Reproducible Research Workshop (Part 2)

Jupyter, Snakemake and Git

Önder Kartal

Table of Contents

- Making Interactive Documents with **Jupyter Notebooks**
- Managing Workflows with **Snakemake**
- Tracking your Content with **Git**

jupyter

Jupyter Notebooks, http://jupyter.org/

The principal goal of scientific publications is to teach new concepts, show the resulting implications of those concepts in an illustration, and provide enough detail that the work is reproducible. In real life reproducibility is haphazard and variable. We rarely see a seismology PhD thesis being redone at a later date by another person.

The reproducibility problem can be largely overcome by standardized software generally available that is not hard to use. A new form of documentation is coming into existence. We call it an ***active document*** *(a-doc). Active documents will serve us far better than paper documents. In an a-doc the author provides programs and a command script for every figure. Readers, students, and customers, can thus verify the calculation and adapt it to new circumstances without laboriously recreating the author's environment.*

Jon Claerbout, 1967

Advantages

- Single document with "live code, equations, visualizations and explanatory text."
- Exploratory analysis
- Documenting your workflow/learning
- Sharing notebooks
- Over 40 programming languages supported

Disadvantages

- Not so good if you want to re-use parts (copy-paste)
- Text editing sub-optimal
- Not good for automatization and generalization of workflows (reusability)

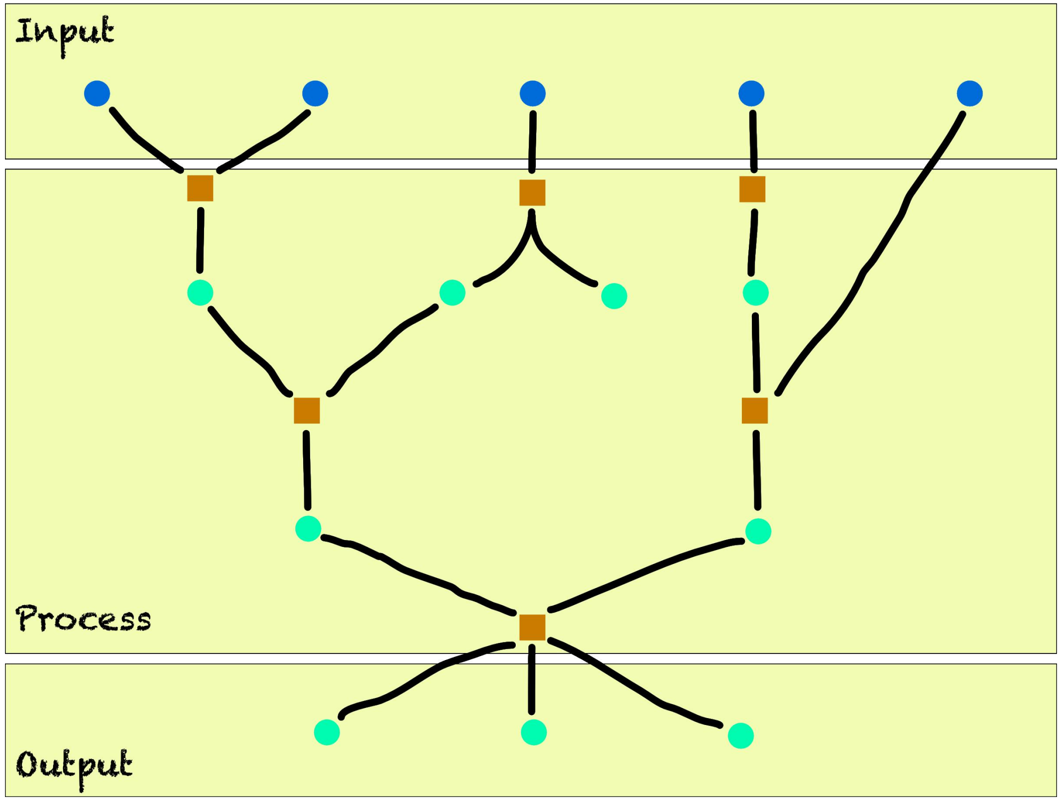
Snakemake

Snakemake workflows, https://snakemake.readthedocs.io/en/stable/

● Raw Data

■ Rule

● Processed Data



Advantages

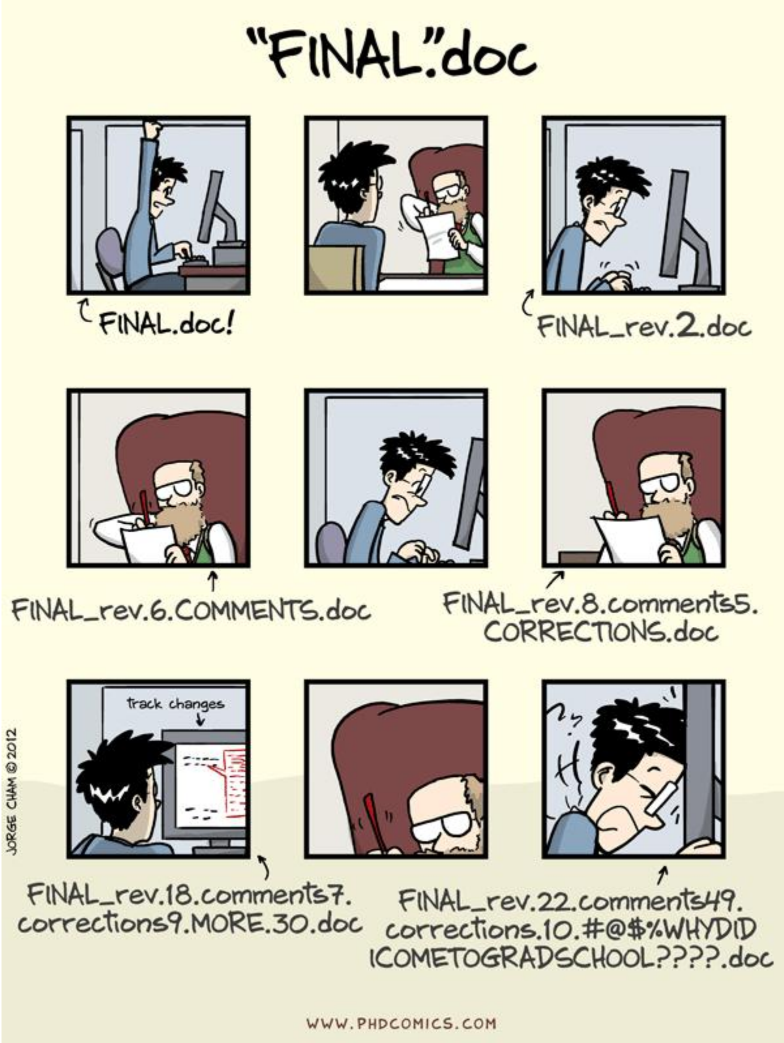
- Automating the creation of results (software, figures, papers, etc.) by running a set of *rules*
- Declarative syntax (rules, inputs, outputs)
- Dependency tracking: checks which rules need to be re-run if some input data has changed
- Efficiently reproduce all results with single command

Disadvantages

- Learning curve
- Needs more time to set up; this is set off by the time you save if you have to re-run often

git

Git - the stupid content tracker, https://git-scm.com/


"FINAL".doc
FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.CORRECTIONS.doc
track changes
FINAL_rev.18.comments7.corrections9.MORE.30.doc
FINAL_rev.22.comments49.corrections.10.#@$%WHYDID ICOMETOGRADSCHOOL????.doc
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

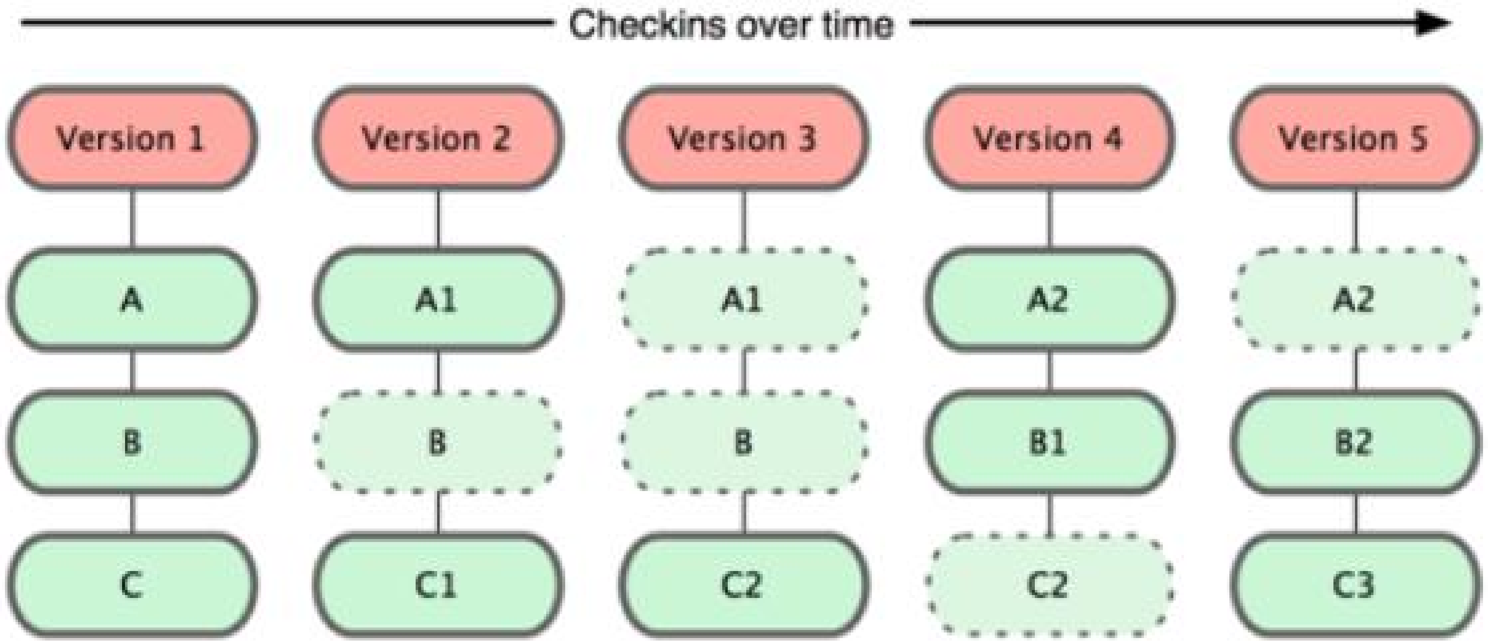
Advantages

- Track changes on steroids! Has become indispensable for open-source projects (Github!)
- Build a history (“log”) of your whole project
- Go to any state of the project repository non-destructively
- Check differences made to each file
- Remote Workflow/Collaboration (not today!)
  - Each collaborator has a full local copy
  - Work in parallel on different *branches*

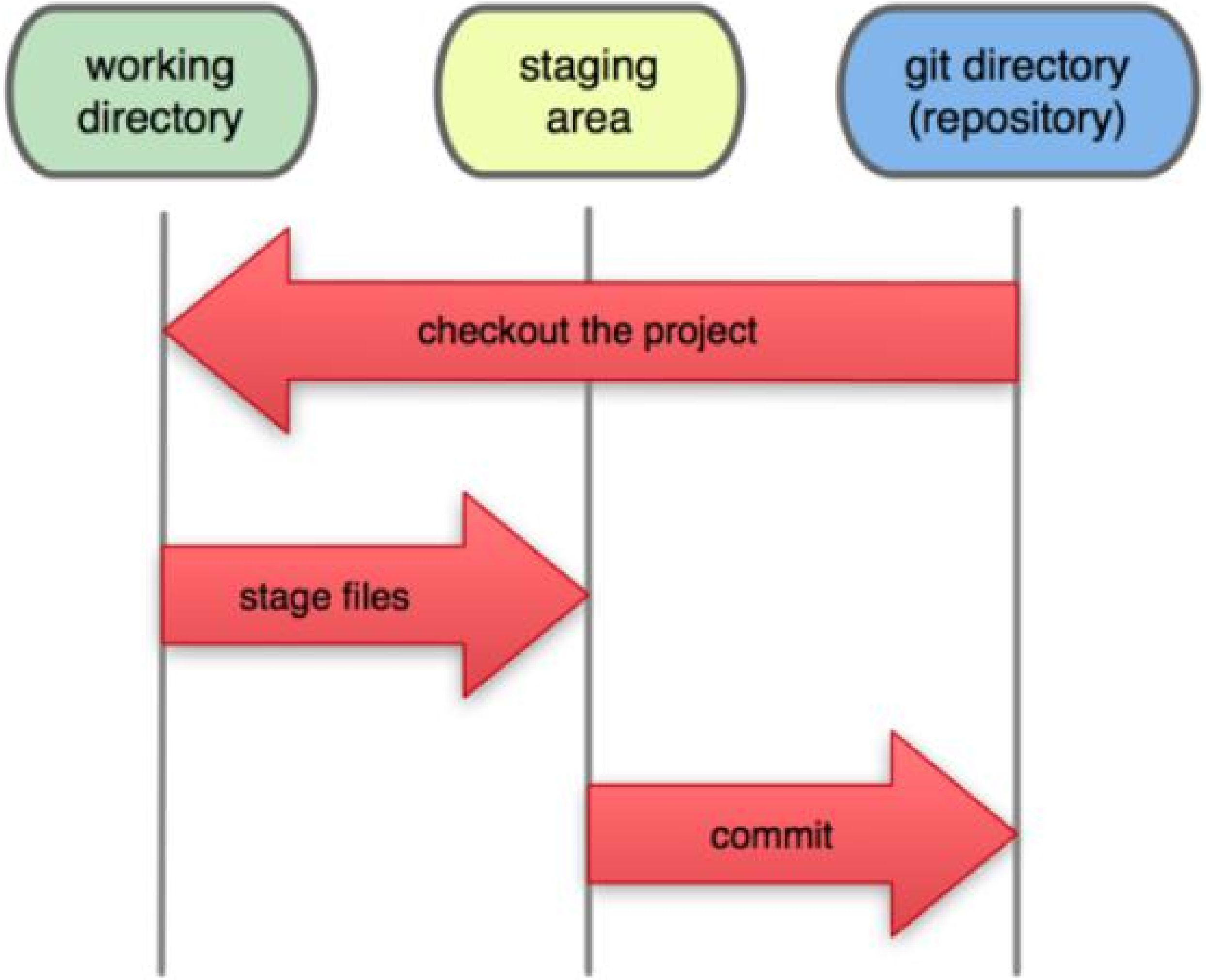
Disadvantages

- Not recommended for binary/big files
- Complicated when merge conflicts occur (for remote operations)

Figure 1.5: Git stores data as snapshots of the project over time.



Local Operations


working directory
staging area
git directory (repository)
checkout the project
stage files
commit