

How To Collaborate on a Manuscript Using GitHub

Geoffrey D Hannigan

Introduction

GitHub offers a lot of great resources for collaborating on scientific processes. This includes the process of writing and editing an manuscript. Although it's great, it is not standard practice across our fields, which means collaborators can be left in the dark on how to contribute. To make the process easier, I created this brief guide to downloading, editing, and submitting changes to project manuscripts.

Downloading the Repository

To get started, you are going to want to download the repository so that you can easily work with it. First you are going to want to **fork** the project repository, which mean copying it to your account. This way you can make your edits without directly impacting the master copy. To fork your repository, simply click the fork button at the top left side of the screen.

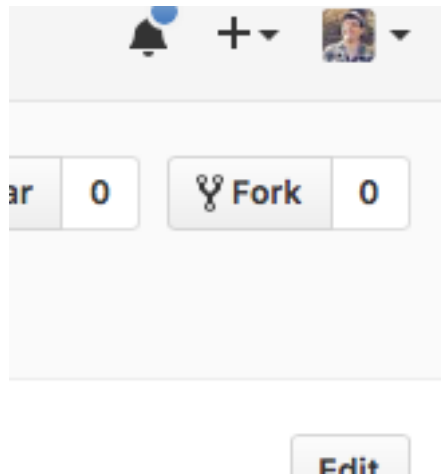


Figure 1: Click the fork button to copy the project repository to your account.

Once the fork is complete, you will have your own copy of the project that is associated with your account. To work with it, you still need to download it to your computer. To accomplish this, you can simply clone the repository to your directory. To do this, first click the **Clone or download** button to copy the required cloning link.

In this case, the link to clone the repository is:

```
https://github.com/Microbiology/HowToCollaborate.git
```

Now that we have the link, we can download the repository to our computer using the following command in the command line. *NOTE:* Be sure your current working directory (`pwd`) is where you want the project repo.

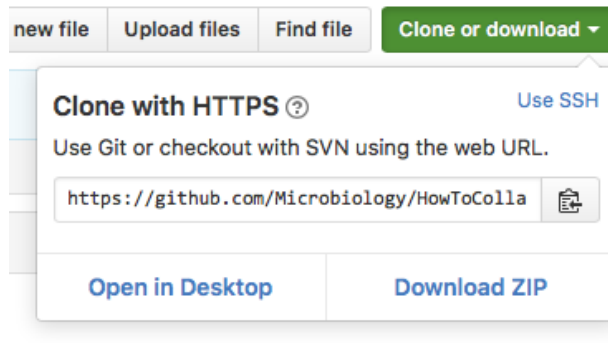


Figure 2: Click the green button, and then copy the https link.

The new project directory will be made in your current working directory. So make sure you are where you want the folder.

```
git clone https://github.com/Microbiology/HowToCollaborate.git
```

Editing the Manuscript

Now that the project repository is on your computer, you can move into the project directory and look at the contents.

```
# Move into the project directory
cd ./HowToCollaborate
# List the contents of the
ls -lth
# Move into the document directory
cd ./doc
```

In general, you will see a few directories, including a **bin** for source code and **data** for data files. You will also see a **doc** directory, which contains the document files. When we move into the document directory, we can look at the fully rendered manuscript as a PDF (**manuscript.pdf**). To edit the manuscript, we need to open the manuscript markdown file (**manuscript.md**) in our favorite text editor. For editing manuscripts, I suggest **Sublime Text** with the **Academic Markdown** package.

At this point you can update the text as you see fit. Be sure to use standard GitHub Markdown syntax (click here for a brief guide to using Markdown). After you have made some edits, you are probably going to want to see what the new manuscript looks like. To do this, you can re-render the manuscript in the command line. Luckily, the **./doc** directory contains a little bash script to make this easy. Just go back to your command line and type the following. *NOTE:* Here I am using pandoc directly because I like the control, but you could also use **Rmarkdown**, which has Pandoc built in with some other functions (and yes, I will be implementing some of that functionality in the future, but that is lower on my priority list right now).

```
./rendermanuscript
```

This script will use **Pandoc** to render the markdown file into a new PDF. This does require that you have Pandoc on your computer, so if you are getting errors, it is probably because you don't have it. You can easily download Pandoc on almost any system (click here for instructions). If you are on a Mac, I suggest the Homebrew option (click here for Homebrew instructions). Remember that you can always edit the manuscript without Pandoc, although you will only be able to edit the text without rendering a new PDF.

Submitting Manuscript Edits

At this point the edits have been made to the manuscript and you are ready to submit them to the group. You will need to commit your changes to the repository and submit a pull request to the admin of the master project folder. Luckily, this is also a pretty easy task.

To start, go ahead and commit/push your changes to the manuscript markdown file. Again go back to your command line and type the following.

```
# Add the markdown file to the commit list
git add ./manuscript.md
# Commit your changes with a helpful message following -m
git commit -m 'I made some really helpful edits to the manuscript.'
# Push the commit
git push
```

At this point you can go back online and see your changes. You can keep committing/pushing more changes without anybody knowing, but once you are ready to submit your changes to the group, you are going to want to create a pull request. This just means that you are requesting your edits be merged into the master project file. To do this, click the pull request tab near the top of the screen.

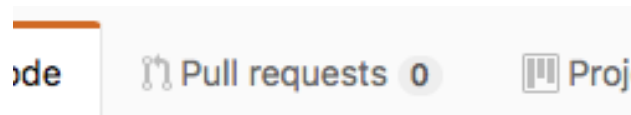


Figure 3: Start the pull request process by opening the pull request tab.

Once you are in the pull request section, you can click the big green button that says **New Pull Request** to start the submission process. At this point you will see a lot of information about the changes you made to the file and how it differs from the master copy. You can click **Create New Pull Request** to submit these changes. Just add an informative name and brief description of what you changed, and click **Create Pull Request**.

Congratulations, you submitted your manuscript changes to the master copy! At this point, the project leader will review the changes and decide to either keep all of the changes, keep some of the changes, or ask for some corrections. Pretty similar to any other manuscript editing process between collaborators. If you decide you want to make more changes, go ahead. You can commit those additional changes and add them to the pull request, or just make a new request if the old one was already accepted. It's a pretty easy process.