Week 8 — Git On

Z620: Quantitative Biodiversity, Indiana University

March 6, 2015

OVERVIEW

A major goal of Quantitative Biodiversity is to provide you with tools to conduct reproducible science. One of these tools is Github. Github helps ensure the integrity of a research project by managing changes to data and code. The version control features of Github are useful for an individual working on a solo project, but also greatly facilitate collaboration among researchers.

This semester we've been using an Enterprise version of Github (https://github.iu.edu/), which is restricted to people with an IU affilition. Although useful for teaching and development, there are a few downsides to using Enterprise versions of Github. First, once you leave IU, you may no longer have access to data or code that is stored in an Enterprise respository. Second, Enterprise versions of Github may hinder your ability to collaborate with colleagues at other institutuions. Last, Enterprise versions of Github do not help us meet the requirements of many journals and funding agencies that our data and code be "open". In this handout, we present a workflow that will assist you with conducting reproducible science upon "graduation" from Quantitative Biodiversity.

After completing this exercise you will know how to:
1. Create a Github respository
2. Make a source-code file
3. Create and modify a README file
4. Create a github.com account (not restricted to IU)
5. Properly license your work
6. Migrate repositories from IU Enterprise to github.com

1) CREATING A GITHUB RESPOSITORY

When you start a new project, you are going to want to create a new Github repository. A repository holds all of your files, along with a history of the changes that have been made. This semester, you have been working with a repository that the instructors created for you (2015-QuantitativeBiodiversity). On the first day of class, you forked and cloned this respository to your local computer. After making modifying the files on your local computer, you saved and commited changes, which were then pushed to your origin (e.g., QB2015_Smith). The final step in the Github version control process is to create a pull request so that changes are merged into the upstream repository.

In the following section, we are going to go through the steps that are involved in creating your own repository:

1. Open an web browser and navigate to https://github.iu.edu/

2. At the top of page there is black banner.

On the right-ahdn side there is + symbol. If you hover over this symbol, it will say "Create new...".

\includegraphics[scale=1]{images/GitCreateNew.png}

3. Click on the + symbol and select "New Repository"

\includegraphics[scale=1]{images/GitCreateNewRepo.png}

4. Give your repository a name. For this exericse, name your repository "QBTools".

5. Decide if you want the repository to be accessible to all IU affilates (Public) or not (Private)

6. Click the button that says "Initialize this repository with a README"

7. Optional: you can click on the "Add .gitignore" button and select R

This will. . . please add text

\includegraphics[scale=0.75]{images/GitRepoInitialize.png}

8. When you have completed the items above, click on the green "Create repository" button

\includegraphics[scale=1]{images/GitCreate.png}

9. Now, we are going to clone your new repository to your local machine. Open a terminal window and type the following:

```sh
cd ~/GitHub
git clone https://github.iu.edu/User_Name/QBTools
cd ./QBTools
git status
```

10. At this point you are ready to go, but it is always good to check your remotes. You should have just one remote: origin. Open a terminal window, `cd` to the correct directory, and type the code below to confirm.

```sh
git remote -v
```

2) CREATE A SOURCE-CODE FILE

Over the past eight weeks, we have created a number of functions in R. Some of them took a long time to write. In this part of the handout, we are going to compile some of these user-defined functions into an R source-code file. As you will recall, the benefit a source file is that it contains "vetted" code, whic can be used across multiple projects. Once created, you will be able to load your source-file into a new project and use all of the functions that you created in Quantitative Biodiversity. Moreovoer, you can add new functions to this source-code file and effectively create a library of commonly used functions. The instructions below will guide you through the steps needed to create an R source-code file:

A. Create a Blank R File

Go to your menu bar and choose: File > New File > R Script. Alternatively, you can use the following shortcut key: Ctrl + Shift + N. *Note*, you want this to be a .R file, not a .Rmd file.

B. Add a File Header

The header of your source-code file should provide all of the information that will allow others to use your script.

Here is an example header:

```
################################################################################
#                                                                              #
# Quantitative Biodiversity Functions Source Code                              #
#                                                                              #
# Written by: Student X                                                        #
#                                                                              #
# Last update: 2015/03/06                                                      #
#                                                                              #
# Notes:                                                                       #
#                                                                              #
################################################################################
```

There are a few things to note about the header.
First, the comment character (#) prevents R from interpreting the information in the header. Second, the header is formatted for stylystic purposes. In our example, the header is 80 characters wide. As a rule of thumb, this is an appropriate width for both your header and code. We use the comment character (#) to box-off the top and sides of the heaer.

C. Add Content to Source-Code File

i. Require commands

In order for function in a source-code file to properly work, we need to make sure that the contributed packages are installed. By default, the following code will try to load a specificied package. If that package is not found, R will install the package.

```
require("vegan")||install.packages("vegan");require("vegan")
```

You can cusotmize this code for any package that you want loaded for our source file

ii. Functions

We are now going to populate our source-code file with functions that we have written this semester. Go through all of your assignments and find the functions that you want to save. Copy those functions into the new source-code file. Add comments about each function before the code. This annotation will help you and others understand how the functions operate. Here is a hypothetical example:

```
# My_Function: returns estimate of beta biodiversity
# Inputs: x, y, z

My_Function <- function(x, y, z){
  ...
}
```

3) CREATE A README FILE

You may recall at the beginning of the course, we had you edit a README file, which gave us some basic information about you. In a more general sense, README files are commonly used and contain important documentation about the files (code, data, etc.) that are contained in a directory. It's good practice to inclucde a detailed README file for all of your Github respositories. A basic README file should contain the following:

** 1.** General information on the content of a repository
* 2.** The version of programs and software that are needed to run code
* 3.** Contact information for the owner of the repository (e.g., name and email)
* 4.** Potential bugs

Take a moment to modify README file in your GitOn directory with your information. Save the file when you are done.

4) CREATE A GITHUB.COM ACCOUNT

We are now in the process of moving your newly created repository and source-code file to github.com. If you don't already have an account, let's create one now. Open up a new browser and navigate to https://github.com/. After supplying a username, your email, and a password, click the green button that says "Sign up for GitHub".

5) FORKING AND CLONING YOUR QBTools REPOSITORY

Unlike the Enterprise version of Github at IU, github.com encourages usesers to keep open respositories. Encourages students to choose "public". If they don't feel comfortable with this for whatever reason, they have the following options

1) don't migrate your files

2) choose private at which time Github is likely to ask for your credit card information

3) make public for now and then request private repos from github

4) make public for now, and delete the repo (or github.com account) after class

If you would prefer not to have your files open, we recommend option 3 and 4 as a short-term, easy solution.

This last one is commonplace among people who develop and distribute software (e.g. ecologists who develop R packages) and among folks who engage seriously in reproducible science.

5) MIGRATING TO GITHUB.COM

notes, add: 1. how to setup 2. create a repo. . . GitHub will ask whether you want to add a README.md. 3. add a license. . . should the licensing section be moved here, as a subsection? 4. add source file

6) LICENSING YOUR CODE

(this should come after "MIGRATING" because you can't automatically license code on IU's GitHub. If you figure out how to add a license, it won't mean much because IU probably owns whatever you put on their Enterprise, i.e., IU rights supercede your license)

When setting up public repositories on GitHub, you will want to license your work because it will be in the public domain. Licenses are what allow you to establish the right to waive responsibilities and liabilities, to attract copyright, and to legally establish intellectual primacy. Licensing allows us to be visible, transparent, and fearless professionals; protected by putting our ideas and work into the public domain while maintaining rights and restrictions to (re)use and (re)distribution.

GitHub offers a broad selection of licenses that range from short and tidy (MIT) to the thorough but lengthy (GPL V3); see http://choosealicense.com/. 9 Simple ways to make it easier to (re)use your data. There's lots of goodies of in it, the repo which is found here: https://github.com/weecology/data-sharing-paper.