Building Your Open Science Toolkit

Cassidy Soloff

Outline

- Roots for Resiliency
- Open Science
- Data management
- GitHub
- Containers
- Remote Computing

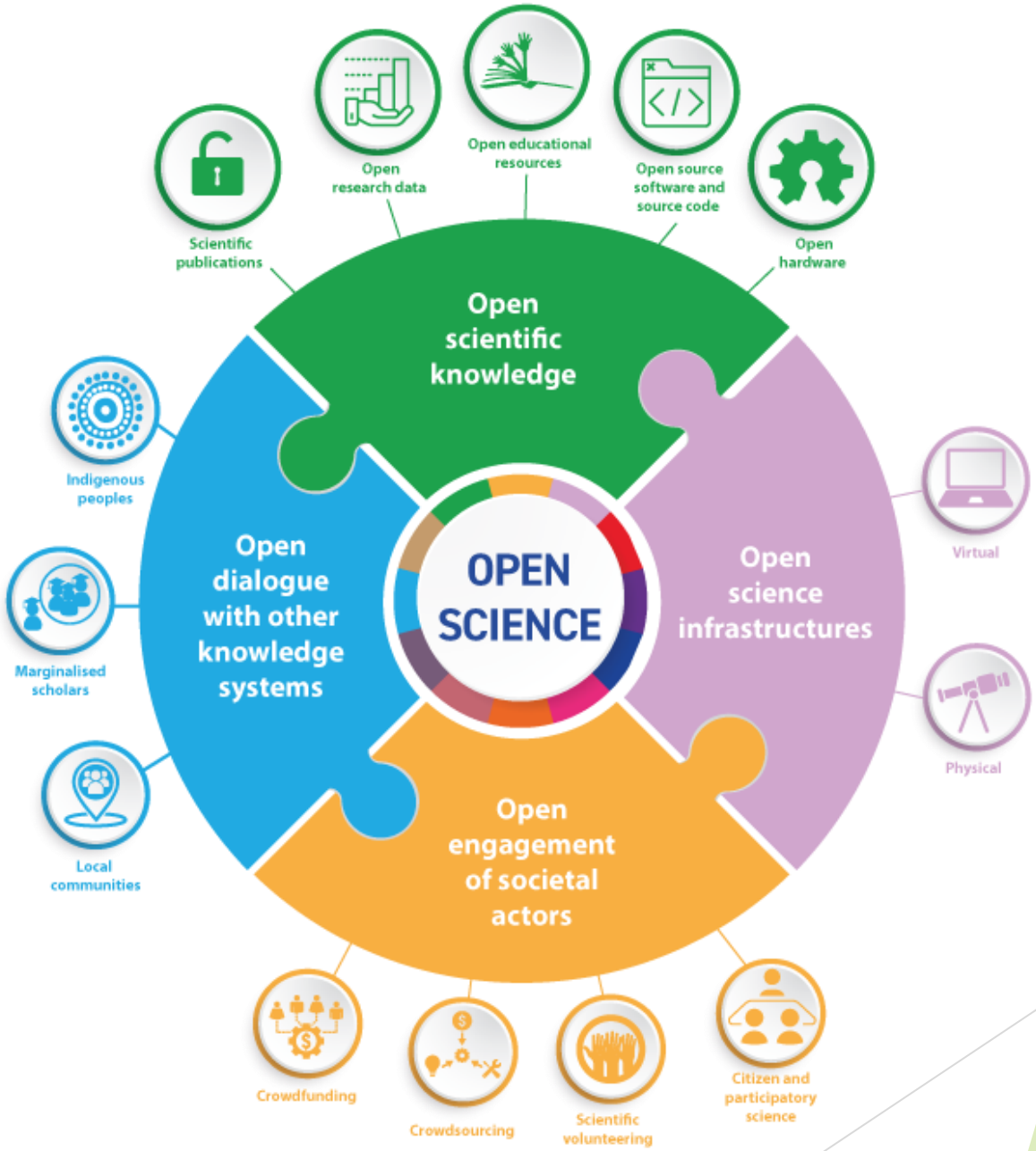
Roots for Resiliency?

- The Roots for Resilience Program provides training and support to select graduate students on open, reproducible science and computational infrastructure tools to enhance research focused on resiliency in the environment
- 13 students from different departments
- Takes place in fall semester
- Meets twice a week
- Each fellow is nominated by department
  - Talk to your advisor if interested in being nominated
  - The process for selection usually starts in April

What is open science?

- Open access to data, analysis, and software
- Reproducibility, accountability and collaboration
- Examples:
  - Government agencies like NASA and NOAA share their data and models globally, giving anyone access to life saving forecasts and the ability to further scientific development
  - Publications often give you the ability to shared code and data used for analysis

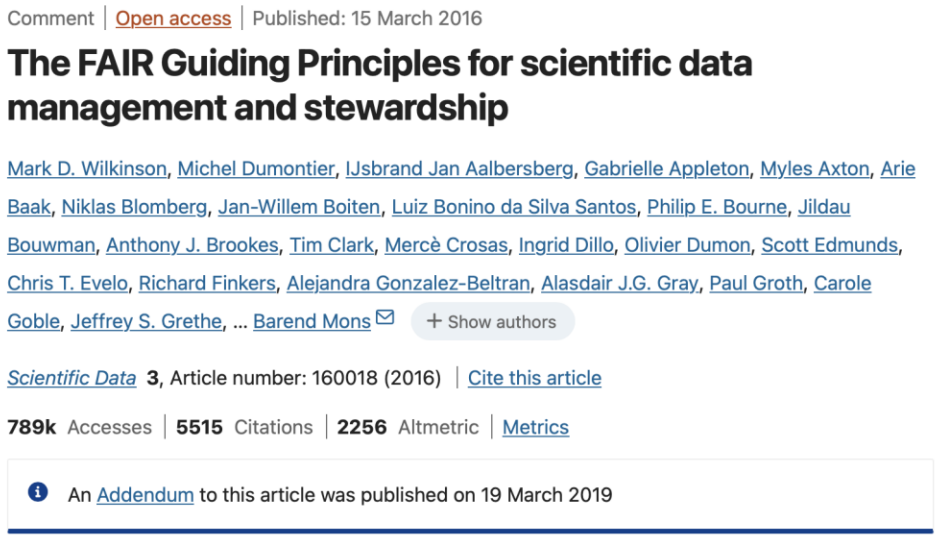
Pillars of Open Science



Credit: United Nations Educational, Scientific and Cultural Organization

Data management

- Data is the foundation of science, so we will start here to in building an open science framework
- FAIR Principles
  - Findable
  - Accessible
  - Interoperable
  - Reusable



Comment | Open access | Published: 15 March 2016

The FAIR Guiding Principles for scientific data management and stewardship

Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J.G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, ... Barend Mons + Show authors

Scientific Data **3**, Article number: 160018 (2016) | Cite this article

789k Accesses | **5515** Citations | **2256** Altmetric | Metrics

An Addendum to this article was published on 19 March 2019

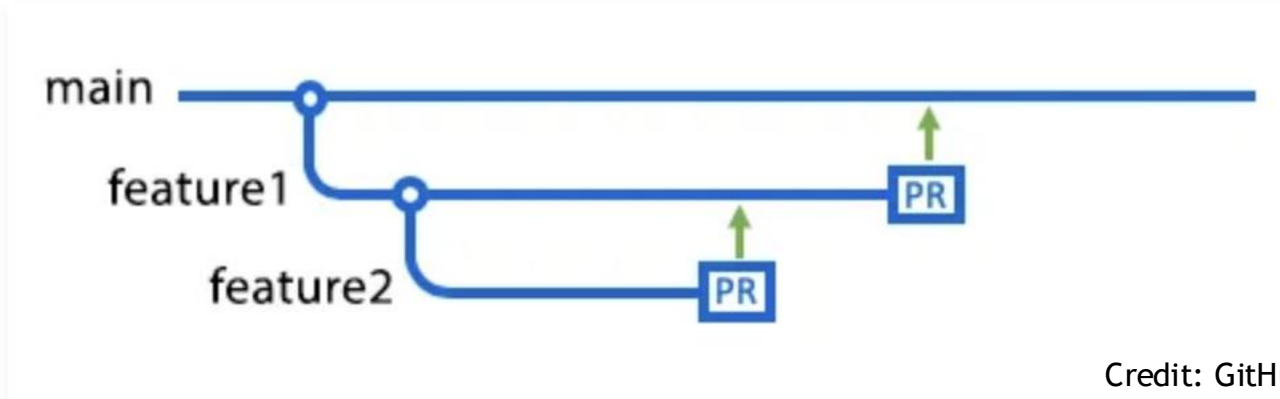
- Ensure both data and meta data follow these principles

Data Manage Plan (DMP) Tools

- Through the university, we have access to Data Manage Plan (DMP) Tools website: https://dmptool.org/
- Public funded research requires you to make a data management plan
  - "DMPs are short, formal, documents outlining what types of data will be used, and what will be done with the data both during and after a research project concludes."
- Through DMP Tools you can…
  - Make a private and public plan
  - Request feedback
  - Create an ORCID
  - Use ReData if you want to use the university to store data

Using GitHub

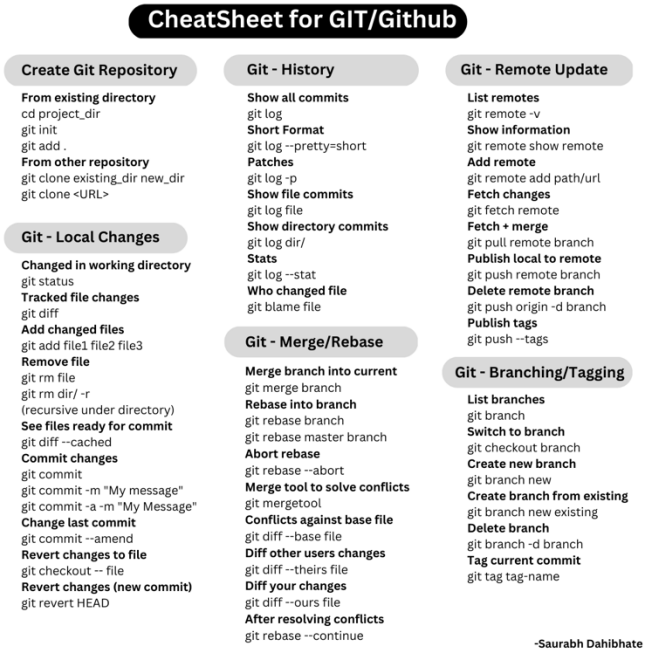
- GitHub is an incredible resource for making creating, editing, and collaborating with code
- You can track changes you have made by pushing new edits to your code to a GitHub repository
- Using branches, you can work on new features of your code without effecting the code on the main branch
- When you merge changes from one branch to another, you create a pull request



Credit: GitHub

GitHub and VS Code

- GitHub commands can be done in command line, but VS Code has powerful integrations that can make the process more intuitive



CheatSheet for GIT/Github

Create Git Repository

From existing directory
cd project_dir
git init
git add .
From other repository
git clone existing_dir new_dir
git clone <URL>

Git - Local Changes

Changed in working directory
git status
Tracked file changes
git diff
Add changed files
git add file1 file2 file3
Remove file
git rm file
git rm dir/ -r
(recursive under directory)
See files ready for commit
git diff --cached
Commit changes
git commit
git commit -m "My message"
git commit -a -m "My Message"
Change last commit
git commit --amend
Revert changes to file
git checkout -- file
Revert changes (new commit)
git revert HEAD

Git - History

Show all commits
git log
Short Format
git log --pretty=short
Patches
git log -p
Show file commits
git log file
Show directory commits
git log dir/
Stats
git log --stat
Who changed file
git blame file

Git - Merge/Rebase

Merge branch into current
git merge branch
Rebase into branch
git rebase branch
git rebase master branch
Abort rebase
git rebase --abort
Merge tool to solve conflicts
git mergetool
Conflicts against base file
git diff --base file
Diff other users changes
git diff --theirs file
Diff your changes
git diff --ours file
After resolving conflicts
git rebase --continue

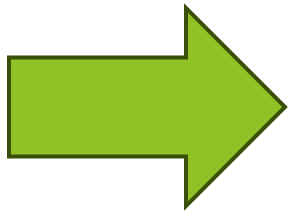
Git - Remote Update

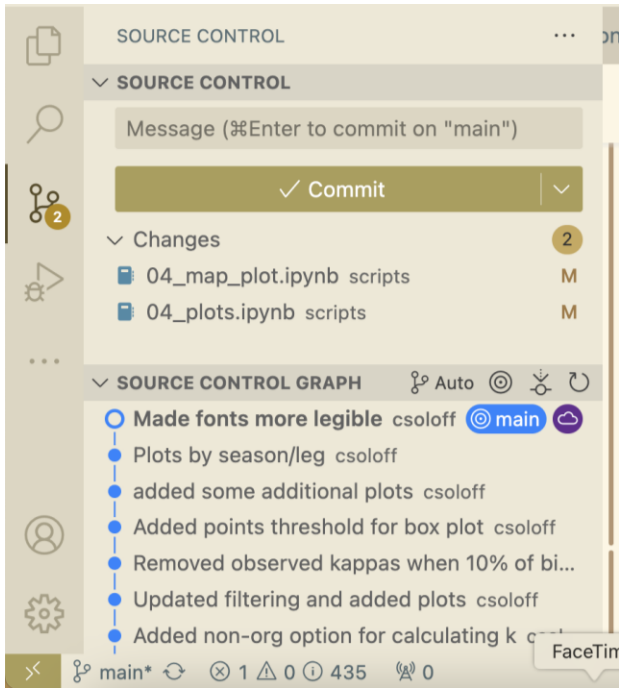
List remotes
git remote -v
Show information
git remote show remote
Add remote
git remote add path/url
Fetch changes
git fetch remote
Fetch + merge
git pull remote branch
Publish local to remote
git push remote branch
Delete remote branch
git push origin -d branch
Publish tags
git push --tags

Git - Branching/Tagging

List branches
git branch
Switch to branch
git checkout branch
Create new branch
git branch new
Create branch from existing
git branch new existing
Delete branch
git branch -d branch
Tag current commit
git tag tag-name

-Saurabh Dahibhate





Making your code reproducible

- Create a README file
- Provides scripts
- Access to necessary data
- Organize your repository with folders
- Specify if scripts should be run in a particular order

Making a README.md

- In GitHub, .md documents stand for Markdown documents
- Markdown has simple syntax that allows you to format the text for web display

Markdown	HTML	Rendered Output
`# Heading level 1`	`<h1>Heading level 1</h1>`	Heading level 1
`## Heading level 2`	`<h2>Heading level 2</h2>`	Heading level 2

Markdown	HTML	Rendered Output
`I just love **bold text**.`	`I just love <strong>bold text</strong>.`	I just love **bold text**.
`I just love __bold text__.`	`I just love <strong>bold text</strong>.`	I just love **bold text**.

Markdown	HTML	Rendered Output
`Italicized text is the *cat's meow*.`	`Italicized text is the <em>cat's meow</em>.`	Italicized text is the *cat's meow*.
`Italicized text is the _cat's meow_.`	`Italicized text is the <em>cat's meow</em>.`	Italicized text is the *cat's meow*.
`A*cat*meow`	`A<em>cat</em>meow`	A*cat*meow

What to include in a README.md

- Description of repository
- Description of file structure
- Description of the contents and purpose of each file
- Steps to set up the coding environment
- How to run the scripts

Coding environments

- To run specific code, you need to make sure all the packages and the correct versions are installed
- Coding environments allow you to reproduce the exact environment that is compatible with a repository of code
- In Python, you can export all the packages required to make your code run to a file called “requirements.txt”

```
pip freeze > requirements.txt
```

```
python -m venv new_env source
new_env/bin/activate
python -m pip install -r requirements.txt
```

Example GitHub repository

- Research repository
  - https://github.com/csoloff/ACTIVATE_CCN_closure
- Website hosting
  - https://github.com/csoloff/csoloff.github.io
  - csoloff.com

Software containers

- Another way to make your code reproducible is to build software containers
- Software containers package your code and software
  - When you run a container, all the required software will install
  - Makes it easier for the recipient to run your code, no need to manually install dependencies
  - Works across operating systems
- Docker is a popular platform for building, testing, and deploying containers

Remote computing

- Remote computing allows you to run your code off a powerful computer, speeding up the time it takes for computationally demanding tasks to run
- The University of Arizona has two amazing resources for remote computing
  - CyVerse
  - High Performance Computer (HPC)

CYVERSE

- “Cyverse has been in existence for 16 years; has spent $120M in research funds; has 135,000 registered users; and has facilitated 1,700 peer-reviewed publications across many scientific fields”
- Data storage: up to 3TB of storage, easy to share data
- Analysis tools: preinstalled analysis applications
  - Executable Apps: Run a script or a series of scripts on your data
  - Interactive Apps: Launch software such as Jupyter notebooks, RStudio, QGIS, VScode, and more
- Cyverse.org

U of A High Performance Computer (HPC)

- Free access to storage and computing
- The power of computing on HPC is due to the high numbers of cores (16 – 94 cores per computer)
  - To harness this power, you must make sure you code is parallelized
    - In Python, I use Joblib
    - In some software, it may be just a simple as checking a box
- It is also possible to use multiple computers (or nodes) at once

Questions?