Version Control Concepts (with Git)

Michael Shyne

Biostatistical Design and Analysis Center
University of Minnesota

October 22, 2018

Why use version control?

There are two main purposes for version control:

- To maintain an historical record of changes to code, documents or data
  - To be able to easily undo changes that led to errors or other problems
  - To be able to return to a version of particular importance (e.g published paper)
- To manage collaboration between a team of developers

Basic concepts of version control

- Version control systems (like Git) maintain **repositories**, stores of documents and their changes over time. Repositories can be local (located on your own computer) or remote (located at a sharable site, like GitHub).
- A version of a document is added to a repository by performing a **commit**. A commit should always be accompanied by a short description of the changes made in this version of the document.
- A **push** will update a remote repository with changes committed to a local repository.
- Important milestones can be marked by **tag**.
- A **branch**, or **fork** is a copy of a repository made a particular point in time. Changes can be committed to a branch without affecting the original (**trunk**).
- A branch can be reintegrated with the trunk by a **merge** or **pull**.

Important git commands

- **`git init`**: Create a local repository
- **`git clone`**: Create a local copy of a remote repository
- **`git add`**: Stage a change, i.e. mark a file to be committed
- **`git commit`**: Add all staged documents to repository
- **`git status`**: Shows the status of changed files, could be untracked, modified or staged.
- **`git push`**: Updates remote repository with commits made to local repository (since last push)
- **`git pull`**: Updates local repository with commits made to remote repository
- **`git merge`**: Combines two branches