Source Control

Installation

Practical 1

Practical 2

Branches

Fork and Pull Request

Write Documents

Source control with git and Github

Eric Marcon

13 October 2024

Source control with git and Github

Eric Marcon

Source Control

Installation

Practical 1

Practical 2

Branches

Fork and Pull Request

Write Documents

Source Control

git

git is a source manager:

- Track changes: much more than a backup!
- Ability to go back in time;
- Several competing versions: branches.

GitHub

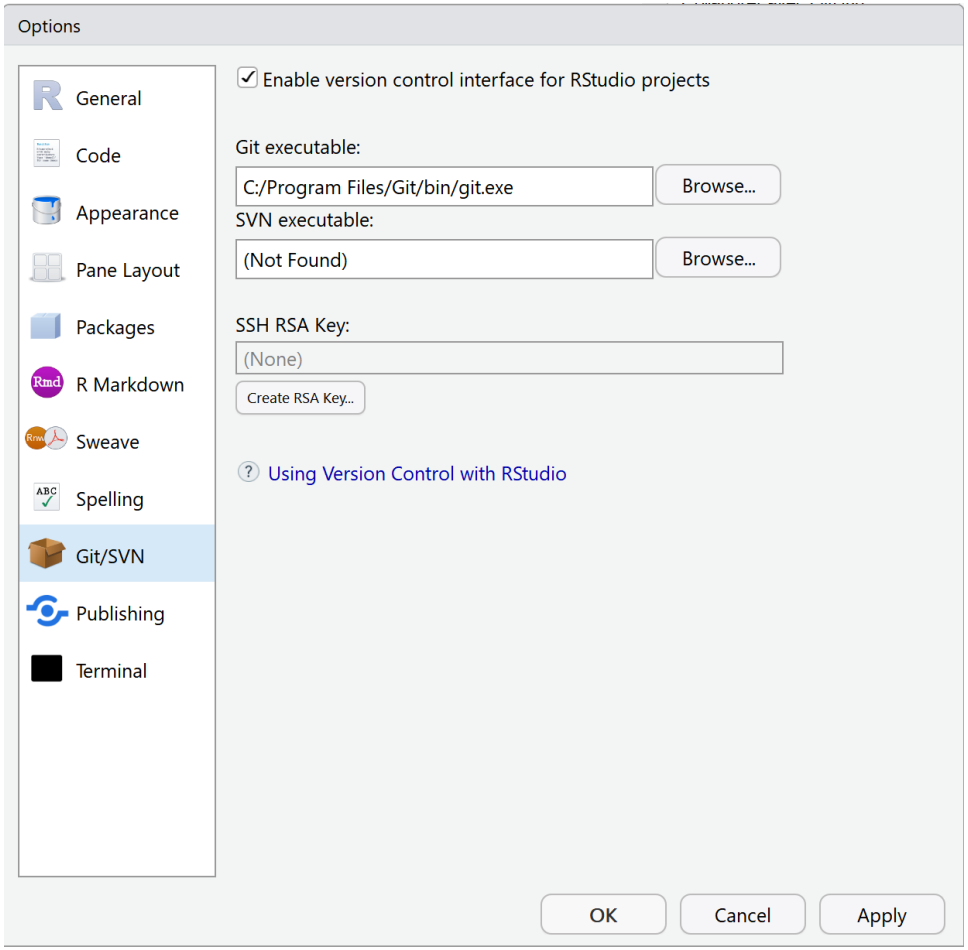
GitHub is a collaboration platform based on Git.

- Multiple developers: the end of attached files!
- Presentation of results: GitHub pages;
- Continuous integration: GitHub Actions.

Installation

git: install

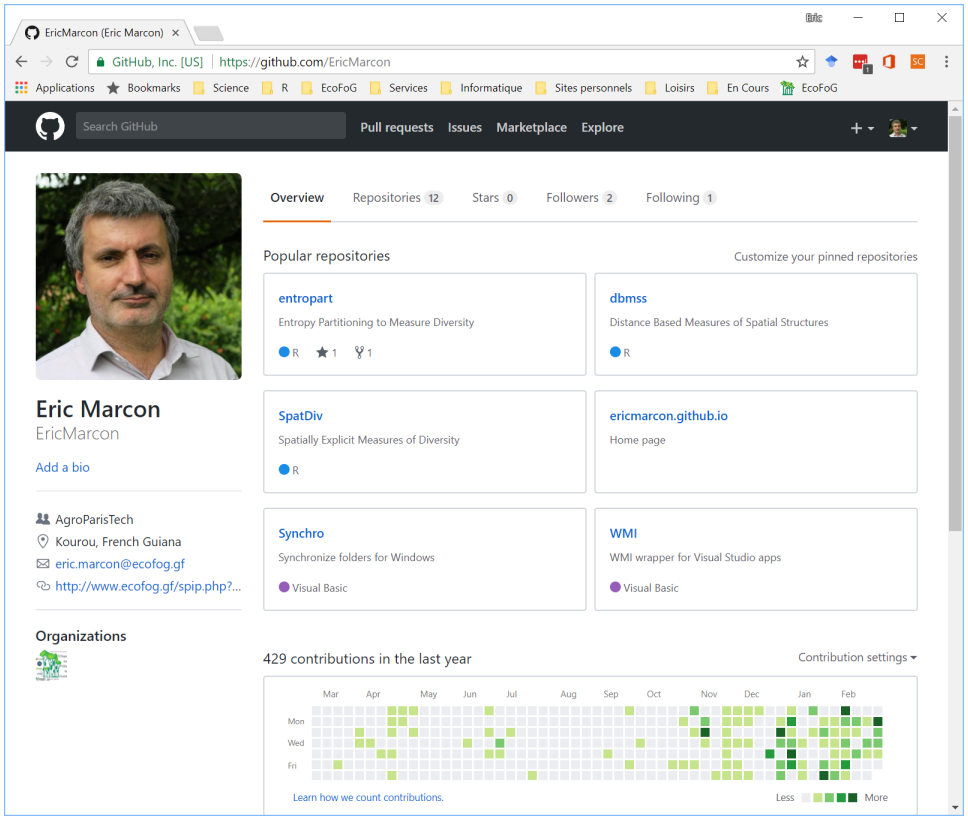
RStudio must detect Git



Otherwise, install it.

GitHub

Open an account on GitHub



GitHub security: SSH

Your computer must be able to write to git. We choose **SSH authentication** for simplicity.

- In RStudio, Tools / Global Options/ Git-SVN, create a new SSH key (no passphrase).
- Click on *View public key*. Copy the key.
- On GitHub, your accounts, Settings, SSH and GPG keys, Add New SSH Key, paste the key. Name it after your computer.

GitHub security: PAT

Create a GitHub token for Continuous Integration:

- User account settings;
- Developer Settings > Personal Access Tokens;
- Generate a token, describe it as "git-RStudio" and give it "repo" authorization.

Save the token!

Source Control

Installation

Practical 1

Practical 2

Branches

Fork and Pull Request

Write Documents

Practical 1

Create a project

From scratch:

- In GitHub:
  - *New Repository*
  - Choose name (no special characters)
  - Do not add anything else
- Copy the URL from *Clone or Download*.
- In RStudio: New project / Version Control / Git, paste the URL.

Create a project

Source control with git and Github

Eric Marcon

Source Control

Installation

Practical 1

Practical 2

Branches

Fork and Pull Request

Write Documents

From an existing RStudio project:

- Put the project under version control:
  - *Tools /Version Control /Project Setup...*
  - Select *Git*.
- Create a repository on GitHub and retrieve its URL: git@github.com:MyAccount/MyRepo.git
- In RStudio Terminal, run :

```
git remote add origin git@github.com:MyAccount/MyRepo.git
git branch -M master
git push -u origin master
```

Basic operations: filter

Modified files are displayed in the RStudio *git* window.

Use .gitignore to hide non-tracked files.

Basic operations: commit

- Create an R file with one line of code and save it.

After each work session, *Commit* the result.

Select files to commit.

Enter a clear message: summary on the first line.

Basic operations: synchronize

Push modifications to GitHub to make them public.

Pull to retrieve changes from GitHub.

Basic operations: collaborate

Declare a collaborator.

Work with two or more people on the same file.

Content of a work session:

- Pull,
- Modify,
- Commit,
- Push.

Conflicts

The elementary data is the line.

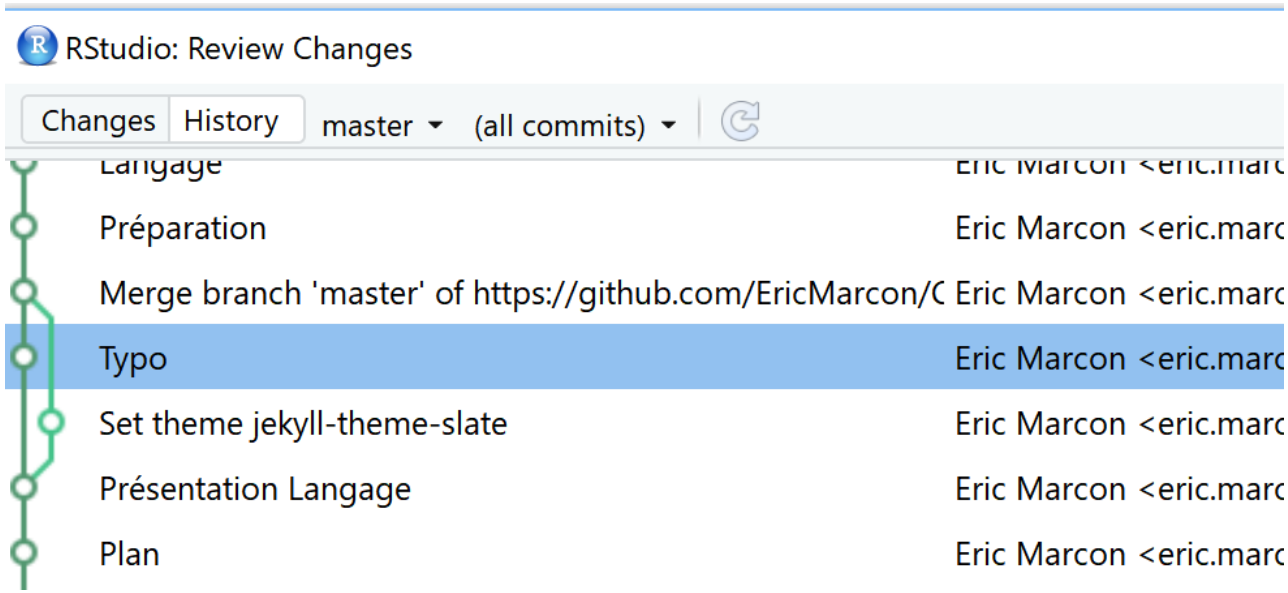
Conflicting changes imply a conflict.

Minimize conflicts: in a text, one sentence = one line.

In case of conflict, decide which code or text to keep.

Project tracking

Clock icon in Git window



Source Control

Installation

Practical 1

Practical 2

Branches

Fork and Pull Request

Write Documents

Practical 2

Generate a conflict

Edit the same line of README.md :

- online on GitHub,
- locally.

Commit, Pull, observe the conflict, solve it.

Collaborate with your neighbor

The neighbor on the left invites the neighbor on the right to GitHub.

Both modify the project.

- Remember to pull before modifying.
- Push quickly to limit conflicts.

Source Control

Installation

Practical 1

Practical 2

Branches

Fork and Pull Request

Write Documents

Branches

Purpose

Modify the project without disturbing its stable state.

Application :

- develop new functionality,
- test it, correct bugs,
- make it visible when finished.

Create a branch

Source control with git and Github

Eric Marcon

Source Control

Installation

Practical 1

Practical 2

Branches

Fork and Pull Request

Write Documents

Click on the New Branch button.

Type its name.

Work in the new branch (pull, commit, push).

Merge

Source control with git and Github

Eric Marcon

Source Control

Installation

Practical 1

Practical 2

Branches

Fork and Pull Request

Write Documents

Move to master branch.

Run:

```
git merge branch_to_merge
```

Source Control

Installation

Practical 1

Practical 2

Branches

Fork and Pull Request

Write Documents

Fork and Pull Request

Fork

Source control with git and Github

Eric Marcon

Source Control

Installation

Practical 1

Practical 2

Branches

Fork and Pull Request

Write Documents

Goal: modify someone else's repository

Fork: create a copy of the repository

Start a new branch, modify it.

Pull Request

Source control with git and Github

Eric Marcon

Source Control

Installation

Practical 1

Practical 2

Branches

Fork and Pull Request

Write Documents

Branch integration request: *Pull Request*.

On GitHub.

Dialogue possible.

If accepted, merge branch and delete.

Source Control

Installation

Practical 1

Practical 2

Branches

Fork and Pull Request

Write Documents

Write Documents

Use memoiR

Install the **memoiR** package.

Create a new document with memoiR: Stylish Document.

Knit it to check everything is OK.

Source Control

Source control with git and Github

Eric Marcon

Source Control

Installation

Practical 1

Practical 2

Branches

Fork and Pull Request

Write Documents

Use git with your document.

Create an appropriate `gitignore`:

```
memoiR::build_gitignore()
```

Store the knitted files into `/docs`

```
memoiR::build_githubpages()
```

Make your first commit.

Upload to GitHub

Source control with git and Github

Eric Marcon

Source Control

Installation

Practical 1

Practical 2

Branches

Fork and Pull Request

Write Documents

Create your repo on GitHub.

Copy the commands to upload an existing git repo:

```
git remote add origin git@github.com:MyAccount/MyRepo.git
git branch -M master
git push -u origin master
```

Declare repo in header

In the header of your Rmd file, update the repository name and the URL.

Commit.

Add a README:

```
memoiR::build_readme()
```

Commit.

Continuous integration

On GitHub, your repo, Settings / Secrets and variables / Action :

- Add two secrets:
  - EMAIL: your email address
  - GH_PAT: your personal access token

In RStudio, run

```
memoiR::build_ghworkflow()
```

Commit, Push and look at Actions.