Version Control with Git

Open Science

Learning Objectives

- Explain how a version control system can be leveraged as an electronic lab notebook for computational work.

> The opposite of “open” isn’t “closed”. The opposite of “open” is “broken”.
>
> — John Wilbanks

Free sharing of information might be the ideal in science, but the reality is often more complicated. Normal practice today looks something like this:

- A scientist collects some data and stores it on a machine that is occasionally backed up by her department.
- She then writes or modifies a few small programs (which also reside on her machine) to analyze that data.
- Once she has some results, she writes them up and submits her paper. She might include her data—a growing number of journals require this—but she probably doesn’t include her code.
- Time passes.
- The journal sends her reviews written anonymously by a handful of other people in her field. She revises her paper to satisfy them, during which time she might also modify the scripts she wrote earlier, and resubmits.
- More time passes.
- The paper is eventually published. It might include a link to an online copy of her data, but the paper itself will be behind a paywall: only people who have personal or institutional access will be able to read it.

For a growing number of scientists, though, the process looks like this:

- The data that the scientist collects is stored in an open access repository like figshare or Zenodo, possibly as soon as it’s collected, and given its own DOI. Or the data was already published and is stored in Dryad.
- The scientist creates a new repository on GitHub to hold her work.
- As she does her analysis, she pushes changes to her scripts (and possibly some output files) to that repository. She also uses the repository for her paper; that repository is then the hub for collaboration with her colleagues.
- When she’s happy with the state of her paper, she posts a version to arXiv or some other preprint

server to invite feedback from peers.
- Based on that feedback, she may post several revisions before finally submitting her paper to a journal.
- The published paper includes links to her preprint and to her code and data repositories, which makes it much easier for other scientists to use her work as starting point for their own research.

This open model accelerates discovery: the more open work is, the more widely it is cited and re-used. However, people who want to work this way need to make some decisions about what exactly "open" means and how to do it.

This is one of the (many) reasons we teach version control. When used diligently, it answers the "how" question by acting as a shareable electronic lab notebook for computational work:

- The conceptual stages of your work are documented, including who did what and when. Every step is stamped with an identifier (the commit ID) that is for most intents and purposes is unique.
- You can tie documentation of rationale, ideas, and other intellectual work directly to the changes that spring from them.
- You can refer to what you used in your research to obtain your computational results in a way that is unique and recoverable.
- With a distributed version control system such as Git, the version control repository is easy to archive for perpetuity, and contains the entire history.

Making Code Citable

This short guide from GitHub explains how to create a Digital Object Identifier (DOI) for your code, your papers, or anything else hosted in a version control repository.

How Reproducible Is My Work?

Ask one of your labmates to reproduce a result you recently obtained using only what they can find in your papers or on the web. Try to do the same for one of their results, then try to do it for a result from a lab you work with.