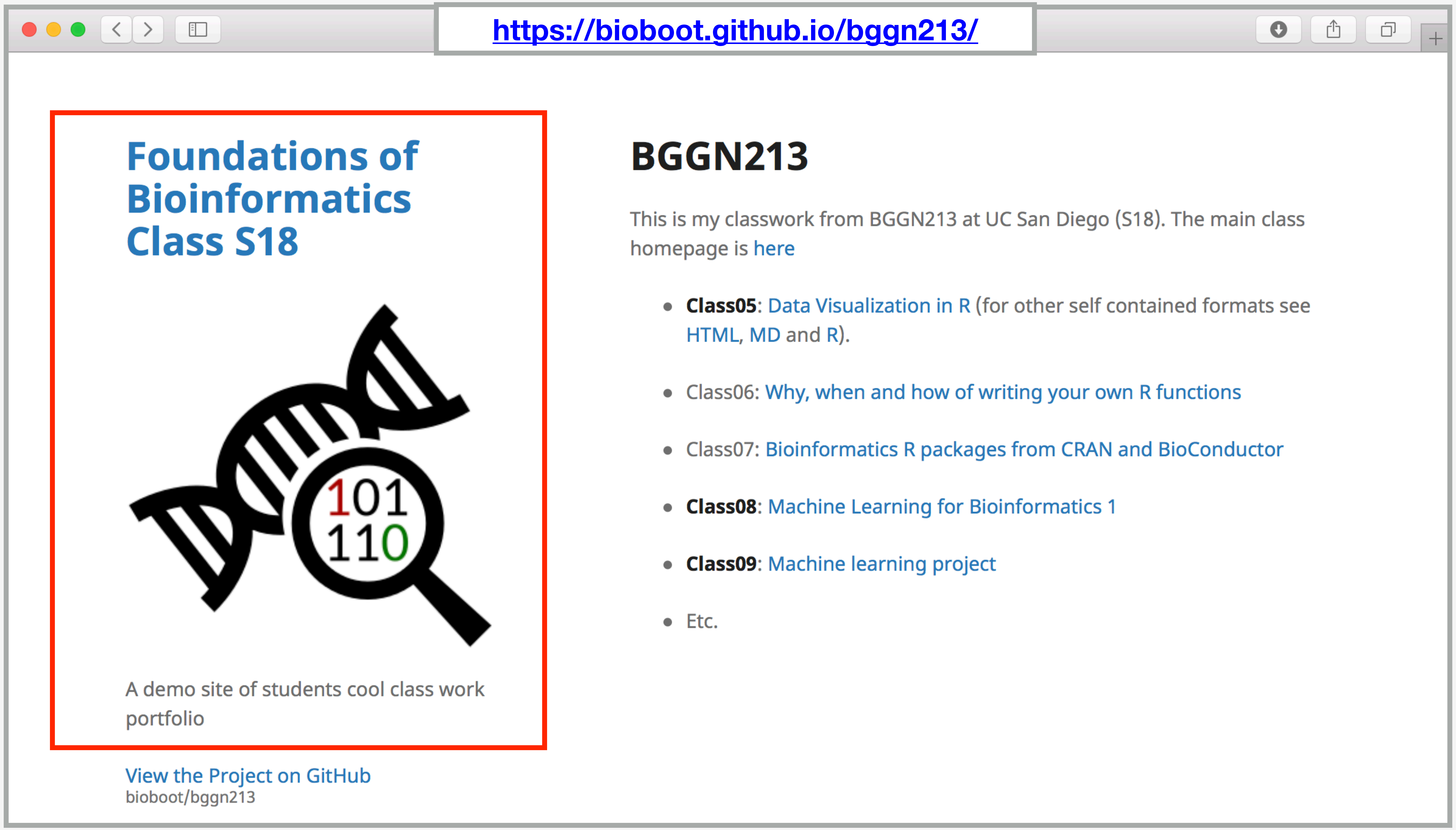
bioboot Update _config.yml 3b72493 just now

1 contributor

4 lines (3 sloc) | 151 Bytes

```yaml
theme: jekyll-theme-minimal
logo: https://bioboot.github.io/bimm143_F18/assets/img/logo.png
title: Serina's Bioinformatics Class (BIMM143, Fall 2018)
```

Here I: (1) Chose the “minimal” theme, (3) Edited _config.yml (adding logo and title), (4) Edited README.md



https://bioboot.github.io/bggn213/

Foundations of Bioinformatics Class S18

A demo site of students cool class work portfolio

View the Project on GitHub
bioboot/bggn213

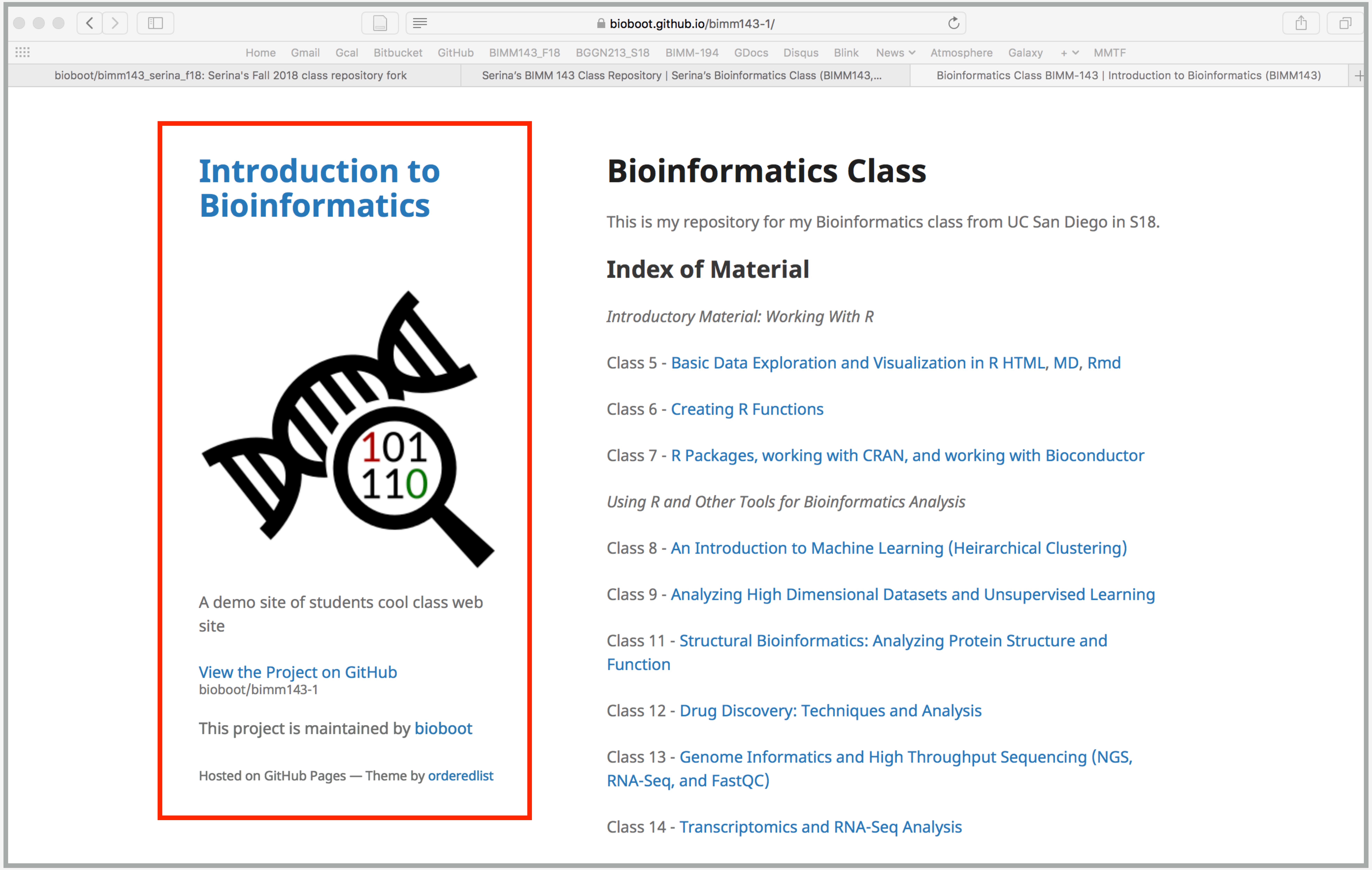
BGGN213

This is my classwork from BGGN213 at UC San Diego (S18). The main class homepage is here

- **Class05**: Data Visualization in R (for other self contained formats see HTML, MD and R).
- Class06: Why, when and how of writing your own R functions
- Class07: Bioinformatics R packages from CRAN and BioConductor
- **Class08**: Machine Learning for Bioinformatics 1
- **Class09**: Machine learning project
- Etc.

Here I: (1) Chose the “minimal” theme, (3) Edited _config.yml (adding logo and title), (4) Edited README.md

https://bioboot.github.io/tmp_test/



bioboot.github.io/bimm143-1/
Home Gmail Gcal Bitbucket GitHub BIMM143_F18 BGGN213_S18 BIMM-194 GDocs Disqus Blink News Atmosphere Galaxy + MMTF
bioboot/bimm143_serina_f18: Serina's Fall 2018 class repository fork
Serina's BIMM 143 Class Repository | Serina's Bioinformatics Class (BIMM143,...
Bioinformatics Class BIMM-143 | Introduction to Bioinformatics (BIMM143)
101
110

Introduction to Bioinformatics

A demo site of students cool class web site

View the Project on GitHub
bioboot/bimm143-1

This project is maintained by bioboot

Hosted on GitHub Pages — Theme by orderedlist

Bioinformatics Class

This is my repository for my Bioinformatics class from UC San Diego in S18.

Index of Material

Introductory Material: Working With R

Class 5 - Basic Data Exploration and Visualization in R HTML, MD, Rmd

Class 6 - Creating R Functions

Class 7 - R Packages, working with CRAN, and working with Bioconductor

Using R and Other Tools for Bioinformatics Analysis

Class 8 - An Introduction to Machine Learning (Heirarchical Clustering)

Class 9 - Analyzing High Dimensional Datasets and Unsupervised Learning

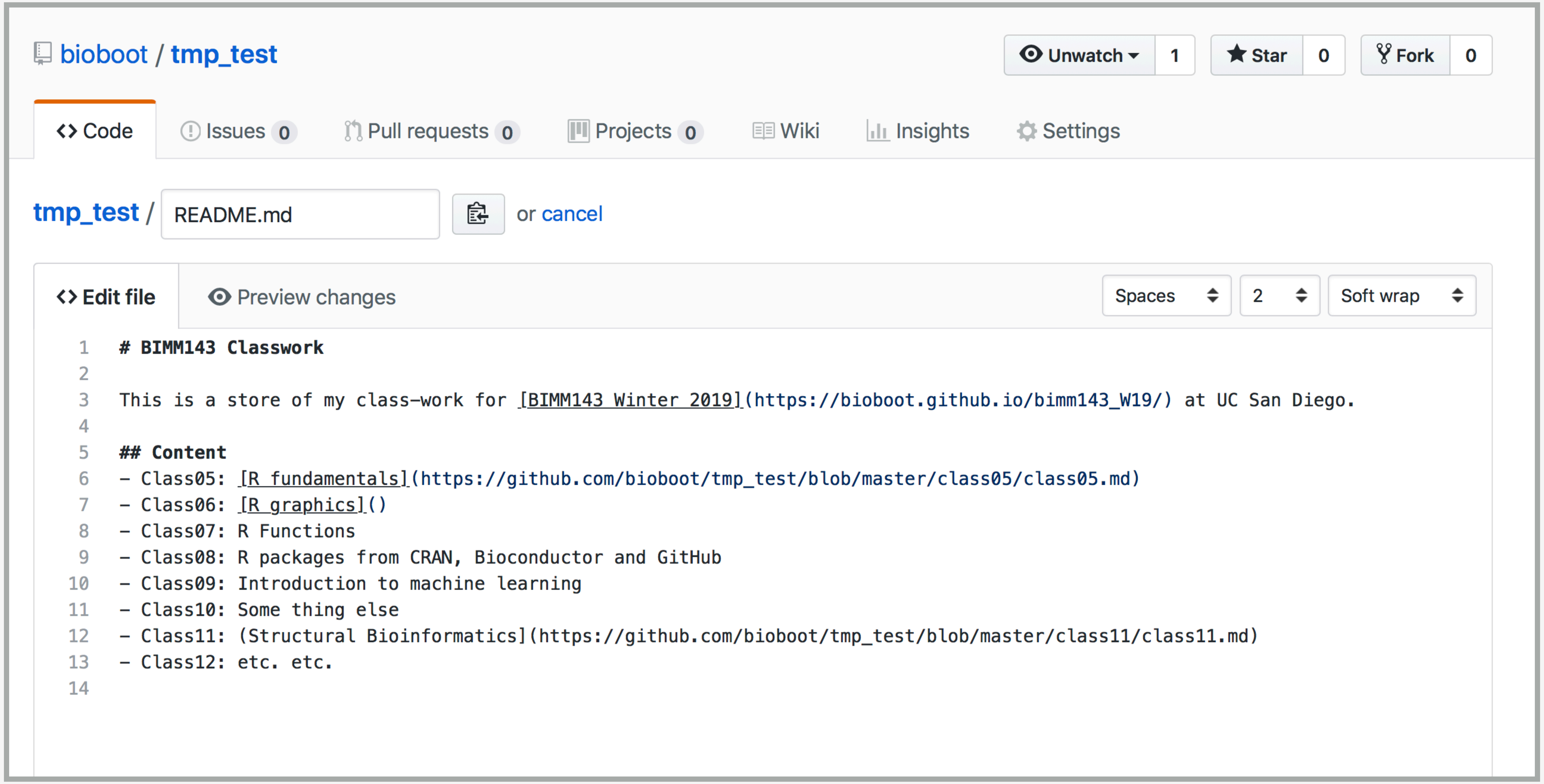
Class 11 - Structural Bioinformatics: Analyzing Protein Structure and Function

Class 12 - Drug Discovery: Techniques and Analysis

Class 13 - Genome Informatics and High Throughput Sequencing (NGS, RNA-Seq, and FastQC)

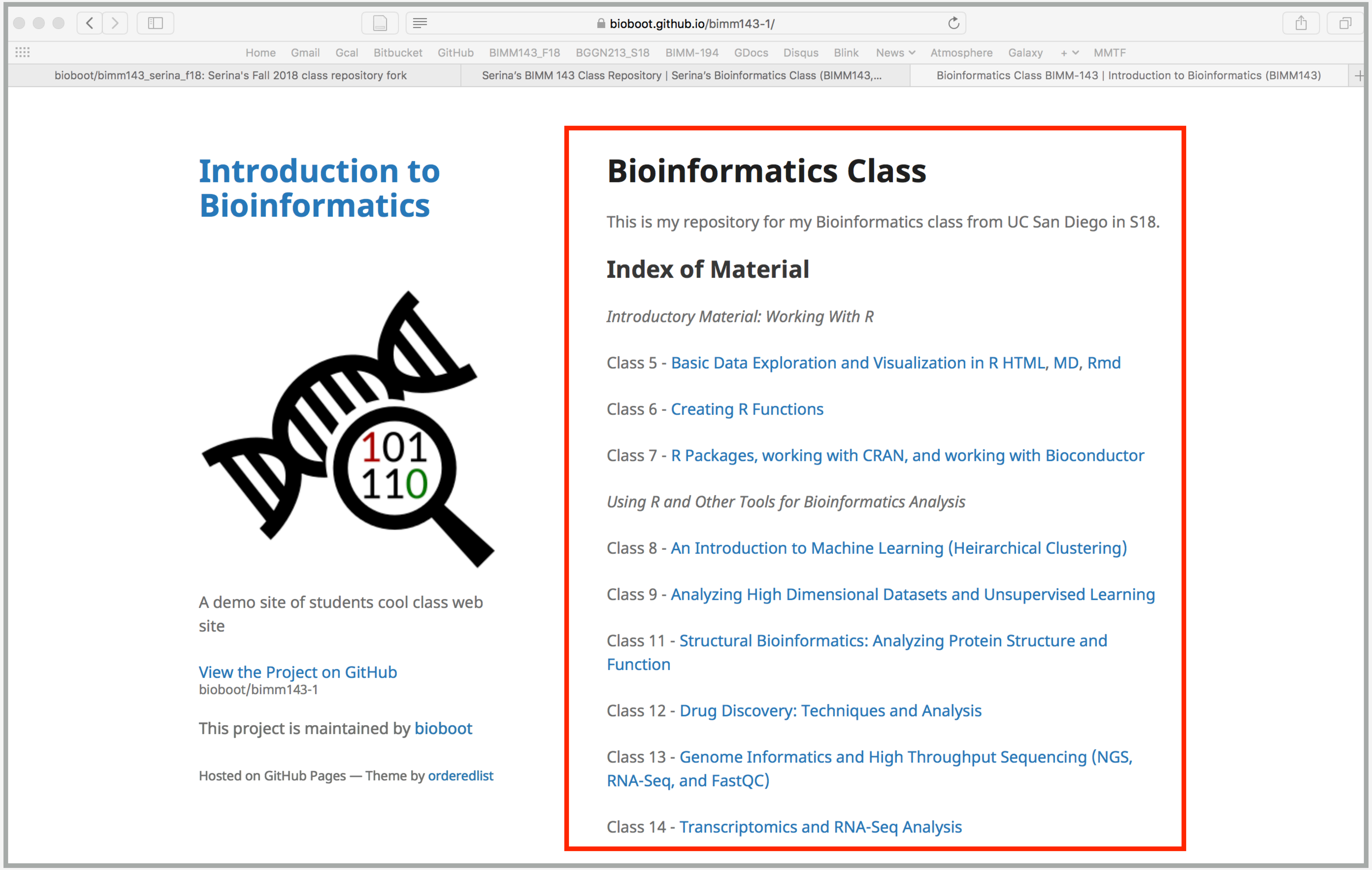
Class 14 - Transcriptomics and RNA-Seq Analysis

Here I: (1) Chose the "minimal" theme, (3) Edited _config.yml (adding logo and title), (4) Edited README.md



```
# BIMM143 Classwork

This is a store of my class-work for [BIMM143 Winter 2019](https://bioboot.github.io/bimm143_W19/) at UC San Diego.

## Content
- Class05: [R fundamentals](https://github.com/bioboot/tmp_test/blob/master/class05/class05.md)
- Class06: [R graphics]()
- Class07: R Functions
- Class08: R packages from CRAN, Bioconductor and GitHub
- Class09: Introduction to machine learning
- Class10: Some thing else
- Class11: (Structural Bioinformatics](https://github.com/bioboot/tmp_test/blob/master/class11/class11.md)
- Class12: etc. etc.

```

Here I: (1) Chose the “minimal” theme, (3) Edited _config.yml (adding logo and title), (4) Edited README.md


bioboot.github.io/bimm143-1/
Home
Gmail
Gcal
Bitbucket
GitHub
BIMM143_F18
BGGN213_S18
BIMM-194
GDocs
Disqus
Blink
News
Atmosphere
Galaxy
MMTF
bioboot/bimm143_serina_f18: Serina's Fall 2018 class repository fork
Serina's BIMM 143 Class Repository | Serina's Bioinformatics Class (BIMM143,...
Bioinformatics Class BIMM-143 | Introduction to Bioinformatics (BIMM143)
101
110

Introduction to Bioinformatics

A demo site of students cool class web site

View the Project on GitHub
bioboot/bimm143-1

This project is maintained by bioboot

Hosted on GitHub Pages — Theme by orderedlist

Bioinformatics Class

This is my repository for my Bioinformatics class from UC San Diego in S18.

Index of Material

Introductory Material: Working With R

Class 5 - Basic Data Exploration and Visualization in R HTML, MD, Rmd

Class 6 - Creating R Functions

Class 7 - R Packages, working with CRAN, and working with Bioconductor

Using R and Other Tools for Bioinformatics Analysis

Class 8 - An Introduction to Machine Learning (Heirarchical Clustering)

Class 9 - Analyzing High Dimensional Datasets and Unsupervised Learning

Class 11 - Structural Bioinformatics: Analyzing Protein Structure and Function

Class 12 - Drug Discovery: Techniques and Analysis

Class 13 - Genome Informatics and High Throughput Sequencing (NGS, RNA-Seq, and FastQC)

Class 14 - Transcriptomics and RNA-Seq Analysis

Here I: (1) forked Serina's Repo, (2) Chose the "minimal" theme, (3) Edited _config.yml (adding logo and title)

Thank you very much!

Please do fill out your CAPs evaluation (Link!) if you get a change.
It is important to the courses we offer in the future and how we teach them!

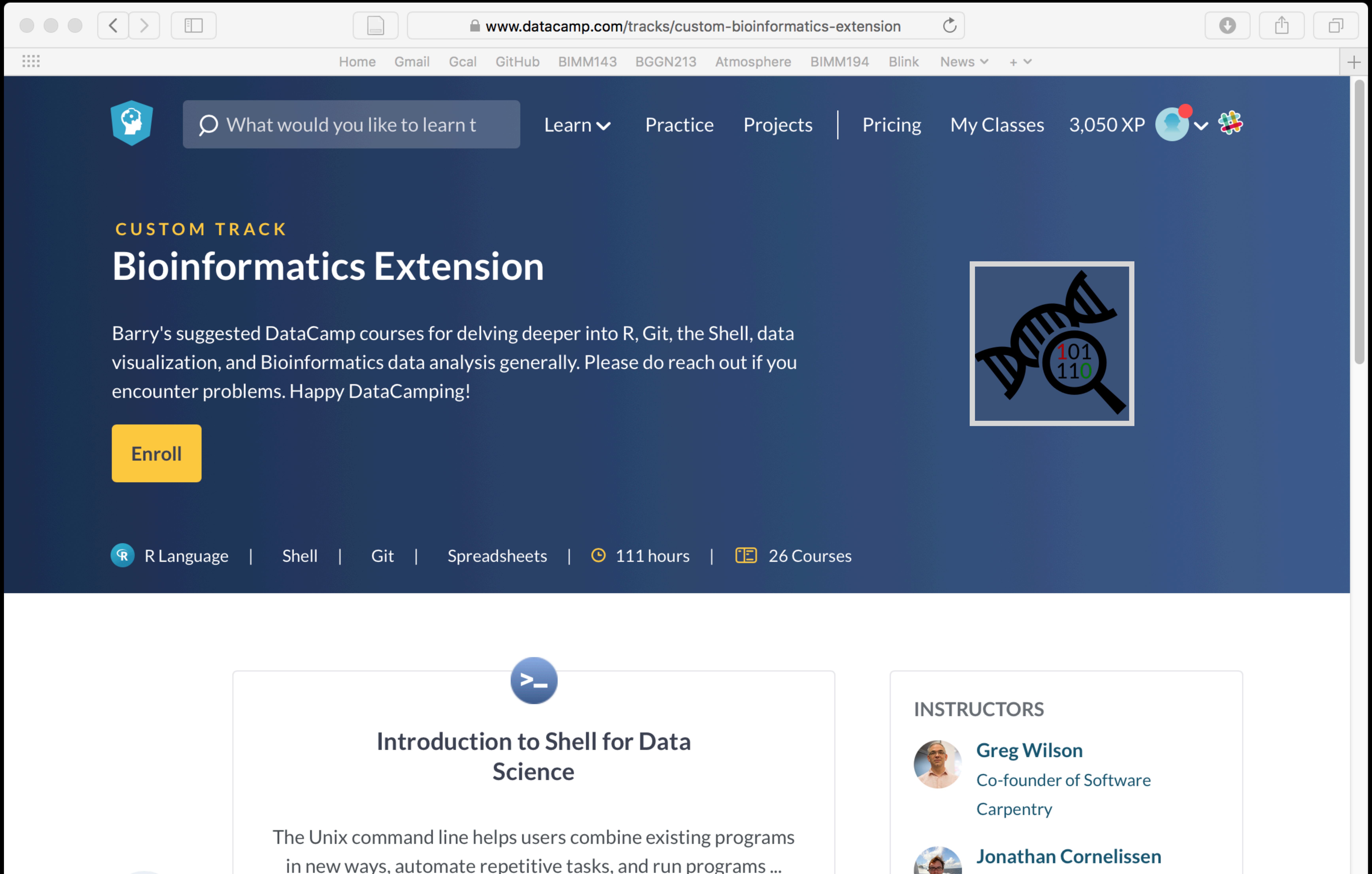
Thank you very much!

Please do fill out your CAPs evaluation (Link!) if you get a change.
It is important to the courses we offer in the future and how we teach them!

Post to GradeScope your GitHub Pages portfolio URL with all classes by this Friday and receive another 5pts credit!

Going Further With DataCamp

https://bioboot.github.io/bimm143_F19/class-material/datacamp_extras.pdf



www.datacamp.com/tracks/custom-bioinformatics-extension
Home Gmail Gcal GitHub BIMM143 BGGN213 Atmosphere BIMM194 Blink News
What would you like to learn t
Learn
Practice
Projects
Pricing
My Classes
3,050 XP

CUSTOM TRACK

Bioinformatics Extension

Barry's suggested DataCamp courses for delving deeper into R, Git, the Shell, data visualization, and Bioinformatics data analysis generally. Please do reach out if you encounter problems. Happy DataCamping!

Enroll

R Language | Shell | Git | Spreadsheets | 111 hours | 26 Courses

Introduction to Shell for Data Science

The Unix command line helps users combine existing programs in new ways, automate repetitive tasks, and run programs ...

INSTRUCTORS

Greg Wilson
Co-founder of Software Carpentry

Jonathan Cornelissen

Thank you very much!

Please do fill out your CAPs evaluation (Link!) if you get a change.
It is important to the courses we offer in the future and how we teach them!

Post to GradeScope your GitHub Pages portfolio URL with all classes by this Friday and receive another 5pts credit!

Announcement!

Bonus:

Bioinformatics & Genomics in Industry

Live Stram Video

Enjoy a set of short open ended guest lectures from leading genomic scientists at **Illumina Inc.**, **Synthetic Genomics Inc.**, and the **La Jolla Institute for Allergy and Immunology**. Feel free to contact these scientists for networking and to have your questions about industry careers in Bioinformatics and Genomics answered.

Learning Resources

- **Set up Git**. If you will be using Git mostly or entirely via **GitHub**, look at these how-tos.
  < https://help.github.com/categories/bootcamp/ >

- **Getting Git Right**. Excellent **Bitbucket** git tutorials
  < https://www.atlassian.com/git/ >

- **Pro Git**. A complete, book-length guide and reference to Git, by Scott Chacon and Ben Straub.
  < http://git-scm.com/book/en/v2 >

- **StackOverflow**. Excellent programming and developer Q&A.
  < http://stackoverflow.com/questions/tagged/git >

Learning git can be painful!

However in practice it is not nearly as crazy-making as the alternatives:

- Documents as email attachments
- Hair-raising ZIP archives containing file salad
- Am I working with the most recent data?
- Archaelogical “digs” on old email threads and uncertainty about how/if certain changes have been made or issues solved

Finally Please remember that **GitHub** and **BitBucket** are **<u>PUBLIC</u>** and that you should cultivate your professional and scholarly profile with intention!

The End!

http://thegrantlab.org

Thank You!

http://thegrantlab.org

Reference Slides

Command Line GIT

Using Command Line Git

1. Initiate a Git repository.
2. Edit content (i.e. change some files).
3. Store a 'snapshot' of the current file state.*

Initiate a Git repository

Do it Yourself!

Initiate a Git repository

```
> cd ~/Desktop
> mkdir git_class   # Make a new directory
> cd git_class      # Change to this directory
> git init          # Our first Git command!
> ls -a             # what happened?
```

Side-Note: The .git/ directory

- Git created a ‘hidden’ **.git/** directory inside your current working directory.

- You can use the ‘**ls -a**’ command to list (*i.e.* see) this directory and its contents.

- This is where Git stores all its goodies - **this is Git!**

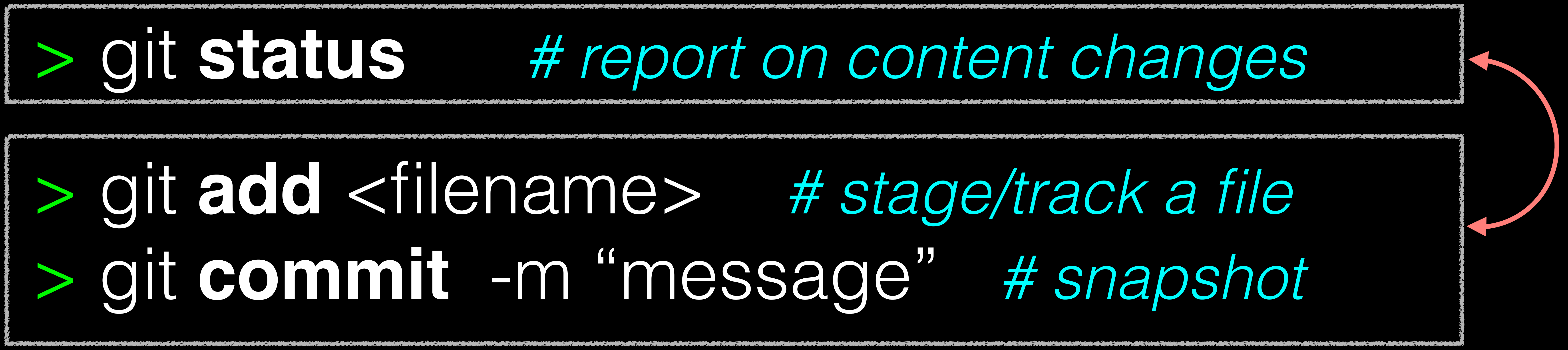
- You should not need to edit the contents of the **.git** directory for now but do feel free to poke around.

Important Git commands

```
> git status        # report on content changes
```

```
> git add <filename>       # stage/track a file
> git commit  -m “message”    # snapshot
```

Important Git commands



```
> git status        # report on content changes
```

```
> git add <filename>      # stage/track a file
> git commit  -m “message”    # snapshot
```

You will use these three commands again and again in your Git workflow!

Git TRACKS your directory content

- To get a report of changes (since last commit) use:

  > **git status**

- You tell Git which files to track with:

  > **git add <filename>**

  This adds files to a so called **STAGING AREA** (akin to a “shopping cart” before purchasing).

- You tell Git when to take an historical **SNAPSHOT** of your staged files (*i.e.* record their current state) with:

  > **git commit -m ‘Your message about changes’**

Example Git workflow

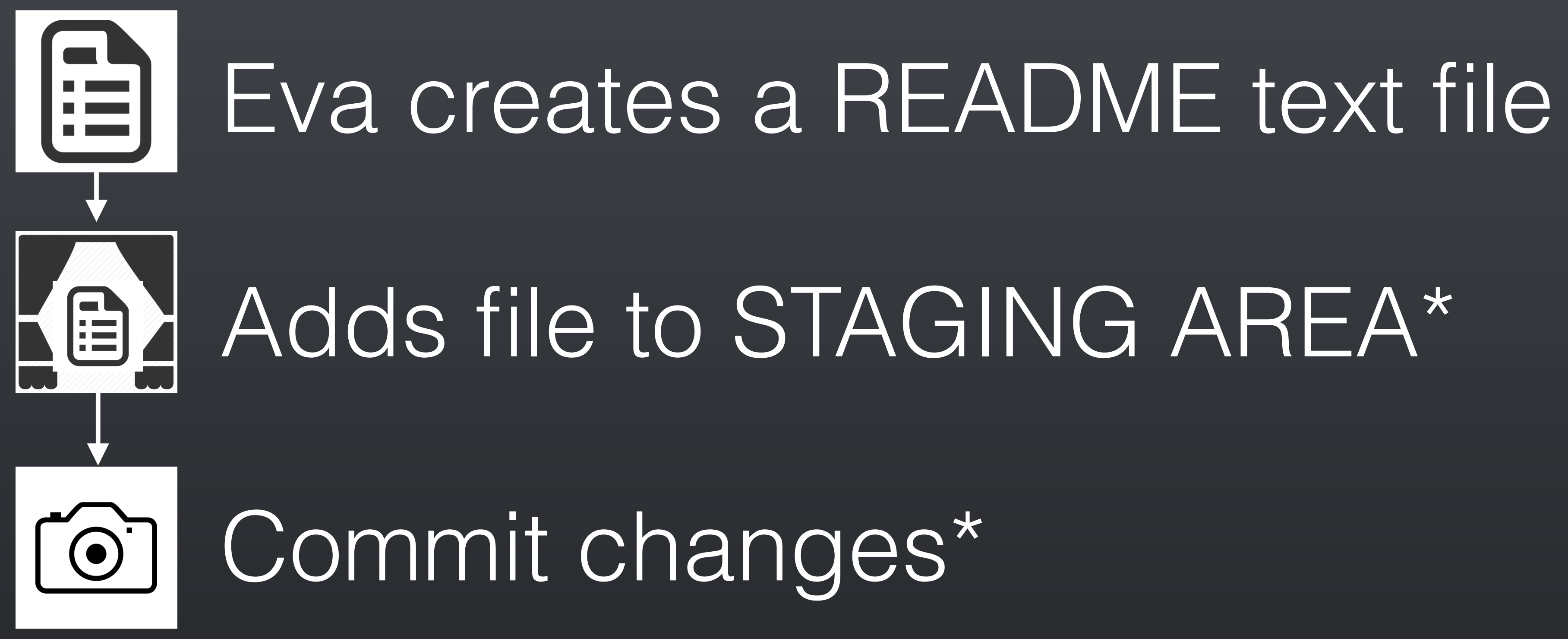
Eva creates a README text file
(this starts as untracked)

Adds file to STAGING AREA*
(tracked and ready to take a snapshot)

Commit changes*
(records snapshot of staged files!)

Hands on example!

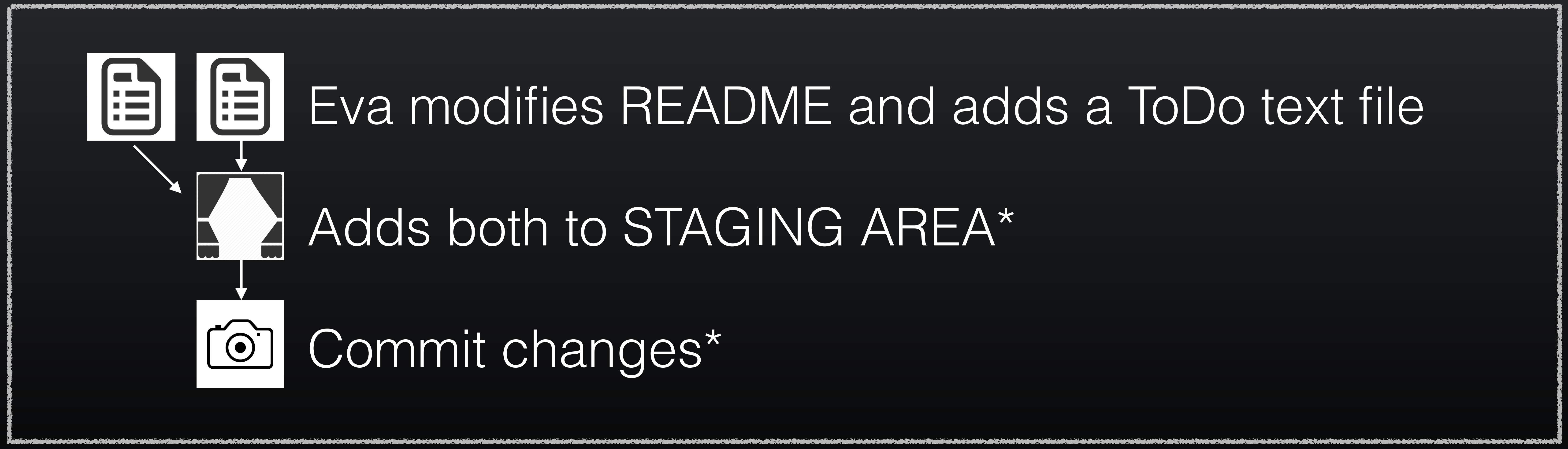
Example Git workflow



Eva creates a README text file

Adds file to STAGING AREA*

Commit changes*



Eva modifies README and adds a ToDo text file

Adds both to STAGING AREA*

Commit changes*

Do it Yourself!

1. Eva creates a README file

```
> # cd ~/Desktop/git_class
> # git init

> echo "This is a first line of text." > README
> git status          # Report on changes
# On branch master
#
# Initial commit
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#     README
#
# nothing added to commit but untracked files present (use "git add" to track)
```

2. Adds to ‘staging area’

```
> git add README      # Add README file to staging area
> git status          # Report on changes

# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#     new file:   README
#
```

3. Commit changes

```
> git commit -m “Create a README file”     # Take snapshot
# [master (root-commit) 8676840] Create a README file
#  1 file changed, 1 insertion(+)
#  create mode 100644 README


> git status                   # Report on changes
# On branch master
# nothing to commit, working directory clean
```

4. Eva modifies README file and adds a ToDo file

```
> echo "This is a 2nd line of text." >> README
> echo "Learn git basics" >> ToDo

> git status          # Report on changes
# 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:   README
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#     ToDo
#
# no changes added to commit (use "git add" and/or "git commit -a")
```

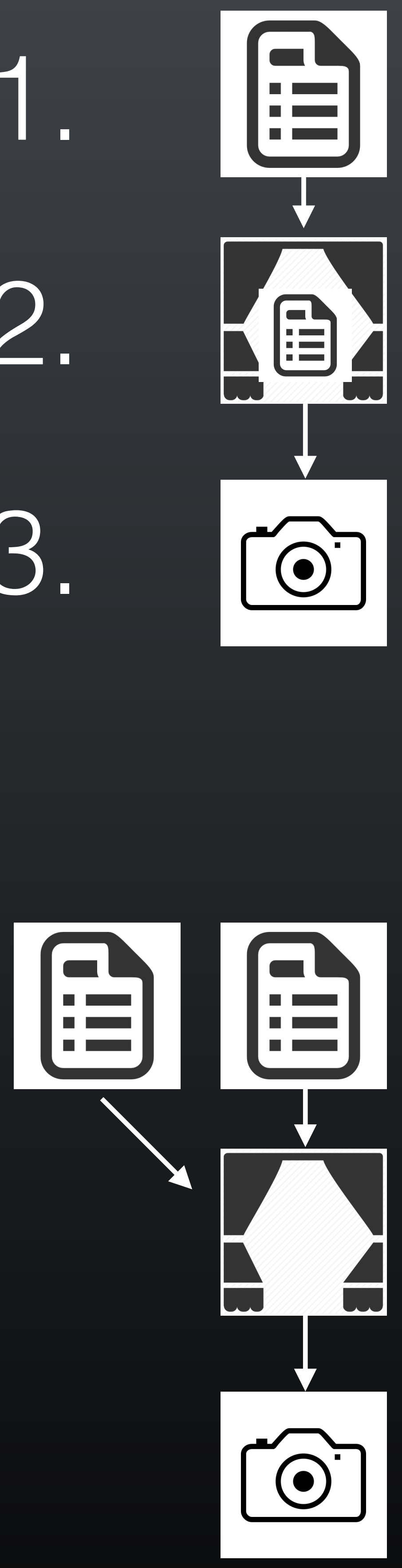
5. Adds both files to ‘staging area’

```
> git add README ToDo     # Add both files to ‘staging area’
> git status              # Report on changes

# On branch master
# Changes to be committed:
#   (use "git reset HEAD <file>..." to unstage)
#
#    modified:   README
#      new file:   ToDo
#
```

6. Commits changes

```
> git commit -m "Add ToDo and modify README"
# [master 7b679fa] Add ToDo and modify README
#  2 files changed, 2 insertions(+)
#  create mode 100644 ToDo

> git status
# On branch master
# nothing to commit, working directory clean
```

Example Git workflow



1. Eva creates a README text file
2. Adds file to STAGING AREA*
3. Commit changes*

4. Eva modifies README and adds a ToDo text file
5. Adds both to STAGING AREA*
6. Commit changes*

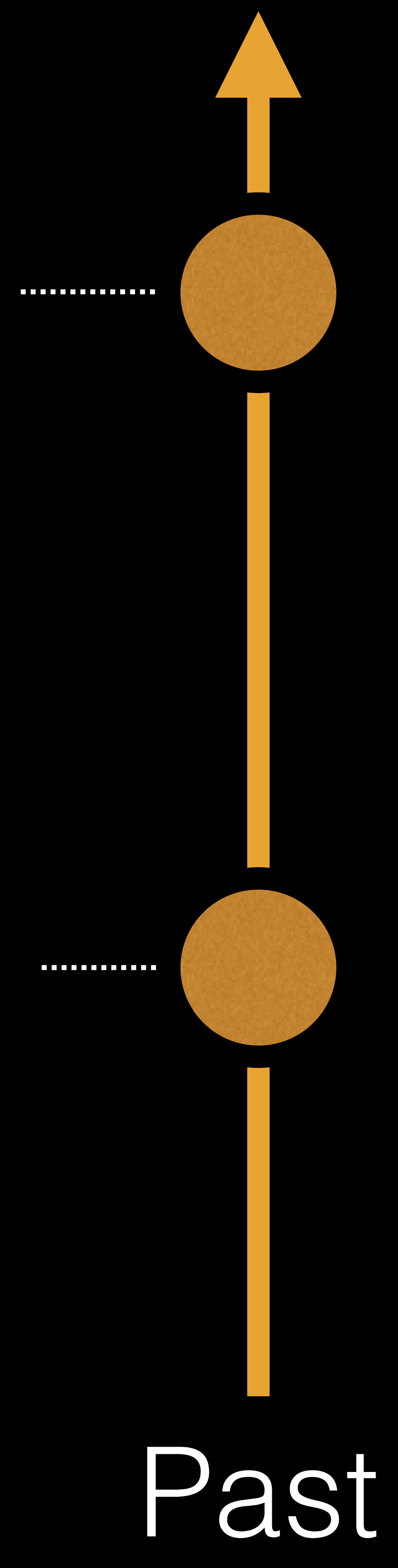
…But, how do we see the history of our project changes?

git log: Timeline history of snapshots (*i.e.* commits)

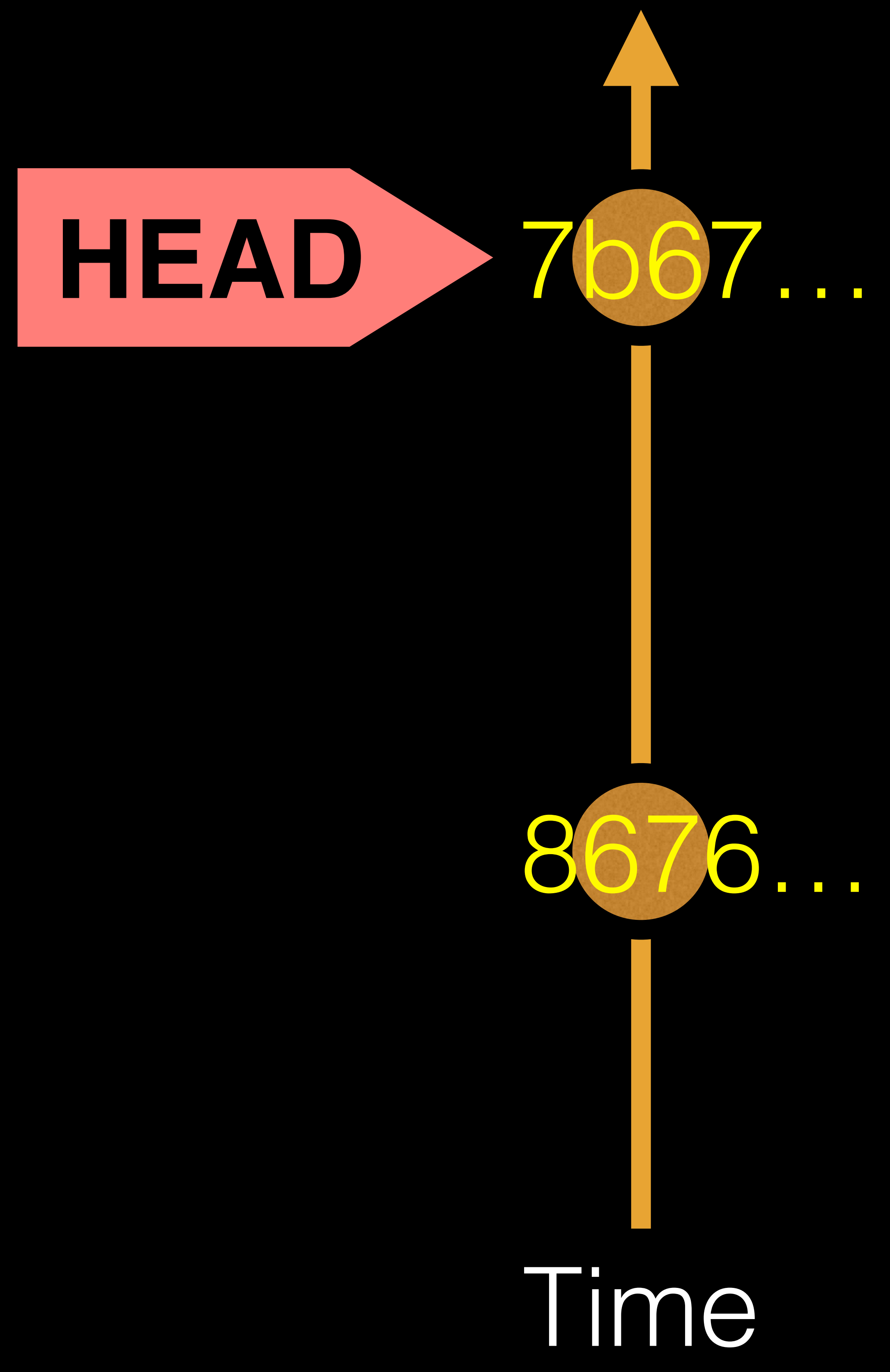
```
> git log
# commit 7b679fa747e8640918fcaad7e4c3f9c70c87b170
# Author: Barry Grant <bjgrant@umich.edu>
# Date:   Thu Jul 30 11:43:40 2015 -0400
#
#    Add ToDo and finished README
#
# commit 86768401610770ae32e2fd4faee07d1d5c68619c
# Author: Barry Grant <bjgrant@umich.edu>
# Date:   Thu Jul 30 11:26:40 2015 -0400
#
#    Create a README file
#
```

git log: Timeline history of snapshots (*i.e.* commits)

```
> git log
# commit 7b679fa747e8640918fcaad7e4c3f9c70c87b170
# Author: Barry Grant <bjgrant@umich.edu>
# Date:   Thu Jul 30 11:43:40 2015 -0400
#
#    Add ToDo and finished README
#
# commit 86768401610770ae32e2fd4faee07d1d5c68619c
# Author: Barry Grant <bjgrant@umich.edu>
# Date:   Thu Jul 30 11:26:40 2015 -0400
#
#    Create a README file
#
```



Side-Note: Git history is akin to a graph

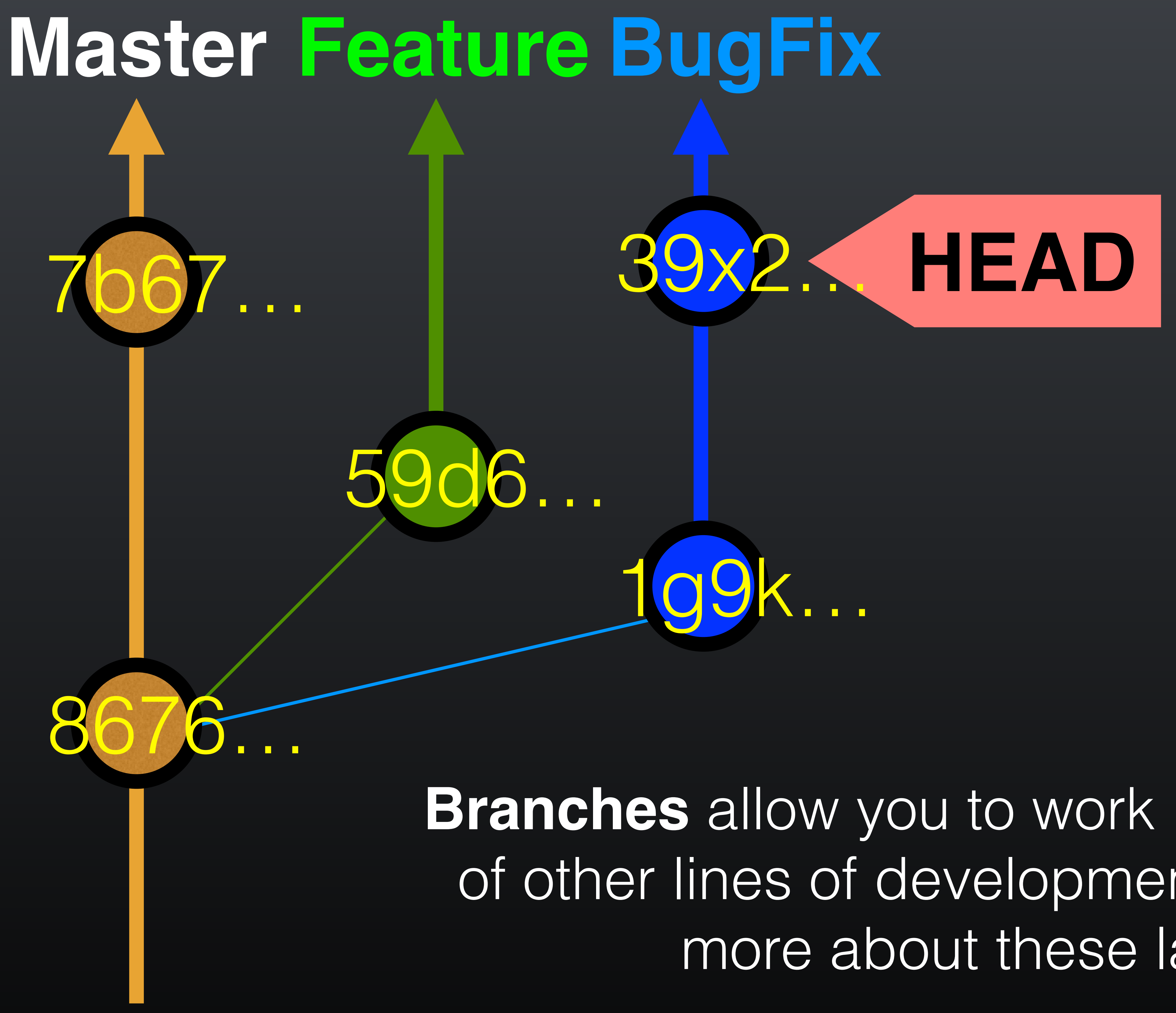


Nodes are **commits** labeled by their unique '**commit ID**'.

(This is a CHECKSUM of the commits author, time, commit msg, commit content and previous commit ID).

HEAD is a reference (or '**pointer**') to the currently checked out commit (typically the most recent commit).

Projects can have complicated graphs due to **branching**



Branches allow you to work independently of other lines of development we will talk more about these later!

Key Points:

You explicitly and iteratively tell git what files to track (“**git add**”) and snapshot (“**git commit**”).

Git keeps an historical log “(**git log**”) of the content changes (and your comments on these changes) at each past commit.

It is good practice to regularly check the status of your working directory, staging arena repo (“**git status**“)

Break

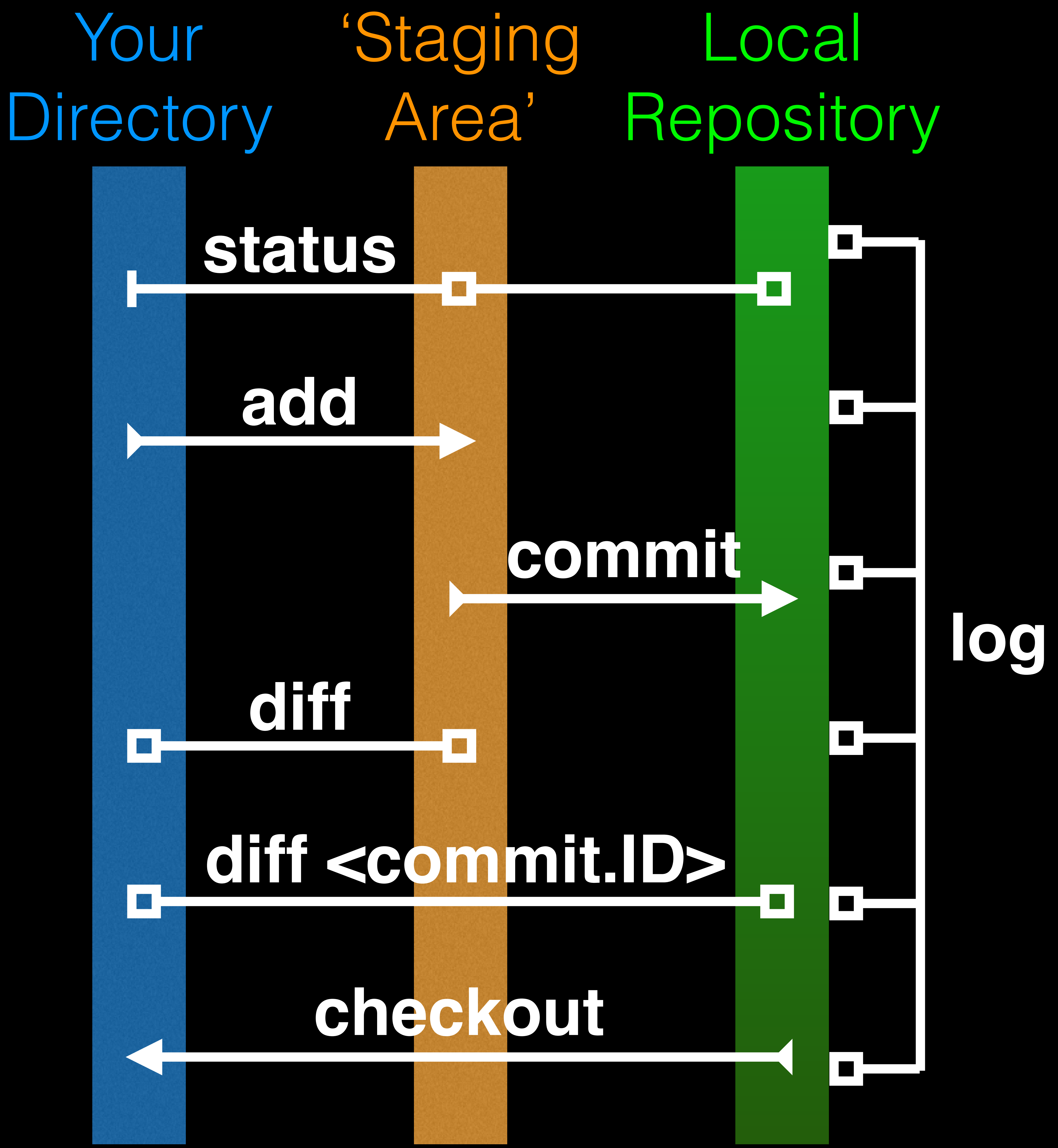
Summary of key Git commands:

```
> git status        # Get a status report of changes since last commit
```

```
> git add <filename>                # Tell Git which files to track/stage

> git commit -m ‘Your message’      # Take a content snapshot!
```

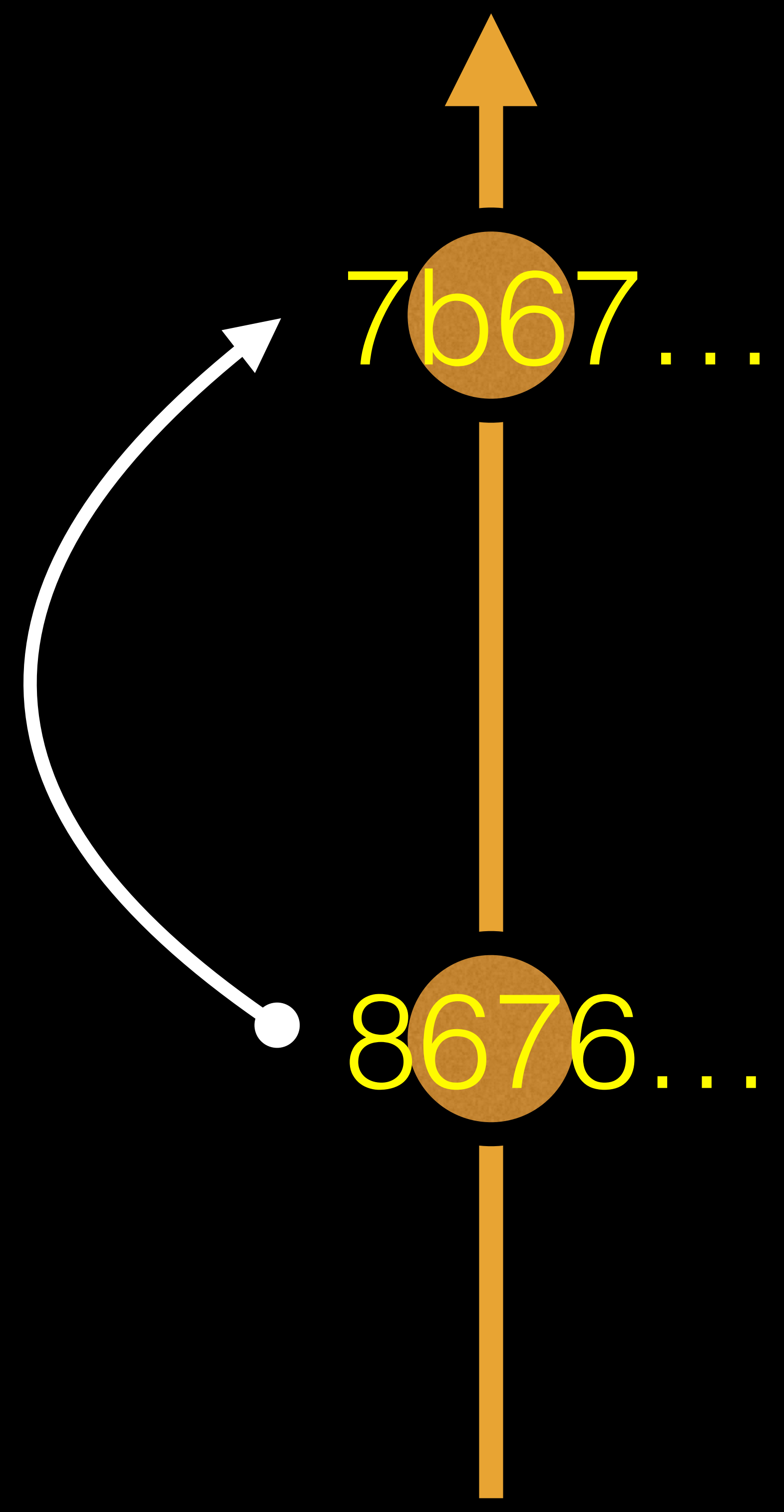
```
> git log                           # Review your commit history
```

```
> git diff <commit.ID>  <commit.ID> # Inspect content differences

> git checkout  <commit.ID>  # Navigate through the commit history
```


Your Directory
‘Staging Area’
Local Repository
status
add
commit
diff
diff <commit.ID>
checkout
log

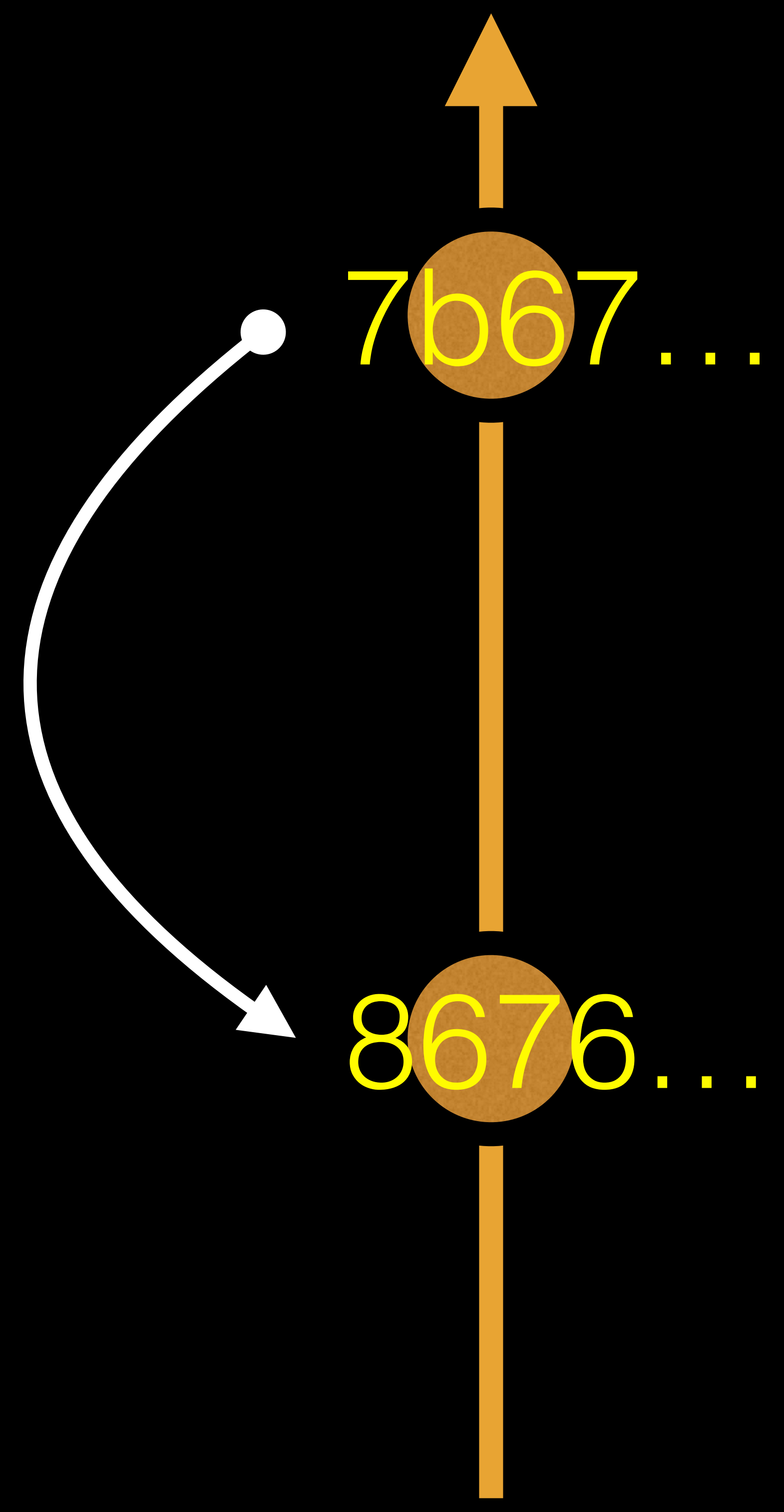
git diff: Show changes between commits

```
> git diff 8676 7b67
# diff --git a/README b/README
# index 73bc85a..67bd82c 100644
# --- a/README
# +++ b/README
# @@ -1 +1,2 @@
#  This is a first line of text.
# +This is a 2nd line of text.

# diff --git a/ToDo b/ToDo
# new file mode 100644
# index 0000000..14fbd56
# --- /dev/null
# +++ b/ToDo
# @@ -0,0 +1 @@
# +Learn git basics
```



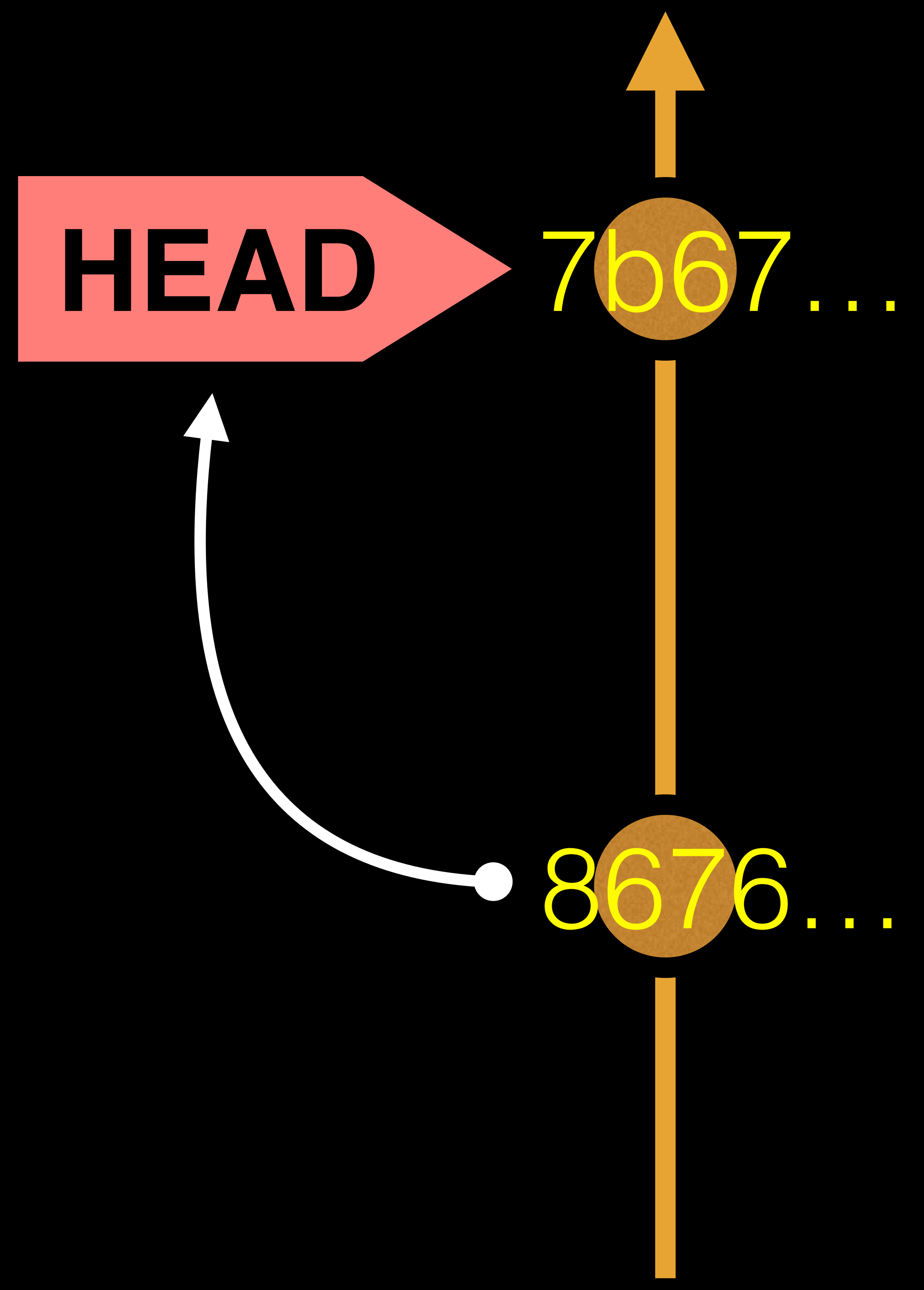
git diff: Show changes between commits

```
> git diff 7b67 8676
# diff --git a/README b/README
# index 67bd82c..73bc85a 100644
# --- a/README
# +++ b/README
# @@ -1,2 +1 @@
#  This is a first line of text.
# -This is a 2nd line of text.

# diff --git a/ToDo b/ToDo
# deleted file mode 100644
# index 14fbd56..0000000
# --- a/ToDo
# +++ /dev/null
# @@ 1 +0,0 @@
# -Learn git basics
```

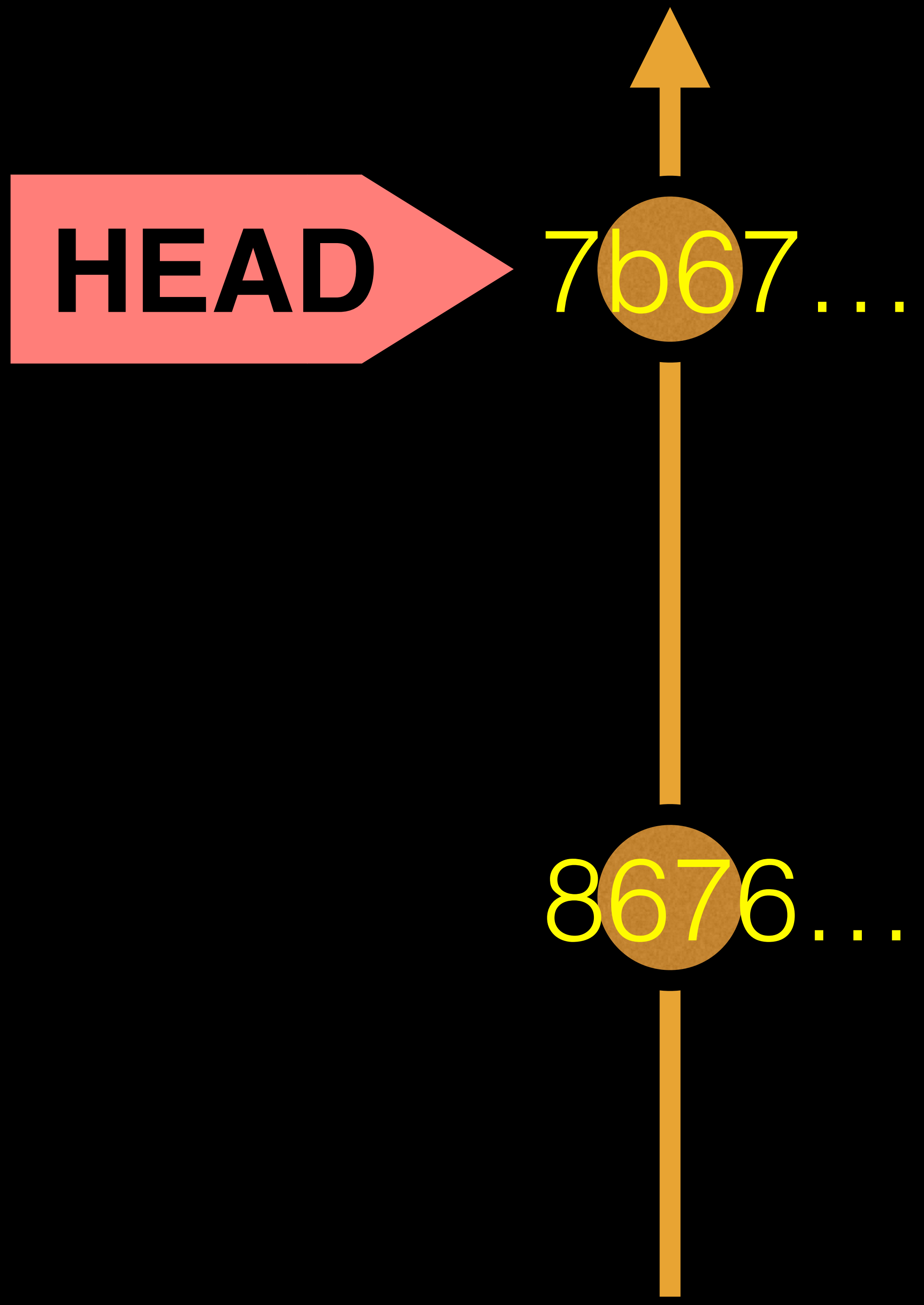


git diff: Show changes between commits

```
> git diff 8676                ## Difference to current HEAD position!

# diff --git a/README b/README
# index 73bc85a..67bd82c 100644
# --- a/README
# +++ b/README
# @@ -1 +1,2 @@
#  This is a first line of text.
# +This is a 2nd line of text.

# diff --git a/ToDo b/ToDo
# new file mode 100644
# index 0000000..14fbd56
# --- /dev/null
# +++ b/ToDo
# @@ -0,0 +1 @@
# +Learn git basics
```



HEAD advances automatically with each new commit

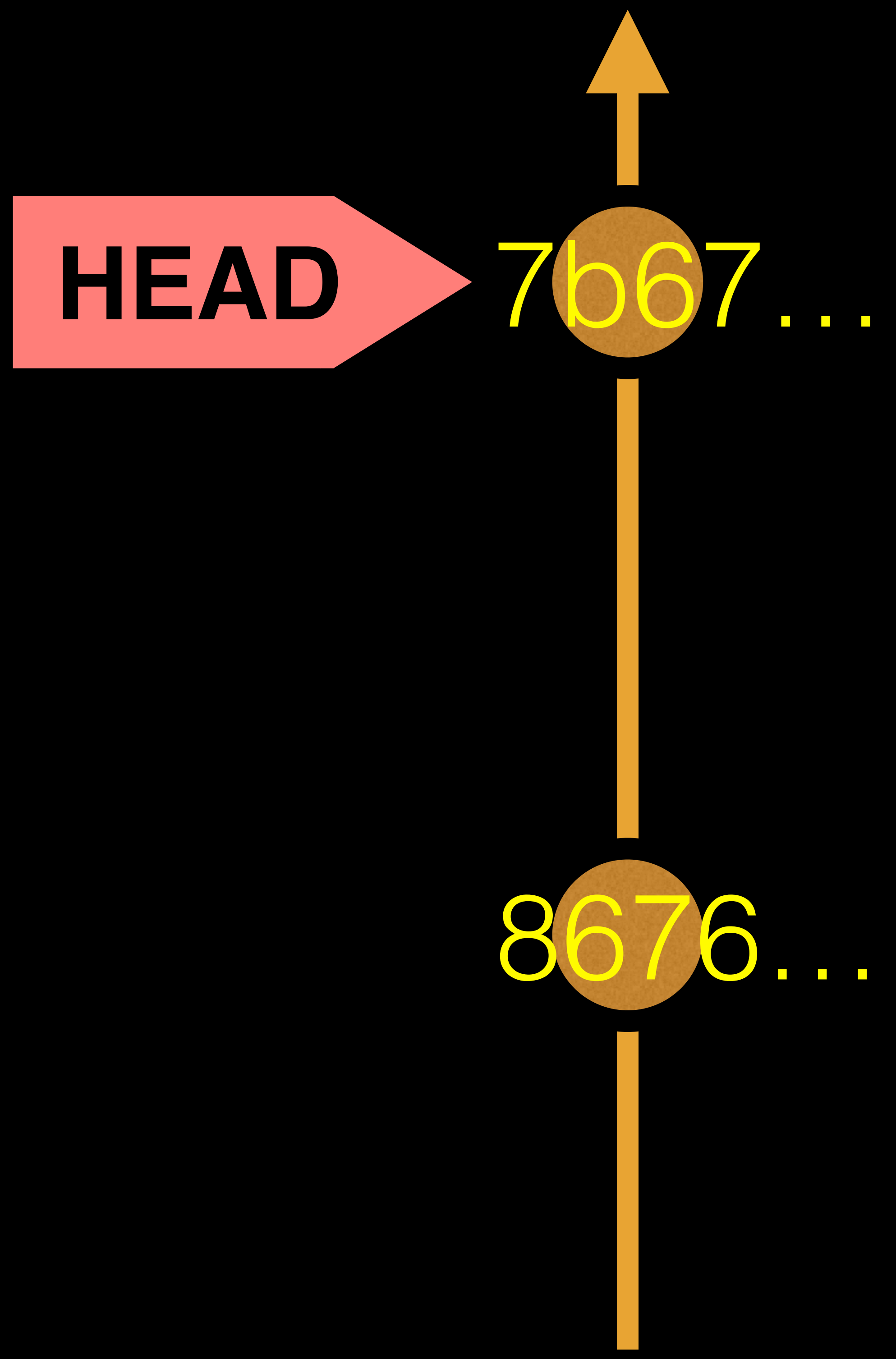


To move **HEAD** (back or forward) on the Git graph (and retrieve the associated snapshot content) we can use the command:

```
> git checkout <commit.ID>
```

git checkout: Moves HEAD

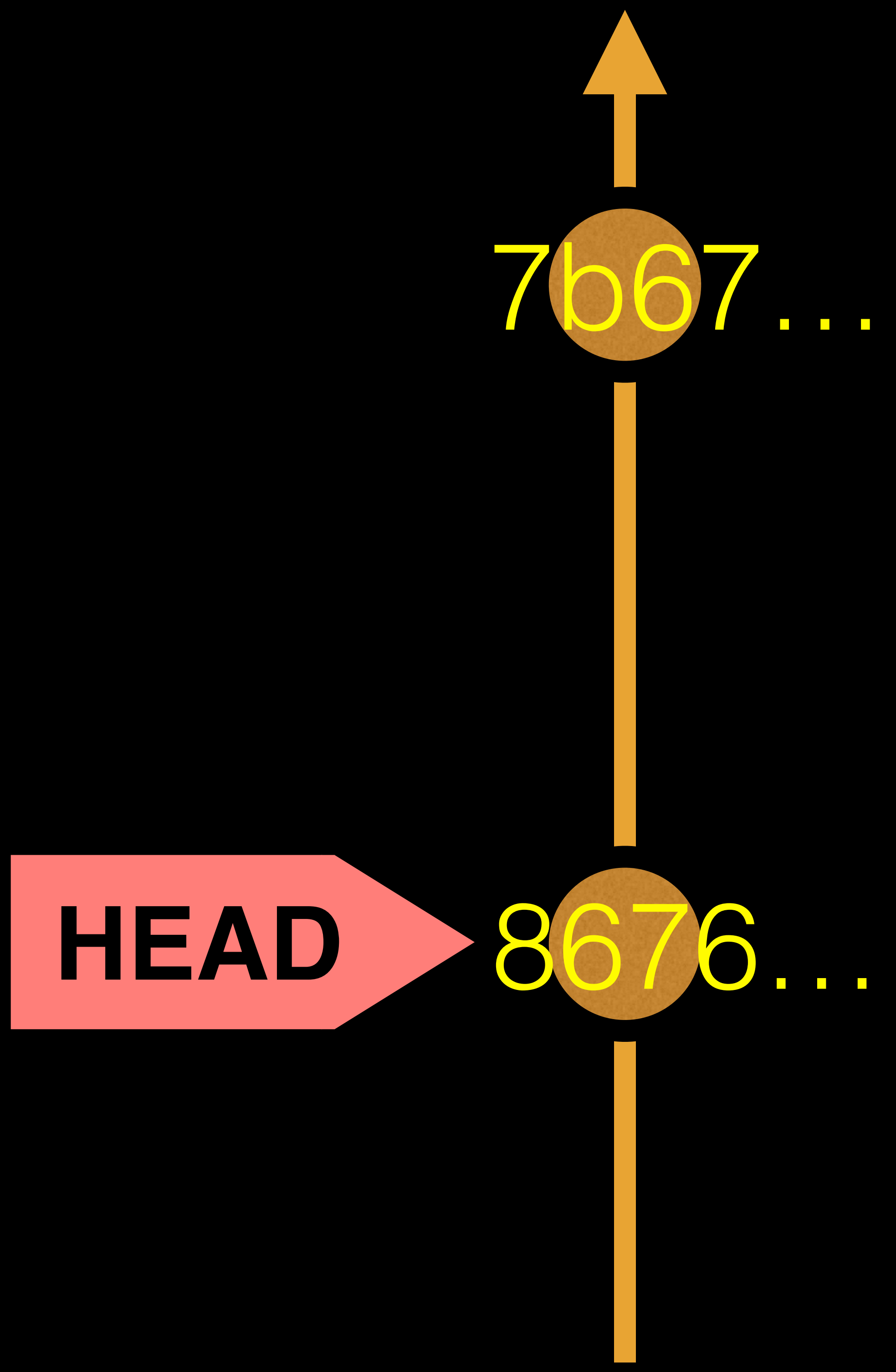
```
> more README
This is a first line of text.
This is a 2nd line of text.

> git log --oneline
# 7b679fa  Add ToDo and finished README
# 8676840  Create a README file
```



Do it Yourself!

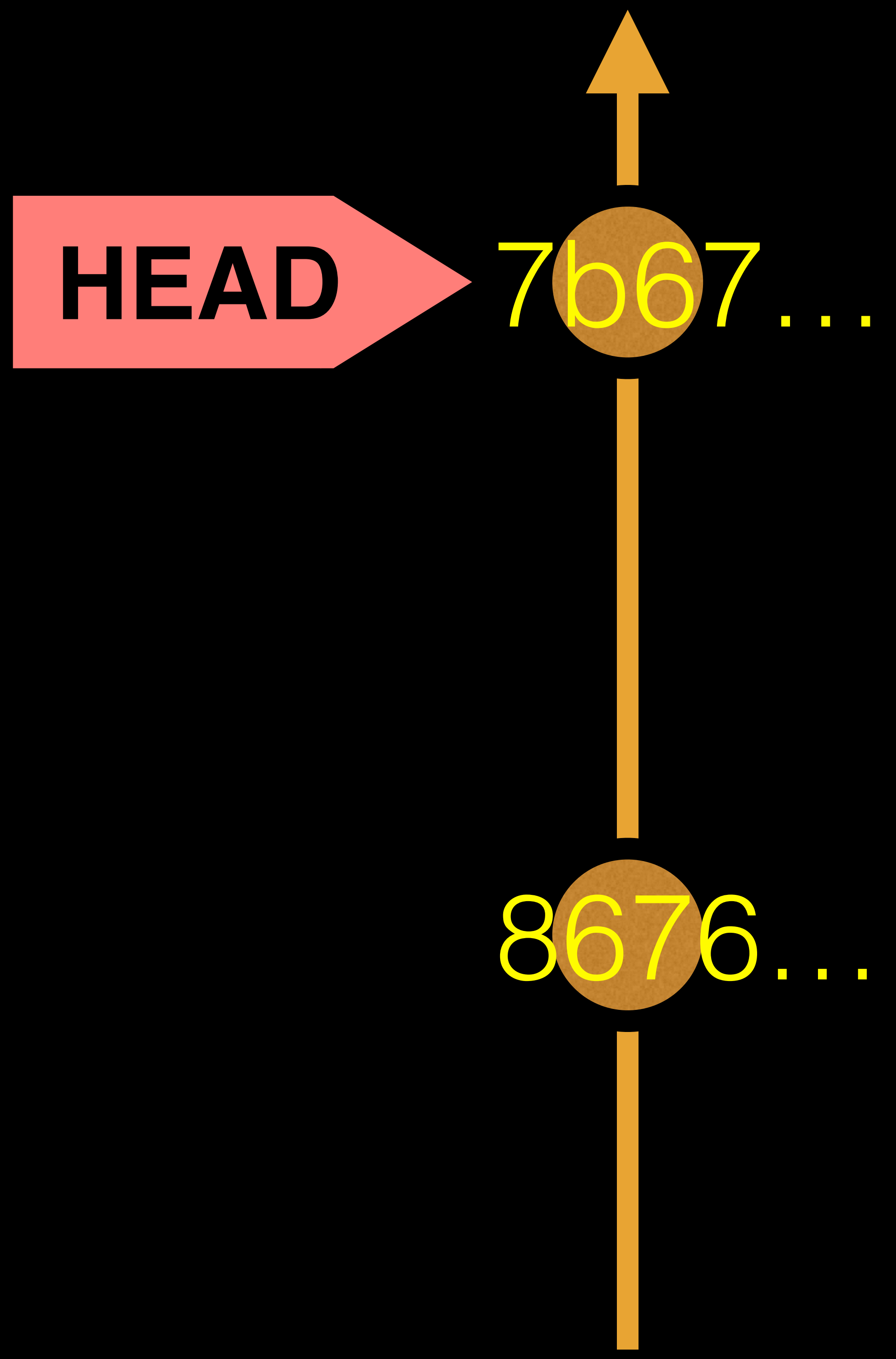
git checkout: Moves HEAD (e.g. back in time)

```
> more README
This is a first line of text.
This is a 2nd line of text.

> git log --oneline
# 7b679fa  Add ToDo and finished README
# 8676840  Create a README file

> git checkout 86768
# You are in 'detached HEAD' state…<cut>…
# HEAD is now at 8676840... Create a README file

> more README
This is a first line of text.

> git log --oneline
# 8676840  Create a README file
```



git checkout: Moves HEAD (e.g. back to the future!)

```
> git checkout master
# Previous HEAD position was 8676840... Create a README file
# Switched to branch 'master'

> git log --oneline
# 7b679fa Add ToDo and finished README
# 8676840 Create a README file

> more README
This is a first line of text.
This is a 2nd line of text.
```



Side-Note: There are two* main ways to use **git checkout**

- Checking out a **commit** makes the entire working directory match that commit. This can be used to view an old state of your project.

```
> git checkout <commit.ID>
```

- Checking out a **specific file** lets you see an old version of that particular file, leaving the rest of your working directory untouched.

```
> git checkout <commit.ID> <filename>
```

You can discard revisions with **git revert**

- The **git revert** command undoes a committed snapshot.
- But, instead of removing the commit from the project history, it figures out how to **undo the changes** introduced by the commit and **appends a new commit** with the resulting content.

```
> git revert <commit.ID>
```

- This prevents Git from losing history!

Removing untracked files with **git clean**

- The **git clean** command removes untracked files from your working directory.

- Like an ordinary **rm** command, **git clean** is not undoable, so make sure you really want to delete the untracked files before you run it.

```
> git clean -n # dry run display of files to be ‘cleaned’

> git clean -f   # remove untracked files
```

Demo Tower

GUIs

Tower (Mac only)
GitHub_Desktop (Mac, Windows)
SourceTree (Mac, Windows)
SmartGit (Linux)
RStudio

https://git-scm.com/downloads/guis