

git and GitHub

[git](#) is a software version control system. git is useful for tracking changes to code, including systematic approaches to adding new features or changing functionality. We use git to track and save code as we're working on it.

[GitHub](#) is one of several online repositories for sharing code (and is commonly used as a remote repository for git). We use GitHub to facilitate collaboration and share code for papers and other projects.

Setting Up Project Repositories

Meta-Repositories

Repository Visibility

It may be prudent to set up initial repositories as private until you're ready to release the code (which should always be done **ahead** of paper submission).

When you start a new project, you should [create a GitHub meta-repository](#) right away. You can do this in your personal account or as part of the [srikrishnan-lab](#) organization. Make sure that your repository name is informative; we will eventually rename this to align with a paper (something like [lead_author-et-al-year](#)).

A meta-repository is a repository which creates a single point of access for all resources needed to create a published work. We use meta-repositories to avoid mixing data, models, and analysis codes, which can inhibit reproducibility, make the repository cluttered, and make it difficult to re-use models for subsequent projects. You can fork [our meta-repository template](#) as a starting point.

Repository Guidelines

Some guidelines on how to structure your meta-repository:

- Model codes and data should be minted and archived somewhere like [Zenodo](#) (this can be [set up to occur automatically upon a GitHub release](#)).
 - You can link to external models and data, particularly if they have a DOI and/or a version number.

- If you developed a model for your project, keep it in its own repository and release a specific version for a paper. Link this from the meta-repository README.
- Generated data should be archived.
 - If you generated a unique dataset for your project, the code for this should be included as part of the meta-repository for the paper.
 - If you're generating data which will be reused for other projects, it's worth making a repository just for the generation code and linking to a released version of that repository in the meta-repository.
- Avoid hard-coding absolute paths. If you use an external model or data, let the user set a path to point to where they've installed those resources, then use paths relative to them in the rest of the code.
- Make separate folders for different parts of the analysis to improve readability (*e.g.* separate data pre-processing, experiment, and figure-generation codes).
- Make individual scripts for each figure you are generating. If all of your figures require large files to generate (for example, the results of a large-scale simulation), make another script which loads the data and then calls the figure-generating scripts in turn. Remember, the goal is readability!

Commits

Commits are how you register a particular code state with `git`. You can see the status of a repository (what files are untracked and which are changed) using `git status`. From the terminal, you can make commits with

Making Commits

```
1 git add <files to commit> # (1)!
2 git commit -m <commit message>
```

1. This adds the files you would like to associate with a given commit. This does not have to be all of the files you've changed if you've made multiple changes that you would like to split across multiple commits.

In general, good commit guidance is:

- Make commits as often as possible.
- Make sure your commit messages are informative. You should be able to tell what changes you made so you can revert if necessary.

Alternatives

There are several alternatives to GitHub, including [Bitbucket](#). You may use whichever online repository you prefer, though we use GitHub for lab-wide management, such as the website and lab manual.

Learn More

- [Version Control with Git](#) from Software Carpentry
- [Github Training](#) from GitHub.
- [Getting Started: Git and GitHub](#) from Water Programming.
- [A collection of git-related posts](#) from Water Programming.

Last update: June 26, 2023