Sharing Scripts & Data

C. Ryan Campbell

Duke University

c.ryan.campbell@duke.edu

28 Sept 2017

Overview

1. Goals
2. Gitting
   - Fetch, Commit, Push & Pull
3. Data
   - Commandline Tools
   - SRA

Today's Goals

- Set up a group git

Today's Goals

- Set up a group git
  - Fetch and Commit

Today's Goals

- Set up a group git
  - Fetch and Commit
  - Push and Pull

Today's Goals

- Set up a group git
  - Fetch and Commit
  - Push and Pull
  - Access files in cluster folder

Today's Goals

- Set up a group git
  - Fetch and Commit
  - Push and Pull
  - Access files in cluster folder
- Download project data

Today's Goals

- Set up a group git
  - Fetch and Commit
  - Push and Pull
  - Access files in cluster folder
- Download project data
  - `wget, curl, sratoolkit`

GitHub

- GitHub is a version control and file sharing service

GitHub

- GitHub is a version control and file sharing service
- Works on your laptops as well as the cluster (via terminal)

GitHub

- GitHub is a version control and file sharing service
- Works on your laptops as well as the cluster (via terminal)
- We're going to use GitHub for the projects so that:

GitHub

- GitHub is a version control and file sharing service
- Works on your laptops as well as the cluster (via terminal)
- We're going to use GitHub for the projects so that:
  - You can all access and edit the same files

GitHub

- GitHub is a version control and file sharing service
- Works on your laptops as well as the cluster (via terminal)
- We're going to use GitHub for the projects so that:
  - You can all access and edit the same files
  - Those files can be transferred easily to the cluster

GitHub

- GitHub is a version control and file sharing service
- Works on your laptops as well as the cluster (via terminal)
- We're going to use GitHub for the projects so that:
  - You can all access and edit the same files
  - Those files can be transferred easily to the cluster
  - You aren't submitting redundant commands

GitHub

- GitHub is a version control and file sharing service

GitHub

- GitHub is a version control and file sharing service
- Works on your laptops as well as the cluster (via terminal)

GitHub

- GitHub is a version control and file sharing service
- Works on your laptops as well as the cluster (via terminal)
- We're going to use GitHub for the projects so that:

GitHub

- GitHub is a version control and file sharing service
- Works on your laptops as well as the cluster (via terminal)
- We're going to use GitHub for the projects so that:
  - You can all access and edit the same files

GitHub

- GitHub is a version control and file sharing service
- Works on your laptops as well as the cluster (via terminal)
- We're going to use GitHub for the projects so that:
  - You can all access and edit the same files
  - Those files can be transferred easily to the cluster

GitHub

- GitHub is a version control and file sharing service
- Works on your laptops as well as the cluster (via terminal)
- We're going to use GitHub for the projects so that:
  - You can all access and edit the same files
  - Those files can be transferred easily to the cluster
  - You aren't submitting redundant commands

GitHub

- This will all take place in the course Git repo

GitHub

- This will all take place in the course Git repo
- You'll need to clone it to your computer

GitHub

- This will all take place in the course Git repo
- You'll need to clone it to your computer
- Then give me your git username so I can make you a collaborator

GitHub

- Go to the GitHub site and copy the “clone URL”

GitHub

- Go to the GitHub site and copy the “clone URL”
- https://github.com/cryancampbell/duke-bio490s.git

GitHub

- Go to the GitHub site and copy the “clone URL”
- https://github.com/cryancampbell/duke-bio490s.git
- Using one of SourceTree/github/terminal clone the repo to your computer

GitHub

- Go to the GitHub site and copy the “clone URL”
- https://github.com/cryancampbell/duke-bio490s.git
- Using one of SourceTree/github/terminal clone the repo to your computer
- `git clone https://github.com/cryancampbell/duke-bio490`

Fetch, Commit, Push & Pull

- Fetch, Commit, Push & Pull

Fetch, Commit, Push & Pull

- Fetch, Commit, Push & Pull
  - The four main commands to send data back and forth to git

Fetch, Commit, Push & Pull

- Fetch, Commit, Push & Pull
  - The four main commands to send data back and forth to git
- Again, they can be used in the terminal:

Fetch, Commit, Push & Pull

- Fetch, Commit, Push & Pull
  - The four main commands to send data back and forth to git
- Again, they can be used in the terminal:
- e.g. - `git fetch`

Fetch, Commit, Push & Pull

- Fetch, Commit, Push & Pull
  - The four main commands to send data back and forth to git
- Again, they can be used in the terminal:
- e.g. - `git fetch`
- Or the buttons on the SourceTree or github

Fetch, Commit, Push & Pull

- Fetch - checks with the repo to see what changes you've missed

Fetch, Commit, Push & Pull

- Fetch - checks with the repo to see what changes you've missed
- Pull - changes your local machine to reflect those changes

Fetch, Commit, Push & Pull

- Fetch - checks with the repo to see what changes you've missed
- Pull - changes your local machine to reflect those changes
- Commit - prepares local changes to be pushed to the repo

Fetch, Commit, Push & Pull

- Fetch - checks with the repo to see what changes you've missed
- Pull - changes your local machine to reflect those changes
- Commit - prepares local changes to be pushed to the repo
- Push - changes the repo to reflect the changes you've committed

Fetch, Commit, Push & Pull

- Fetch - checks with the repo to see what changes you've missed

Fetch, Commit, Push & Pull

- Fetch - checks with the repo to see what changes you've missed
- Pull - changes your local machine to reflect those changes (aka "merge")

Fetch, Commit, Push & Pull

- Fetch - checks with the repo to see what changes you've missed
- Pull - changes your local machine to reflect those changes (aka "merge")
- Commit - prepares local changes to be pushed to the repo (it isn't tracking every change you make)

Fetch, Commit, Push & Pull

- Fetch - checks with the repo to see what changes you've missed
- Pull - changes your local machine to reflect those changes (aka "merge")
- Commit - prepares local changes to be pushed to the repo (it isn't tracking every change you make)
- Push - changes the repo to reflect the changes you've committed

GitHub

- This folder is for SCRIPTS not raw data

GitHub

- This folder is for SCRIPTS not raw data
- (Otherwise we'd all need GB of laptop space)

GitHub

- This folder is for SCRIPTS not raw data
- (Otherwise we'd all need GB of laptop space)
- So remember to send downloaded data to your own folder on the cluster

GitHub

- This folder is for SCRIPTS not raw data
- (Otherwise we'd all need GB of laptop space)
- So remember to send downloaded data to your own folder on the cluster
- e.g. - `/work/cc216/490S/<your folder>`

Data

- Many journals and grants require data to be public

Data

- Many journals and grants require data to be public
- PLOS, NSF, NIH

Data

- Many journals and grants require data to be public
- PLOS, NSF, NIH
- The information for where data can be found is often in:

Data

- Many journals and grants require data to be public
- PLOS, NSF, NIH
- The information for where data can be found is often in:
  - The “Methods” section of a paper

Data

- Many journals and grants require data to be public
- PLOS, NSF, NIH
- The information for where data can be found is often in:
  - The “Methods” section of a paper
  - An appendix

Data

- Many journals and grants require data to be public
- PLOS, NSF, NIH
- The information for where data can be found is often in:
  - The “Methods” section of a paper
  - An appendix
  - The author’s lab website

Data

- Many journals and grants require data to be public
- PLOS, NSF, NIH
- The information for where data can be found is often in:
  - The “Methods” section of a paper
  - An appendix
  - The author’s lab website
- We’ll talk about how to get data from each of these places

Downloand Commands

- `curl`

Downloand Commands

- `curl`
- `wget`

Downloand Commands

- `curl`
- `wget`
- `fastq-dump` - subset of the SRA Toolkit

Downloand Commands

- `curl`
- `wget`
- `fastq-dump` - subset of the SRA Toolkit
- Each is useful in different situations

curl

- Downloads a given url site

```
Usage:  curl [options...]  <url>
```

- Options include username & password
- Useful for sftp servers
- An example:

```
curl -o ESPNfrontpage.html http://www.espn.com/
```

- saved as `ESPNfrontpage.html`

wget

- Downloads a given url site

```
Usage:  wget [OPTION]...  [URL]...
```

- Same options as curl
- An example:

```
wget http://www.bzip.org/1.0.6/bzip2-1.0.6.tar.gz
```

- Often used for downloading software (you'll see later...)

Sequence Read Archive

- Holds raw sequence data from published articles

Sequence Read Archive

- Holds raw sequence data from published articles
- Sorted by Experiment and Project

Sequence Read Archive

- Holds raw sequence data from published articles
- Sorted by Experiment and Project
- Use SRA Toolkit to access the files

Sequence Read Archive

- Often directly reference in papers

Sequence Read Archive

- Often directly reference in papers
- e.g. “data are available in SRR#########”

Sequence Read Archive

- Often directly reference in papers
- e.g. “data are available in SRR#########”
- Sometimes they are well organized, sometimes not

Sequence Read Archive

- Often directly reference in papers
- e.g. “data are available in SRR#########”
- Sometimes they are well organized, sometimes not
- You should be able to follow the paper’s methodology to separate the data into samples (if it wasn’t kept that way on SRA)

SRA Toolkit

- So, how do we download the data?

SRA Toolkit

- How do we get the software onto the cluster?

SRA Toolkit

- How do we use this file?

SRA Toolkit

- How do we use the toolkit?
- What is the toolkit (file, command, etc)?

Group Work

Group steps

1. Pick a group ID/Name
2. Make a group folder with that name within:
3. `duke-bio490s/projects/<group name here>`

- This is where you'll keep scripts to download, trim, etc data

Group Work

- You should have everything you need now to download your own data

Group Work

- You should have everything you need now to download your own data
- TIPS:

Group Work

- You should have everything you need now to download your own data
- TIPS:
  - Run the line of code in the terminal FIRST so you can troubleshoot it

Group Work

- You should have everything you need now to download your own data
- TIPS:
  - Run the line of code in the terminal FIRST so you can troubleshoot it
  - Instead of downloading the whole file use `head` to check output

Group Work

- You should have everything you need now to download your own data
- TIPS:
  - Run the line of code in the terminal FIRST so you can troubleshoot it
  - Instead of downloading the whole file use `head` to check output
  - Once your output looks correct THEN submit to the cluster

Group Work

- You should have everything you need now to download your own data
- TIPS:
  - Run the line of code in the terminal FIRST so you can troubleshoot it
  - Instead of downloading the whole file use `head` to check output
  - Once your output looks correct THEN submit to the cluster
  - This will save a lot of headaches and waiting

Group Work

- You should have everything you need now to download your own data
- TIPS:
  - Run the line of code in the terminal FIRST so you can troubleshoot it
  - Instead of downloading the whole file use `head` to check output
  - Once your output looks correct THEN submit to the cluster
  - This will save a lot of headaches and waiting
  - And most importantly:
    DON'T write data to the git repo

The End