Week 1 Exercise: Git and GitHub

Z620: Quantitative Biodiversity, Indiana University

January 16, 2015

Git

Git is a software system designed to allow collaborators to simultaneously work on text documents, project data, and computing code for websites and for analyzing data and conducting scientific analyses. Git is a form of *version control*.

Version control

Version control is an approach to developing text, data, and code that allows users the ability to examine, comment on, and revert back to any changes within the life of a document. In short, version control works when one make changes to a *local* version of a project (code, data, or text), and then *pushing* those changes to one's personal online version. Having one's own online version keeps your project (or your version of a group's project) safe from loss or damage. If working with collaborators, one can then submit a *pull request* for others to check and *pull* the changes into the group's project.

In short, Git is a version control system allows multiple people in remote locations to work on the same code, data, and text without irretrievably deleting or writing over others' work.

GitHub

GitHub is a web-based service for hosting projects that use the Git version control system. GitHub provides a graphical interface for viewing and managing a project's code, data, and text files, whether in collaboration with others or working individually.

GitHub allows users to *fork* a personal version of a project. Nowadays, downloading new open-source software is often done by forking a GitHub repository. Forking immediately creates a copy of a project on your personal GitHub page or other GitHub page that you have permissions for. But, forking does not create a local copy of a project on your local machine.

To create local copies, say, on your laptop, you would need to *clone* the repository. Cloning a repository can be done via GitHub (the graphical web-interface) or by working with Git directly on your local machine.

GitHub at Indiana University

IU has an Enterprise GitHub system that is a university managed version of GitHub available to faculty and student. So, IU basically has basically forked a restricted version of the GitHub project. IU's Enterprise version GitHub is only visible to members of the IU community.

Goals

- Install and Configure Git and other software
- Create an GitHub account using IU's GitHub Enterprise
- Learn what a Repo is and how we will be using Repos during in Quantitative Biodiversity

Git Installation

If you do not have a current Git installation (already installed in computer lab), please do the following:

1. Naviate to git-scm.com/download/
2. Select the appropriate operating system
3. The download should start automatically
4. Open the installer and follow the onscreen directions

On Mac: You will need to make sure you have Xcode Command Line Tools installed.
On Windows: This process will install Git Bash (msysGit). We recommend this over editing your path to use Git Tools in Command Prompt or PowerShell (If you are an experienced user you can decided for yourself).

Nano Installation

In addition to Git, we will also be using the program Nano. Nano (http://www.nano-editor.org/docs.php) is a non-GUI text editor that is useful when working in the command line and on UNIX based systems. During the course we will use it to quickly edit files. Because Nano is a command-line interface (i.e. not a GUI), it relies on keyboard commands ONLY. Most Nano commands are invoked by holding down the Ctrl key (that is, the control key), and pressing one of the other keys. While using Nano, the control key is referred to using ˆ. For example, ˆX means "*hold down the CTRL key and press the x key*". Most of the important commands are listed at the bottom of your screen (and ˆG = nano help)

If you are a **Mac User** you already have Nano. If you are a **Windows User**: 1. Download the WinNT/9x binary, .zip format nano from http://www.nano-editor.org/download.php 2. Unzip the archive and rename the unzipped folder from nano-2.x.x to nano 3. Now, move this nano folder to the Program Files folder (e.g. Program Files (x86)) 4. Edit your User Path to include Nano: 1. From Control Panel: Go to User Accounts 2. Select: *Change My Environmental Varaibles* 3. Highlight *PATH* and select Edit 4. Add the following to the end: `C:\Program Files (x86)\nano\;` (make sure that there is a ";" separating each PATH entry) 5. Open Command Prompt and Type "nano" to test ("Ctrl + X" will exit)

Git Setup

1. We recommend that you create a folder in your user directory (> cd ~) called '*GitHub*' to make this and future assignments easier to manage. (Windows Users: Do this from GitBash)

   ```
   cd ~
   mkdir ./GitHub/
   cd ./GitHub
   ```

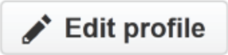
2. You will need to configure your local Git installation

   ```
   git config --global user.name "John Doe"
   git config --global user.email johndoe@example.com
   git config --global push.default simple
   git config --global credential.helper store
   ```

   You are now ready to ***Git*** **!!!**

Create User with IU's Enterprise GitHub Service

1. Navigate to https://github.iu.edu
2. Sign in with your IU Username and Passphrase
3. On the top right of your screen, click on your Username
4. Click on the Edit Profile Icon to edit your profile

Fork and Clone a Repo

1. Navigate to and click on your student repository (repo) on https://github.iu.edu/2015-QuantitativeBiodiversity
2. Fork your repo by clicking on the Fork Icon in the top right of your screen

   You should now see the repo on your GitHub page.
3. Clone the repo onto your local machine using the command line (terminal)

   ```
   cd ~/(Path to GitHub Folders, e.g. ~/GitHub/)
   git clone https://github.iu.edu/[User Name]/[Repo]
   cd ./[Repo]
   git status
   ```

   The repo should have downloaded onto your local machine and the status should stay "all up to date". You should also see that the only thing in your repo is a file named README.md

   We recommend that you create a folder in your user directory (> cd ~) called '*GitHub*' to make this and future assignments easier to manage.
4. Open and edit the README.md file

   ```
   nano README.md
   ```

   Edit the file as needed (we will demonstrate). Update your '*Student Name*' with your full name. Enter your email address. Write a short Bio about yourself (~ 1 paragraph). List three to five course expectations. Hint: View the Markdown guide to learn about formatting and making ordered lists (https://guides.github.com/features/mastering-markdown/). When you are done, save the close the document

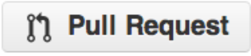
   ```
   [ctrl + o]  # Saves document
   [ctrl + x]  # Closes document
   ```
5. Now we need to add and commit our changes to git

   ```
   git status
   git add ./README.md
   git commit -m "Updated README.md with student information"
   ```
6. Now push the changes to GitHub

```
git push master
```

You should now see the repo on your GitHub page.

7. Navigate to your GitHub page to make sure that the file was uploaded correctly. If so, submit a Pull Request to submit your file to the course



The course instructors can now see your work