Linking GitHub, Overleaf, and a local directory

June 30, 2020

1 Goals

If you'd like to be able to work collaboratively on your manuscript, keep a record of changes that have been made to your manuscript, and work on writing both offline and online, use these instructions to help you work easily on the same version of your document(s) between Overleaf, GitHub, and a LaTeX distribution on your computer.

2 Requirements

You will need:

- A GitHub account and command line git tools
- An Overleaf account (free from MIT)
- A LaTeX distribution on your computer (I use TeXShop, but there are many)

3 Setup

After following these instructions, you will have a directory on your computer, GitHub repository, and Overleaf project that all contain the same files.

3.1 Starting from a directory on your computer

If you already have a folder with your .tex file on your computer, you need to link it to a repository on GitHub.

1. Create a new repository on GitHub (click the green button on the Repositories tab on your account)
2. In terminal, navigate to the directory on your computer where your .tex file is stored

3.
   ```
   git init
   ```
   to initialize this directory as a git repo (not linked to your GitHub yet)

4.
   ```
   git add .
   ```
   to add all of the files from the directory to be tracked. See below for how to set up a .gitignore to skip adding some files.

5.
   ```
   git commit -m "[commit message]"
   ```
   commit the files

6.
   ```
   git remote add origin https://github.com/JuliaWilcots/overleaf-github-example.git
   ```
   link to your GitHub repo that you created in step 1 (change my name and repo name to yours)

7.
   ```
   git push -u origin master
   ```
   push the files you added to GitHub
8. Check your GitHub (online) to make sure you see the files there.
9. In Overleaf, click "New Project" > "Import from GitHub" > click the repository you want to add.

10. You are good to go!

(* is a regular expression that means "any number of characters, any characters", so "*.aux" tells git to ignore any files with the .aux extension.)

3.2 Starting from a directory created on Overleaf

1. On Overleaf, in your project folder, go to "Menu" > "GitHub" > "Create a GitHub Repository" > Follow the instructions
2. In your newly created GitHub repo, click "Clone" and copy the link to your repo to your clipboard
3. In terminal, navigate to the parent directory you want to keep your manuscript folder (the new github repo) in.
4. then type:
   ```
   git clone https://github.com/JuliaWilcots/new-repo.git
   ```
   (replace "new-repo" with your repo name)
5. You should be all set!

3.3 Starting from a repository on GitHub

Choose one of the two above options and work in one direction first (e.g. add a "New Project" on Overleaf) and then go in the other direction.

3.4 Adding a .gitignore

When you typeset a latex doc on your computer, a bunch of accessory files are created. These have extensions like .aux, .gz, .log. You do *not* need to push these to your git repository, so you will want to tell git to **ignore** them. To do this add a file called ".gitignore" to your repository on your computer. To do this:

1. type `vim .gitignore` into terminal while you are in the right parent directory
2. type `i` (to insert into the file you just created)
3. `*.aux` (press enter)
4. `*.gz` (press enter)
5. `*.log` (press enter)
6. add any other file names that are in your local directory that you don't want on GitHub/Overleaf
7. press 'esc' and then type `:wq` and press enter to exit.

(* is a regular expression that means "any number of characters, any characters", so "*.aux" tells git to ignore any files with the .aux extension.)

4 Workflow

4.1 Computer to GitHub to Overleaf

If you have been editing on your local machine (this is great if you have a poor internet connection, if the Overleaf servers are not responding, or if you like your TeX distribution more than Overleaf), follow these instructions to update your GitHub repo and Overleaf folders **after every time to edit your manuscript**!

1. In terminal, navigate to the (git) directory your manuscript is stored in.

2.
   ```
   git add .
   ```
   to add all files
   or
   ```
   git add filename.ext
   ```
   to add a specific file

3.
   ```
   git commit -m ”commit message”
   ```
   commit changes

4.
   ```
   git push origin master
   ```
   Push to GitHub

5. Your GitHub repo will now be up to date with local

6. in Overleaf, in your project, click “Menu” > “Github” > “Pull GitHub changes into Overleaf”

7. That’s it!

4.2 Overleaf to GitHub to Computer

Especially if you are collaborating with co-authors, make sure to push the edits you make in Overleaf to GitHub when you are done. Also, if you will want to work offline later, make sure to complete the process and pull your updated files from GitHub to your computer.

1. In your Overleaf project, go to “Menu” > “Github” > “Pull GitHub changes into Overleaf”

2. In terminal, navigate to your manuscript’s directory

3.
   ```
   git pull
   ```
   get edits from GitHub

4. Good to go

5 Notes

At any point, you can type `git status` to see how up-to-date your local directory is compared to your last commit to GitHub.

The setup I’ve described here only works if you are editing in **one** location at a time. You cannot easily make changes in Overleaf and on your local machine at the same time. (There are other ways of doing this, with multiple branches in GitHub, but I haven’t described those here.)