

FAIR_bioinfo for bioinformaticians

Introduction to the tools of reproducibility in bioinformatics

C. Hernandez¹ T. Denecker¹ J.Sellier² C. Toffano-Nioche¹

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

²Institut Français de Bioinformatique
à compléter

Sept. 2020



Introduction to code versioning

Really need of a files history?

"FINAL".doc



FINAL.doc!



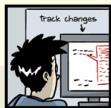
FINAL_rev.2.doc



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
ICOMETOGRADSCHOOL????.doc



WWW.PHPCOMICS.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'."

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

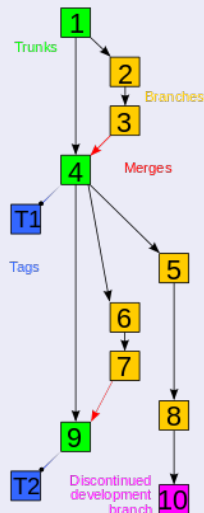
Each revision is associated with a timestamp and the person making the change. Revisions can be compared, restored, and merged.

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

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

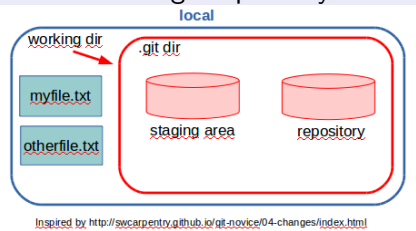
<https://www.powershellmagazine.com/2015/07/13/git-for-it-professionals-getting-started-2/>

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:

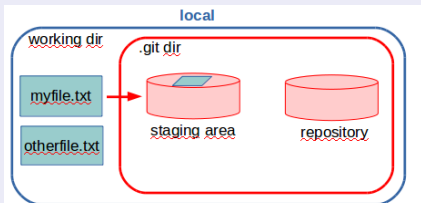


3 ways to initialize a `.git` repository:

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

Tracking file

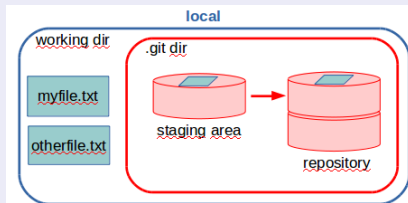
git add command for myfile.txt:



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

Objective

- install git
- initialize git
- create a git repository
- use the basic git commands for tracking changes
- copy another repository from github
- use branching and merging to manage code change

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 file

create 2 files, check their git status

```
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
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:

```
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
- as our previous git project, 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!

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
```

go into the new "branch1"

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

work inot 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

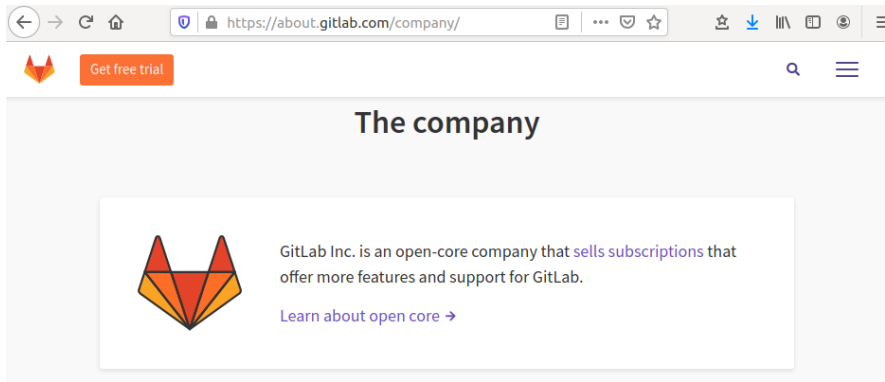
Quizz

- 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?



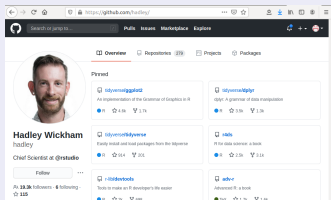
The screenshot shows a web browser window with the address bar displaying `https://about.gitlab.com/company/`. The page header includes the GitLab logo (a stylized fox head) and a "Get free trial" button. The main heading is "The company". Below this, there is a white box containing the GitLab logo and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



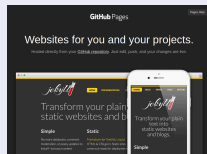
Quizz

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

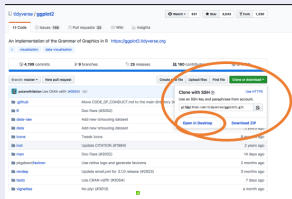
a social network



a tool to create websites



a desktop application



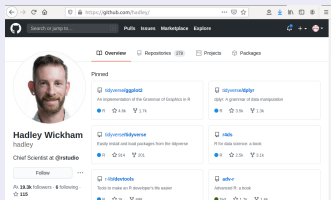
a stable repository ...

Popularity | [view](#)

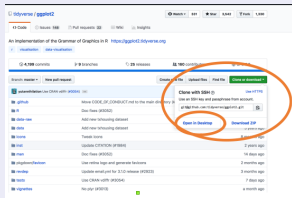
Name	Users	Projects	Alexa rank (lower = more popular)
Assembly	Unknown	526,381 ⁽¹⁸⁾	33,434 as of 28 July 2020 ⁽¹¹⁾
Bitbucket	5,000,000 ⁽¹⁴⁾	Unknown	1,341 as of 28 July 2020 ⁽¹¹⁾
Buddy	Unknown	Unknown	39,857 as of 28 July 2020 ⁽¹¹⁾
CloudPurge	Unknown	Unknown	402,888 as of 28 July 2020 ⁽¹¹⁾
Gitex	Unknown	Unknown	216,332 as of 28 July 2020 ⁽¹¹⁾
GitLab	31,000,000 ⁽¹²⁾	100,000 ⁽¹³⁾	78 as of 28 July 2020 ⁽¹¹⁾
GitLab	100,000 ⁽¹²⁾	\$4,000 ⁽¹³⁾	2,710 as of 28 July 2020 ⁽¹¹⁾
GNU Savannah	\$3,340 ⁽¹⁶⁾	3,940 ⁽¹⁶⁾	162,054 as of 28 July 2020 ⁽¹¹⁾
Launchpad	\$3,985,288 ⁽¹⁶⁾	40,881 ⁽¹⁶⁾	11,337 as of 28 July 2020 ⁽¹¹⁾
OSDN	\$4,020 ⁽¹⁷⁾	6,294 ⁽¹⁷⁾	8,708 as of 28 July 2020 ⁽¹¹⁾
Ourproject.org	6,313 ⁽¹¹⁾	1,840 ⁽¹¹⁾	1,083,012 as of 28 July 2020 ⁽¹¹⁾
OW2 Consortium	Unknown	Unknown	1,506,877 as of 28 July 2020 ⁽¹¹⁾
Openstack	Unknown	Unknown	68,029 as of 28 July 2020 ⁽¹¹⁾
SVN	Unknown	Unknown	1,602,812 as of 28 July 2020 ⁽¹¹⁾
SourceForge	3,700,000 ⁽¹⁷⁾	500,000 ⁽¹⁷⁾	470 as of 28 July 2020 ⁽¹¹⁾
Name	Users	Projects	Alexa rank (lower = more popular)

[en.wikipedia](https://en.wikipedia.org/wiki/List_of_source-code_hosting_facilities), 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:

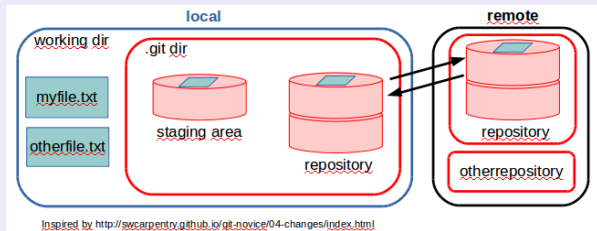
- R script: 
- Python script: 
- pdf file: 
- fastq file: 



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 added functionalities
- git-based: so git concepts and commands are retained
- commands for the "sharing step": `git push origin master` (from local to remote) and `git pull origin master` (from 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. Like a place to compare and discuss the differences introduced on a branch with reviews, comments, integrated tests, etc

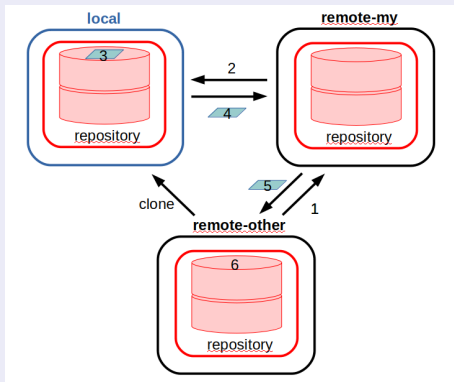
Clone vs. Fork?

- 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

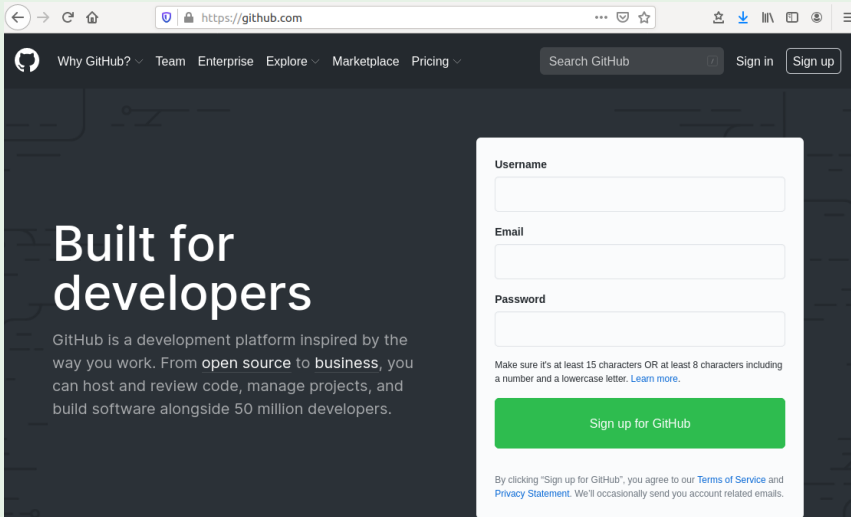


(direct clone from github don't allow to collaborate)

- 1: fork a repository if interest in your github account
- 2: clone from your github account to your local place
- 3: make change (branch, add, commit)
- 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

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 web browser. The browser's address bar displays <https://github.com>. The page header includes the GitHub logo, navigation links (Why GitHub?, Team, Enterprise, Explore, Marketplace, Pricing), a search bar, and 'Sign in' and 'Sign up' buttons. The main content area features the heading 'Built for developers' and a paragraph about GitHub's mission. A white sign-up form is overlaid on the right side of the page, containing fields for Username, Email, and Password, along with a green 'Sign up for GitHub' button and a disclaimer at the bottom.

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.

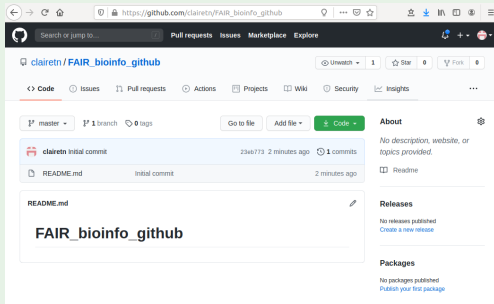
GitHub: fork a project

Objective

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

1: Fork in our gituhb account

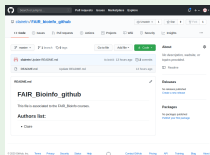
With a browser, go to the url of the initial project, [FAIR_bioinfo_github](https://github.com/clairertrn/FAIR_bioinfo_github) and click to "Fork" (upper right):



Tabs

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

Code



Pull Requests



Wiki



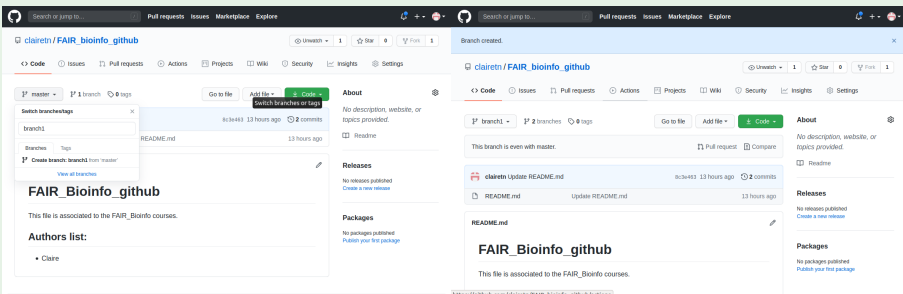
Previous Exercise with git

- 1 copy a github repo. (git clone)
- 2 go to the local repo. (cd)
- 3 create branch (git branch)
- 4 go to branch (git checkout)
- 5 make change (edit file)
- 6 stage change (add)
- 7 version change (commit)
- 8 go to master (git checkout)
- 9 merge branch (git merge)
- 10 delete branch (git branch -d)

Steps with github interface

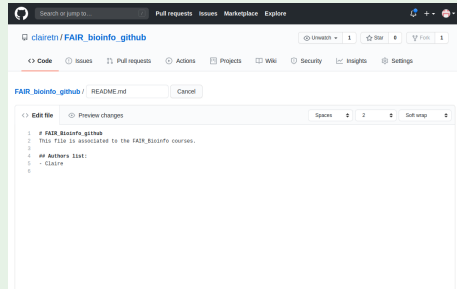
- 1 fork a github repo. (just done)
- 2 create branch
- 3 make change (Edit file)
- 4 version change (commit but.)
- 5 compare branch to master
- 6 merge branch
- 7 ask for merging (Pull Request)
- 8 delete branch

2: create branch1



The screenshot shows the GitHub web interface for the repository `clairertrn/FAIR_bioinfo_github`. The left sidebar shows the repository structure with a file `README.md` and a commit history of 2 commits. A modal window titled "Switch branches/tags" is open, showing a list of branches: `master` and `branch1`. The `branch1` branch is selected, and a button "Create branch: branch1 from 'master'" is visible. The main content area shows the `README.md` file content, which includes the repository name `FAIR_Bioinfo_github` and a description: "This file is associated to the FAIR_Bioinfo courses." The right sidebar shows the repository's metadata, including the number of stars (1) and forks (1).

3: make change by Edit file + 4: commit



Search or jump to... Pull requests Issues Marketplace Explore

clairetn / FAIR_bioinfo_github

Unwatch 1 Star 0 Fork 1

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

FAIR_bioinfo_github / README.md Cancel

Edit file Preview changes Spaces 2 Soft wrap

```
1 # FAIR_Bioinfo_github
2 This file is associated to the FAIR_Bioinfo courses.
3
4 ## Authors list:
5 - Claire
6
```



Commit changes

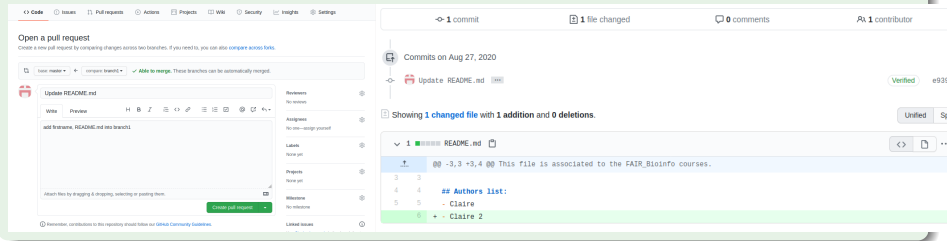
Update README.md

add firstname: README.md into branch1

- ☒ Commit directly to the `branch1` branch.
- ☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes Cancel

4: compare branch1 to master



The screenshot displays a GitHub pull request interface. On the left, the 'Open a pull request' section is visible, with a dropdown menu showing 'base: master' and 'compare: branch1'. The 'Add to merge' button is highlighted. Below this, the 'Update README.md' section shows a diff view with a green bar indicating a change. The main content area shows the diff for 'README.md', with a green bar indicating a change. The diff shows a new line added: '# Authors list: - Claire'. The right sidebar shows the 'Commits on Aug 27, 2020' section, with a commit titled 'Update README.md' and a diff view showing the same change.

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: branch1 • **Add to merge**. These branches can be automatically merged.

Update README.md

Write Preview H B I T E S G C A

add framework, README.md into branch1.

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

Create pull request

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

1 commit **1 file changed** **0 comments** **1 contributor**

Commits on Aug 27, 2020

Update README.md **Verified** e93...

Showing **1 changed file** with **1 addition** and **0 deletions**.

1 README.md

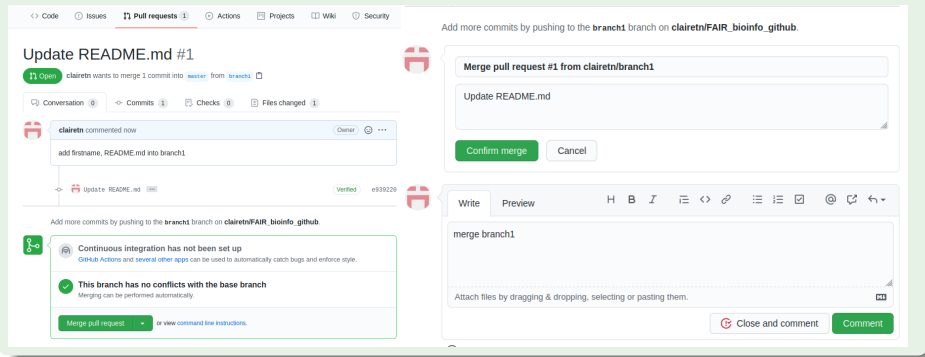
Diff **-3,3 +3,4** This file is associated to the FAIR_Bioinfo courses.

```

3 3
4 4  ## Authors list:
5 5  - Claire
6 6  + - Claire 2

```

5: Merge branch1 + 6: Pull Request

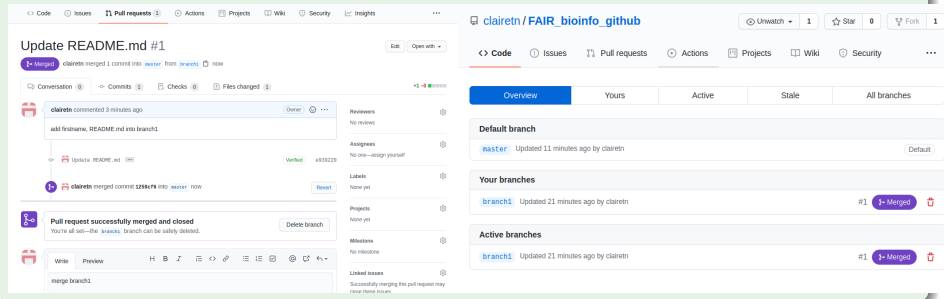


The screenshot displays a GitHub Pull Request (PR) titled "Update README.md #1". The PR is created by user "clairetn" and targets the "master" branch from the "branch1" branch. The PR description states "Update README.md".

On the left sidebar, the "Conversation" tab is active, showing a comment from "clairetn" stating "add firstname, README.md into branch1". Below the comment, a status box indicates that "Continuous integration has not been set up" and "This branch has no conflicts with the base branch". A "Merge pull request" button is visible at the bottom of this section.

On the right, the "Merge pull request" dialog is open, showing the title "Merge pull request #1 from clairetn/branch1" and the description "Update README.md". Below the dialog, the "Write" tab is active, showing the commit message "merge branch1". A "Close and comment" button and a "Comment" button are at the bottom right of the dialog.

Succed + 7: delete branch1



The screenshot shows a GitHub interface with two main sections. The left section displays a pull request titled "Update README.md #1" by user "clairetn". It shows a commit history with the message "add firstline, README.md into branch1" and a status of "Merged". Below this, a message states "Pull request successfully merged and closed". The right section shows the repository "clairetn / FAIR_bioinfo_github" with tabs for Code, Issues, Pull requests, Actions, Projects, Wiki, and Security. It lists the "Default branch" as "master" and "Your branches" as "branch1".

GitHub Exercise 2

extrait du programme IFB

Github/Gitlab (en linéaire) :

FORK d'un projet existant

Invitation collaborateur des intervenants

Ajouter son nom dans le README

Commit en local puis push sur Github

Pull request (à valider par un collaborateur)

Github collaboratif :

Clone du projet principal

Création d'une branche

Ajout de son nom dans le README en local

Demande de révision (Pull Request)

Merge de la branche

Récupération du README avec tous les noms (toutes les branches ont été mergées)



Objective

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

- fork an existing project to a local folder
- made a change by adding our name in the README file (local)
- save the change (local) and github (personal remote)
- create a pull request and waiting until its validation (remote)

Web interface

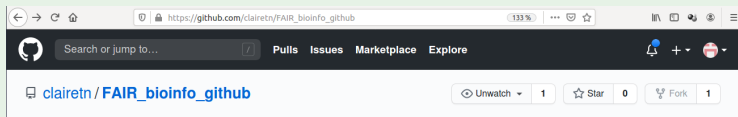
During this exercise, most of the actions that will be performed will be done through github's web interface, i.e. a lot of "click-buttons".

GitHub: Fork a repository

Repository to fork:

https://github.com/clairetn/FAIR_bioinfo_github

Click on the fork button:



Result:

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



GitHub: Add a change to the project

Clone our fork:

Make a copy of the forked repository in our local computer to be able to work on the project.

By command line with `git clone` or by the "clone" button on the GitHub interface.

Work on the project:

Edit the README file and add your name at the end of the file.

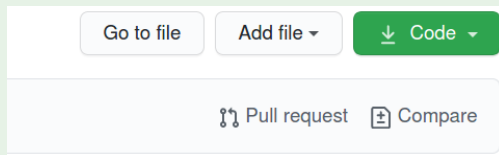
Git add, commit and push by GUI or command lines:

```
1 git add README.md
2 git commit -m "add name"
3 git push origin master
```

GitHub: Propose your change into the initial project

In your forked repository

"Compare" and then "Pull request" your issue (explain your proposals as much as possible):



The pull request asks the maintainer(s) to review your work, provide comments, request edits, etc. If your change will be approved, the maintainer(s) add your change into the code.

Wait for validation from the initial repository

...

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

ajouter éditeur intégrés avec git

Ressources

- <https://nbis-reproducible-research.readthedocs.io/en/latest/git/>
- <https://swcarpentry.github.io/git-novice/>: Learning Git by Software Carpentry
- <https://services.github.com/on-demand/resources/cheatsheets/>: git Cheat Sheets
- <https://jules32.github.io/2016-07-12-Oxford/git/>: step-by-step progression to link RStudio and GitHub
- <https://cupnet.net/git-github/>: Pierre Poulain fr ressources