**GitHub and RStudio**

*Helpful Resources*

<http://happygitwithr.com/>

<https://support.rstudio.com/hc/en-us/articles/200532077-Version-Control-with-Git-and-SVN>

(I modified a lot of instructions from these websites—the former is super extensive and helpful, the latter is a little more brief but has the basics.)

*Install Git and interface with RStudio*

1. In order to use GitHub with RStudio, you have to install Git. For Windows and OS X: <https://git-scm.com/downloads>.
2. Go to Tools -> Global Options. Click the Git/SVN option on the left. Check the box for Enable version control interface for RStudio projects.
3. You may need to enter the path for your Git executable. Mine looks like this “C:/Program Files/Git/bin/git.exe” (without the quotation marks).
4. You should now have a tab labeled Git next to Environment and History in your RStudio.

*Creating a new GitHub repository*

1. Go to <github.com/KerkhoffLab> (or whatever GitHub account you are using) and under the “Repositories” tab click the green “New” button.
2. Enter repository name, description, etc. and create the repository.

*Creating a new GitHub repository*

1. Go to the repository and click on the “Settings” tab.
2. Change the name and click “Rename.”

*Creating a new project through GitHub*

1. Create a new project: File -> New Project
2. Select Version Control and then Git.
3. Copy and paste the URL of the Git repository you would like to clone. (Cloning lets you have a copy of that project on your computer. You can then edit this copy and then “push” the changes to the copy on GitHub.) If you would like to clone the old repository for our lab, the URL is <https://github.com/kerkhoffa/kerkhofflab>. Now we have a new GitHub where you can have separate repositories for each new project: <https://github.com/KerkhoffLab>.
4. Now there will be a clone of whichever repository you choose. [If you clone the original repository for our lab (<https://github.com/kerkhoffa/kerkhofflab>), there will be a copy of the entire Kerkhoff Lab GitHub on your computer (there’s no way to clone part of it). When you open a file from part of that, e.g. when I open Biome-Transitions and then the corresponding R project, you now have the option to make your edits part of the repository that we can all access.]

*Making Changes*

1. So you’ve made some edits to a file from the Kerkhoff Lab GitHub. You can see these edits in the Git tab from RStudio (next to History and Environment). You will see the files you have modified, so the R script file if you’ve changed that or any additional files you have added or deleted.
2. Click the box next to any files for which you would like to review the changes. Press Commit. This will show you exactly what edits have been made since you last committed.
3. Add a Commit message describing these edits. This will make life easier if you want to go back through previous versions. When you’re satisfied with the changes and your description, press Commit. It will now tell you how many commits you are ahead of the master, i.e. how many times you’ve committed since pushing these edits through.
4. When the commit has gone through, you have the option of pushing those edits to the copy of the folder we all can view. Simply press Push and your changes will be incorporated to the online version!
5. Word to the wise—do not either commit or push when the changes involved adding a very large file. GitHub does not like this (I think there is a maximum file size) and your changes will not go through. If you have committed this, you will also be unable to make any further commits since Git pushes the commits in the order you made them.

*Fixing problems*

Problems like the above will happen (I’ve done that at least three times), so there are some ways to fix things. Most of these solutions involve using the shell:

1. Press More (with the little gear next to it), and select Shell. This opens Git Bash, where you can enter Git-specific commands to fix whatever problem you might have.
2. I’ll update this document with specific instructions for common problems I’ve had that required the Shell! Most of the time I just Google what went wrong and then try things until I find a workable solution. There are a lot of online resources so I’ll add ones I find useful (and you guys can too!). The main thing is to do what you can to ensure you have a copy of all the important changes in case something goes wrong and they are lost in RStudio/GitHub.
3. Deleting a commit: <https://sethrobertson.github.io/GitFixUm/fixup.html>. This link takes you to a “Git choose your own adventure” where you can select exactly what you want to delete or fix. There are commands for fixing a variety of these problems.