Using GitHub for Collaborative Research

William Brasic

The University of Arizona

Open Science

Open Science

Open Science refers to the practice of making scientific research, data, and dissemination accessible to all levels of society, amateurs, and professionals alike. It promotes transparency, collaboration, and reproducibility in research.

- Emphasizes data sharing and open access publications.
- Encourages reproducibility of results.
- Reduces barriers between academia and the public.

Principles of Open Science

- **Open Data:** Data should be freely available for reanalysis.
- **Open Source:** Code and methods should be publicly accessible.

Advantages for PhD Researchers

- **Collaboration:** Simplifies working with advisors, co-authors, and collaborators, even remotely.
- **Reproducibility:** Facilitates documenting and sharing data, code, and results for transparent research.
- **Efficiency:** Automates version control, saving time during revisions and updates.
- **Visibility:** Increases the accessibility and impact of research through open repositories.

Examples of Use Cases

- Managing data analysis scripts for dissertation work.
- Sharing code and results for journal peer reviews.
- Collaborating on multi-author papers or research projects.
- Publishing supplementary materials for academic papers.

Git

Git

Git is a version control system that allows multiple contributors to collaborate on projects while tracking changes in the codebase.

- Keeps track of changes in files over time.
- Facilitates branching and merging.
- Used widely in software development and research projects.

GitHub

GitHub

GitHub is a web-based platform that uses Git for version control, enabling collaborative development and project hosting.

- Hosts repositories online.
- Provides tools for issue tracking and project management.
- Enables seamless collaboration through pull requests.

Repository

Repository

A repository, or "repo", is a central location where a project's files and version history are stored.

- Can be public or private.
- Includes code, documentation, and other project files.

README.md

README.md

A README.md is a file describing your repository.

- Should be a fairly thorough description
- Uses markdown syntax

Fork

Fork

A fork is a copy of a repository that allows independent development of the project without affecting the original repository.

- Useful for contributing to open-source projects.
- Enables experimentation without altering the main codebase.

Clone

Clone

A clone is a local copy of a repository that is downloaded from GitHub or another hosting platform.

- Allows developers to work offline.
- Keeps the local repository synchronized with the remote one.

Pull Request

Pull Request

A pull request is a request to merge changes from one branch or fork into another, often used for code review and collaboration.

- Facilitates peer review before merging changes.
- Provides a platform for discussion and improvement.

Pull Operation

Pull Operation

A pull operation downloads the changes made my others to your local device.

Commit

Commit

A commit is a snapshot of changes in the codebase, saved to the version control history with a descriptive message.

- You can go back to your work at any previous commit.
- Records the state of files at a specific point in time.
- Requires a meaningful message to describe the changes.
- Forms the building blocks of a repository's version history.

Push Request

Push Request

A push request is the process of uploading local changes from the local repository to a remote repository, ensuring the remote repository is updated.

- Keeps the remote repository in sync with local changes.
- Enables team members to access the latest updates.
- Commonly used after committing changes locally.

GitHub Desktop

GitHub Desktop

GitHub Desktop is a user-friendly graphical interface for Git and GitHub, simplifying repository management without requiring command-line usage.

- Allows visual management of commits, branches, and pull requests.
- Ideal for beginners learning Git and GitHub.
- Synchronizes local and remote repositories seamlessly.
- Compatible with both macOS and Windows.

Advantages of GitHub Desktop

- No need for command-line expertise.
- Visualize changes with a side-by-side diff view.
- Simplifies staging and committing changes.
- Easy management of multiple branches and repositories.

Question 1: Pull Request vs Pull Operation

What is a Pull Request?

Isn't a pull request just pulling changes from Git that other contributors could have made?

Question 1: Pull Request vs Pull Operation

What is a Pull Request?

Isn't a pull request just pulling changes from Git that other contributors could have made?

Pull Request vs Pull Operation

No, a pull request is a request to merge changes from one branch or fork into another. It is used for code review and collaboration, not for pulling changes into your local repository. To pull changes made by others into your local repository, you use the 'git pull' operation.

Question 2: What is a Branch?

What is a Branch?

What is a branch, and why would I use it?

Question 2: What is a Branch?

What is a Branch?

What is a branch, and why would I use it?

Definition and Use of a Branch

A branch is a parallel version of your codebase that allows you to work independently on a specific feature or fix. It prevents incomplete or experimental work from affecting the main branch. You use branches to isolate changes, collaborate, and test work before merging it into the main branch.

Question 3: Committing vs Saving

Saving vs Committing

When I save my script, doesn't that save changes locally?

Question 3: Committing vs Saving

Saving vs Committing

When I save my script, doesn't that save changes locally?

Difference Between Saving and Committing

Saving a script only updates the file on your local machine; it does not add those changes to Git's version history. Committing in Git records the changes as a snapshot in the repository, along with a descriptive message, making it part of the version history you can go back to at any time.

Question 4: Commit Visibility

Visibility of Commits

Can everyone see commits in their history tab?

Question 4: Commit Visibility

Visibility of Commits

Can everyone see commits in their history tab?

Who Can See Commits

Only you can see commits in your local history until they are pushed to the remote repository. Once pushed, all collaborators with access to the repository can see the commits in their remote history tab.

Question 5: Commit vs. Push

When to Commit but Not Push

When would you commit changes locally, but not push them to the remote repository?

Question 5: Commit vs. Push

When to Commit but Not Push

When would you commit changes locally, but not push them to the remote repository?

Reasons to Commit Locally Only

- Incomplete Work: Save progress locally without sharing unfinished changes.
- Offline Work: No internet access but still need to track progress.
- Experimental Changes: Test ideas without affecting the shared repository.

Benefits of Using GitHub

- **Version Control:** Tracks changes, enabling reversion to earlier versions.
- **Collaboration:** Allows multiple contributors to work simultaneously.
- **Transparency:** Encourages open science by sharing code and data.
- **Documentation:** Provides tools to create README files for documentation.

Best Practices

- Write a thorough README.md file.
- Write meaningful commit messages.
- Regularly synchronize local and remote repositories.

GitHub Workshop!