QB Student Repo Updating

Nathan Winsoski and Jay T. Lennon

28 February, 2017

OVERVIEW

This brief tutorial outlines how Assignments, Handouts, data files, etc. are "pushed" to student repositories via GitHub.

STEP 1: SETTING UP SSH KEY

We found that some of our scipts did not work unless we enabled SSH key. If this is not set up on your computer already, you can do so with the following steps. In the terminal window generate a new SSH key with the following command:

```
ssh-keygen -t rsa -C "your_email@example.com"
```

You will be asked to enter and re-enter a passphrase. After that, you need to add the new key to the SSH-agent using the following commands, which will generate an agent pid.

```
eval "$(ssh-agent -s)"
```

```
ssh-add ~/.ssh/id_rsa
```

Next you need to add your SSH key to your GitHub account. To obtain your SSH key, type the following command at the terminal. (Note: your key may be named one of the folllowing instead of id_rsa.pub: id_dsa.pub, id_ecdsa.pub or id_ed25519.pub)

```
pbcopy < ~/.ssh/id_rsa.pub
```

Now go to https://github.com/settings/keys. Click "new SSH key", make a title for your key, and paste your key into the "key" area, and then click the "add key" button. You will then be asked to supply your GitHub password. Finally, you should test your SSH key by typing the following in the terminal window. When asked if you want to continue, type "yes".

```
ssh -T git@github.com
```

STEP 2: CREATE STUDENT REPOSITORIES

An instructor will need to create an upstream repository for each student in the QBstudents GitHub organization (https://github.com/QBstudents).

STEP 3: SETUP ON YOUR LOCAL MACHINE

Create a GitHub Directory (e.g., QBstudents) that contains files found in the QuantitativeBiodiversity/InstructorResources/GitAutomation folder (see https://github.com/QuantitativeBiodiversity/InstructorResources). In that folder there is a file called `participation_repos.txt`. This file should contain all of the names of the students' respositories that you created. The contents of that file should look something like this:

```
QB2017_Beidler
QB2017_Benavidez
QB2017_Bennett
QB2017_Gibson
QB2017_Kuo
QB2017_Moger-Reischer
QB2017_Test
```

Note you will also have QB2017_Test, which is for instructors use as it cannot be accessed by students.

STEP 4: CLONE STUDENT REPOSITORIES TO YOUR COMPUTER

You need to clone student repositories to your local computer so you can push content to them. In the terminal, navigate to the location where you would like to set up the student repositories. Make sure you know the path to the shell scripts so you can access them. Now we will use the `GitCloneQB.sh` script to clone:

```
sh GitCloneQB.sh participant_repos.txt
```

This is where we ran into problems in 2017 when we did not have the SSH key enabled. Otherwise you may try changing the following line in `GitCloneQB.sh` to work with html instead of ssh:

```
git clone git@github.com:QBstudents/$EachLine.git
```

STEP 5: UPDATE STUDENT REPOSITORIES

In the following sections, we outline how to push content to the each student repository. You'll need to know the paths to the files you would like to push.

A) Make Recipient Directories

First, we need to make the recipient directory (e.g., `Week7-PhyloCom`) that you'll be pushing to with the `GitMkdirQB.sh` script. You will need to supply the relative path (starting from each student repository, e.g., `QB2017_Test`). So, for example, to create Phylocom, we'll run this line of code:

```
sh GitMkdirQB.sh Week7-PhyloCom participant_repos.txt
```

And to create a data folder within the Week7-PhyloCom directory, you would run the following:

```
sh GitMkdirQB.sh Week7-Phylocom/data participant_repos.txt
```

B) Copy Files to Newly Created Directory

To copy files, we'll use the `GitCopyQB.sh` script. This take the following arguments: {path to file to copy} {path to recipient directory} {student repo list}. For example, to copy the Phylocom assignment (replace the path to file with the path on your machine):

```
sh GitCopyQB.sh ~/Github/QuantitativeBiodiversity/QB-2017/Week7-PhyloCom/PhyloCom_assignment.Rmd Week7-
```

You can arrow-up and modify file (use option left arrow for efficiency) to add another file.

C) Add and Commit Files

Now use `GitAddCommitQB.sh` script to add and commit files you just added. The first argument can be the path to a specific file (e.g., `Week6-PhyloTraits/PhyloTraits_handout.Rmd`) or a flag that git recognizes, e.g., `-A` for all files. The next argument is a git commit message (e.g., what you would normally type after `-m` in a git commit). Don't type `-m` though. Last, you'll need to supply the list of the student repositories.

Here's an example of how to add Week7 materials:

```
sh GitAddCommitQB.sh -A 'Adding Week7 Materials' participant_repos.txt
```

D) Push to Student Repos on GitHub

Pushing to student repos is easy, just run `GitPushQB.sh` and supply the list of student repositories:

```
sh GitPushQB.sh participant_repos.txt
```

STEP 6: OTHER SCRIPTS

There are a few other scripts you may need to use. See below for descriptions on how to use them.

A) Remove Files

If you need to remove a file from each student's repo, use the `GitRmQB.sh` script in a similar way:

```
sh GitRmQB.sh path/to/file participant_repos.txt
```

B) Moving Files

If you need to move (or rename) a file in each repo, use the `GitMvQB.sh` script:

```
sh GitMvQB.sh path/of/file/to/move path/of/place/to/move/it participant_repos.txt
```

C) Updating .gitignore Files

To update students' gitignore files (e.g., to ignore .DS_Store files):

```
sh GitUpdateGitIgnore.sh '.DS_Store' participant_repos.txt
```