Introduction to Git and GitHub

Nichole Wespe

NCCR Open Research Data Symposium

Basel | 02 June 2023

Agenda

- Slide presentation explaining Git and GitHub (20 min)
- Workshop setup: install Git, configure Git and GitHub access (10 min)
- Break (5 min)
- Workshop activities (30 min)

My goal for this workshop

- Each participant will understand enough about Git to decide if it is a tool that would benefit their work
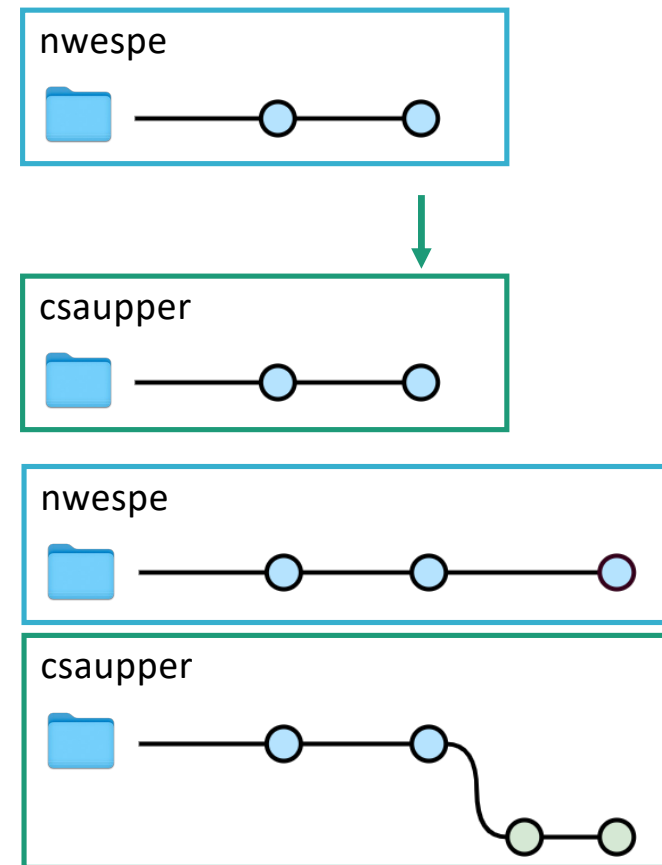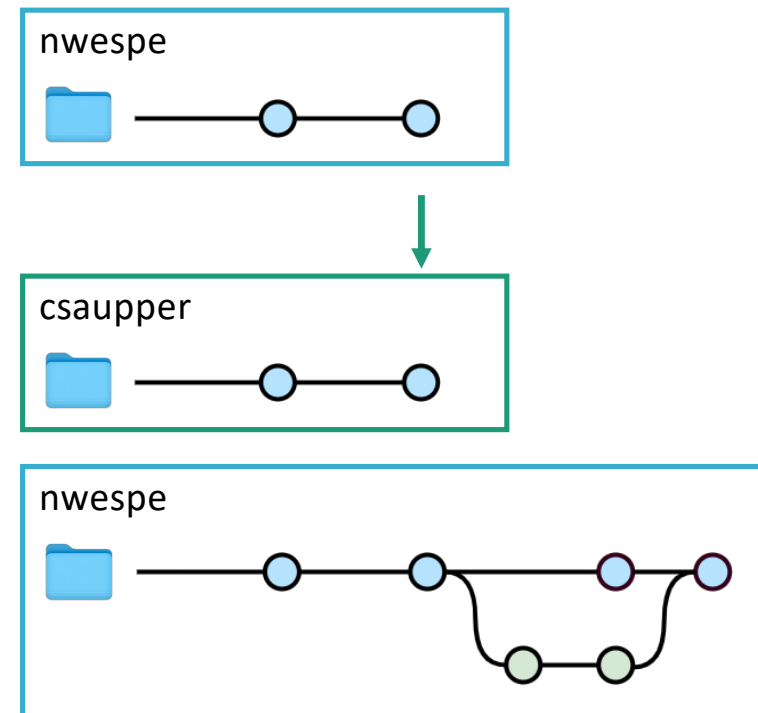
What is Git?

Git is version control software.

- It records a history of changes to a text document.
- A single tracked Git project is called a *repository* and this is set at the level of a directory, not a file*.
- You use Git to create "snapshots" over time of your repository. Each snapshot is called a *commit*.

It is a *distributed* versioning system.

- The full version history is contained in every copy of the repository across different machines.

It has a concept called *branches*.

- Branches are parallel versions of a repository.

What is Git?

Git is version control software.

- It records a history of changes to a text document.
- A single tracked Git project is called a *repository* and this is set at the level of a directory, not a file.
- You use Git to create "snapshots" over time of your repository. Each snapshot is called a *commit.*

It is a *distributed* versioning system.

- The full version history is contained in every copy of the repository across different machines.

It has a concept called *branches*.

- Branches are parallel versions of a repository.
- Git enables you to combine changes made on different branches and to reconcile any conflicts.

What does Git version control look like?

git blame

What does Git version control look like?

git diff

What is Git? Real Talk

With great power comes a steep learning curve.

Git can be a major pain in the you-know-what. Here is my advice:

> learn as you go

> focus on the workflows you need

> use online tutorials

What is GitHub?

GitHub is a website that serves as a collection of Git repositories.
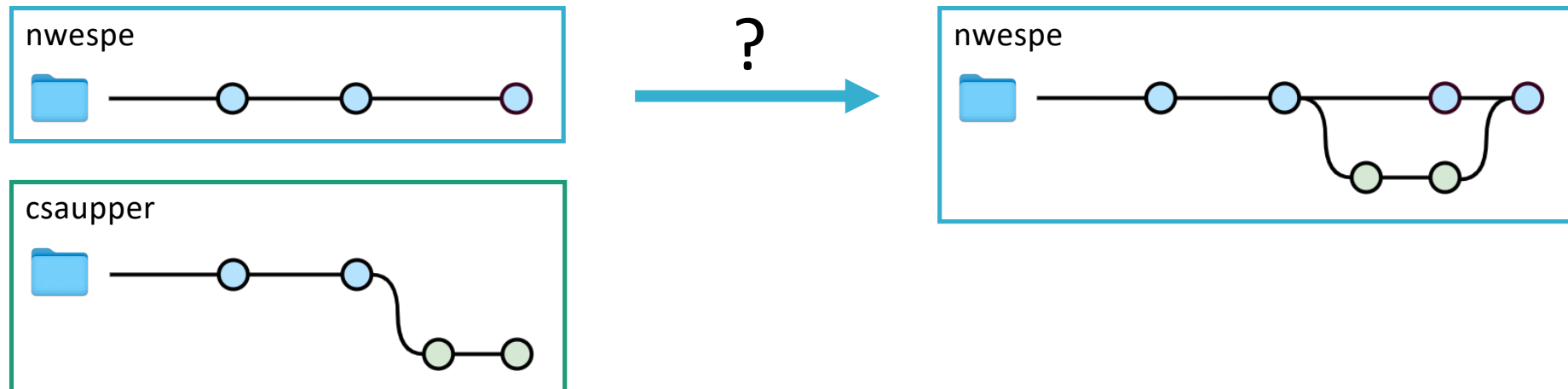
What is GitHub?

GitHub is a website that serves as a collection of Git repositories.
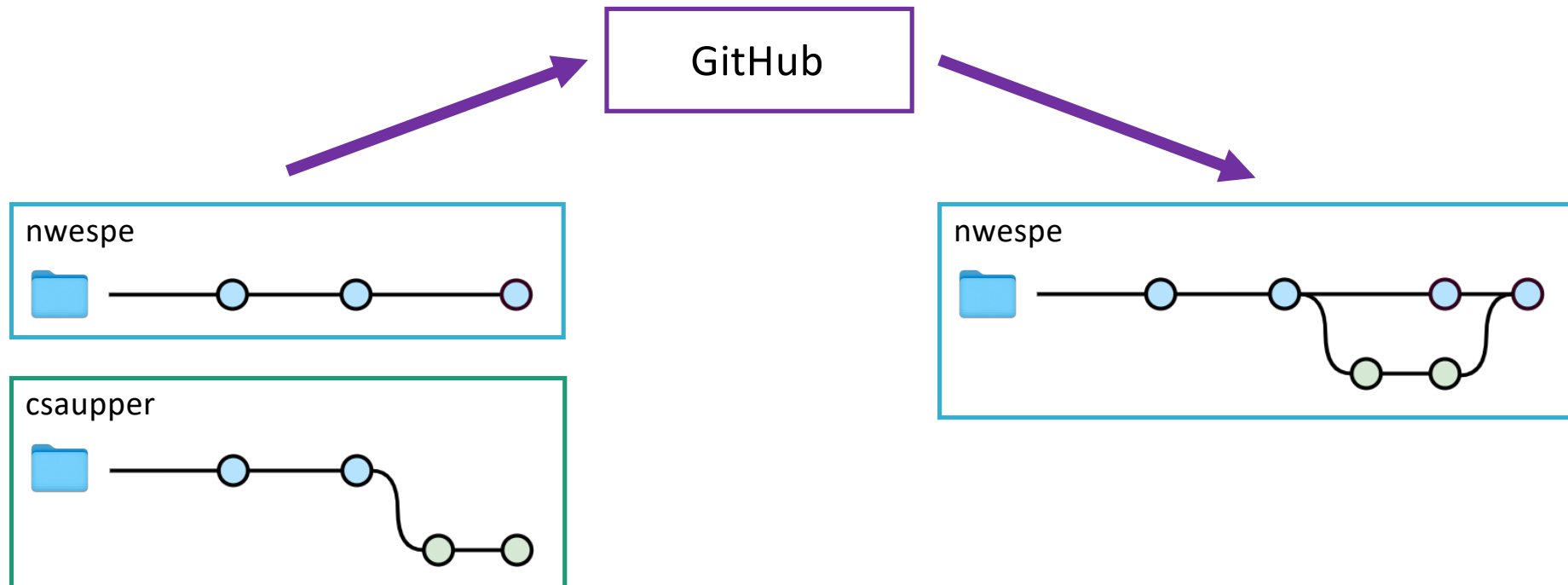
What can I use Git for?

Text documents that have **multiple contributors** and/or **change over time**

- Code
- Manuscripts in progress (in private repo)
- Lab protocols

Why is it useful?

- Version control!
  - ➢ rolling back changes
  - ➢ reconciling conflicts
  - ➢ tracking provenance
- Protection around editing through managed collaboration
- Persistent documentation for changes

What should I NOT use Git and GitHub for?

- passwords or access codes

- large data files (>100 MB not allowed, warning for files >50 MB)

- not necessary for datasets that don't change over time

- not ideal for jupyter notebooks

- not considered a FAIR repository
  - does not assign "a globally unique and persistent identifier"

How do I use Git?

- Command-line interface (CLI) commands
  https://git-scm.com/docs

- Many coding IDEs have Git integration
  e.g., VSCode and PyCharm

- GitHub Desktop and other GUIs

Workshop materials are provided with step-by-step instructions for both CLI and GitHub Desktop
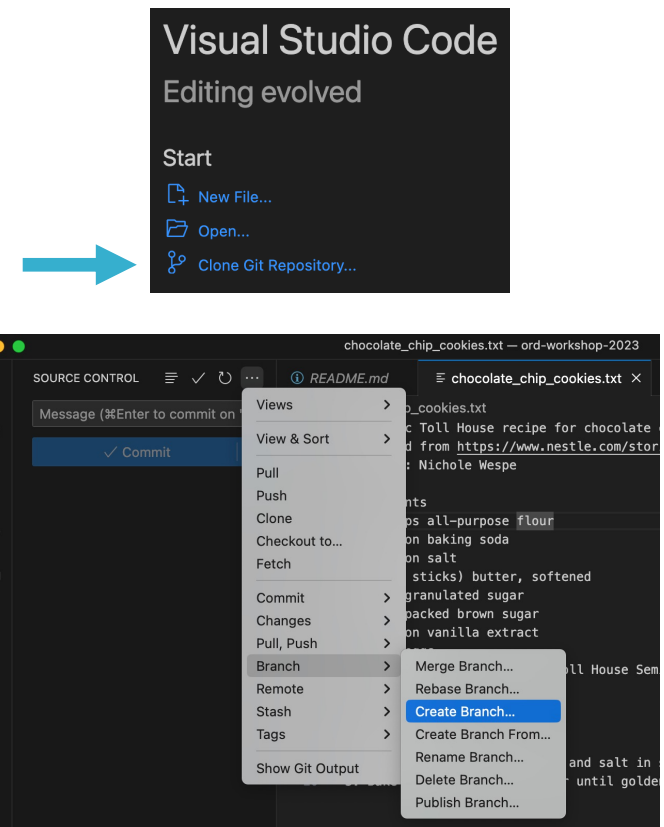
Common Workflows

Getting a repo to work in:

- clone an existing repo for collaboration or personal use
- create a new repo

Working in a repo:

1. create a new branch
2. make some changes (work work work)
3. stage and commit changes
4. merge changes into main branch
   - ➢ Option 1: merge branch into main locally
   - ➢ Option 2: push commit to remote repo and create pull request against remote main branch

	COMMENT	DATE
○	CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
○	ENABLED CONFIG FILE PARSING	9 HOURS AGO
○	MISC BUGFIXES	5 HOURS AGO
○	CODE ADDITIONS/EDITS	4 HOURS AGO
○	MORE CODE	4 HOURS AGO
○	HERE HAVE CODE	4 HOURS AGO
○	AAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFJ	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

Workshop Setup

1. Install Git
2. Create GitHub account
3. Configure Git

Workshop Activity: Recipes

Collaborate on an existing repo

1. clone the workshop repository: nwespe/ord-workshop-2023
2. create a new branch named as your GitHub username
3. make changes: add your recipe file to the folder and add your name to the README.md file
4. stage and commit changes
5. push branch to remote repo (requires being added as a collaborator)
6. create pull request on GitHub
7. merge changes into main branch once PR is approved
8. resolve conflicts (if any)

Workshop Activity: Recipes

Fork an existing repo for personal development

- fork an existing repo (the now-populated recipes repo) to personal account

Workshop Activity: Recipes

Create a new repo

1. create a new git repo locally or through GitHub

2. if local: import to GitHub; if in GitHub: clone locally

3. practice branching, editing, committing and merging