A short introduction to Git (not GIT)

Pieter Belmans

August 8 2023

Goal

1. explain what Git is
2. explain how it's useful for writing papers
3. explain how it's useful for the Stacks project

From the website:

> *Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.*

Tools for collaborative paper writing

maybe you haven't collaborated yet, but in case you have: *what did you use?*

- emailing files back and forth
- Overleaf
- Dropbox

any others?

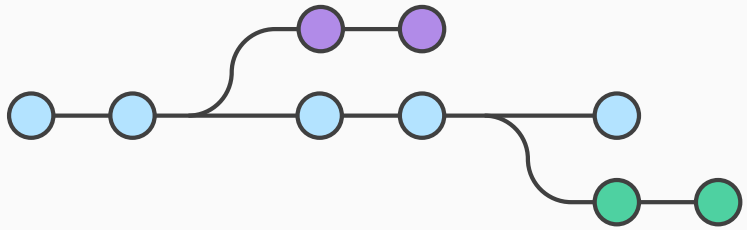
What do you (or I) not like about them?

- lack of conflict resolution
- clutter of temporary files (with Dropbox)
- only with internet access (for Overleaf)
- no real versioning: arXiv v1, submitted to journal, first revision, ...
  (unless you remember to save them at the time, or use Overleaf's basic history functionality)
- no diffs: when your coauthor edits things, there is no description of the changes, and no way to easily see what has changed

What is Git?

Think of it as a *much* better form of Dropbox (but it's not that at all to be honest):

1. never have `paper.tex`, `paperv2.tex`, `paperv3.tex`, `paperv3final.tex`, `paperv3submitted.tex`, ... (and the temporary files)
2. description of changes
3. good handling of conflicts: merge if possible, conflict markers if not
4. decentralised (so without internet access)
5. ...



```
git diff
X$ is a regular
a squence of mo
a sequence of m
```

it has no interactive LaTeX editing like Overleaf though, you can use

Using Git for writing papers

Git is a complicated beast, and it's a commandline tool... Use other tools!

GitHub a website for managing your projects (= repositories): collaboration, backup

GitHub Desktop a graphical tool for interacting with your repositories

Alternatives:

1. BitBucket, GitLab, ...
2. SourceTree, SmartGit, ...

   It's the best thing since sliced bread!

   —Anonymous co-author of mine.

Learning Git

1. a 15 minute (interactive!) commandline tutorial:
   `https://try.github.io`
2. a 10 minute read on using GitHub, including pull requests:
   `https://guides.github.com/activities/hello-world/`
3. ...

it's really best learned by doing

An extremely short showcase

How to start collaborating on a new paper: I need

- a volunteer with a GitHub account
- a title for the paper

What I will do:

1. create the project
2. push to GitHub
3. add a collaborator
4. start editing

What the volunteer needs to do

1. clone the project
2. start editing

Some tips

- use it also for personal projects
- consider using short lines: better diffs, and better insight into sentence structure
- commit often: atomic commits are best
  you can do lots of work, then do commits of chunks, and then push
- if you program something: include it, even if it's just a few lines of code

More resources

- `https://idrissi.eu/post/git-1-preliminaries`, `https://idrissi.eu/post/git-2-theory`, `https://idrissi.eu/post/git-3-practice`
- `https://www.math.cmu.edu/~gautam/sj/blog/20130929-git-quickstart.html`
- if you want to become an expert: `https://git-scm.com/book/en/v2` (this is like reading Hartshorne when you want to solve a quadratic equation)

fun thing to do: use GitHub Actions to build pdf's upon each push

after this tutorial: ask me anything!

The Stacks project

Interacting with the Stacks project

This is not required.

I repeat, this is not required. You can (depending on the type of change)

0. scan handwritten suggestions, and email these
1. make comments on the website
2. download the TEX files, make changes, and email these to
3. use pull requests on GitHub

Interacting with the Stacks project (without Git)

For options 2 and 3:

`https://github.com/stacks/stacks-project`

The Stacks project has big files, making some editors crash, so be careful

The Stacks project uses a strict coding style:

`http://stacks.math.columbia.edu/download/coding.pdf`

Interacting with the Stacks project (with Git)

Via pull requests: see
`https://help.github.com/articles/about-pull-requests/`

fork make your own copy

clone use this copy on your computer

commit make edits, then commit

push make them public

pull request suggest that your changes are pulled into the Stacks project

`https://github.com/stacks/stacks-project/pulls`

word of advice: use small commits (allows cherry-picking)

Social event

- ask Git questions
- make Stacks project suggestions:
  - website
  - mathematics
- ask mathematics questions
- eat the food and drink the drinks