

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

General information

Practical information:

- Dates: June 28th - 30th
- Location: Institut des Systèmes Complexes, 113 rue Nationale, 75013-Paris
- Courses: 9:00 to 17:30
- Meal: 12:30-14:00
- Pauses: 10:30-11:00 + 15:30-16:00
- 2 days of courses + 1 day of course building

Round table:

- Teachers
- Learners

Ressources:



- GitLab
- L^AT_EX

Training schedule

Day 1:

- Introduction to reproducibility
- History management (3 Practical Sessions, git, GitHub)
- Control your development environment (1 PS, CONDA)
- Encapsulation (2 PS, docker)

Day 2:

- Workflow (2 PS, SNAKEMAKE)
- Traceability with notebooks (2 PS, jupyter, zenodo)
- IFB resources (2 PS, slurm, singularity)
- Sharing and disseminating (GitHub, zenodo)
- Conclusion

Day 3:

- Empowerment and improvement of resources

Table of contents

- 1 Introduction to reproducibility
- 2 History management
 - Introduction
 - Git
 - GitHub
- 3 Control your development environment
- 4 Workflow
- 5 Tracability with Notebook
- 6 IFB resources
- 7 Sharing and dissemination
- 8 Conclusion
- 9 3rd Day

Introduction to code versioning

Really need of a files history?

"FINAL".doc

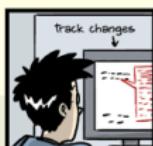


FINAL_rev.2.doc



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.#@\$%WHYDIDICOMETOGRADSSCHOOL????.doc

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

Files history = good practice for reproducible research

"Rule 4: Version Control All Custom Scripts"

OPEN  ACCESS Freely available online



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

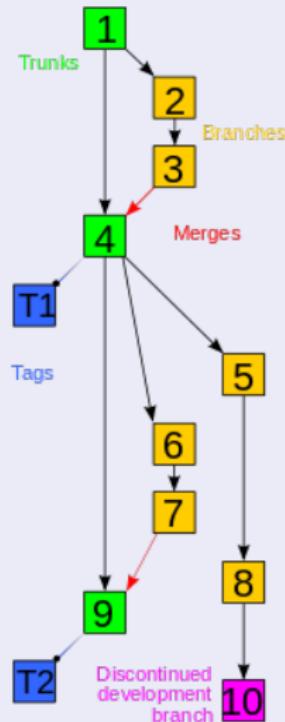
Each revision is associated with a timestamp and the person making the change.
Revisions can be compared, restored, and merged.

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 the 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/> A set of small, light blue navigation icons typically found in presentation software like Beamer.

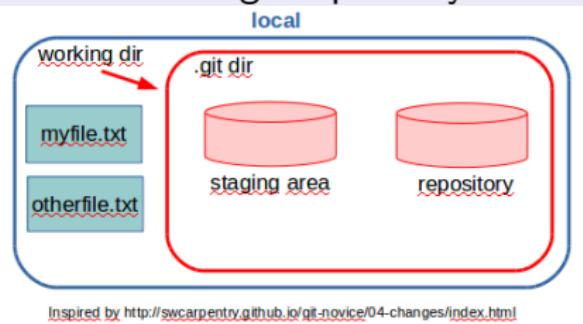
git Setup

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:

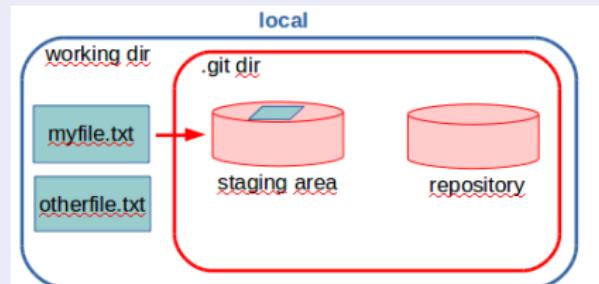
- `git init`: inside an existing folder (possibly containing files)
- `git init myproject`: create folder "myproject" + initializes the .git subfolder inside it
- `git clone /gitfolder/path /new/path`: copy the existing git repository to a new one



Tracking file

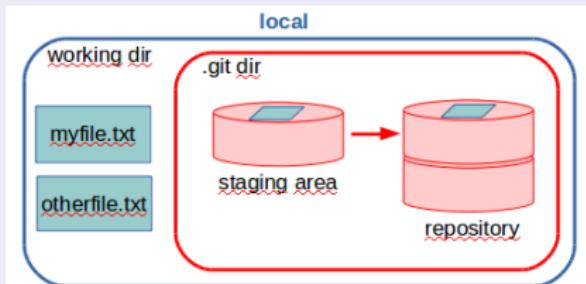
git add command for myfile.txt:

git commit -m "my reason":



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

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



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

git Objective

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"?
```

git commit: objective 4

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.

git cloning: objective 5

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 checked 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



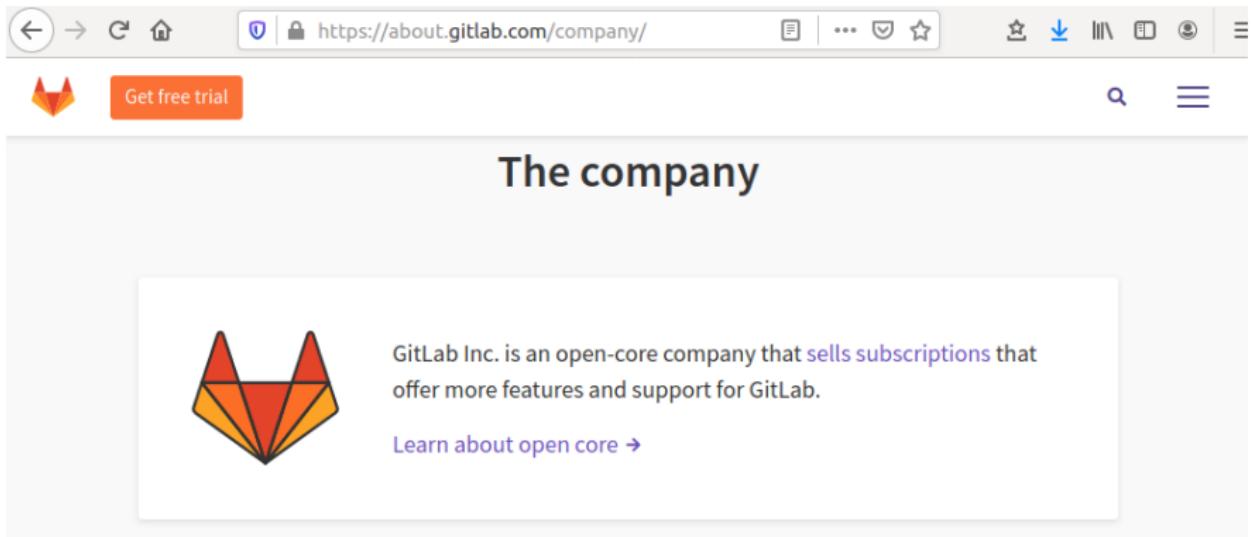
Quizz

- 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 displaying the GitLab company page at <https://about.gitlab.com/company/>. The page has a header with a logo, a "Get free trial" button, and navigation icons. The main section is titled "The company". It features a graphic of a stylized mountain or flame made of orange and red triangles. Below the graphic, text explains that GitLab Inc. is an open-core company that sells subscriptions for more features and support. A link to learn about open core is provided.

GitLab Inc. is an open-core company that [sells subscriptions](#) that offer more features and support for GitLab.

[Learn about open core →](#)

Quizz

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



a social network ✓

A screenshot of a GitHub user profile for Hadley Wickham. The profile picture shows a man with a beard. The bio reads: "Hadley Wickham
Hadley
Chief Scientist at @rstudio". Below the bio, it says "Follow" and "13.3k followers - 6 following - 115". The main content area shows several repositories: "dplyr" (4.4k stars), "gridverse" (3.5k stars), "tidyverse" (1.1k stars), "rlang" (2.3k stars), "advr" (3.1k stars), and "gridextra" (1.1k stars). At the bottom, there's a note: "An implementation of the Grammar of Graphics in R".

a desktop application ✓

A screenshot of a GitHub repository page for "dplyr2". The repository has 366 stars and 29 branches. A red oval highlights the "Clone with SSH" button. The repository description is: "An implementation of the Grammar of Graphics in R". The code tab shows 4,700 commits. The issues tab shows 1 open issue. The pull requests tab shows 30 open pull requests.

a tool to create websites ✓

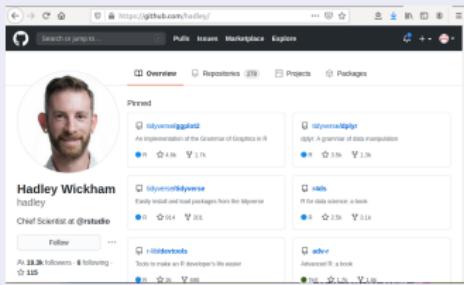
A screenshot of the GitHub Pages interface. It features a header "GitHub Pages" and a sub-header "Websites for you and your projects. Hosted directly from your GitHub repository. Just edit, push, and your changes are live." Below this, there are two examples of websites: "jekylly" and "jekylly". Both examples show a landing page with a "Transform your plain static websites and blogs" message. The "jekylly" example also includes a note about "Static" and "Single" site types.

a stable repository ...

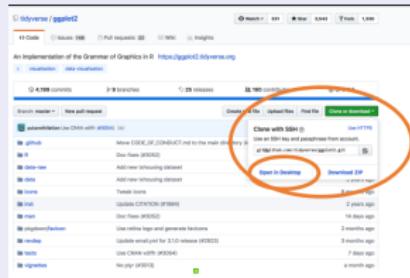
Popularity [edit]					
Name	Users	Projects	Alexa rank (lower = more popular)		
Assembler	unknown	526,551 (18)	33,434 as of 28 July 2020 (41)		
Bitbucket	5,931,060 (45)	Unknown	1,341 as of 20 July 2020 (91)		
Buddy	Unknown	Unknown	39,857 as of 28 July 2020 (50)		
CloudForge	Unknown	Unknown	402,884 as of 28 July 2020 (9)		
Gitsa	Unknown	Unknown	236,332 as of 28 July 2020 (51)		
Gitmix	31,000,000 (5)	180,000 (99)	78 as of 28 July 2020 (4)		
GitLab	100,000 (51)	546,000 (98)	2,710 as of 20 July 2020 (51)		
GNU Savannah	93,346 (18)	3,949 (18)	162,054 as of 20 July 2020 (9)		
Lazishredder	5,985,288 (30)	49,883 (1)	11,533 as of 28 July 2020 (51)		
OSDN	54,026 (93)	6,294 (1)	8,700 as of 20 July 2020 (91)		
OurProject.org	6,333 (91)	1,849 (1)	1,085,012 as of 28 July 2020 (91)		
OW2 Consortium	Unknown	Unknown	1,569,677 as of 28 July 2020 (91)		
Rosetta code	Unknown	Unknown	96,029 as of 28 July 2020 (51)		
SEUL	Unknown	Unknown	3,002,812 as of 28 July 2020 (91)		
SourceForge	3,790,060 (70)	500,000 (70)	470 as of 20 July 2020 (51)		
Name	Users	Projects	Alexa rank (lower = more popular)		

https://en.wikipedia.org/wiki/Comparison_of_source-code-hosting_facilities en.wikipedia,

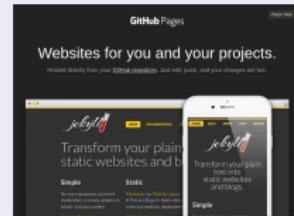
a social network ✓



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:

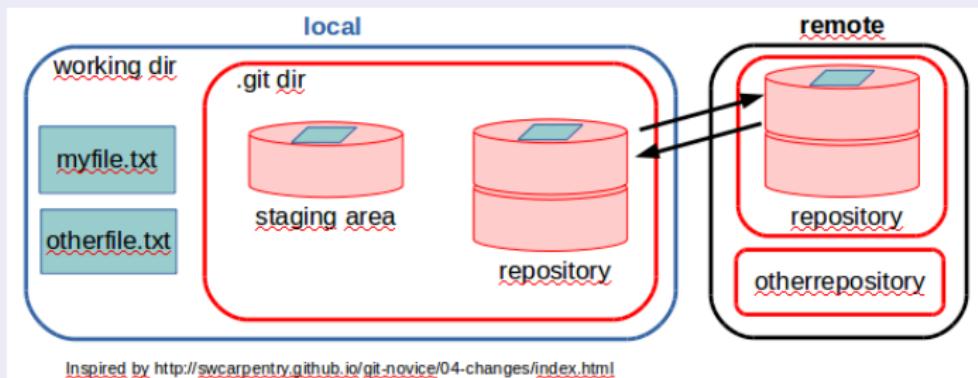
- R script: ✓
- Python script: ✓
- pdf file: ✗
- fastq file: ✗



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 on 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

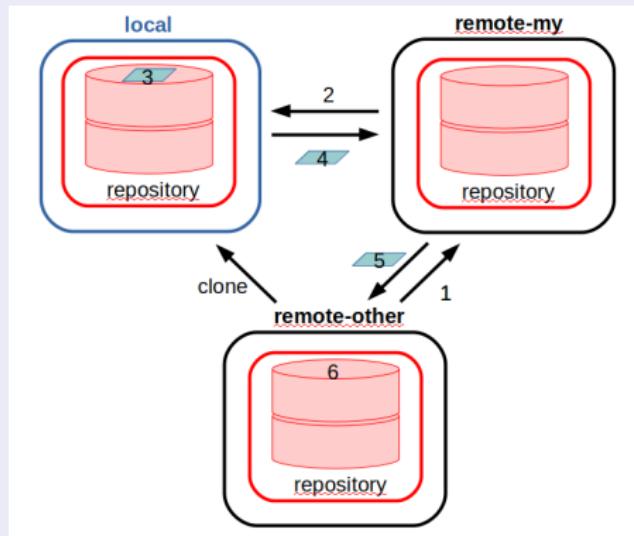


Clone vs. Fork?

- 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

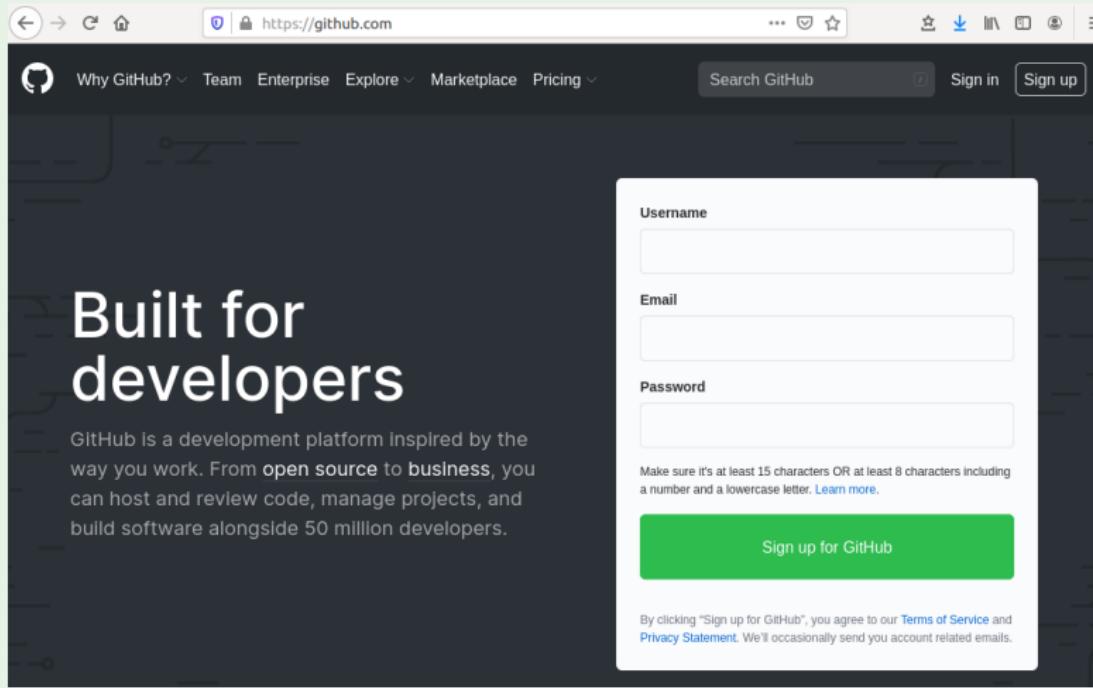
The objective of this exercise is to propose change to an existing project.
We will:

- fork an existing project to our GitHub account
- create a branch
- make 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 with a dark background. On the left, there's a large white text area with the heading "Built for developers". Below it, a paragraph describes GitHub as a development platform. On the right, there's a prominent sign-up form with three input fields: "Username", "Email", and "Password". Below the password field is a note about password requirements. At the bottom of the form is a green button labeled "Sign up for GitHub". A small note at the very bottom of the page states that by clicking the button, you agree to the Terms of Service and Privacy Statement.

Why GitHub? Team Enterprise Explore Marketplace Pricing

Search GitHub

Sign in Sign up

Built for developers

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.

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.

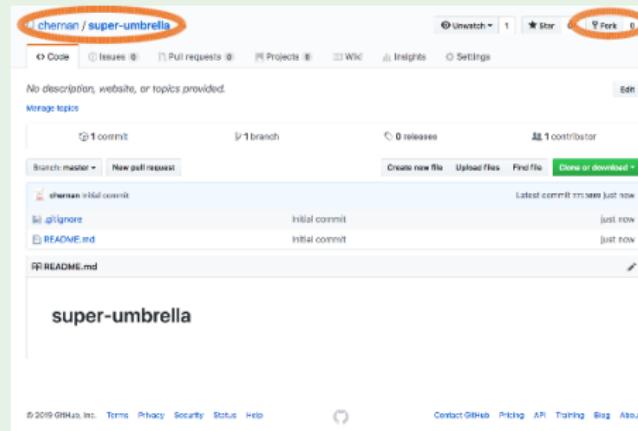
GitHub: fork a project

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):





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".

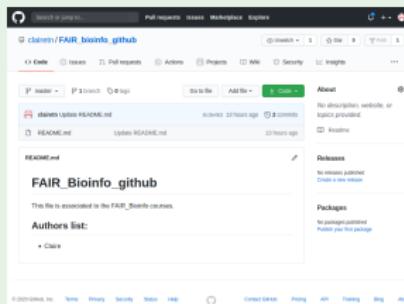
result of the fork chernan, super-umbrella project:

The screenshot shows the GitHub Overview page for the user clairetn. The top navigation bar includes 'Overview' (which is highlighted), 'Repositories' (6), 'Projects', and 'Packages'. On the left, there's a profile section for clairetn with an 'Edit profile' button and a 'Highlights' section listing 'Arctic Code Vault Contributor'. The main content area displays three repositories: 'FAIR_bioinfo.github' (1 commit), 'FAIR_bioinfo_docs' (Forked from chernan/FAIR_bioinfo_docs, TeX), and 'super-umbrella' (Forked from chernan/super-umbrella).

Tabs

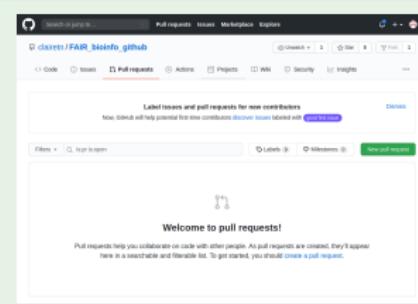
8 Tabs offered by GitHub for each repository:
Code, Pull Requests, Actions, Projects, Wiki, Security, Insights, Settings.
Mainly focus on 3 of them:

Code



This screenshot shows the GitHub Code tab for the repository 'claires/FAIR_Bioinfo.github'. The page displays the README files (README.md and Update-README.md), which have been updated 20 hours ago. It also shows the READEme file and the FAIR_BioInfo_github package. The 'About' section indicates that there are no descriptions, website, or topics provided. The 'Releases' section shows that no releases have been published. The 'Packages' section shows that no packages have been published.

Pull Requests



This screenshot shows the GitHub Pull Requests tab for the repository 'claires/FAIR_Bioinfo.github'. The page displays a message encouraging users to label issues and pull requests for new contributors. It features a 'Welcome to pull requests!' section with instructions on how pull requests help collaborate on code. There are filters for 'Labels' and 'Milestones'.

Wiki



This screenshot shows the GitHub Wiki tab for the repository 'claires/FAIR_Bioinfo.github'. The page displays a welcome message for the FAIR_Bioinfo repository, encouraging users to keep the repository up-to-date, show the current status, and document software releases. It includes a 'Welcome to the FAIR_Bioinfo.github wiki' section and a 'Edit this page' button.

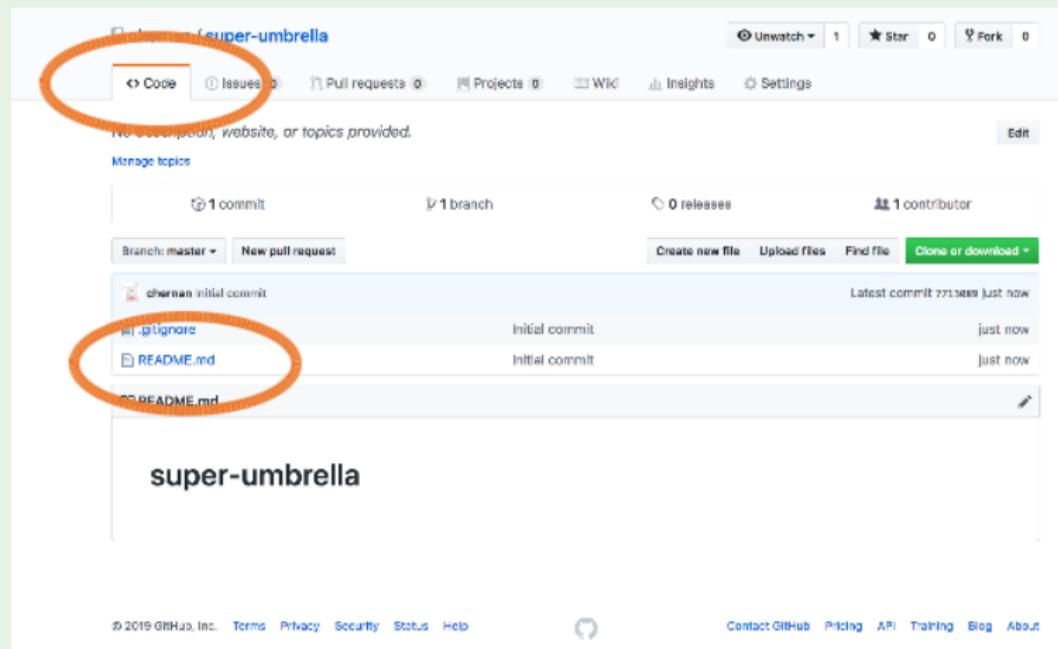
Previous exercises with git

- 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:

- ➊ fork a github repo. (just done)
- ➋ create branch
- ➌ make change (edit file)
- ➍ version change (commit)
- ➎ compare branch to master
- ➏ merge branch
- ➐ ask for merging (Pull Request)
- ➑ delete branch

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



The screenshot shows a GitHub repository page for 'super-umbrella'. At the top, there is a navigation bar with links for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Insights, and Settings. The 'Code' link is highlighted with an orange oval. To the right of the navigation bar are buttons for Unwatch (1), Star (0), Fork (0), and a Settings gear icon.

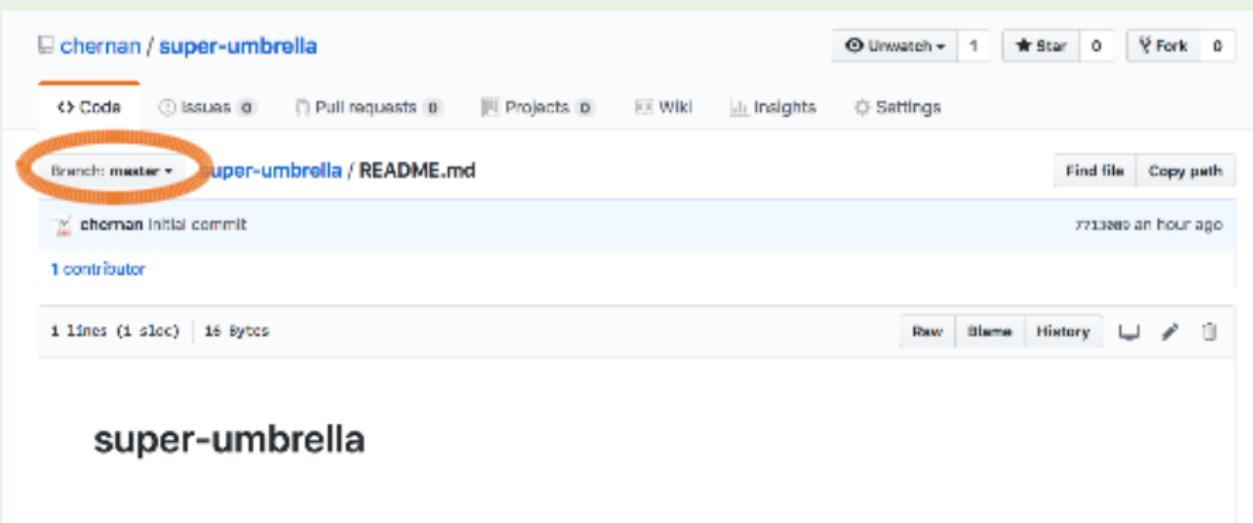
The main content area displays a message: 'No description, website, or topics provided.' Below this, there is a 'Manage topics' section and a summary of repository statistics: 1 commit, 1 branch, 0 releases, and 1 contributor. A 'Branch: master' dropdown and a 'New pull request' button are also present.

The repository's history is shown in a list of commits:

- chernan initial commit (just now)
- .gitignore (just now)
- README.md (just now)
- super-umbrella (just now)

At the bottom of the page, there is a footer with links for Contact GitHub, Pricing, API, Training, Blog, and About. The footer also includes the GitHub logo and navigation icons.

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



The screenshot shows a GitHub repository page for 'chernan / super-umbrella'. At the top, there are navigation links for 'Code', 'Issues', 'Pull requests', 'Projects', 'Wiki', 'Insights', and 'Settings'. A dropdown menu labeled 'Branch: master' is highlighted with an orange circle. Below the dropdown, the file 'super-umbrella / README.md' is shown, with a commit message from 'chernan' stating 'Initial commit'. The commit was made an hour ago. There is one contributor listed. The file content is displayed below, showing the text 'super-umbrella'.

3: edit README.md to make change

chernan / super-umbrella

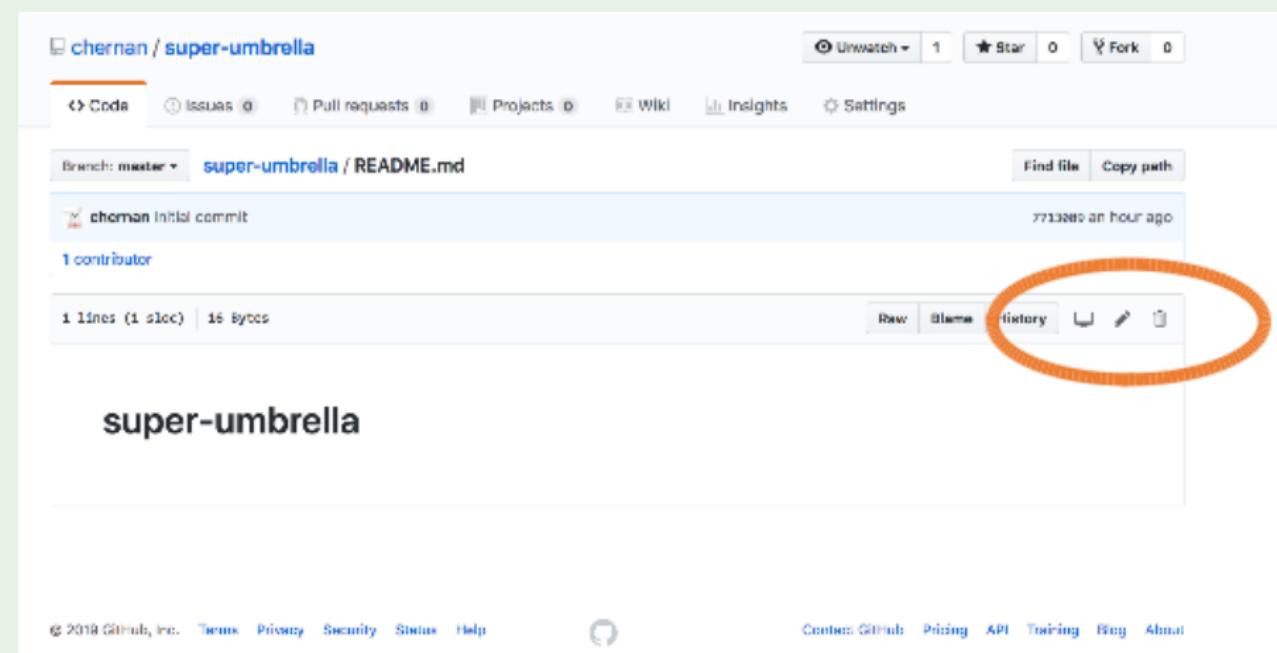
Code Issues Pull requests Projects Wiki Insights Settings

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

chernan Initial commit 7733289 an hour ago

1 contributor

1 lines (1 sloc) | 16 Bytes Raw Blame history

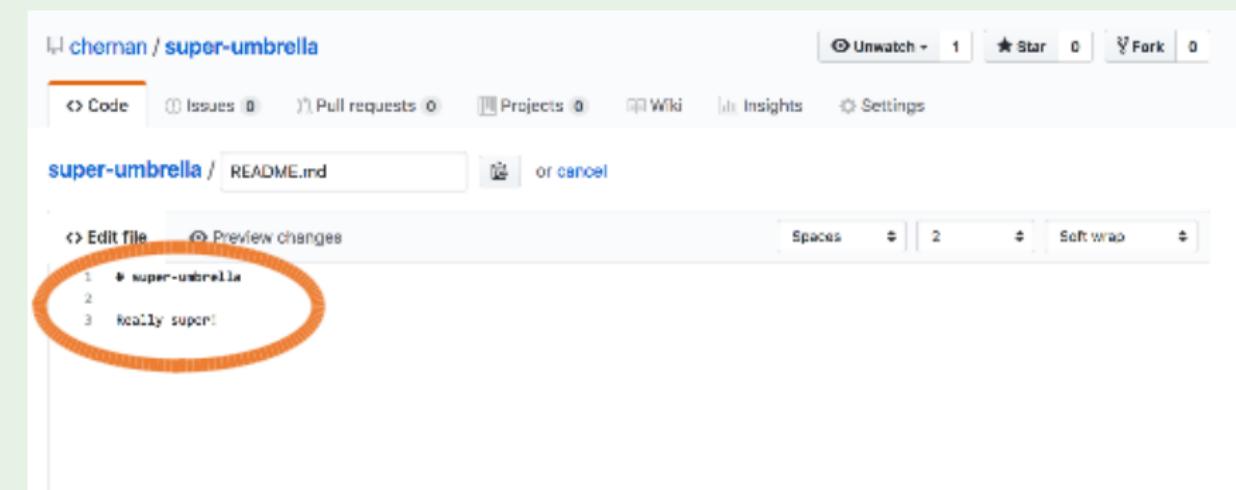


super-umbrella

© 2018 GitHub, Inc. Terms Privacy Security Status Help

Contact GitHub Pricing API Training Blog About

3: make change

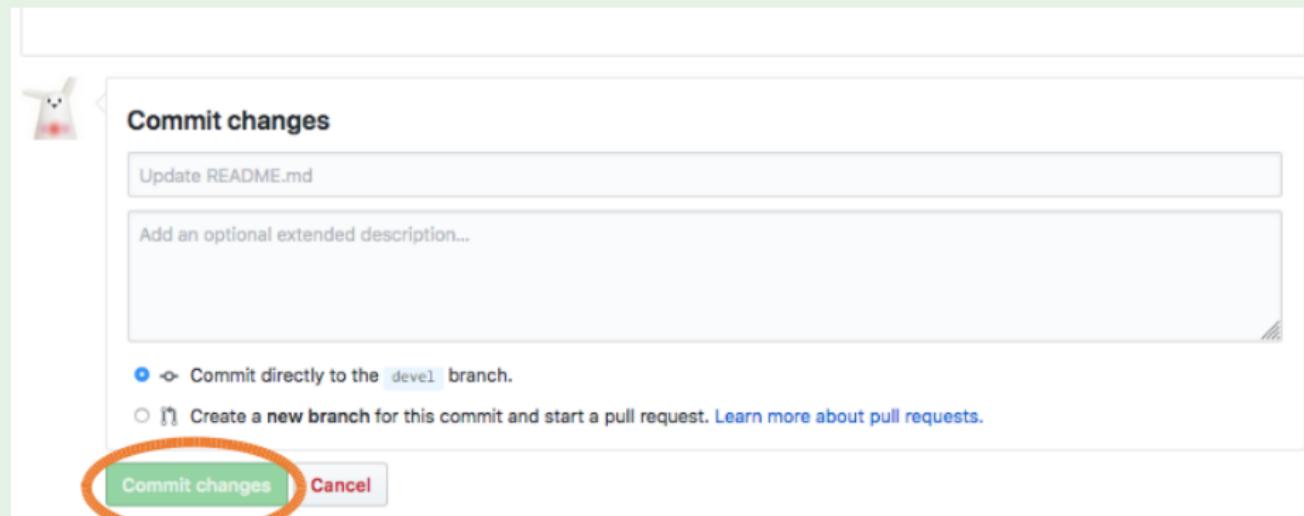


The screenshot shows a GitHub repository page for 'cherman / super-umbrella'. The 'Code' tab is selected. In the center, there's a modal window for editing the 'README.md' file. The modal has two tabs: 'Edit file' (selected) and 'Preview changes'. The preview area contains the following text:

```
1 * super-umbrella
2
3 Really super!
```

A large orange oval highlights the first line of the commit message, '* super-umbrella'.

4: commit



Commit changes

Update README.md

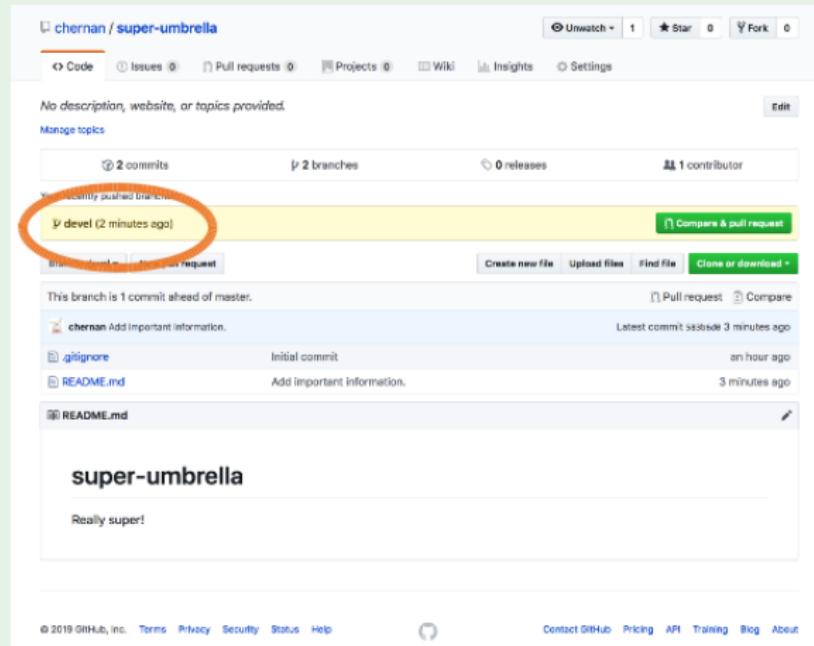
Add an optional extended description...

➔ 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



The screenshot shows a GitHub repository page for 'chernan / super-umbrella'. At the top, there are navigation links for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Insights, and Settings. Below the header, it says 'No description, website, or topics provided.' and has a 'Manage topics' link. It displays summary statistics: 2 commits, 2 branches, 0 releases, and 1 contributor. A red circle highlights the 'dvel (2 minutes ago)' branch under 'Your recently pushed branches'. Below the branches, there's a button for 'Compare & pull request'. The main content area shows the 'dvel' branch with one commit ahead of 'master'. The commit details are: 'chernan Add Important Information.' (Initial commit, added 3 minutes ago) and 'chernan Add Important Information.' (Latest commit, added 3 minutes ago). There are also file lists for 'ignore', 'README.md', and 'README.md'. The repository title 'super-umbrella' is at the bottom, followed by the text 'Really super!'. At the very bottom, there are links for Terms, Privacy, Security, Status, Help, Contact GitHub, Pricing, API, Training, Blog, and About.

5: pull request, compare

chernan / super-umbrella

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:

dvel (2 minutes ago) 

Branch: dvel ▾ 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 3 minutes ago

ignore Initial commit an hour ago

README.md Add important Information. 3 minutes ago

README.md

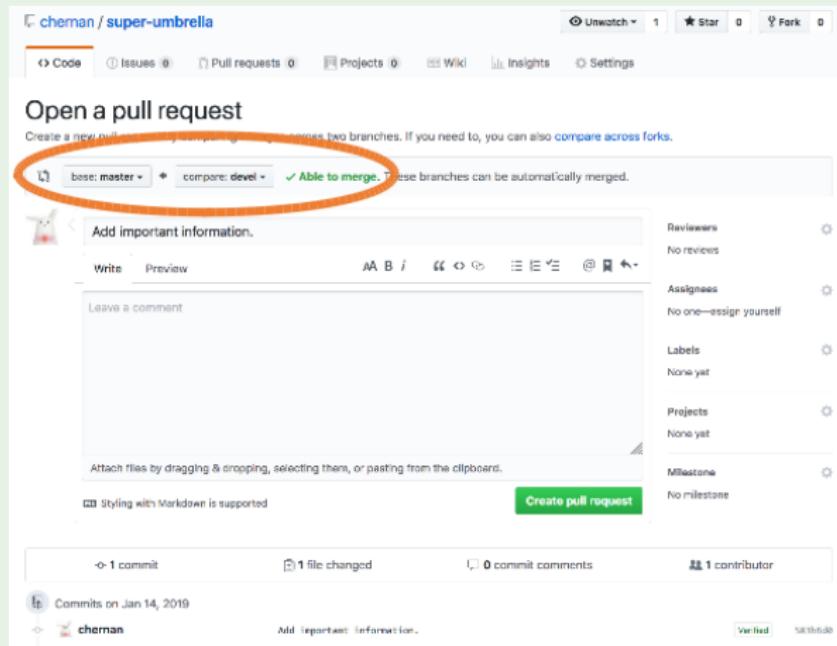
super-umbrella

Really super!

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

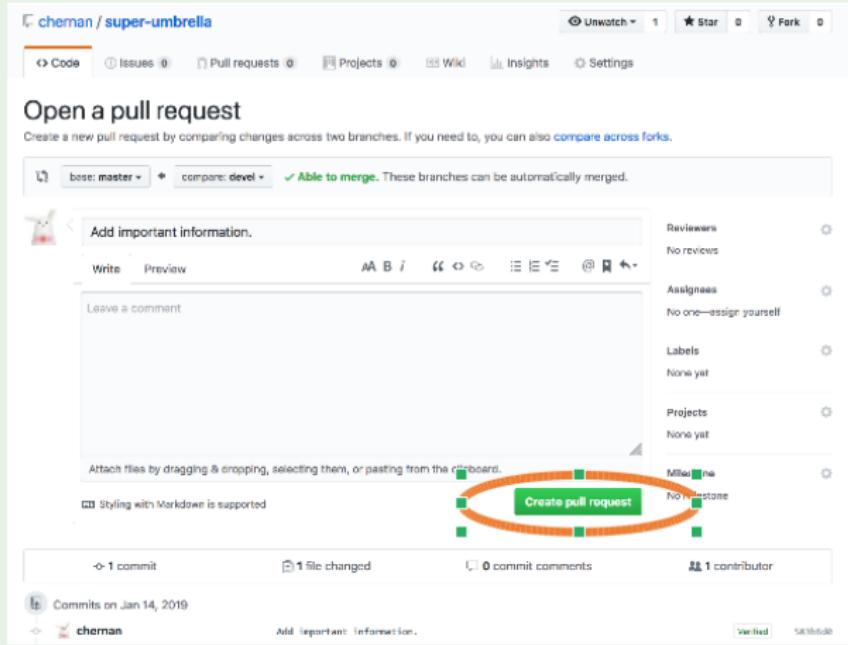


5: pull request, able to merge



The screenshot shows the GitHub interface for creating a pull request. At the top, there's a navigation bar with links for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Insights, and Settings. Below the navigation, a large button says "Open a pull request". A callout box highlights the "Able to merge" status, which is indicated by a green checkmark and the text "These branches can be automatically merged". The main form area has sections for "Add important information" (with Write and Preview tabs), "Leave a comment", and "Attach files". On the right side, there are fields for Reviewers (No reviewers), Assignees (No one—assign yourself), Labels (None yet), Projects (None yet), and Milestone (No milestone). At the bottom, there are summary statistics: -1 commit, 1 file changed, 0 commit comments, and 1 contributor. The contributor is listed as "cherman" with a commit made on Jan 14, 2019.

5: merge and pull request



cherman / super-umbrella

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

Leave a comment

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

Styling with Markdown is supported

1 commit 1 file changed 0 commit comments 1 contributor

Commits on Jan 14, 2019

cherman Add important information. Verified Screenshot

Reviewers No reviews

Assignees No one—assign yourself

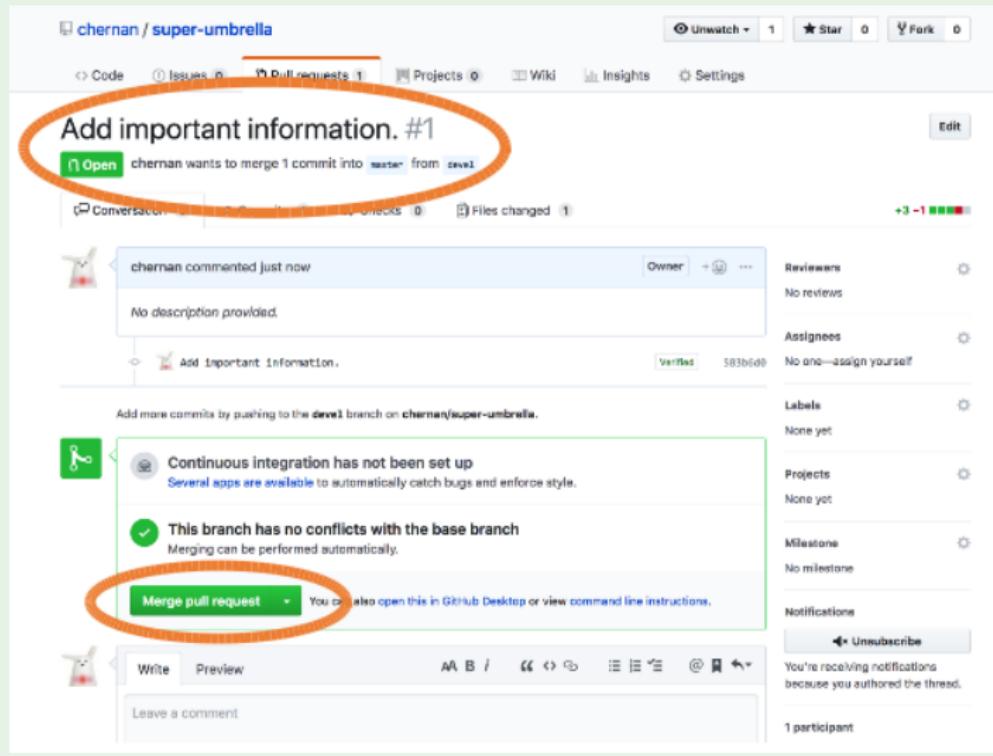
Labels None yet

Projects None yet

Milestones No milestone

Create pull request

5: merge and Pull request



The screenshot shows a GitHub pull request page for the repository `chernan / super-umbrella`. The pull request has been merged, indicated by the green "Merge pull request" button at the bottom.

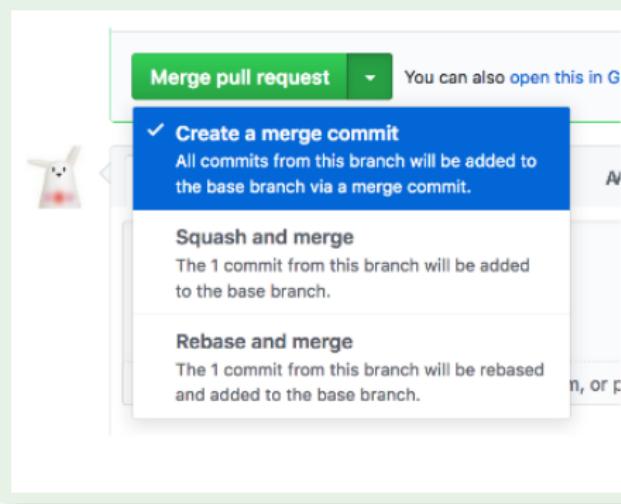
Annotations:

- A red oval highlights the title "Add important information. #1".
- A red oval highlights the green "Merge pull request" button at the bottom of the main content area.

Page Content:

- Header:** chernan / super-umbrella, Unwatch, 1 star, 0 forks.
- Navigation:** Code, Issues (0), Pull requests (1), Projects (0), Wiki, Insights, Settings.
- Title:** Add important information. #1
- Description:** chernan wants to merge 1 commit into `master` from `devol`.
- Comments:** chernan commented just now: "No description provided." (verified, 583b6d9)
- Actions:** Add Important Information.
- CI Status:** Continuous integration has not been set up. Several apps are available to automatically catch bugs and enforce style.
- Branch Status:** This branch has no conflicts with the base branch. Merging can be performed automatically.
- Buttons:** Merge pull request (green button), Write, Preview, AA B /, @, etc.
- Notifications:** You're receiving notifications because you authored the thread. 1 participant.

5: merge and Pull request



Merge pull request You can also open this in G

- ✓ **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.

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

Code Issues Pull requests Projects 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 green red

chernan commented 7 minutes ago

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.

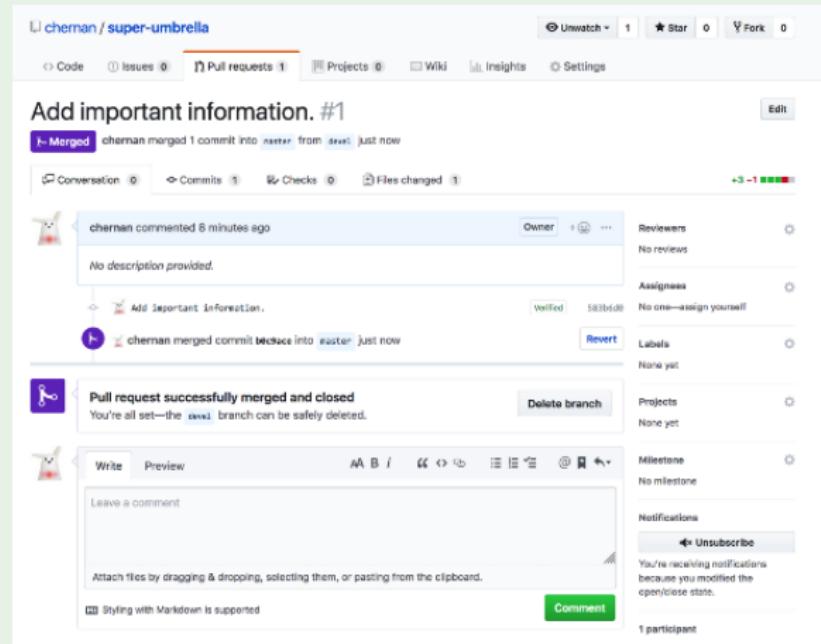
Confirm merge Cancel

Write Preview AA B i CC @ Unsubscribe

Leave a comment

1 participant

7: the merge delete the branch



The screenshot shows a GitHub pull request page for the repository "cherman / super-umbrella". The pull request has been merged, as indicated by the "Merged" status and the message "cherman merged 1 commit into master from dev". The pull request details include a conversation, commits, checks, and files changed. A prominent "Delete branch" button is visible next to the merge commit message. Below the pull request, a message states "Pull request successfully merged and closed" and "You're all set—the dev branch can be safely deleted." A "Delete branch" button is also present here. At the bottom, there is a comment section with a "Comment" button and a note about Markdown support.

See version network, Insights tab

chernan / super-umbrella

Code Issues Pull requests Projects Wiki Insights Settings

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

Owners Jan chernan 13



Jan

13

chernan

master

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.

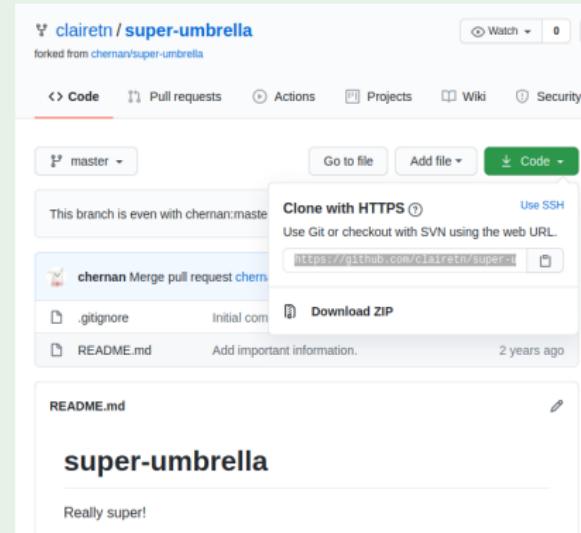
Steps

- 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):



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
```

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

Bonus

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

Ressources

- Learning Git by [Software Carpentry](#):
<https://swcarpentry.github.io/git-novice/>
- [Git Cheat Sheets](#): <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-Oxford/git/>
- [Pierre Poulain fr ressources](#): <https://cupnet.net/git-github/>

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, ...