Earth Sciences
New Zealand

Git Basics Training

Hands-on Workshop on the Fundamental concepts of Git version control and Github

Faye Nielsen, Mario Krapp, Jack Drummond, Quyen Nguyen

Agenda

1. Intro & Key Concepts
2. Git (local) Fundamentals
3. GitHub and Collaboration
4. Discussion & Wrap up

What is Git and Github?

Git is known as a version control system

Version control *is a* **system** *that records changes to a file or set of files over time so that you can recall specific* **versions** *later.*

Github is an online platform to store files in the cloud which enables backed up storage, sharing/collaborating with others, easy integration with other cloud softwares

Why is it useful?

Git is a time machine for your code that creates snapshots of your work, letting you easily revert to previous versions and experiment safely.

Example:

You're analysing earthquake data with analyse_data.py. You add a new visualisation function, but it breaks your entire analysis. What changed?

Without Git: analyse_data_v1.py, analyse_data_v2_final.py etc.

With Git: One file. Instant revert to the last working version with a single command. Essential when working with complex research code or collaborating with others.

How does it work?

Essentially, Git is simply a turbocharged save button.

A save in Git is called a commit. When you make a commit, you take a snapshot of all the files that are in the folder which you have initialised git in – otherwise known as your git repository, or repo for short.

This snapshot saves everything in it's **current state**, and that particular commit can then be accessed further down the line if you mess anything up.

Repository (repo for short)

A folder/directory that is tracked by git (i.e. it has a .git folder)

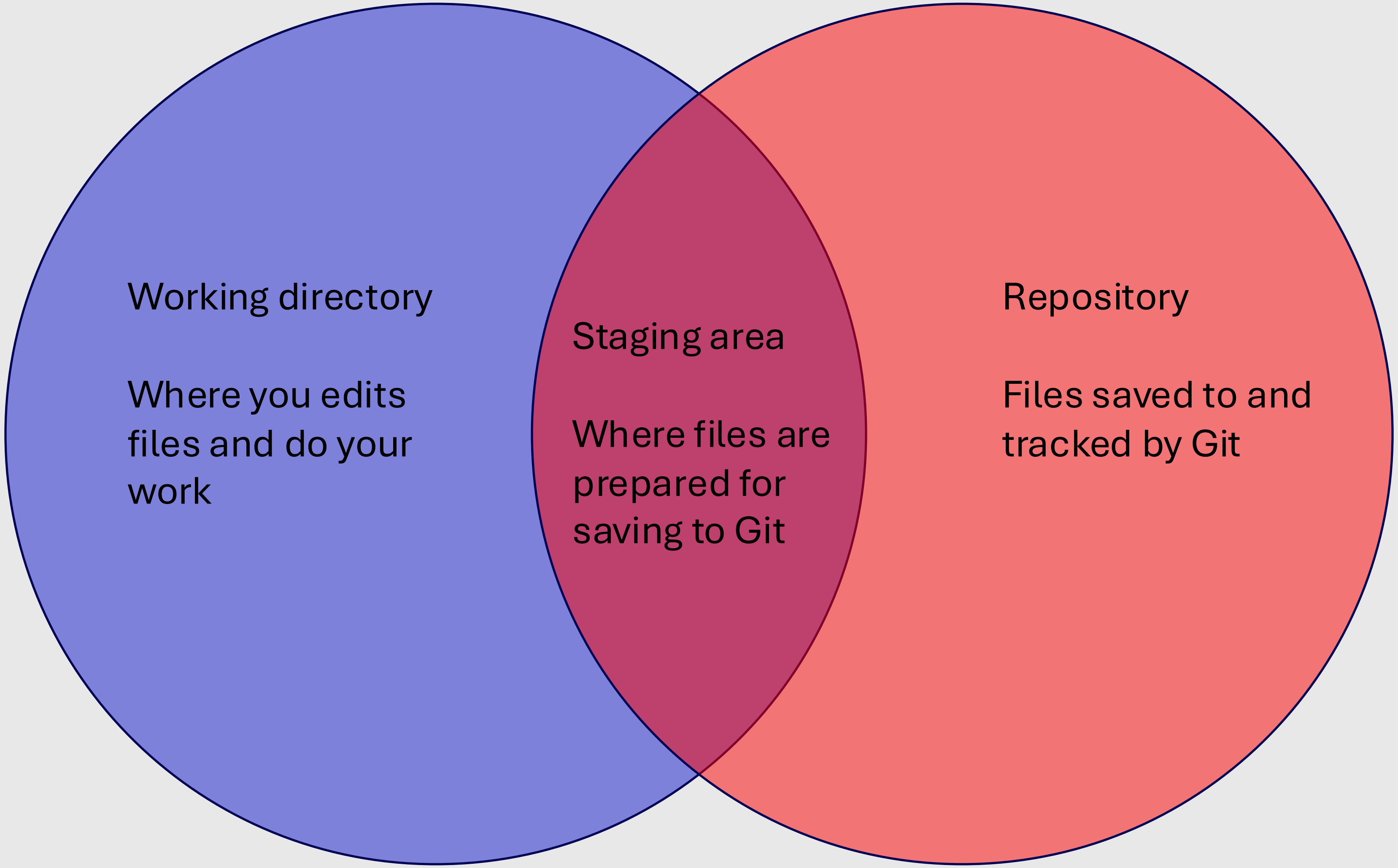
Repository (repo for short)

A folder/directory that is tracked by git (i.e. it has a .git folder)

A local repository is on your local machine e.g. laptop, PC

A remote repository is hosted on a server e.g. Github, Gitlab

The Three States / Areas



Working directory

Where you edits files and do your work

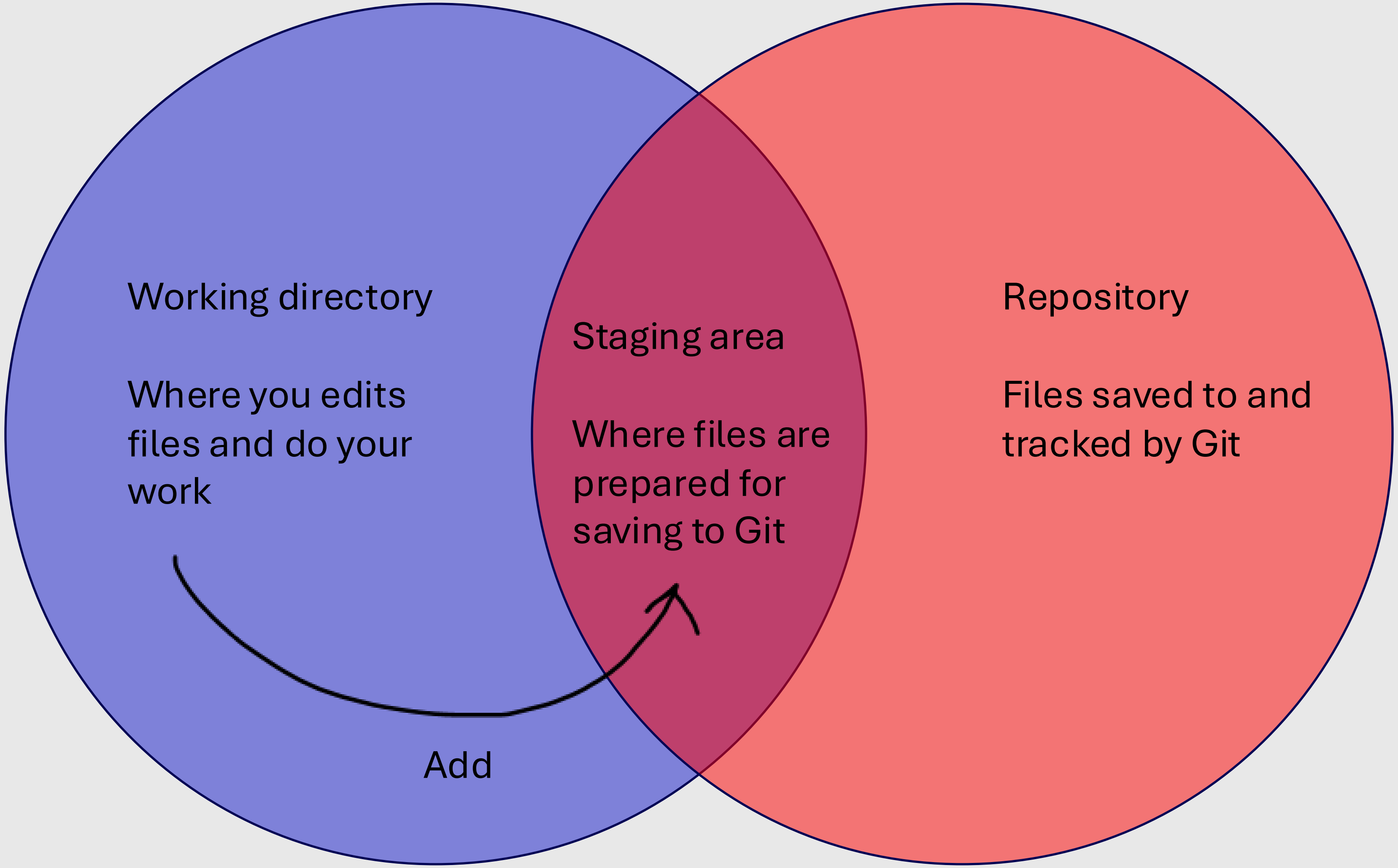
Staging area

Where files are prepared for saving to Git

Repository

Files saved to and tracked by Git

The Three States / Areas



Working directory

Where you edits files and do your work

Staging area

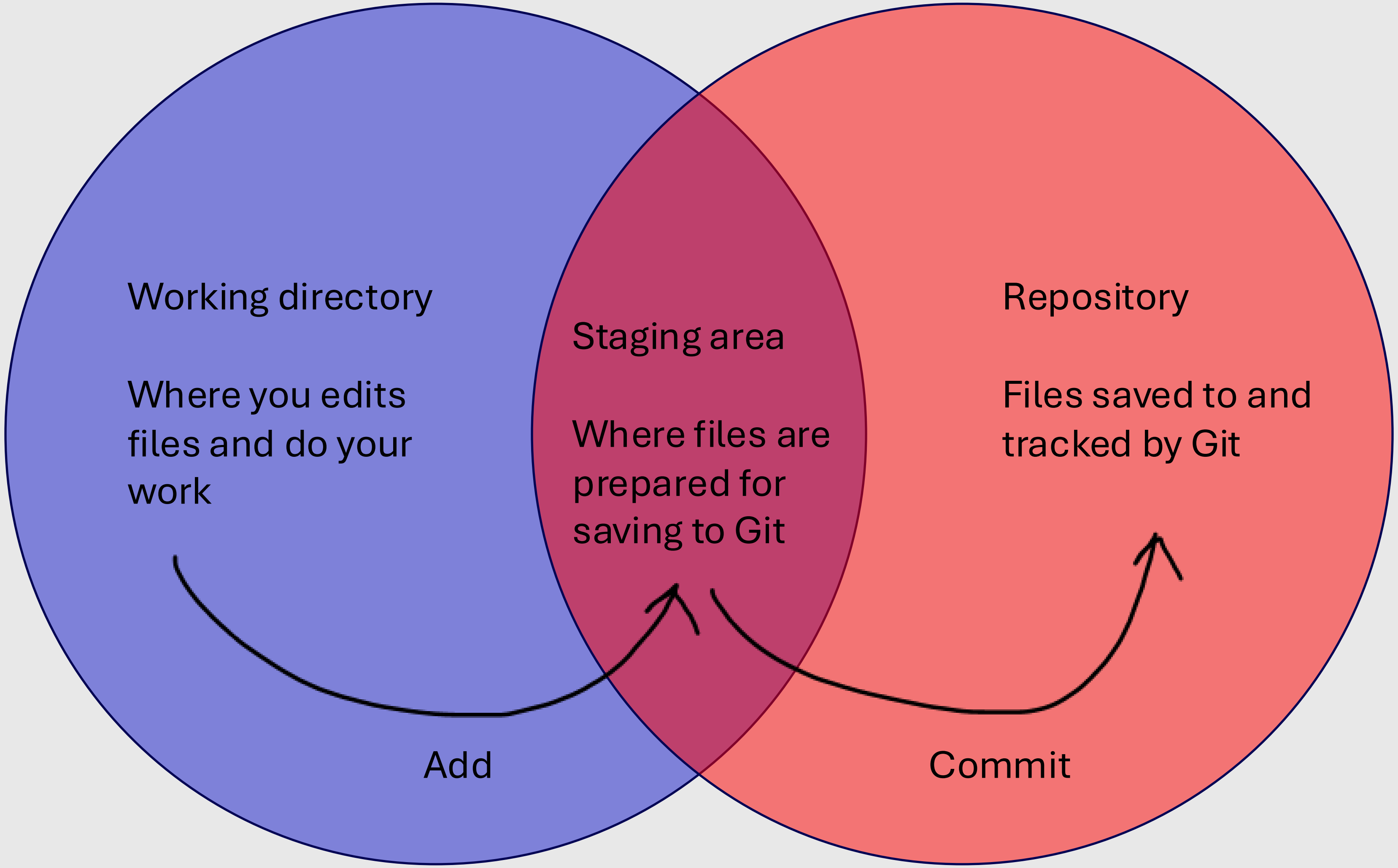
Where files are prepared for saving to Git

Repository

Files saved to and tracked by Git

Add

The Three States / Areas



Working directory

Where you edits files and do your work

Staging area

Where files are prepared for saving to Git

Repository

Files saved to and tracked by Git

Add

Commit

Add

‘Stage’ files

Prepare files to track in Git by selecting which ones you want to save

This is useful as you might only want to save one file, and leave the others out as they are not needed

Commit

Snapshot of all files at that moment in time, including the changes you ‘staged’ by using the ‘add’ command

These are the building blocks of your version control history, when you look back through them you will see what changed between one commit and the next

Has a unique ID, message, author and timestamp

Branch

Parallel version of a project

Enables you to work on different features on different branches, rather than everything all together where it can get confusing

Enables multiple people to work on the same codebase without disrupting/breaking what someone else is working on

Gets merged back into the ‘main’ branch

Merge

Merging branches back into the ‘main’ branch