Cheatsheet: GitHub

Luke Johnston

2015-03-23

Git is pretty powerful tool for managing projects, especially scientific projects. An added benefit of using Git is you can use the amazing GitHub, a git repository hosting service, to keep an online backup of your project and optionally share it with the world or with your team. There are increasing arguments and support for science to be more open and publicly accessible. GitHub can help with getting your code and/or data out into the public domain. While the following commands are used for dealing with GitHub, they are not exclusive to GitHub.

Before using Git and GitHub: Initial setup

Before you start trying these commands, do the initial setup found on the Git cheatsheet page. These need to be completed before using these commands.

Git commands for dealing with GitHub

`git clone <repository-name>`

Cloning is basically downloading a new git repository that is online/in GitHub. You take an existing git repository and duplicate/copy/clone it onto your own computer. For example, if you wanted the files for this workshop series, you would first fork our GitHub Code As Manuscript workshop repo so you have your own copy of it on your account and then you clone it into your computer.

Example code:

```
## First fork our Code As Manuscript and then:
cd /path/to/where/you/want/the/repo
git clone https://github.com/your-name/workshops.git
## You now have the workshop files on your computer.
```

`git remote add <name> <server-url>`

A remote, in git terminology, is a server or online location. Remote is the opposite of local. A remote repository is a repository that is *not* on your computer, while a local repository is. Think of the remote as an external hard drive. When you add a remote, you use a name (by convention the name is usually "origin", which I *strongly* encourage you to use as well) that will tell git that that is the remote name. The server url can generally be found on the GitHub or BitBucket page, usually in the top or bottom right corner. For instance, the server url for my own 'test' project would be `https://github.com/lwjohnst86/test.git`. to describe your Git project that you want to store on the server such as Github.

Example code:

```
cd /path/to/your/git/repo
## The actual URL can be found on GitHub, usually in the corner.
git remote add origin https://github.com/yourname/yourproject.git
git push
```

`git push`

Push is essentially the same as uploading your local git repository to the remote (GitHub) repository. Pushing is more powerful than a simple upload, as git checks the remote repository, compares it to the local repository, making sure that the integrity of the files is preserved and that nothing is lost or overwritten (unlike Dropbox for example). This is especially important for your research files!

Example code:

```
cd /path/to/your/git/repo
## ... edit a file ...
git commit -am "Edited a file"
git push
```

`git pull`

This is essentially a command to download the remote (GitHub) repository contents and merge it into your own local git repository. This is only ever used if you a) work on a project that is only more than one computer and you use git and GitHub to sync the files across computers, or b) if you work on a team and one or more other person(s) are making changes to the remote (GitHub) repository. Pulling then syncs the updated remote content with your local content. This is where git really shines when you collaborate with others on a project (ie: this workshop series)!

Example code:

```
cd /path/to/your/git/repo
## Someone has added stuff to the remote repo
git pull
```