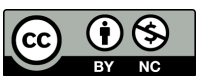
Introduction to Github

Jerome Mathieu & Rosalie Bruel
April 2023
Sorbonne University



Intro : what is GitHub

Roughly speaking, GitHub can be seen as online folder, just like a dropbox or a GoogleDrive. You can upload documents from your computer into online folders (called "**repositories**") or, do the opposite: download them into your computer.
The repository can be shared with other people, allowing them to download and upload documents, and make modifications to the files.

But there are also strong differences with a traditional online folder.

1° There is a controlled **versioning system**. This means that the system tracks the history of each folder and file. Modifications need to be accepted manually by the admins of the repository. This ensures that the changes in the files are wanted and trackable, and allows dealing with conflicts (eg when several people modified the same file at the same moment).

So contrary to Dropbox and similar cloud folders, there is an extra manual step during the synchronizing between the local and remote folders. This step ensures that modifications are wanted.
Modifications either from the cloud to the computer or from the computer to the cloud need to be accepted by the user.

2° **There are two ways to interact with a GitHub repository**

- If you are the owner or an administrator of the repository
  → In this case you can accept/rejects the modifications in the folder. See chapter II for details.

- If you are not the owner/admin of the repository
  → In this case you are not allowed to make modifications. What you can do is copy ("**fork**') the folder on your computer, modify your version of the folder (now called a '**branch**'), and at some point, propose to merge your branch to the original folder ("**pull request**"). See chapter 4 for details.The admin can accept or reject your proposal

Good to know

- GitHub repositories can be directly published on Zenodo and get a DOI, which makes the code citable.
- Repositories can be open to everyone, or restricted.
- GithHub can host any type of file.
- You can host one html website on github for free.

1 - Get started with your GitHub account

1.1 Create an account on GitHub

https://github.com/

1.2 Get familiar with your space

In the top right of the web explorer,

- go to “your profile” → this is your public profile
- go to “Repositories” → This is where all your repositories will be accessible

2 Exploring a repository on Github.com

2.1 Finding a repository

- Click on the search bar near the GitHub logo in the top left of the screen (1)
- write the key words
- Choose where to search (2): either on the entire website or in your folders

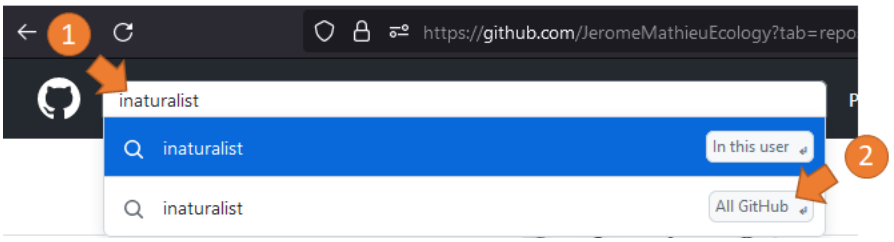


Figure 1 *How to search for repositories*

2.2 Structure of the page of a GitHub repository

There are many sections on the webpage of a repository. We'll focus here on the ones which are needed for a user. By the way, **it is strongly recommended to interact with files (donwload, upload) from your computer, not from the interface (see section 2).** Use the interface mostly to understand the history of the repo, check/make open issues, track pull requests or participate in discussions.

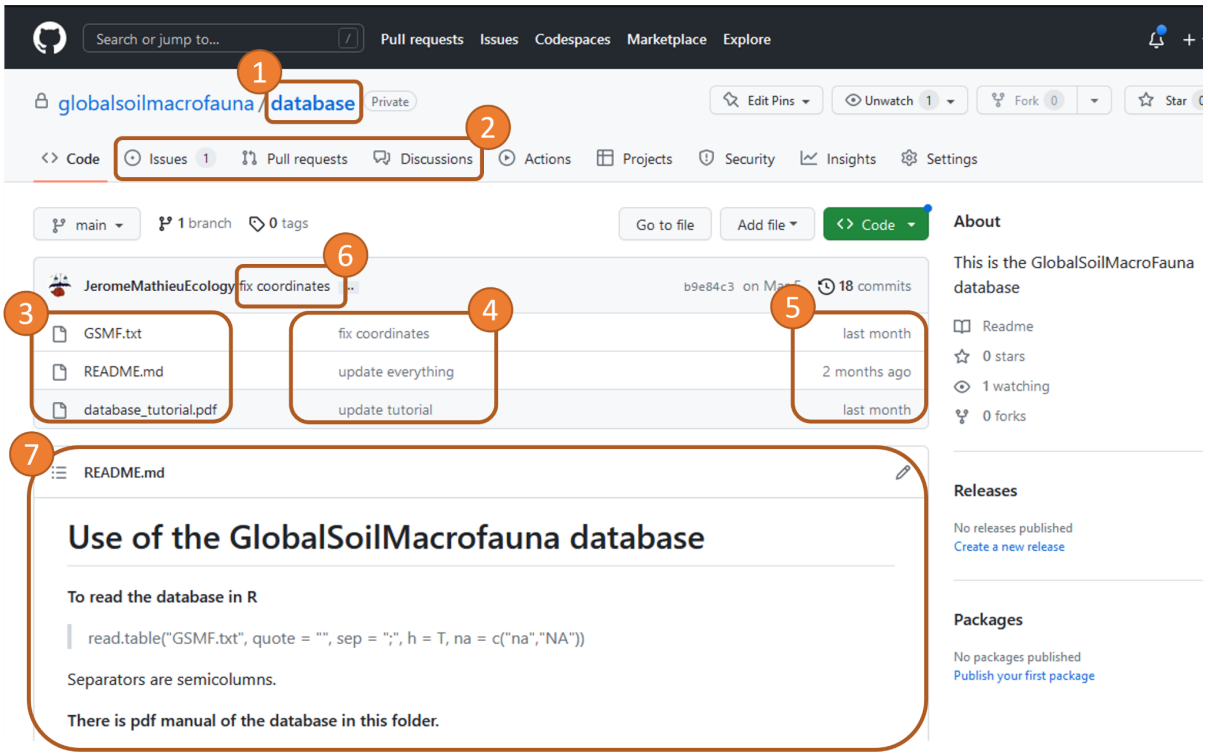


***Figure 2** Structure of the webpage of a repository*

(1) Name of the repository (here 'database')
(2) Places to interact with the owner of the repo and other users (see section 4 for details)
(3) List of all the folders and files in the repository.
(4) Name of the latest commit* in which the file (in 3) was modified.
(5) Date of the last commit on the file.
(6) Name of the last commit in the repository, all files considered.
(7) Section giving any useful detail about the repository. This text is actually stored in the REAMDE.md file

* A **commit** is a new version of a file or of a repository. Basically an update. In GitHub, all updates have a name and date, indicated here in 3 and 5 for the files, and 6 for the repository

2.3 Open/download a file

As mentioned in the intro, it is strongly recommende to download files from a GitHub application on your computer (section 3) instead of downloading manually.
The way to download the files in the repo depends on the nature of file. In particular some types of files, such as .md, pdf, csv, can be displayed, while other can't. But the general idea is:
1° Click on the name of the file you want to download (1) in Figure 2
A new page should open.
2° Click on download (2).
3° If you don't have the possibility to save right away the file with the button 'download' then go back to the page with the download button.
4° click on raw file, or view raw (3). The file should appear in a new window. Do a right-click on the new page and click 'save as'.

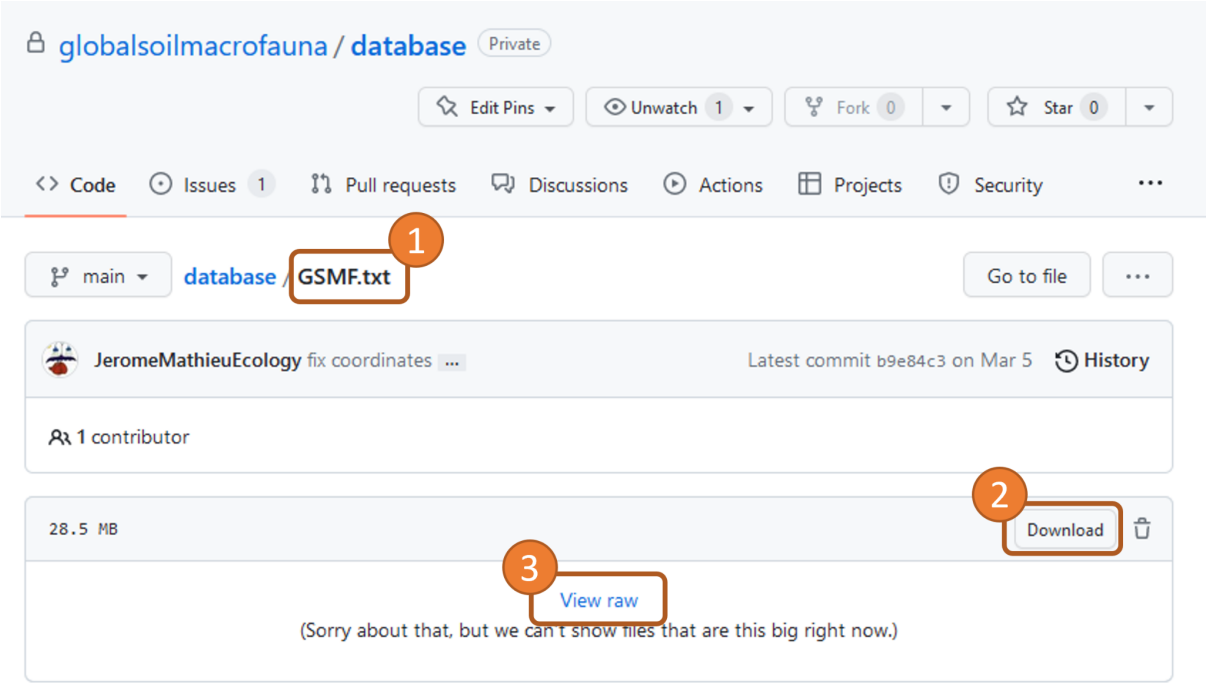


Figure 3 *Webpage of file with the download and view raw buttons. This last button might change depending on the nature of file*

1) Name of the file
2) Dowload Button
3) Link to the raw file

2.4 Explore the history of changes in a repo

Due to the versioning system, you can track the whole history of a repository. You can look at file by file, commit by commit, and so on. It can be a bit tricky to navigate in the history of a repo, and get confused with the old and updated version of the repo, so be careful.

See the past versions of a file

Once again, the way to proceed will depend on the nature of the file. But generally speaking,

- you first need to click on the name of the file (3 in Fig. 2) to show the page of the file (Figure 4).
- Then click 'history', on the page of the file. (1, Figure 4)

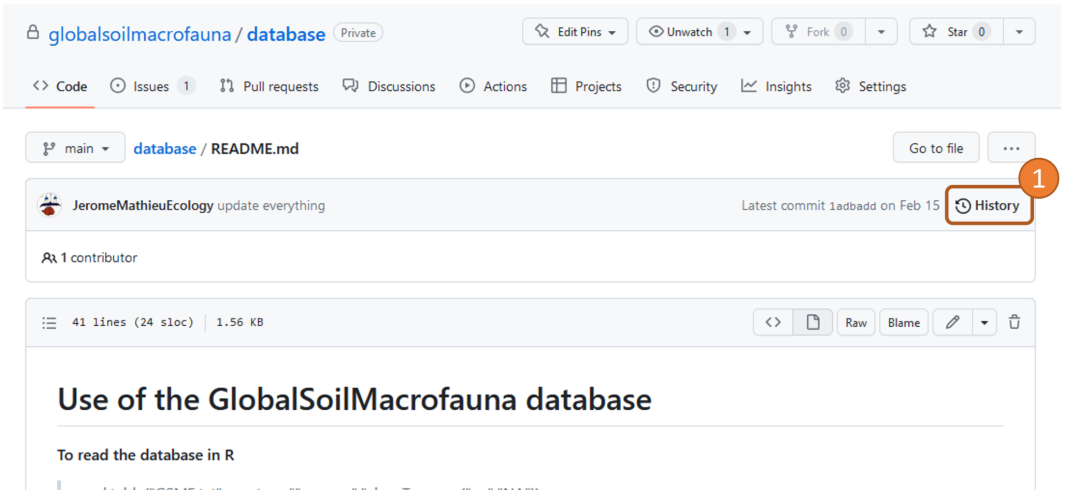


Figure 4 *Button to get access to the history of a file, with the different versions*

A new page will open with the different versions of the file and the correspond commits.

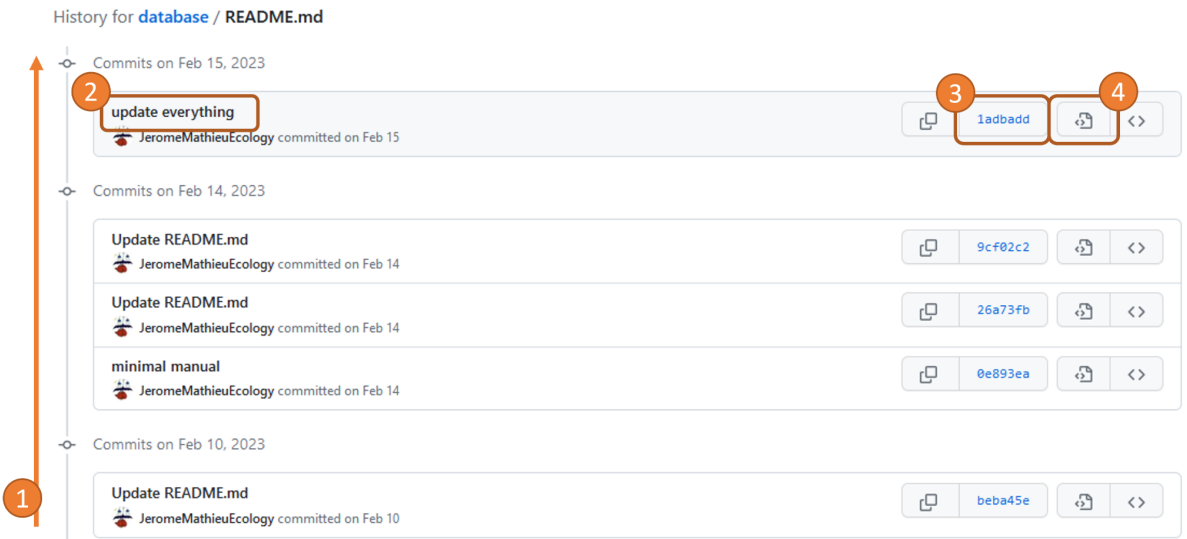


Figure 5 *page of the history of a file*

(1) Updates (commits) are listed by date

(2) Name of the last update of the file
(3) ID of the commit, gives access to the details of the changes of the file since the version before (see below)
(4) Gives access to the download page of the version concerned. If you click on it, you will have a download page with the same options as usual.

So to download a former version of a file, you need to click on button 4 in Fig. 5.

See the detailed changes of files ('diff')

When you click on the commit ID on the history page of a file (button 3 Fig.5), you will have access to the details of the change of the file since the last version, and also of all files that were changed during the same update (same commit, Fig. 6).

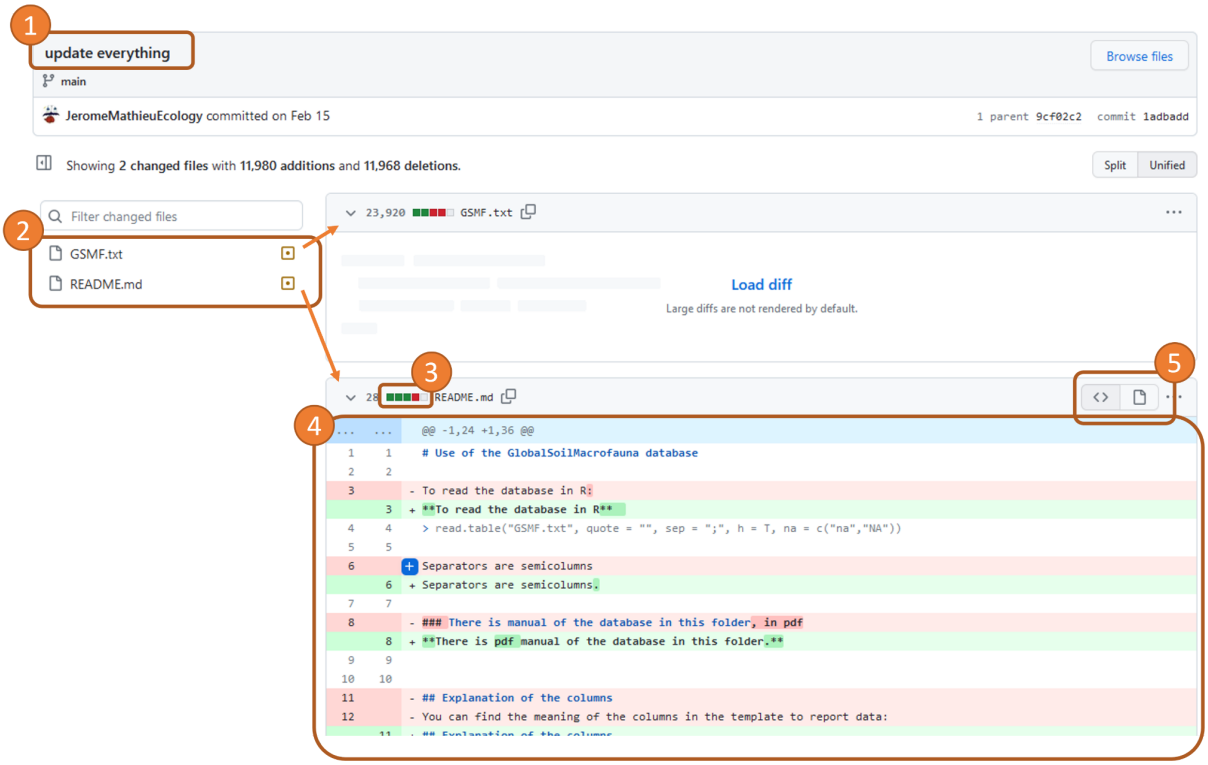


Figure 6 Webpage displaying the difference in the files since the last commit (last update)

1) Name of the commit
2) Files that were updated during this commit. The details of each file re given on the right of 2) (points 3-5)
3) Indicators of the type of change since last commit: green amount of new lines of code, red: amount of deletion of lines
4) Detailed changes: in red: deletion, in green: addition of text
5) Switch between raw display (as on the figure) and between formatted display, when available (it depends on the type of file).

Explore the history of changes of a repo (‘commits’)

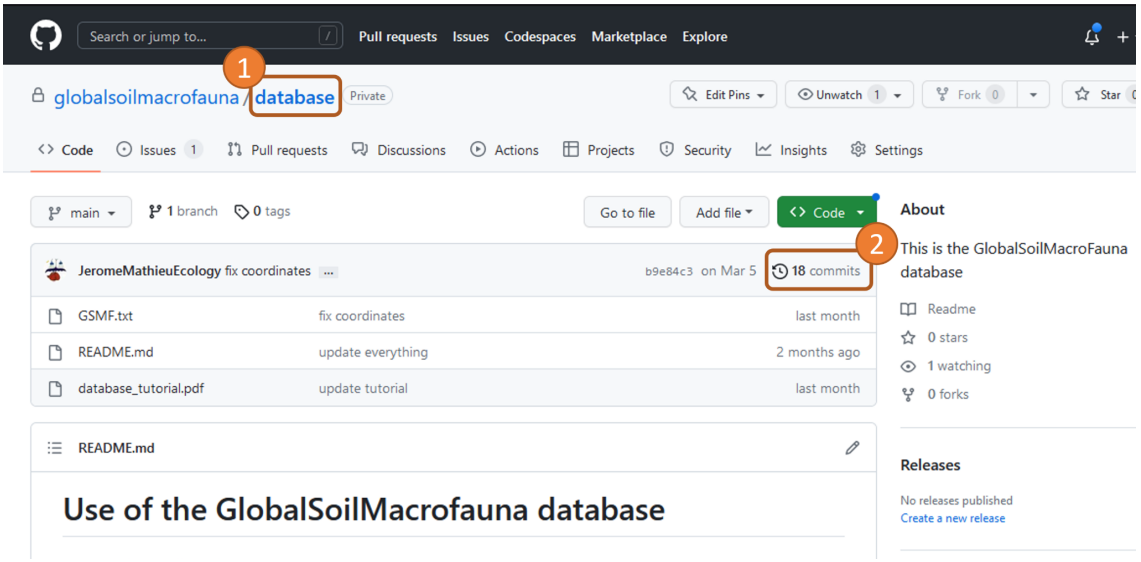


Figure 7 Button to get access to the history of a repository

1) Name of the repository.
2) Button to open the history page of the repo.

To get access to the history of a repo, go to the webpage of the repo and then click on button 2 fig. 7. A new page will open with the list of commits that happened (Fig. 8)

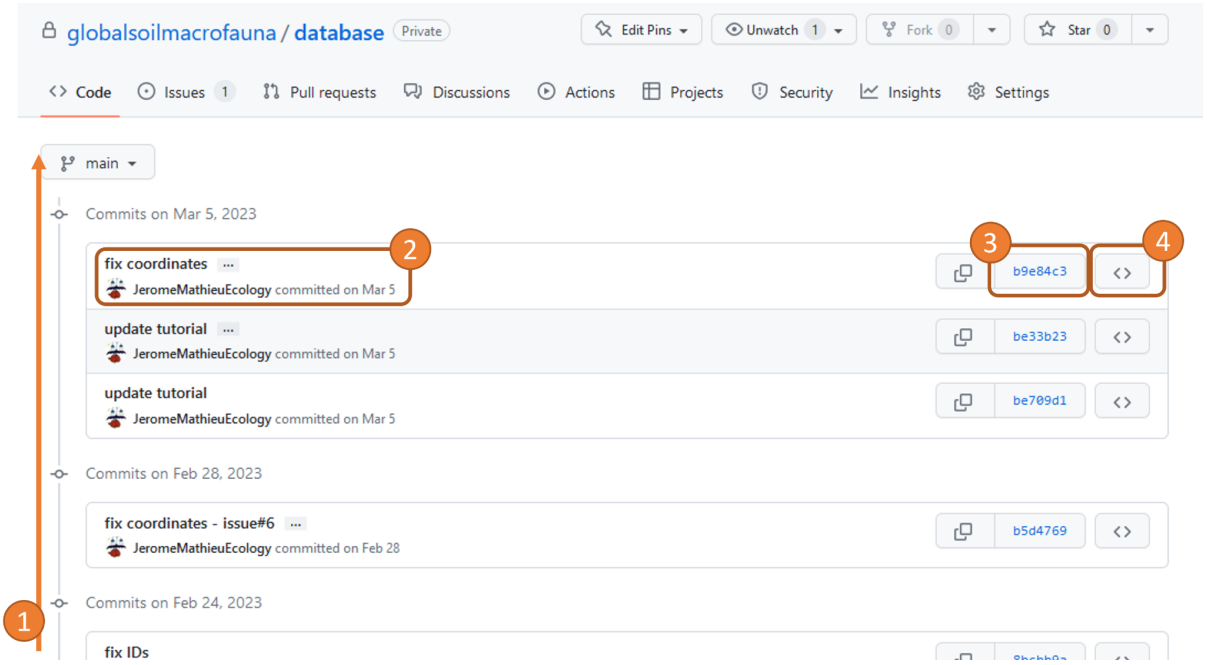


Figure 8: Webpage of the history of a repository

1) Commits are listed by date
2) Name of a specific commit
3) ID of the commit. Gives access to all the detailed differences between this version and the version before (like in Fig.6)
4) Button that gives access to the complete repository in state at the date of the commit (see next section)

See the different versions of a repository

Click on button 4 Fig 8, you will see the whole repository in its state at a given point of history.

Warning

the interface will look exactly the same than the updated repo (that can be confusing!), but you are indeed browsing the repository as it was at that point in history (Fig. 9).

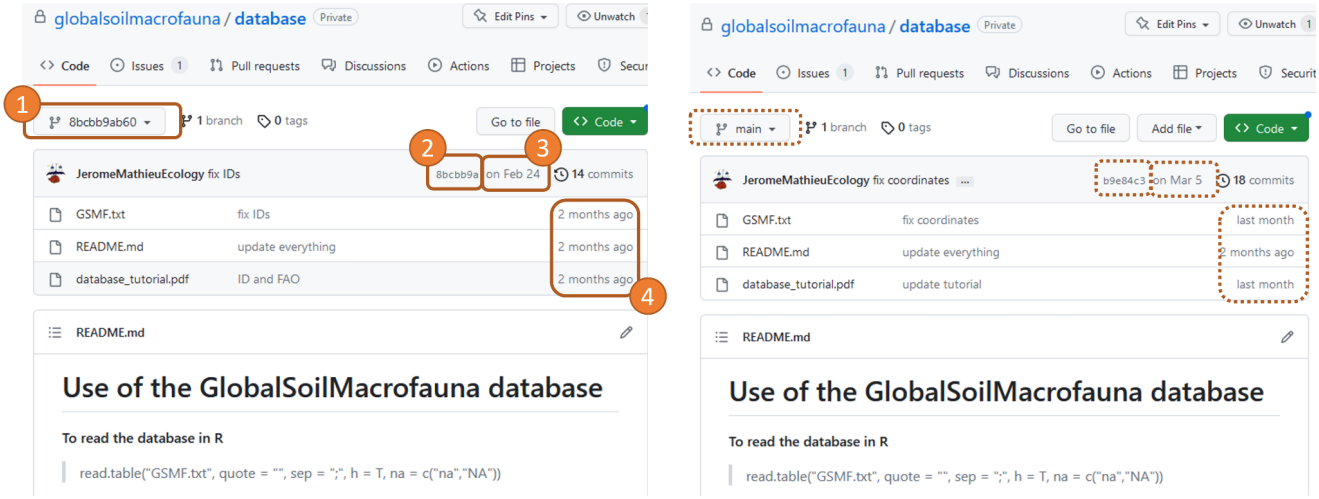


Figure 9 Comparison of a former state of the repo (on the left) and the present state (on the right). The versions can be differentiated from the names and date of the commits:

1) ID of the branch (= commit)
2) ID of the commit
3) Date of the commit of the repo
4) Date of the last commit of the file

3 - Manage GitHub files from your computer

Now that you are familiar with the GitHub web interface, let's see how to access files from your computer, and upload files (if you are admin). Recall that this is the recommended way to use GitHub.

3.1 Applications to interact with GitHub

First, you need to install Git on your computer

You can use Git with
R studio : https://r-pkgs.org/git.html#git-setup
GitHub Desktop : https://desktop.github.com/

(other resources are available, including GitHub for Atom https://github.atom.io/)

GitHub with R Studio

The documentation is already well done elsewhere, so we just provide the links

> **Create your own repository**
> https://r-pkgs.org/git.html#git-init
>
> **Do a local modification in the repo: "commit"**
> https://r-pkgs.org/git.html#git-status
>
> **Send your modification on GitHub: "push"**
> https://r-pkgs.org/git.html#github-init

GitHub with GitHub Desktop

Once you installed GitHub desktop, you need to open the application and make the link to your account: *File>Options> Accounts*.

At this point, your GitHub space in the App should be empty.

You can either download repositories that you already own, that are open, or that are shared with you. Another option is to create from scratch a repository (see next section).

The rest of the tutorial will detail the functioning of this application.

3.2 Get a GitHub repository

We will see how to import an existing repository and how to create one from scratch

Import a repo from GitHub.com to your computer

Downloading an existing repository to your computer is called “cloning”. To do so, go to *File>Clone repository*

You can download repositories from your GitHub account or from other people, provided they are open or shared with you. The repos that you created before or that are shared with you will appear in 3 fig. 10, when you click on Button 1. To clone open repos, you need to get their URL and click on Button 3.

Be extra careful with the location of the folder where the repository will be downloaded. This folder **should not** be placed in a synchronized folder such as a drobpox or google drive folder.

If you own the repository or you are an admin of the repository, your copy will be considered as the **main branch**.
If you are not admin of the repository, your copy will be considered as a **branch**. If you make a modification to your copy, your branch will become a ‘**fork’**, which means a parallel version of the main branch (the official repository).

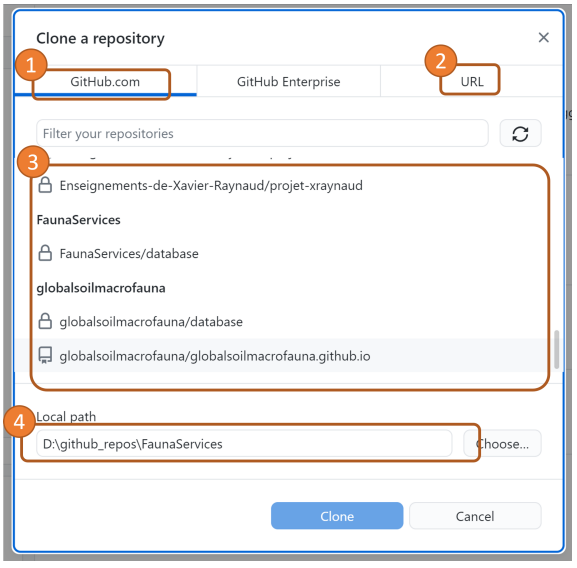


Figure 10 Cloning a repository to your computer

go to *File>Clone repository then:*

1) Access to your repositories, or the shared ones
2) Access to open repositories (you need the URL)
3) List of repos in your GitHub account
4) Folder where the repository will be saved on your computer

When you hit clone you will have on your computer a copy of the online repository.

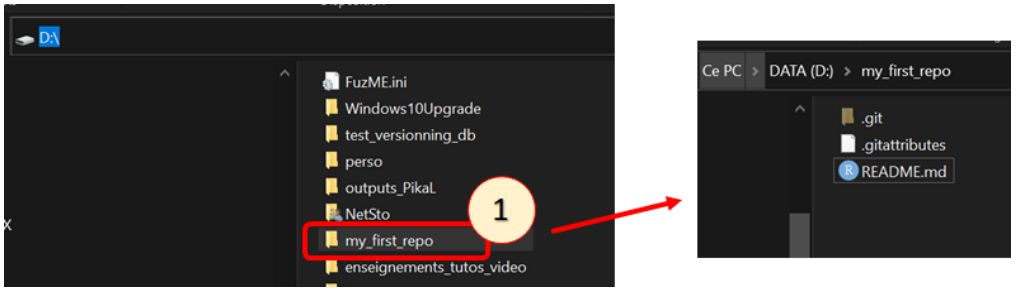
Create a repository on your computer and upload it to GitHub

1° Hit *File>New repository*

2° Fill in the *Dialog box*
Name = name of the folder *(do not put it in a synchronized folder such as a dropbox folder)*
Path = root of the folder
Initialize this repository with a README creates a readme file (recommended)

3° *You should then have a new repo* in your computer

In your PC :



This repo should appear in GitHub Desktop(1)

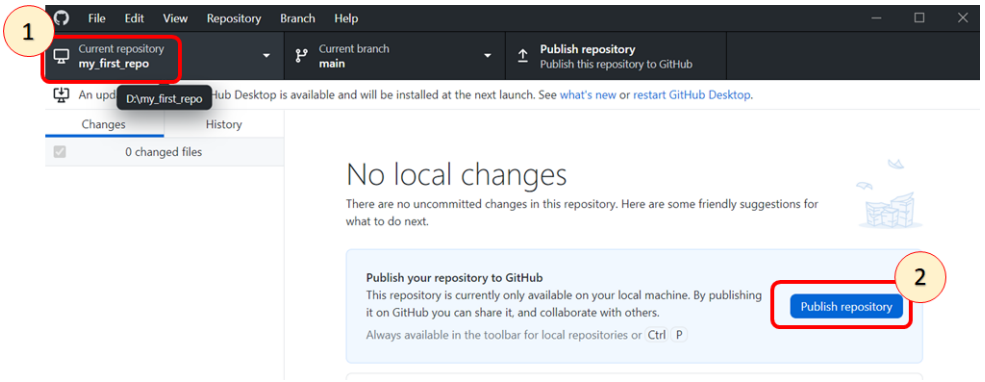


Figure 11 Publishing a new repository on GitHub.com

4) *To Publish your repo online*, hit 'Publish repository' (2) → You're done !

3.3 Interact with GitHub to update the local or the online version of a repository

Before diving into the process, let's clarify the GitHub terminology for the downloading and uploading processes. Recall that compared to automatic online folders, there is an extra step to validate updates, either during uploading and downloading.

Downloading files from the cloud to your computer (3.4)
There are two steps(see the fig. 12):

1. **Fetch changes**: Identify the modifications in the remote folders that are not so far updated on your computer.
2. **Pull changes**: accept and apply these modifications to your local files (on your computer).

Uploading files from your computer to the cloud (3.5)
There are also two steps (see the figure):

1. **Commit changes**: list the modifications that you made in your local folder since the last update.
2. **Push changes**: send these modifications to the online folder to update it.

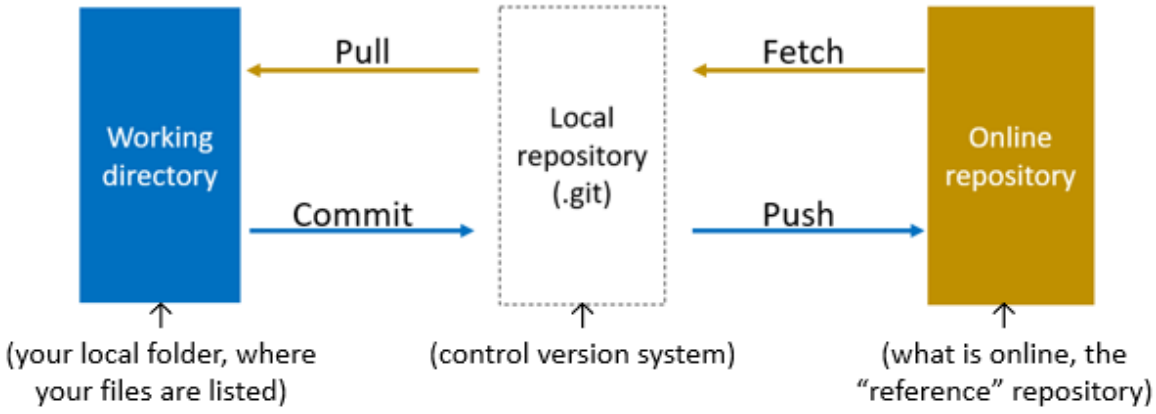


Figure 12 The different steps to download or upload updates in GitHub

3.4 Update your online repository from your local repository

There are two steps

1° Accept last modifications in your local versioning system: 'commit'

When you modify a GitHub folder in your computer, the gitHub application will automatically detect these changes and propose you to manage these changes: accept (='commit") or reject.

In the application, If you click on a file, the changes in the file since last commit (the "diffs") will be displayed on the main panel (2)

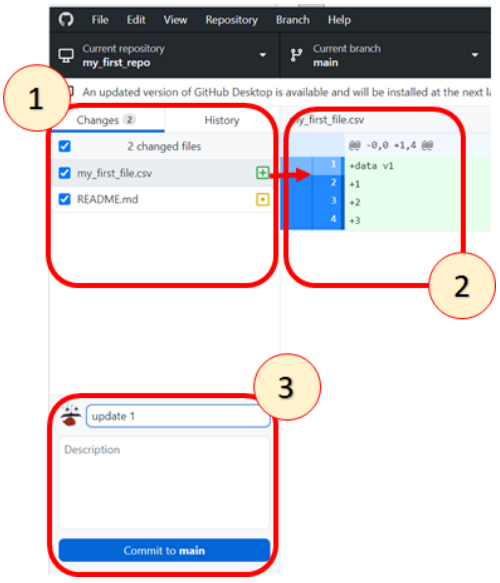


Figure 13 Presentation of the changes in files and folder since the last commit

To accept these change into the versioning system of your computer (nothing will happen on the online repository), you need to

- give a name, and optionally a description - to the update “commit” this new version to the Git System (3)
- Hit the ‘**Commit to main**’ button (3).

At this stage no interactions were done with the online folder. Updates were only done within your local versioning system. The next step is to synchronize the online folder with your local folder.

2° Send your modifications on GitHub.com : “push”

Once you accept modifications in your local Git system, you may want to update the online version of your folder accordingly to these changes.

To send the modifications to GitHub.com, click “**Push Origin”**
This should send the modifications to the online folder.
If there is any conflict between the local and online versions, a pop up will open, asking you how to deal with these conflicts.

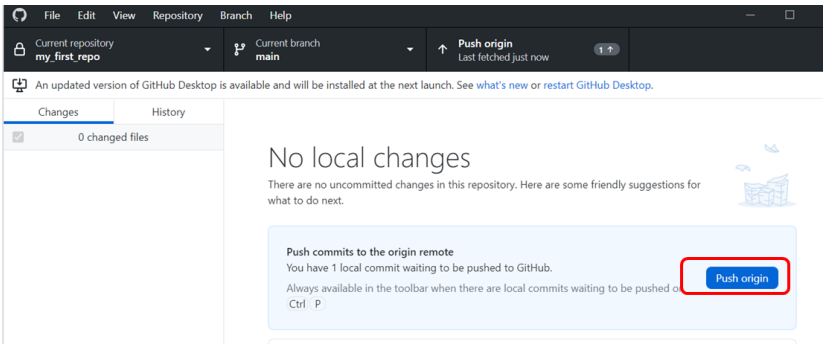


Figure 14 Pushing commits to GitHub.com

3.5 Update your local repository from the online GitHub

Here you want to update your local copy of a repository from the online repository.

There are two steps:

1° ‘Fetch’: Check for existing updates of the online repository

In this step you check all modifications that were done one the repository that were not applied to your local copy so far.
First select the repository in GitHub desktop (1 Fig below)
The synchronizing wheel should roll (2 in Fig. below), with the message ‘Fetching origin’

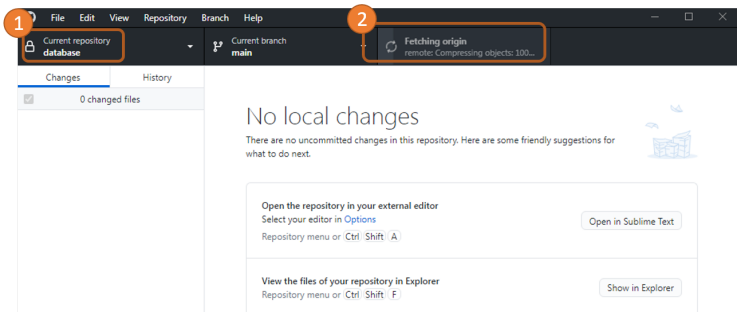


Figure 15 Fetching updates for GitHub.com

2° ‘Pull’ : Accept this modifications and apply them to your local copy of the repository

Once the synchronization is done, the synchronization wheel has gone and the main panel of the application is updated:

1) Number of commits that were applied on the online folder since the last time you updated your local copy of the folder
2) You can browse in details these changes with different tools
3) If you are OK with the changes, hit ‘Pull Origin’. This will update your local folder.

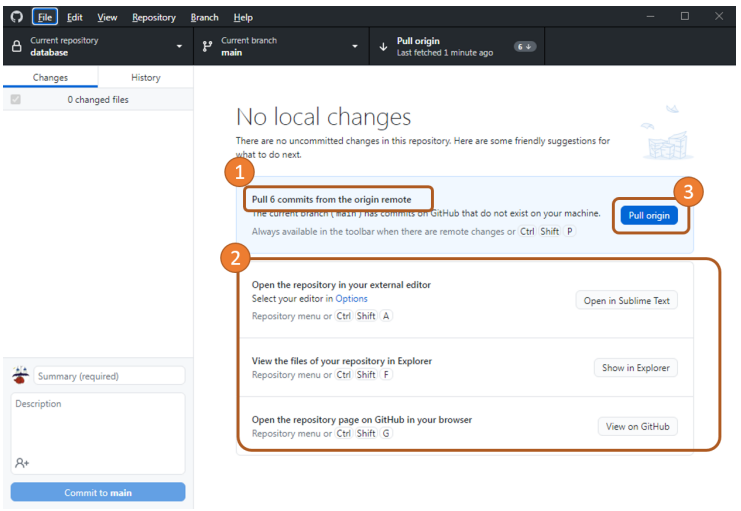


Figure 16 Pulling changes from gitHub.com to your computer

3.6 Remove / delete a repository

Go to Repository/remove

You can just remove the folder from the git system (the folder stays in your pc), or you can completely delete the folder by checking “delete the folder”.

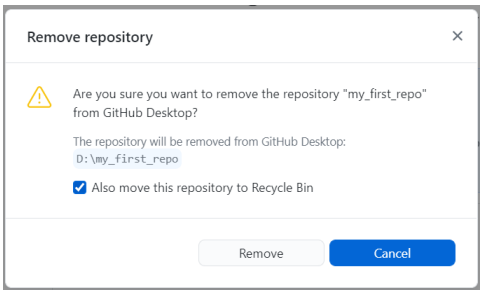


Figure 17 Deleting a repository in your computer

4 - Work with others

Several tools were developed to allow different people to work on the same repository.

Recall that only admins can modify the main branch. Others can only clone, fork and ask to inject their modifications to the main branch (‘**pull request**’).

Here will not cover pull requests, but focus on tools to interact with the admins and other users of the repo. In particular there are two interesting tools.

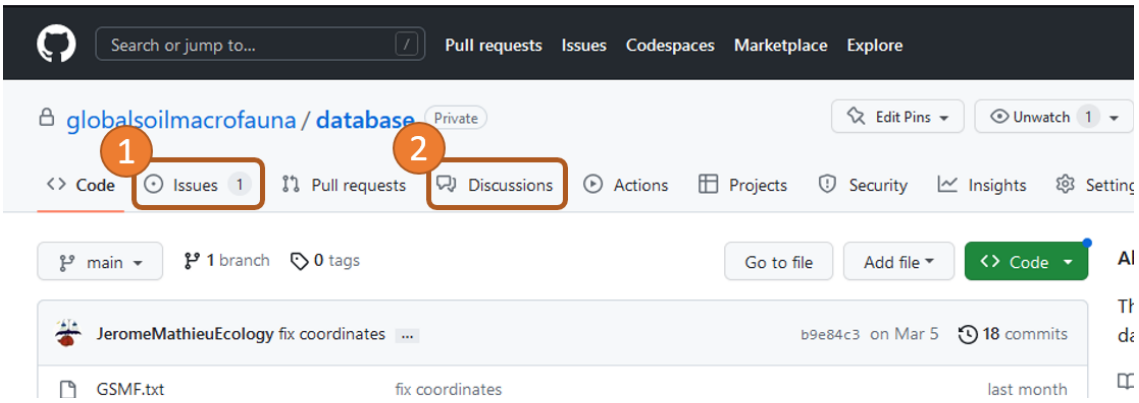


Figure 18 ‘Issues’ and ‘Discussions’ Buttons

(1) Button to open an issue
(2) Read the discussion threads

4.1 Report a bug in the repo: open an ‘issue’

This is the case when something is not working like it should. In other words, the system is not behaviouring like expected. This requires a modification of the code or data.
If you ask an issue, make sure that the problem is clearly stated. If possible make a reproducible example with lines of codes. Use rich formatting to improve the display.

Feature requests can be asked here too.

But this is not a place to ask questions about the functioning of the system. This goes to the discussion section.

Open issues ask an action on the repo from the admins.

4.2 Ask questions about the repository: ‘Discussion’

This is the place to ask questions about the way to use the repo, and get feedback from others. It is a good place to share experience, synchronize with other users. It doesn’t imply any action on the repo.

4.3 Follow a repository: Get notified when a new commit or any activity is done

In GitHub, following a repository is called '**watching**'.
To watch a repo, go to its page, and hit the watch button (1) Fig. 19.

This is very useful to be advised when the repository is updated, or when an issue is opened.

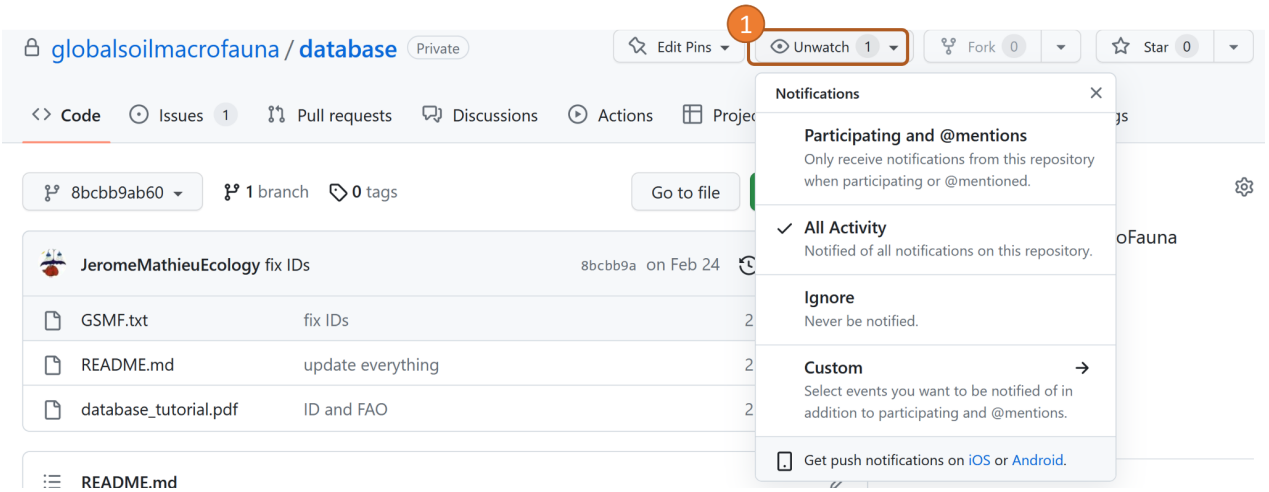


Figure 19 Watching a folder to track changes

Watching a repository allows you to be updated on any activity in a repository

4.3 See the branching history of a repo

It can be useful to track contributions to a repo,in particular in complex repositories.

From the home page of a project, click on *Insight>Network* and see the different branches for the same project.

Example: https://github.com/JeromeMathieuEcology/club-iEES/network

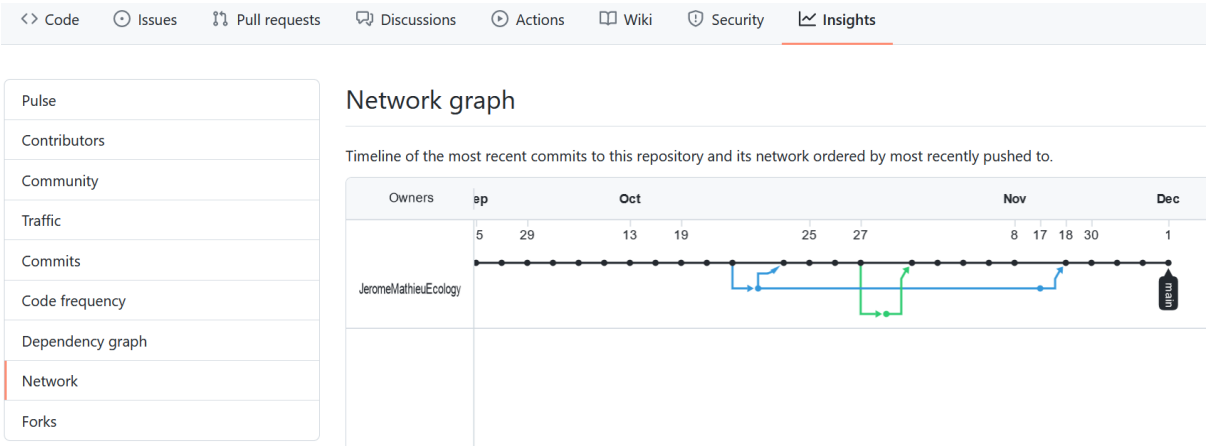


Figure 20 Map of branches of a repository

5. Resources for an in depth understanding of Git :

https://gist.github.com/peterhurford/4d43aa5d6de114c0c741ba664c9c5ff5

https://docs.github.com/en

glossary : https://docs.github.com/en/get-started/quickstart/github-glossary