

FAIR_bioinfo : Open Science and FAIR principles in a bioinformatics project

How to make a bioinformatics project more reproducible

C. Hernandez¹ T. Denecker² J. Sellier² G. Le Corguillé²
C. Toffano-Nioche¹

¹Institute for Integrative Biology of the Cell (I2BC)
UMR 9198, Université Paris-Sud, CNRS, CEA
91190 - Gif-sur-Yvette, France

²IFB Core Cluster taskforce

June 2021



Introduction to code versioning

Really need of a files history?

"FINAL".doc



FINAL.doc!



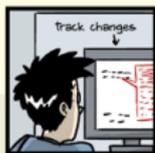
FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



FINAL_rev.18.comments7.
corrections9.MORE.30.doc



FINAL_rev.22.comments49.
corrections.10.#@\$%WHYDID
ICOMETOGRAD SCHOOL?????.doc

WWW.PHDCOMICS.COM

"Most researchers are primarily collaborating with themselves," [Tracy] Teal explains. "So, we teach it from the perspective of being helpful to a 'future you'."

"Rule 4: Version Control All Custom Scripts"

OPEN ACCESS Freely available online

 PLOS COMPUTATIONAL BIOLOGY

Editorial

Ten Simple Rules for Reproducible Computational Research

Geir Kjetil Sandve^{1,2*}, Anton Nekrutenko³, James Taylor⁴, Eivind Hovig^{1,5,6}

1 Department of Informatics, University of Oslo, Blindern, Oslo, Norway, **2** Centre for Cancer Biomedicine, University of Oslo, Blindern, Oslo, Norway, **3** Department of Biochemistry and Molecular Biology and The Huck Institutes for the Life Sciences, Penn State University, University Park, Pennsylvania, United States of America, **4** Department of Biology and Department of Mathematics and Computer Science, Emory University, Atlanta, Georgia, United States of America, **5** Department of Tumor Biology, Institute for Cancer Research, The Norwegian Radium Hospital, Oslo University Hospital, Montebello, Oslo, Norway, **6** Institute for Medical Informatics, The Norwegian Radium Hospital, Oslo University Hospital, Montebello, Oslo, Norway

Replication is the cornerstone of a cumulative science [1]. However, new tools and technologies, massive amounts of data, interdisciplinary approaches, and

We further note that reproducibility is just as much about the habits that ensure reproducible research as the technologies that can make these processes efficient and

than to do it while underway). We believe that the rewards of reproducibility will compensate for the risk of having spent valuable time developing an annotated



Version control

Definition

version control, revision control, source control, or source code management: class of systems responsible for managing changes to files.

Feature

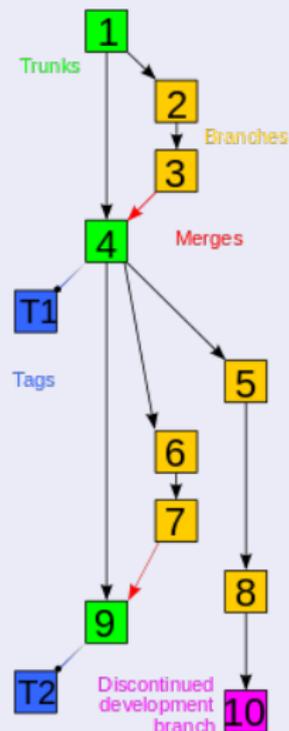
Each revision is associated with a timestamp and the person making the change. Revisions can be compared, restored, and merged.

Software

SVN, Git, Mercurial, GNU arch, etc

[wikipedia source](#)

Revisions graph



Git and GitHub

Git



- will track and version your files
- enables you to collaborate with ... yourself
- open source license GPL (GNU General Public License)
- created in 2005 by Linus Torvalds for the development of the Linux kernel

GitHub



- stores your  **git** repositories online
- enables you to collaborate with others (and yourself)
- first commit in 2007 by Chris Wanstrath, founded in feb. 2008, Microsoft Corporation still 2018

Git



Concepts, objects

- working directory: a user private copy of a whole repository of interest
- commit: a git object, the snapshot of your entire repository compressed into a SHA (also the command that saves changes by creating the snapshot)
- staging area: list of files of the working directory that will be considered for next commit (ie. could be not all the modified files)
- branch: a lightweight movable pointer to a commit
- HEAD: pointer representing your current working directory. Can be moved (`git checkout`) to different branches, tags, or commits
- merge: combines remote tracking branches into current local branch
- clone: a local copy of a repository (include all commits and branches), the original repository can be local, or remote (http access)

https://www.tutorialspoint.com/git/git_quick_guide.htm

<https://www.powershellmagazine.com/2015/07/13/git-for-it-professionals-getting-started-2/>

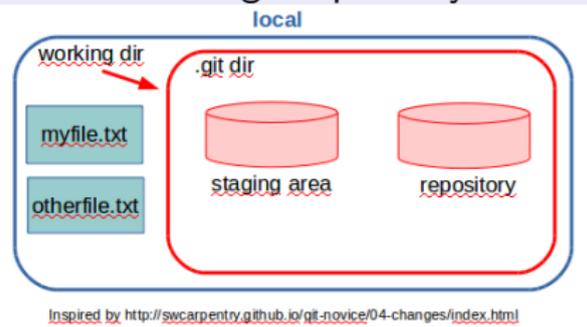


Git configuration: if not yet done, tell git our identity

```
1 git config --global user.name 'Your Name'
2 git config --global user.email 'Your Email'
```

Git repository initialisation

The initialisation (red arrow) is the creation of a `.git` repository:

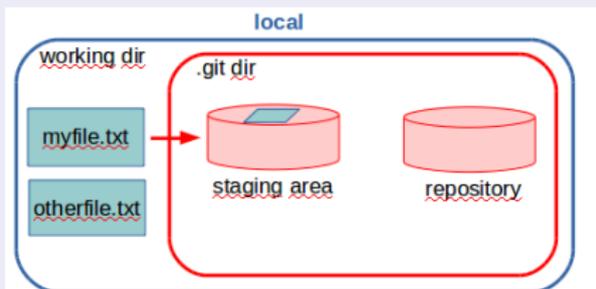


3 ways to initialize a `.git` repository:

- `git init`: inside an existing folder (possibly containing files)
- `git init myproject`: create folder "myproject" + initializes the `.git` subfolder inside it
- `git clone /gitfolder/path /new/path`: copy the existing git repository to a new one

Tracking file

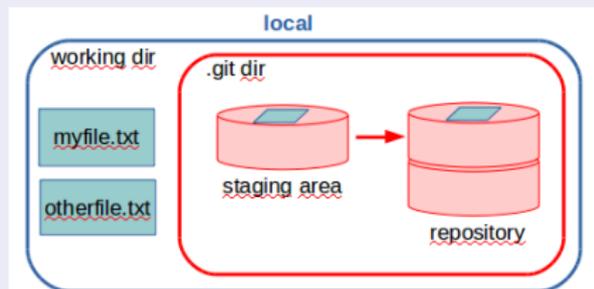
git add command for myfile.txt:



Inspired by <http://swcarpentry.github.io/git-novice/04-changes/index.html>

<http://swcarpentry.github.io/git-novice/fig/git-staging-area.svg>

git commit -m "my reason":



Inspired by <http://swcarpentry.github.io/git-novice/04-changes/index.html>

Git file states

Checking the file status: `git status`

File goes from untracked to tracked state (init), unstaged to staged state (add) and finally, to a committed state (commit).

Git Exercise

1st exercise

- 1 access and configure git
- 2 initialize a git repository
- 3 create files in this repository
- 4 use the basic git commands for tracking files changes (status, add, commit)

2nd exercise

- 5 copy another repository from github (clone)
- 6 use branching (branch) and merging (merge) to manage code changes

git setup: objectives 1 & 2

Git access by doker

```
1 docker run -i -t -v ${PWD}:/data continuumio/miniconda3
```

Git configuration

Global configuration (checking user.name with: `git config --list`):

```
1 git config --global user.name 'Your Name'  
2 git config --global user.email 'Your Email'
```

Git repository initialization

On a new dedicated folder run:

```
1 git init # observe the .git folder (ls -la)  
2 git status # find the current branch, "nothing to commit"
```

git adding files: objective 3

create 2 files, check their git status: obj

```
1 for i in 1 2 ; do echo "file"${i}" text" > file${i}.txt ;  
   done  
2 git status # observe list of untracked files
```

add file1 to staging area

```
1 git add file1.txt  
2 git status # observe the changing status of file1: untracked  
   => staged
```

change file1 text

```
1 sed 's/text/text change/' file1.txt > tmp ; mv tmp file1.txt  
2 git status # observe the 3 states, why file1 appears in "to  
   be committed" and also in "not staged for commit"?
```

stage all files

```
1 git add file1.txt file2.txt # all files
2 git status
```

commit

```
1 git commit -m "1st commit + file1 change" # always add a
   message, use present time to explain the change
2 git status # all ok
```



So far, we have initiated a new project whose code is versioned by git: we have created files and all their successive changes were saved thanks to git.

We will now create a 2nd project by copying an already existing one. We're going to bring this project from an online git project site, e.g. github.

copy of a project: clone

To download a project from github, we use the `git clone` command:

```
1 git clone https://github.com/clairetn/FAIR_bioinfo_github.git
```

observe result

- a new folder has been created (check with the shell `ls` command)
- its name is directly deduced from the url used
- this `FAIR_bioinfo_github` folder contains a `.git` repository and also a `README.md` file (see with `ls -la FAIR_bioinfo_github/`)
- it is a minimal project!

git branching: objective 6

We plan to change the README file by adding our firstname at the authors list. With a git versioning system, a good practice is to create a branch to reserve the initial code until we validate our change.

create a branch named "branch1"

```
1 cd FAIR_bioinfo_github
2 git branch branch1
```

list all branches

```
1 git branch # find the star
```



git branching: objective 6

go into the new "branch1"

```
1 git checkout branch1
2 git branch # find the star
3 git status # find the branch
```

work into branch: change a file and keep change

Edit the README.md file and add your firstname to the "Authors list"

```
1 git status # file README.md is modified
2 git add README.md ; git commit -m "add my firstname in
   branch1"
```

return to master branch

```
1 git checkout master
2 more README.md # Is README.md modified or initial version?
```

We have check that our change is valid, so we now plan to move it into the master branch.

merge branch, then delete branch

```
1 git merge branch1
2 more README.md # what README.md version?
3 git branch -d branch1 # -d for delete
```

GitHub

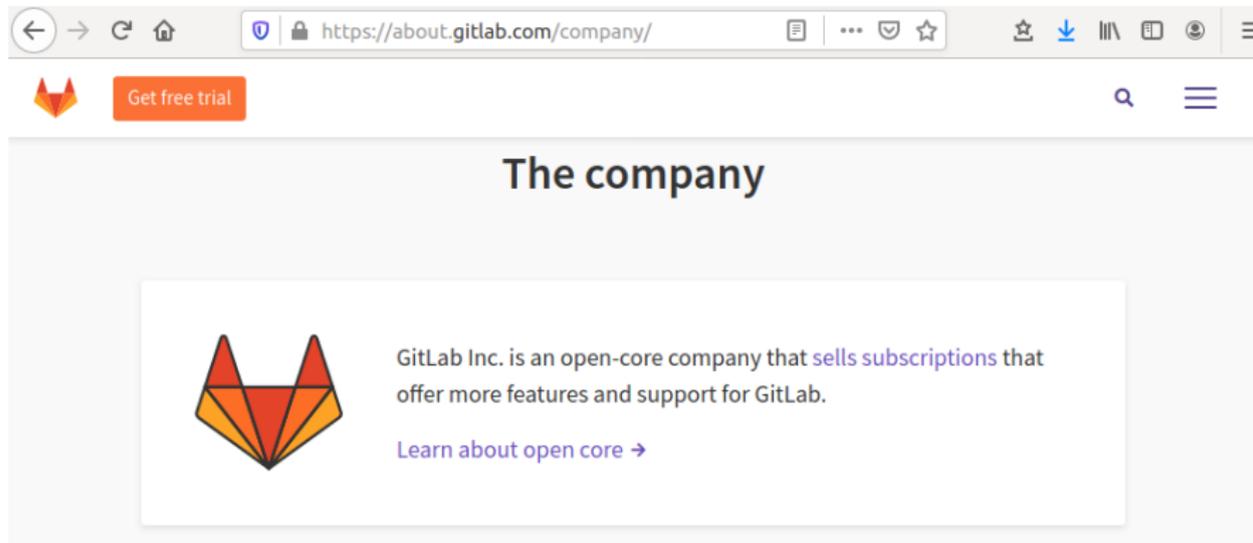
Quizz

- 1 public institute (governmental)?
- 2 semi-public institute?
- 3 not-for-profit organisation?
- 4 private company?

Response

See <https://github.com/about>: Careers' paragraph, you'll see a "company" word

GitLab, a GitHub alternative?

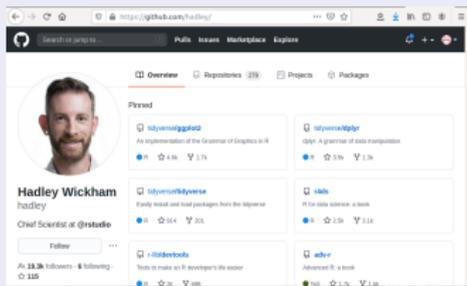


The screenshot shows a web browser window with the URL `https://about.gitlab.com/company/`. The page features the GitLab logo (a stylized fox head) and a "Get free trial" button. The main heading is "The company". Below this, there is a section with the GitLab logo and the text: "GitLab Inc. is an open-core company that [sells subscriptions](#) that offer more features and support for GitLab." A link "Learn about open core →" is also present.

Quizz

- 1 social network?
- 2 desktop application?
- 3 tool to create websites?
- 4 stable repository to publish any file?

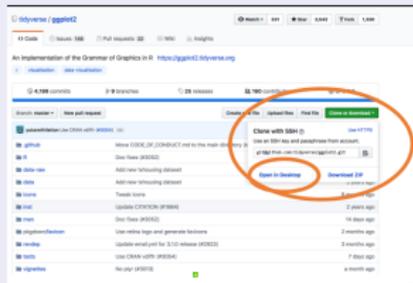
a social network



a tool to create websites



a desktop application



a stable repository ...

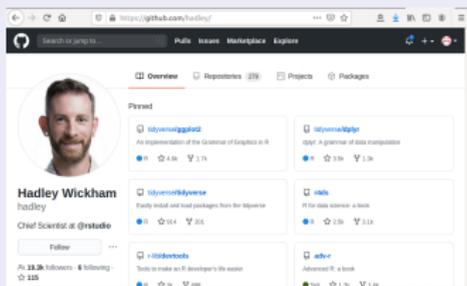
Popularity 

Name	Users	Projects	Alexa rank (lower = more popular)
Assembla	Unknown	526,361 ⁽⁹¹⁾	33,434 as of 28 July 2020 ⁽⁹¹⁾
Bitbucket	5,099,000 ⁽⁴⁾	Unknown	1,341 as of 28 July 2020 ⁽⁹¹⁾
Buddy	Unknown	Unknown	39,857 as of 28 July 2020 ⁽⁹¹⁾
CloudForge	Unknown	Unknown	402,888 as of 28 July 2020 ⁽⁹¹⁾
Gitex	Unknown	Unknown	236,332 as of 28 July 2020 ⁽⁹¹⁾
GitLab	31,000,000 ⁽¹⁾	108,000,000 ⁽¹⁾	78 as of 28 July 2020 ⁽⁹¹⁾
GitLab	100,000 ⁽⁷⁾	548,000 ⁽⁶⁾	3,710 as of 28 July 2020 ⁽⁹¹⁾
GNU Savannah	93,340 ⁽⁶⁾	3,940 ⁽⁶⁾	162,054 as of 28 July 2020 ⁽⁹¹⁾
Launchpad	3,985,288 ⁽¹⁾	40,882 ⁽¹⁾	11,531 as of 28 July 2020 ⁽⁹¹⁾
OSDN	54,020 ⁽¹⁾	6,294 ⁽¹⁾	8,700 as of 28 July 2020 ⁽⁹¹⁾
OurProject.com	6,313 ⁽¹⁾	1,840 ⁽¹⁾	1,083,012 as of 28 July 2020 ⁽⁹¹⁾
OW2 Consortium	Unknown	Unknown	1,506,477 as of 28 July 2020 ⁽⁹¹⁾
Source code	Unknown	Unknown	68,029 as of 28 July 2020 ⁽⁹¹⁾
SRM	Unknown	Unknown	1,602,812 as of 28 July 2020 ⁽⁹¹⁾
SourceForge	3,700,000 ⁽¹⁾	500,000 ⁽¹⁾	470 as of 28 July 2020 ⁽⁹¹⁾
Name	Users	Projects	Alexa rank (lower = more popular)

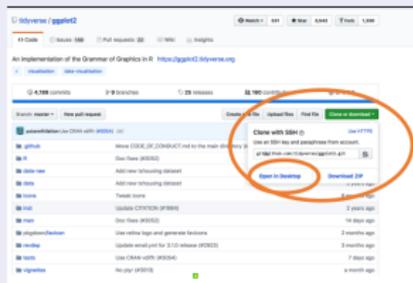
https://en.wikipedia.org/wiki/Comparison_of_

[source-code-hosting_facilities](https://en.wikipedia.org/wiki/Comparison_of_source-code-hosting_facilities)[en.wikipedia,](https://en.wikipedia.org/wiki/Comparison_of_source-code-hosting_facilities)

a social network ✔



a desktop application ✔



a tool to create websites ✔



... to publish any file ✔ ✘

Files for which git can calculate the difference between versions. Usually txt files of reasonable size:

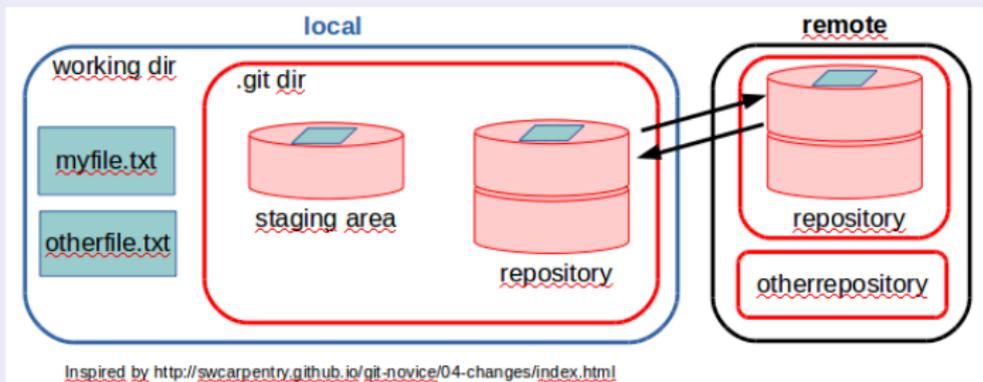
- R script: ✔
- Python script: ✔
- pdf file: ✘
- fastq file: ✘



GitHub main usage: sharing code with others

GitHub:

- so used that Microsoft was interested in it ([bought](#) in june 2018)
- web-based: graphical interface + many more features than git
- git-based: git concepts and commands are retained
- commands for "sharing": `git push origin master` (local to remote) and `git pull origin master` (remote to local):



Concepts, objects

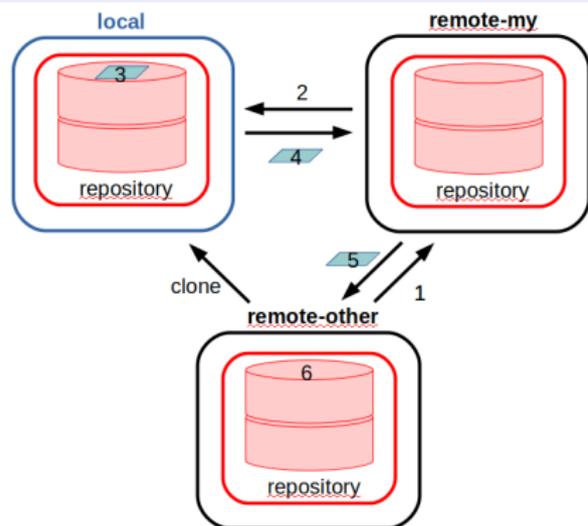
- user: your account on GitHub (unlimited for academics)
- organization: account for one or more user (e.g., swcarpentry)
- local GitHub: copies of GitHub files located your computer
- remote GitHub: your GitHub files located on <https://github.com>
- fork: a copy of a GitHub repository to your own GitHub account
- push: send changes on the working repository to your remote GitHub repository
- pull: copy changes on the remote GitHub repository to your local GitHub repository (useful when multiple people make changes)
- pull request: propose your changes to the initial forked GitHub repository. Also a place to compare and discuss the differences introduced on a branch with reviews, comments, integrated tests, etc

Clone vs. Fork?

- clone is git, fork is github
- all 2 copy a .git repository: clone copy it in your local machine, fork in your github account (do a clone)
- good practice: work (change files) in the local copy, not in the github copy (only for minor changes)
- to share your changes with the original repository, need a fork (by the way of a pull request)

See [here](#) an historical point of view of those 2 words.

Recommended flow to collaborate



(direct clone from github don't allow to collaborate)

- 1: fork a repository of interest in your github account
- 2: clone from your github account to your local place
- 3: make change (branch, add, commit, merge)
- 4: push change to your github account
- 5: pull request to propose your change to the initial project
- 6: wait (discuss) for integrating your change or not

GitHub Exercise 1

Objectives

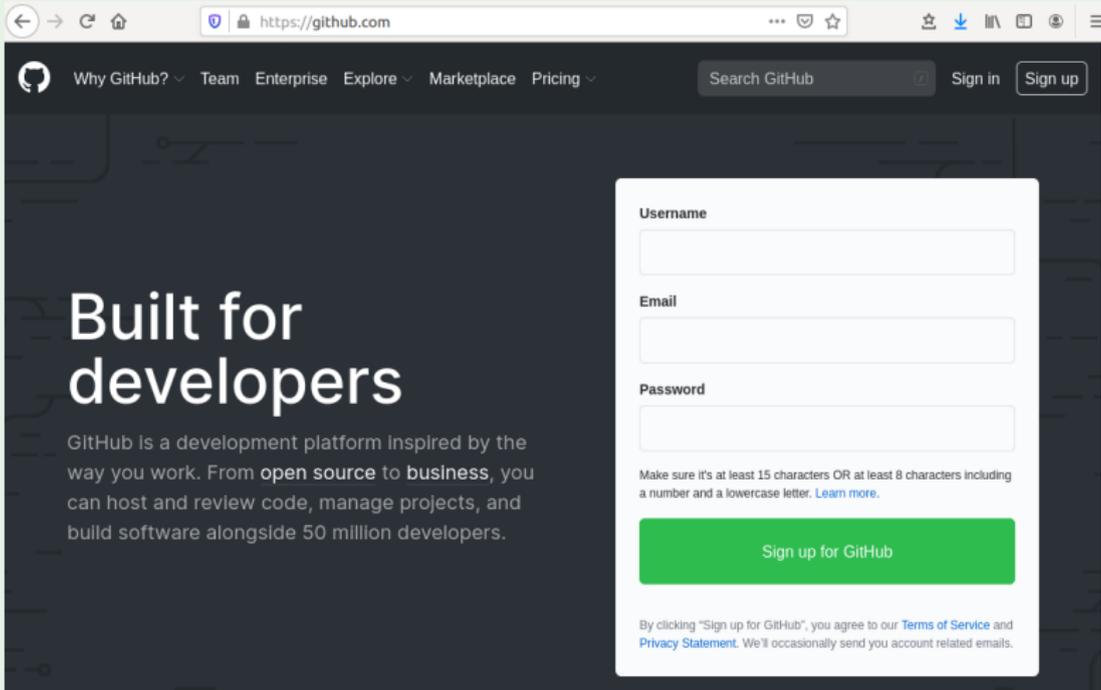
The objective of this exercise is to propose change to an existing project. We will:

- fork an existing project to our GitHub account
- create a branch
- made a change in the branch
- save change into the change
- merge the branch

Web interface

During this exercise, most of the actions that will be performed will be done via the GitHub web interface, i.e. with many button clicks. The following pages will guide us to the next action.

With a browser, go to github (<https://github.com>). If not already yet, sign up and create your github account, otherwise sign in



The screenshot shows the GitHub homepage in a browser. The address bar displays <https://github.com>. The navigation bar includes links for "Why GitHub?", "Team", "Enterprise", "Explore", "Marketplace", and "Pricing", along with a search bar and "Sign in" and "Sign up" buttons. The main content area features the heading "Built for developers" and a subheading "GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 50 million developers." On the right, a white sign-up form is overlaid, containing fields for "Username", "Email", and "Password", a "Sign up for GitHub" button, and a disclaimer: "By clicking 'Sign up for GitHub', you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails."

Username

Email

Password

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

[Sign up for GitHub](#)

By clicking "Sign up for GitHub", you agree to our [Terms of Service](#) and [Privacy Statement](#). We'll occasionally send you account related emails.

GitHub: fork a project

Objective

For this exercise, we will replay the addition of our first name, but by using the user interface proposed by github.

Fork in our gituhb account

With a browser, go to the url of the initial project, [super-umbrella](#) and click to "Fork" (upper right):

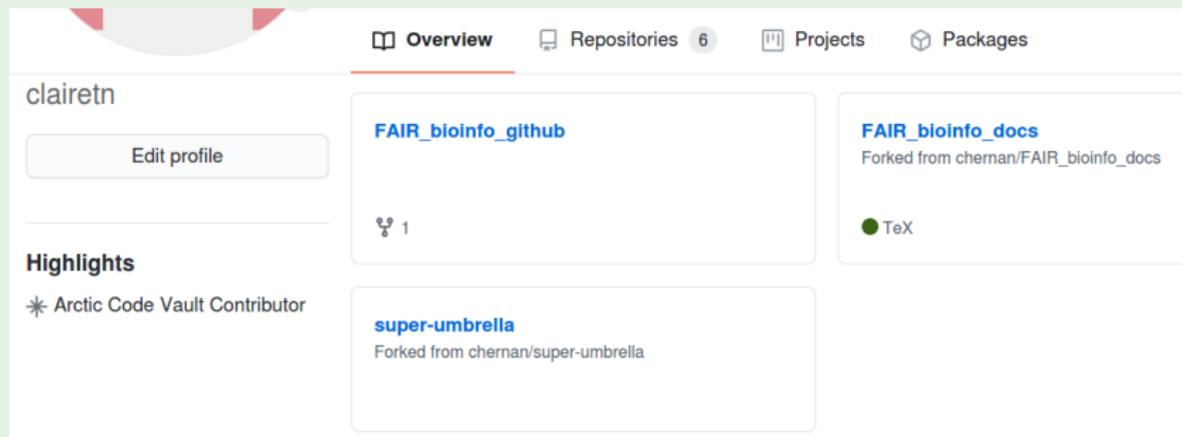
The screenshot shows the GitHub repository page for 'chernan / super-umbrella'. The repository name is circled in orange. In the top right corner, the 'Fork' button is also circled in orange. The page displays the repository's metadata, including 1 commit, 1 branch, 0 releases, and 1 contributor. Below this, there are navigation options like 'Search: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A table shows the commit history for 'README.md', with the latest commit by 'chernan' at 'initial commit' just now. The main content area shows the 'README.md' file with the text 'super-umbrella'.

GitHub: the forked repository

Result:

You can see the result in your Github Overview: you have a new repository, named FAIR_bioinfo_github and entitled "forked from chernan/super-umbrella".

result of the fork chernan, super-umbrella project:

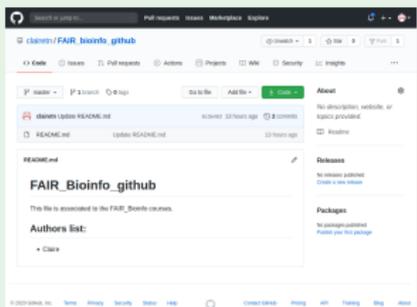


The screenshot shows the GitHub profile overview for user 'clairetn'. The profile name 'clairetn' is at the top left, with an 'Edit profile' button below it. Under the 'Highlights' section, it lists 'Arctic Code Vault Contributor'. The main content area displays three repository cards: 'FAIR_bioinfo_github' (forked from chernan/super-umbrella), 'FAIR_bioinfo_docs' (forked from chernan/FAIR_bioinfo_docs), and 'super-umbrella' (forked from chernan/super-umbrella). The navigation bar at the top includes 'Overview', 'Repositories 6', 'Projects', and 'Packages'.

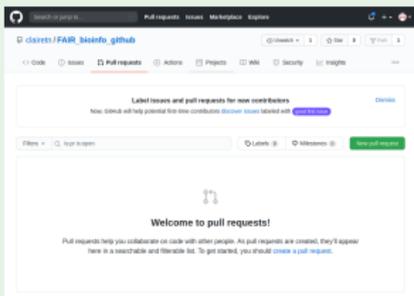
Tabs

8 Tabs offered by GitHub for each repository:
Code, Pull Requests, Actions, Projects, Wiki, Security, Insights, Settings.
Mainly focus on 3 of them:

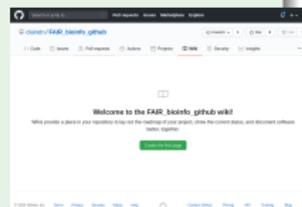
Code



Pull Requests



Wiki



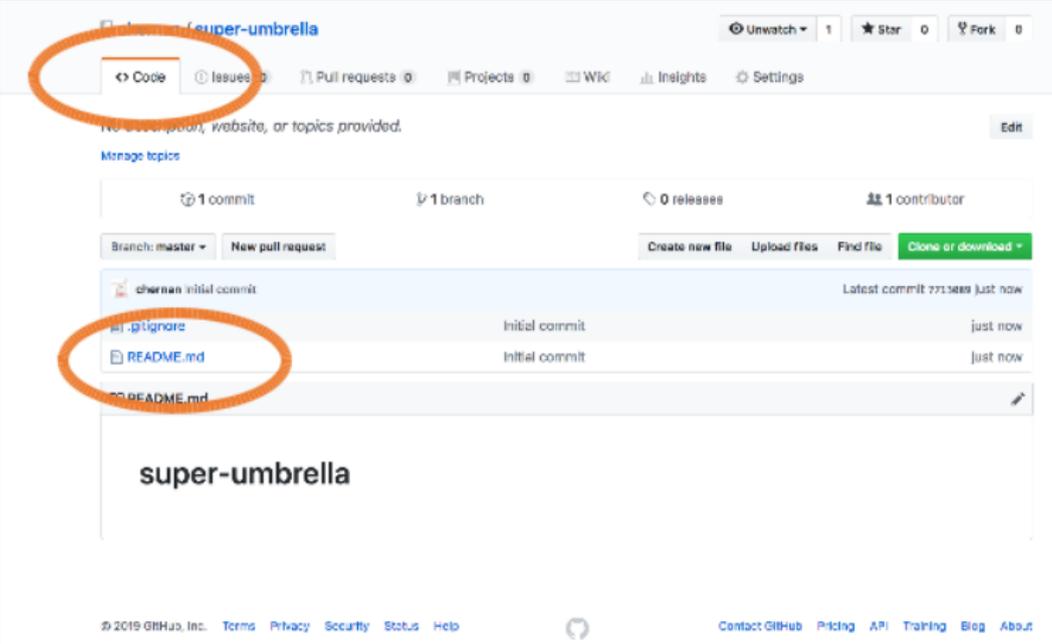
Previous exercises with git

- copy a github repo. (git clone)
- go to the local repo. (cd)
- create branch (git branch)
- go to branch (git checkout)
- make change (edit file)
- stage change (add)
- version change (commit)
- go to master (git checkout)
- merge branch (git merge)
- delete branch (git branch -d)

Next steps with github GUI:

- 1 fork a github repo. (just done)
- 2 create branch
- 3 make change (edit file)
- 4 version change (commit)
- 5 compare branch to master
- 6 merge branch
- 7 ask for merging (Pull Request)
- 8 delete branch

1: fork chernan, super-umbrella, see the README.md file:



The screenshot shows the GitHub repository page for 'super-umbrella'. The 'Code' button in the top navigation bar is circled in orange. Below the repository name, the file list shows a commit by 'chernan' with files '.gitignore', 'README.md', and 'README.md'. The 'README.md' file is also circled in orange. The repository content area displays the text 'super-umbrella'.

Repository: **super-umbrella** (Unwatch 1 | Star 0 | Fork 0)

Navigation: **Code** | Issues | Pull requests | Projects | Wiki | Insights | Settings

Repository status: 1 commit | 1 branch | 0 releases | 1 contributor

Actions: Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

File	Commit	Latest commit
.gitignore	Initial commit	just now
README.md	Initial commit	just now
README.md		

Repository content: **super-umbrella**

Footer: © 2019 GitHub, Inc. | Terms | Privacy | Security | Status | Help | Contact GitHub | Pricing | API | Training | Blog | About

2: create a new branch, named "devel-your-name"

chernan / super-umbrella

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Projects Wiki Insights Settings

Branch: master super-umbrella / README.md

Find file Copy path

cherman Initial commit

7713889 an hour ago

1 contributor

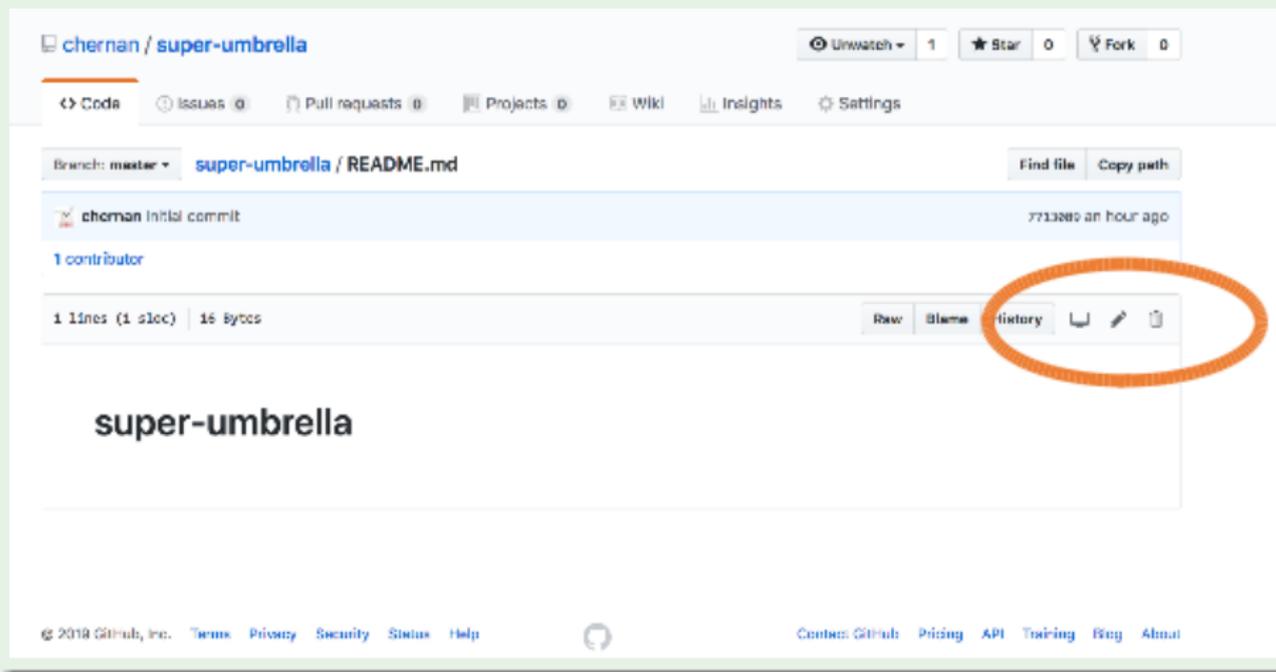
1 lines (1 sloc) 16 Bytes

Raw Blame History

super-umbrella



3: edit README.md to make change



chernan / super-umbrella

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Projects Wiki Insights Settings

Branch: master super-umbrella / README.md Find file Copy path

choman initial commit 7713289 an hour ago

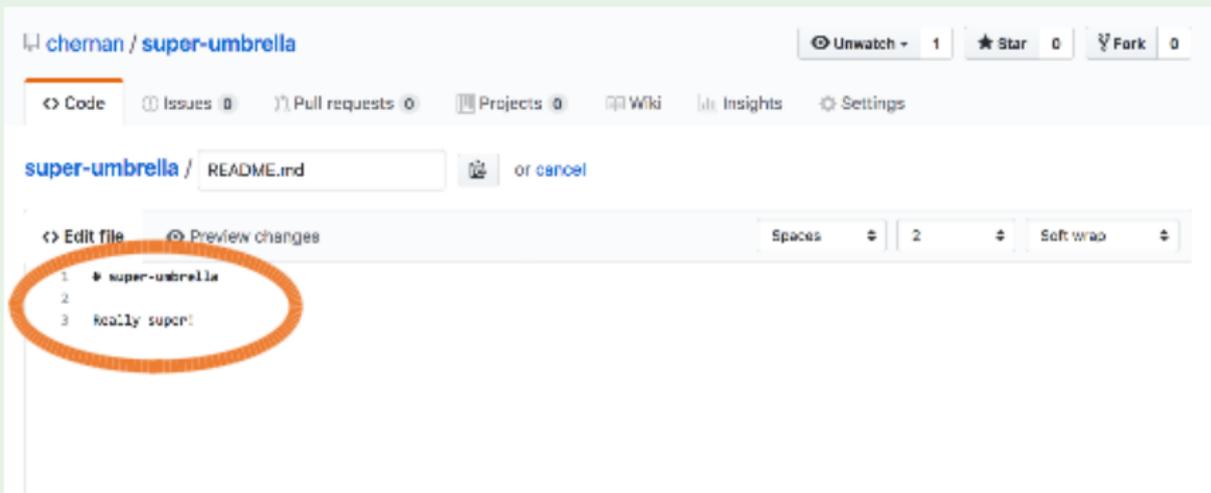
1 contributor

1 lines (1 slot) 16 bytes Raw Blame History

super-umbrella

© 2018 GitHub, Inc. Terms Privacy Security Status Help Contact: GitHub Pricing API Training Blog About

3: make change



The screenshot shows the GitHub interface for the repository 'cheman / super-umbrella'. The file 'README.md' is selected for editing. The code editor shows three lines of text: '1 super-umbrella', '2', and '3 really super!'. The first line is circled in orange. The interface includes navigation tabs for Code, Issues, Pull requests, Projects, Wiki, Insights, and Settings. At the top right, there are buttons for Unwatch (1), Star (0), and Fork (0). Below the file name, there are options to 'or cancel' and a 'Spaces' dropdown set to 2. The editor has 'Edit file' and 'Preview changes' tabs.

4: commit



Commit changes

Update README.md

Add an optional extended description...

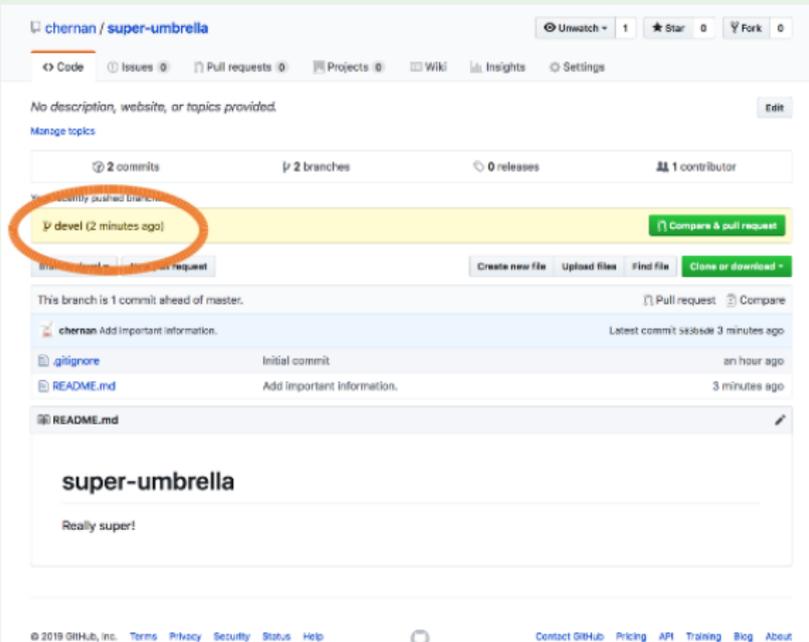
- Commit directly to the `devel` branch.
- Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes

Cancel



4: commit and pull request



chernan / super-umbrella

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Projects Wiki Insights Settings

No description, website, or topics provided. [Edit](#)

Manage topics

2 commits 2 branches 0 releases 1 contributor

Recently pushed branches

dev (2 minutes ago) [Compare & pull request](#)

Create new file Upload files Find file Clone or download

This branch is 1 commit ahead of master. [Pull request](#) [Compare](#)

chernan Add important information. Latest commit `8e30d` 3 minutes ago

- `.gitignore` Initial commit an hour ago
- `README.md` Add important information. 3 minutes ago

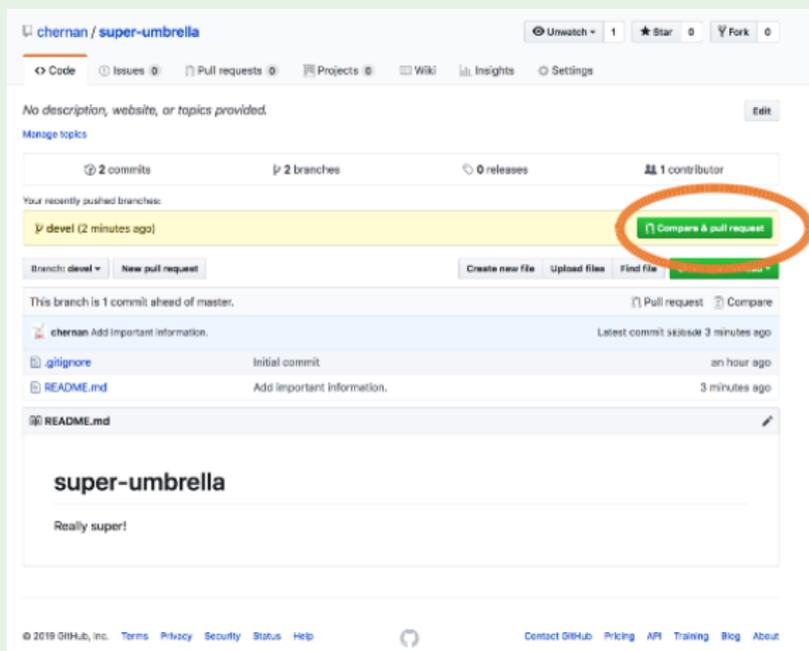
README.md

super-umbrella

Really super!

© 2019 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#) [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

5: pull request, compare



chernan / super-umbrella

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Projects Wiki Insights Settings

No description, website, or topics provided. [Edit](#)

Manage topics

2 commits 2 branches 0 releases 1 contributor

Your recently pushed branches:

- devel (2 minutes ago) [Compare & pull request](#)

Branch: devel [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#)

This branch is 1 commit ahead of master. [Pull request](#) [Compare](#)

chernan	Add Important Information.	Latest commit sksisk 3 minutes ago
gitignore	Initial commit	an hour ago
README.md	Add Important Information.	3 minutes ago

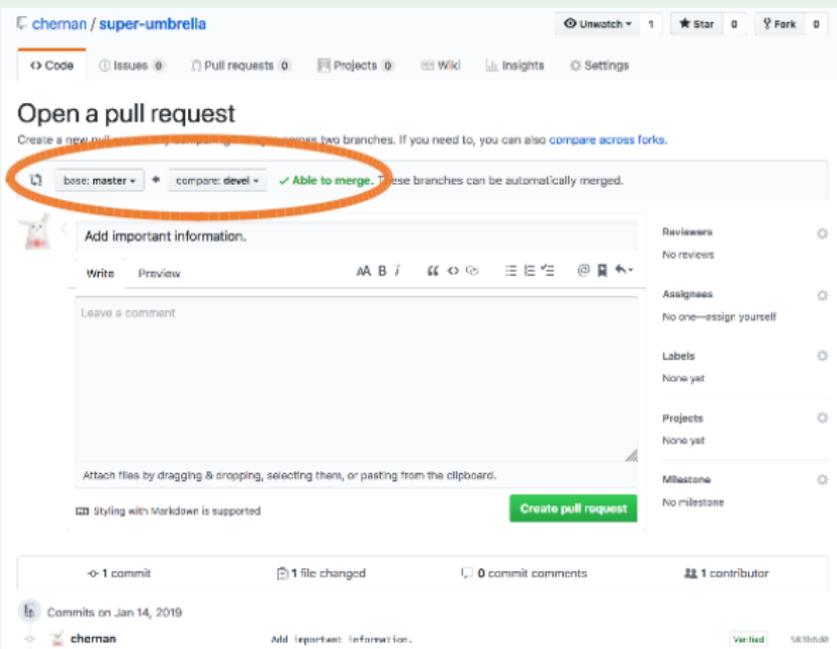
README.md

super-umbrella

Really super!

© 2019 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#) [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

5: pull request, able to merge



cherman / super-umbrella

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Projects Wiki Insights Settings

Open a pull request

Create a new pull request by comparing branches. If you need to, you can also [compare across forks](#).

base: master + compare: devel ✓ Able to merge. These branches can be automatically merged.

Add important information.

Write Preview AA B I 

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

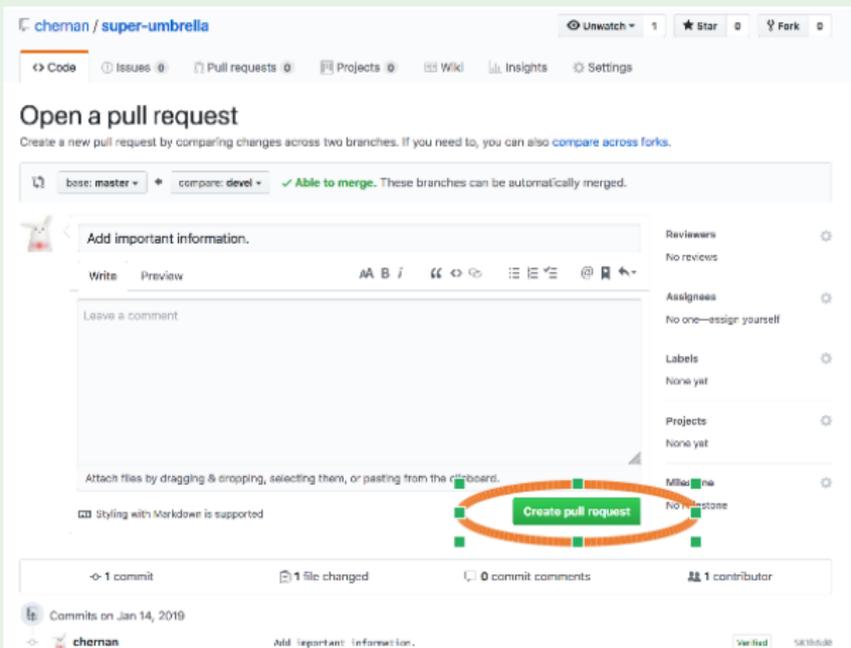
Create pull request

1 commit 1 file changed 0 commit comments 1 contributor

Commits on Jan 14, 2019

cherman Add important information. Verified skibidib

5: merge and pull request



cherman / super-umbrella

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Projects Wiki Insights Settings

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: master compare: devel ✓ Able to merge. These branches can be automatically merged.

Add important information.

Write Preview AA B i ↵ ↻ ↺ ⋮ ⋮ ⋮ @ 📎 ↶ ↷

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

Reviewers: No reviews

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestones: No milestones

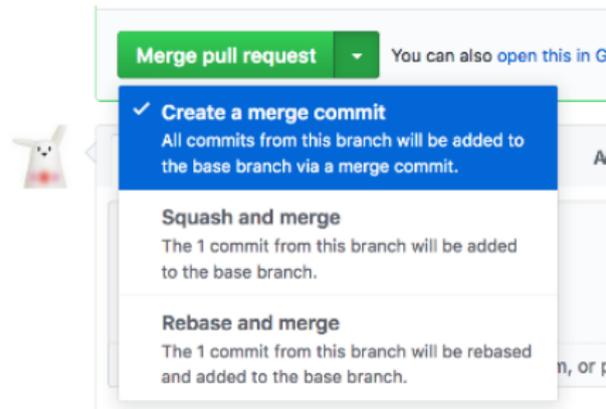
Create pull request

< 1 commit 1 file changed 0 commit comments 1 contributor

Commits on Jan 14, 2019

cherman Add important information. Verified 5835d38

5: merge and Pull request



The screenshot shows the GitHub merge options dropdown menu. At the top is a green button labeled "Merge pull request" with a dropdown arrow and the text "You can also open this in G". Below it are three options:

- ✓ Create a merge commit**
All commits from this branch will be added to the base branch via a merge commit.
- Squash and merge**
The 1 commit from this branch will be added to the base branch.
- Rebase and merge**
The 1 commit from this branch will be rebased and added to the base branch.

Blue arrows point from the "Create a merge commit" option to the first text box, from "Squash and merge" to the second text box, and from "Rebase and merge" to the third text box.

All individual commits are kept, branches are merged.

All commits are combined into one, and propagated to the base.

All commits are propagated to the base branch.

6: merge

chernan / super-umbrella

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 1 Projects 0 Wiki Insights Settings

Add important information. #1

Open chernan wants to merge 1 commit into master from devel

Conversation 0 Commits 1 Checks 0 Files changed 1 +3 -1

chernan commented 7 minutes ago **Owner**

No description provided.

Add important information. **verified** 583b6d8

Add more commits by pushing to the devel branch on chernan/super-umbrella.

Merge pull request #1 from chernan/devel

Add important information.

Confirm merge Cancel

Write Preview AA B / ↵ ↶ ↷ ⋮ ⋮ ⋮ @ 🗉 ↵

Leave a comment

Reviews: No reviews

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

Notifications: Unsubscribe

You're receiving notifications because you authored the thread.

1 participant

7: the merge delete the branch

chernan / super-umbrella

Unwatch 1 Star 0 Fork 0

Code Issues Pull requests Projects Wiki Insights Settings

Add important information. #1

Merged chernan merged 1 commit into master from devel just now

Conversation Commits Checks Files changed +3 -1

chernan commented 8 minutes ago

No description provided.

Add important information. Verified 583b6d8

chernan merged commit be2ace into master just now

Pull request successfully merged and closed

You're all set—the devel branch can be safely deleted.

Delete branch

Write Preview

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

Comment

Reviews: No reviews

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

Notifications: Unsubscribe

You're receiving notifications because you modified the open/close state.

1 participant

See version network, Insights tab

chernan / super-umbrella

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

- Pulse
- Contributors
- Community
- Traffic
- Commits
- Code frequency
- Dependency graph
- Alerts
- Network**
- Forks

Owners	Jan
chernan	13 

GitHub GUI

- With this exercise, we modified a file in a directory of our own GitHub account.
- BUT: reserve this click button mode only for minor modifications (relies on a stable and smooth network connection!)
- Also, we collaborated only with ourselves
- In the next exercise, we will do this task again with a "git command line" mode and by collaborating all together.

GitHub Exercise 2

Previous exercise

In the previous exercise, we added change in the fork of our GitHub account through the GitHub GUI.

Objective

Now we will again modify a file but using a local working copy, so that we can work independently of the internet connection.

We will also collaborate all together (eg. the final README.md file should contains the name of all of us)

Steps of modifications will be done with git on a local clone while steps for collaborative building will be done through the GitHub interface.

Repository to fork:

We could use the previous forked repository to do the collaborative part, but as we want to practice changes in a local copy, we will fork another repository: [clairetn](#), [FAIR_bioinfo_github](#)

Be added as collaborator to the repository

To work in a collaborative mode we will invite our neighbors as collaborator in the Settings tab of this repository: need to exchange our github login.

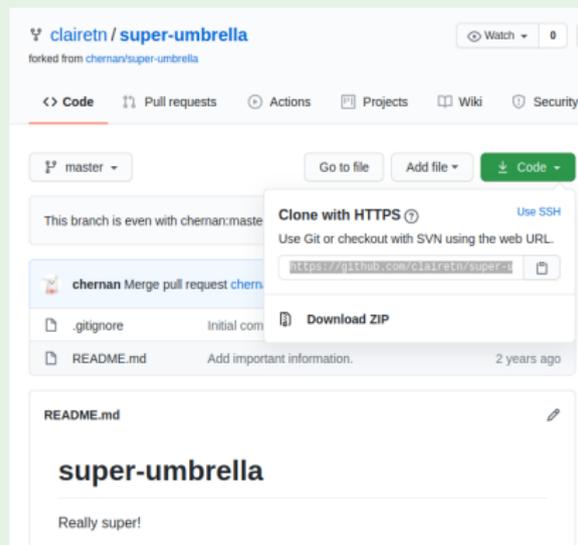
Steps

- 1 fork the repository on your github account (github fork)
- 2 invite your left and right neighbors to collaborate in your fork
- 3 clone your own forked repository in a new local working repository (git, local)
- 4 create a new branch (git, local)
- 5 do the modification (add your name in the README.md file) (local)
- 6 merge the branch (git local)
- 7 push the actual local version to our github repository (git local)
- 8 pull request the original github repository of our changing (github)
- 9 as a collaborator, push your changing in the original upstream repository (github)

GitHub: Cloning one of your fork

Github url:

- We clone a fork with the git command `git clone` followed by the url of the repository.
- This url is accessible with our mouse from the github repository (green "Code" button):



The screenshot shows the GitHub interface for a repository named 'clairetn / super-umbrella', which is a fork of 'chernan/super-umbrella'. The page includes navigation tabs for Code, Pull requests, Actions, Projects, Wiki, and Security. A 'Code' button is highlighted in green, and a dropdown menu is open, showing options to 'Clone with HTTPS' (with a 'Use SSH' link), 'Download ZIP', and 'Use Git or checkout with SVN using the web URL'. The HTTPS URL 'https://github.com/Clairer/n/super-umbrella' is entered in the input field. Below the dropdown, a 'README.md' file is visible, containing the text 'super-umbrella' and 'Really super!'.

`https://github.com/clairetn/FAIR_bioinfo_github.git`

GitHub: Git commands

GUI Github → local git (just done):

```
1 git clone <url_of_your_github_account>
```

Local work:

```
1 git branch mybranch
2 git checkout mybranch
3 # do change, eg. add your name in README.md file
4 git add README.md
5 git commit -m "add name"
6 git checkout master
7 git merge mybranch
8 git delete mybranch
```

Local git → GUI Github:

```
1 git push origin master
```

GitHub: Propose your change into the initial project

From your forked repository

- "Compare" and then "Pull request" your issue (explain your proposals as much as possible)
- conflicts when one change the same line
- manage possible conflicts with the Github GUI

Many small commits

⇒ do many small commits easier to merge than a big unique one

Challenge

- make a (voluntary today) "error" by suppressing the new dedicated repository created for this git exercise
- retrieve your code with the git clone command on your github repository

Conclusion

Git points

- no possibility to make merge when the file are not in text format
- ignore some files from tracking: create a `.gitignore` file containing one file/repository name by line (wildcards accepted)
- tagging a commit: fix a version as a reference

Github points

- github offers more than git embedded commands: sharing code with others, web pages, continuous integration, and more
- pull requests is the way to collaborate
- editors with integrated github: Atom, Visual Studio Code, ...



Resources

- Learning Git by [Software Carpentry](https://swcarpentry.github.io/git-novice/):
`https://swcarpentry.github.io/git-novice/`
- [Git Cheat Sheets](https://services.github.com/on-demand/resources/cheatsheets/): `https://services.github.com/on-demand/resources/cheatsheets/`
- A step-by-step progression to [link RStudio and GitHub](https://jules32.github.io/2016-07-12-0xford/git/):
`https://jules32.github.io/2016-07-12-0xford/git/`
- [Pierre Poulain fr resources](https://cupnet.net/git-github/): `https://cupnet.net/git-github/`

