QB Student Repo Updating

Jay T. Lennon

26 February, 2021

OVERVIEW

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

STEP 1: SETTING UP SSH KEY

We found that some of our scripts 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
```

(2021: ran into problems with the eval command above, so skipped it and seemed to be OK for creating key)

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 following 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). For archiving, student repos from previous years were downloaded to local computers and then deleted.

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). (Jay forked and cloned InstructorResources to local machine then copied the contents of the GitAutomation folder to a new directory – QBstudents – that he created, or you can just have student folders in GitAutomation where shell scripts are located). In the GitAutomation folder there is a file called `participation_repos.txt`. This file must contain all of the names of the students' repositories that you created. Make sure there are no spaces before or after names as this will create errors. Also make sure that repos match up with `participation_repos.txt` (order and content?) The contents of `participation_repos.txt` should look like this:

```
QB2019_Miller
QB2019_Bolin
QB2019_Brewer
QB2019_Caple
QB2019_Crawley
QB2019_Hibbins
QB2019_Mueller
QB2019_Peckenpaugh
QB2019_Rios
QB2019_Phillips
QB2019_Test
```

Note you will also have QB2021_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 (e.g., ~/GitHub/QBstudents/). (Jay did this in the QBstudents folder that contained the copied GitAutomation contents) 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. (no issues with this in 2019, so did not have to monkey with next step)

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: MERGE AND PULL FROM UPSTREAM

Before you update the student repositories (i.e, add files), you should first merge any pull requests that were made by students. After that, you need to pull (i.e., fetch + merge) changes, which might include completed worksheet; otherwise, you will likely encounter a conflict. The following script will perform a batch pull:

```
sh GitPullQB.sh participant_repos.txt
```

(There is also another file in the GitAutomation folder named `GitPullQB2.sh`, but it is not clear what this script does)

STEP 6: 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., `QB2019_Test`). So, for example, we can create a new directory with the following code at the Terminal:

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

And to create a data folder within the Week7-PhyloCom directory, you would run the following: (OK, but is this necessary?)

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

In the following example, you could make a subdirectory ("1.Introductions") in each student's respository (e.g., "QB2019_Miller"), which contains a folder named "2.Worksheets". (In this case stuent respositories and GitMkdirQB.sh are in the same QBstudent root directory):

```
sh GitMkdirQB.sh 2.Worksheets/1.Introductions participant_repos.txt
```

B) Copy files to newly created directory

To copy files, we'll use the `GitCopyQB.sh` script. This takes 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-
```

Here's what Jay did; involved frustration with not getting path exactly right

```
sh GitCopyQB.sh /Users/lennonj/GitHub/QB-2019/2.Worksheets/5.AlphaDiversity/5.AlphaDiversity_Worksheet.
```

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 7: 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
```

You may need to remove a file or folder that no longer exists in the QB_2019 repo. In this case, you can just cd to the appropriate directory containing student repos, and for example, type the following to remove a directory. In this case, we're using wildcads to remove "8.Phylodiversity" from each of the students's repos.

```
rm -r QB2019_*/2.Worksheets/8.Phylodiversity
```

After that, following instructions above, you would type:

```
sh GitAddCommitQB.sh -A 'removed 8.Phylodiversity' participant_repos.txt
```

and

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

The old phylodiversity file will now be removed from all of the students's repos.

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
```