git/git-annex/DataLad/Forgejo-aneksajo: a pragmatic data collaboration ecosystem

Matthias Rißе

Version Control for Research Data Management

What does reproducible research mean in practice?

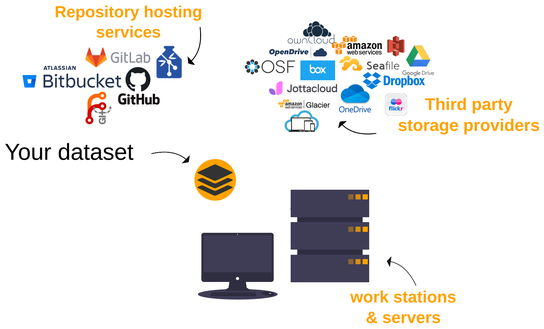
Every publication should have an accompanying publishable artifact that fully describes the data, code and computational environment used to produce it. Anyone (with sufficient determination) should be able to go from e.g. a plot in a paper back through the processes used to produce it, be able to follow any intermediate steps, and arrive at the data, or even the data acquisition process.

Personal opinion: this artifact should be a natural by-product of the work process, otherwise it won't exist / be complete / be up-to-date.

So you have to be able to answer a lot of questions:

- What data did you use?
- Where did it come from?
- What kind of processing did you apply to generate a given result?
- Did you keep your results up-to-date when you updated the input data?
- Did you keep your results up-to-date when you updated your processing code?
- What's the computational environment that was used, and how can it be reproduced?
- ... did you check that an update to your computational environment did not change your results, or did you update your results with the new environment?

This list can be continued for a long time.

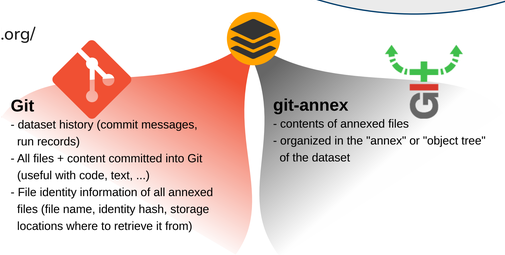


And what does DataLad bring to the mix?

DataLad improves features important to RDM:
- Better nested repository support for re-use of data
- Automatic provenance tracking of program execution
- Streamlined common workflows
- Best practices for data analyses (YODA principles) and how to handle many TiB's of data
- Really good documentation: https://handbook.datalad.org/

The "Trinity" of distributed version control for data

> git annex



Instead you might end up with something like this:

```
├── code/
│   ├── code_final/
│   │   ├── final_2/
│   │   │   ├── main_script_fixed.py
│   │   │   └──takethisscriptformostthingsnow.py
│   │   ├── utils_new.py
│   │   ├── main_script.py
│   │   ├── utils_new.py
│   │   ├── utils_2.py
│   │   └── main_analysis_newparameters.py
│   └── main_script_DONTUSE.py
├── data/
│   ├── data_updated/
│   │   └── dataset1/
│   │       └── datafile_a
│   ├── dataset1/
│   │   └── datafile_a
│   ├── outputs/
│   │   ├── figures/
│   │   │   ├── figures_new.py
│   │   │   └── figures_final_forreal.py
│   │   ├── important_results/
│   │   ├── random_results_file.tsv
│   │   ├── results_for_paper/
│   │   ├── results_for_paper_revised/
│   │   └── results_new_data/
│   ├── random_results_file.tsv
│   ├── random_results_file_v2.tsv
[...]
```

It doesn't have to be that way!

In the software world these issues have been solved:
- Use a version control system to keep track of changes
- Git has become the de-facto standard
  - Your main branch contains your canonical development
  - Other branches can contain experimental changes
  - A "final" version might just be a tagged revision
  - ...

=> The repository at some revision is an artifact containing all history of how this state came to be

... but git can't deal with large files, so how do we version control data?

Git-annex extends git with large file support by using git as an index into local or remote storage spaces, providing a standardized interface to access data across different sources.

Collaboration

Data curation and analysis in a team has some challenges...

- Keeping track of the evolution of files - who changed what, when, how, and why
- Organizing collaboration with multiple people - who works on what, discussions on the best approaches
- Keeping track of what needs to be done - known issues and plans for the future
- Work asynchronously in different directions and consolidate the results

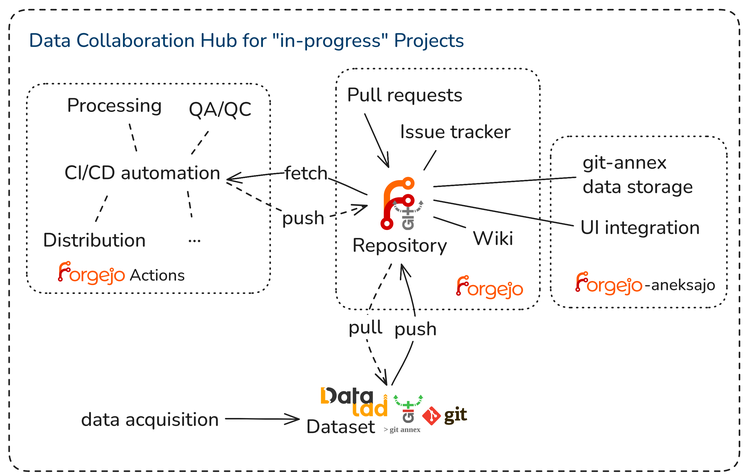
... which are the same encountered in software development.

And there are tools to assist with this kind of collaboration:
- Distributed Version Control Systems
- Issue Trackers
- Change proposals
- Versioning and Releases
- CI Systems for automation
- Wikis

Forgejo is an established and popular free software project implementing a "git forge" that provides these tools (and more).

Forgejo-aneksajo is a soft-fork that extends Forgejo with git-annex support, making it a seamless git-annex/DataLad forge for collaboration on data projects.

Try it out at: https://demo.forgejo-aneksajo.org/



DataHub
Research Field Earth and Environment