Version control with


SciLifeLab

NBIS

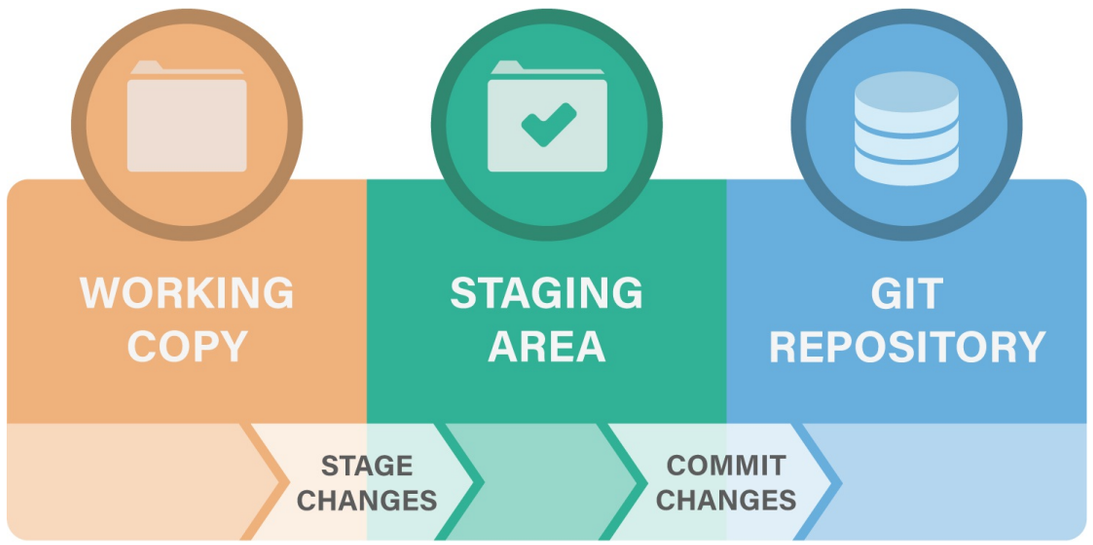
# What is Git?

- A widely used system for distributed version control
- Keeps a complete history the changes you make to your files
- Each point in the history can be re-visited and compared with others
- Git tracks who contributed what to your code
- Git can help you version, backup and share your code and documents
- Kind of like Dropbox, but you decide when each version is saved (plus a lot of more advanced features)
- Git is mainly used for text files, not large or binary files

# Git nomenclature

- A repository is a collection that encompasses all the files and directories of a project
- A commit is a snapshot of a repository's history, i.e. a point in development time
- Development can be separated into branches, allowing for concurrent work on the same repository with simple transitions between functional and work-in-progress code
- Uploading changes to a remote repository is called pushing, while downloading changes is called pulling

# Tracking code in three steps




1. Do some coding (i.e. add or change contents of files)
2. Stage the changes (i.e. specify which changes should be stored)
3. Commit the changes (storing them in the repository's history)

# Git is highly versatile

- Ensures reproducibility of your analyses, regardless whether you've made additional changes to your code after the analysis is run
- Easily fix mistakes by reverting files to previous versions
- Improves your coding by giving you additional structure
- Your code has a backup in your remote repository
- Easily share your code and collaborate with your colleagues

# Questions?

SciLifeLab

NBIS