
Versioning of data and code using Git

Introduction to Data Management Practices course

NBIS DM Team

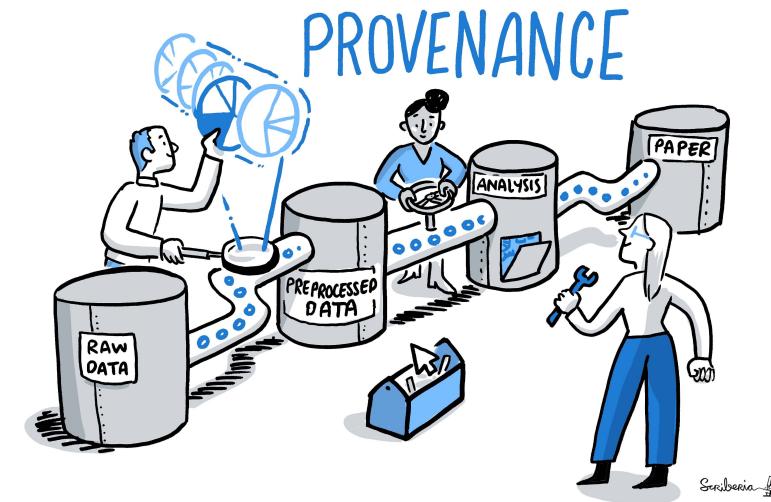
data@nbis.se

<https://nbisweden.github.io/module-versioning-dm-practices/>



Outline

1. The fundamentals of versioning
2. Distributed versioning on the web using GitHub.com
3. Local versioning on your computer using GitHub Desktop
4. Collaborative versioning using GitHub.com and GitHub Desktop



- Data has a life cycle

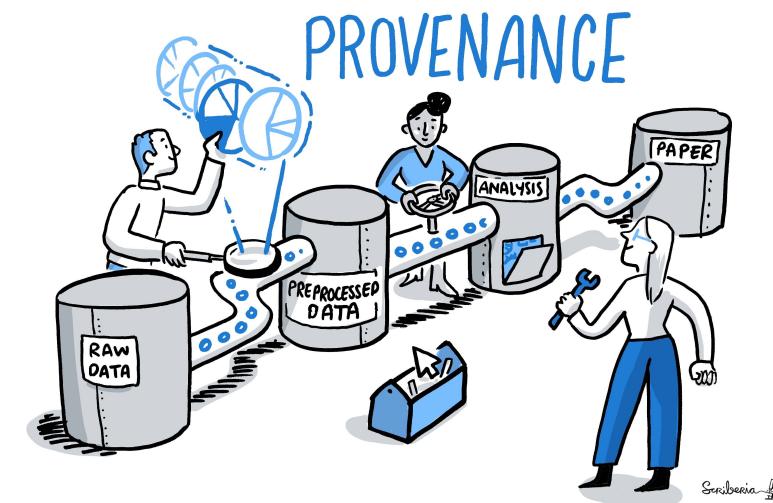
Raw (experiment) data – produce, collect, license, get access, ...

Processed – generate, clean, aggregate, label, transform, analyse, ...

Archived – document, select, convert, package, submit, ...

Published – FAIRify, promote reuse, ...

- Maintain data integrity and authenticity
- Plan a storage strategy
- Plan a backup and disaster recovery strategy



- **Keep original (raw) versions** of data files, or keep documentation that allows the reconstruction of original files
- **Track the location of files** if they are stored in a variety of locations
- Establish **terms and conditions of data use** within the project team and beyond
- Keep a ‘master file’ of the data and take measures to preserve its authenticity
- Decide **how many and which versions to keep** for how long
- **Document changes** that were made in any version
- **Record relationships between items** where needed, for example between code and the data file it is run against

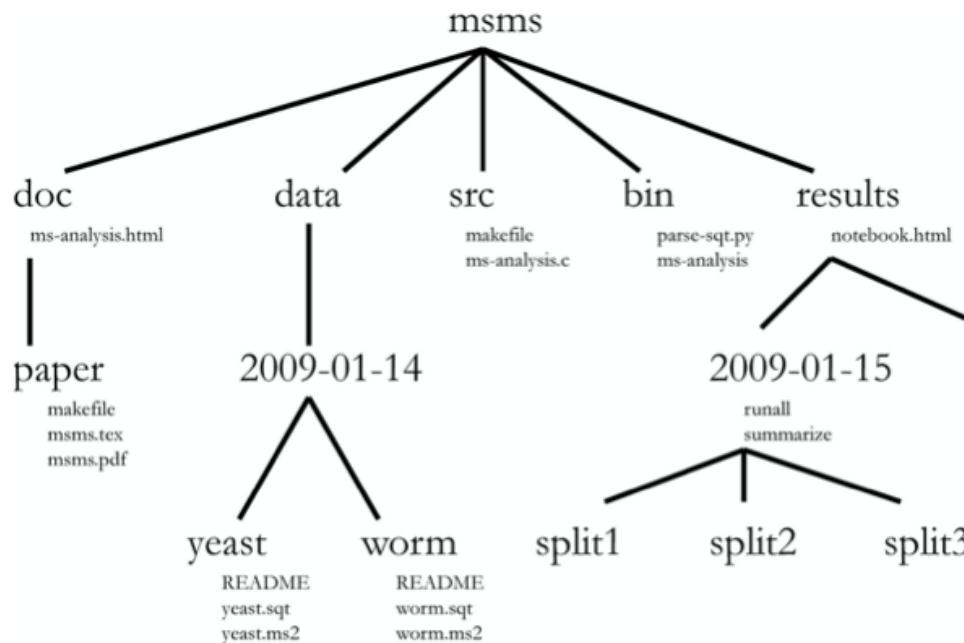
Local storage

- When working on different (local) workstations, e.g. laptop at home and the desktop in the office:
 - **always make sure that you are working on the most current version, for example with the help of versioning software or guidelines**
 - make sure that the most **current version is always backed up somewhere else**
- Only suitable as a primary storage for projects involving very few people
- Avoid if data will be moved back and forth between personal computers frequently

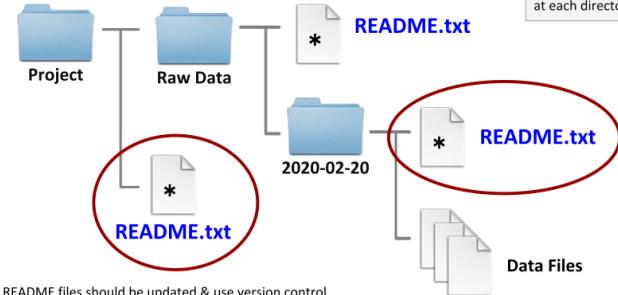
Cloud storage

- Granting **shared, remote and easy access to data and other files** to all involved in the project
 - **Read the terms of service.**
Especially focus on rights to use content given to the service provider
 - **Opt for European, national, or institutional** cloud services which store data in Europe if possible
 - **Not your only storage and backup solution**
 - **Not for unencrypted (sensitive) personal data**
- Also be careful with passwords and other secrets!*

Organising files and folders



Create Documentation Files



Example of a project folder with README documentation files at each directory level

Folder structure, docs and naming conventions are important!

- Snapshot projects and files

Infinite undo, traceability, reproducibility
Software, data, documents, scripts,
manuscripts, ...

- Copy–rename–describe

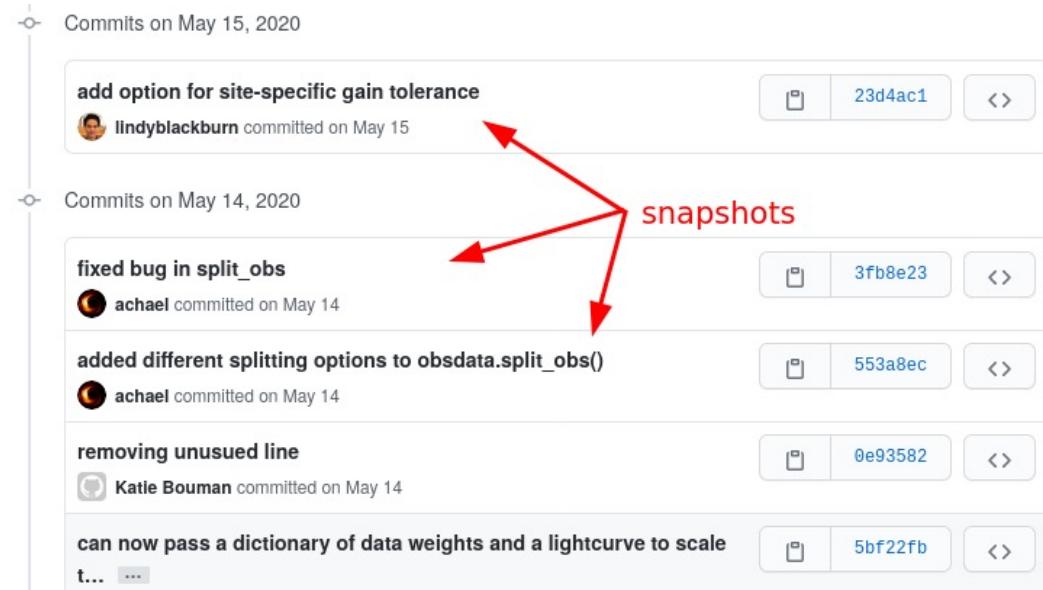
Duplicate / rename files and folders
Changes.txt

- Software assisted

Projectplace, Google Docs, Sharepoint,
Dropbox, but there is more...

- Collaborative versioning

On your computer, on the web, from a
single line of text to a complete project



Git is free and widely used

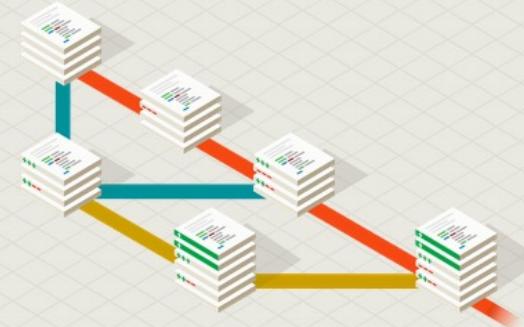


git --distributed-even-if-your-workflow-isnt

Search entire site...

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.



About

The advantages of Git compared to other source control systems.



Documentation

Command reference pages, Pro Git book content, videos and other material.



Downloads

GUI clients and binary releases for all major platforms.



Community

Get involved! Bug reporting, mailing list, chat, development and more.



<https://git-scm.com/>

Git is just one component

 PHD Comics
@PHDcomics

Follow

Final.doc phdcomics.com/comics.php?f=1...



FINAL.doc!
FINAL_rev.2.doc
FINAL_rev.6.COMMENTS.doc
FINAL_rev.8.comments5.CORRECTIONS.doc
FINAL_rev.18.comments7

10:35 AM - 1 Feb 2017

1,197 Retweets 1,750 Likes

39 1.2K 1.8K

 Nicola Gigante
@gignico

Follow

Replying to @PHDcomics

Solution: #LaTeX instead of Word, #Git to handle revisions, @github issues to handle comments from supervisor.

10:42 AM - 1 Feb 2017

5 Retweets 12 Likes

2 5 12

 yaniv brandvain @yanivbrandvain · 1 Feb 2017

Replying to @gignico @PHDcomics @github

commit -m 'another final version'

1 1 1

<https://twitter.com/phdcomics/status/826861642507882496>

<https://twitter.com/phdcomics/status/826861642507882496>

Versioning on the web

- GitHub, GitLab, Bitbucket, etc...
- Initialize a **repository** for storing versions of your files
- Create and **commit** a new file
- Edit and **commit** a changed file
- Make a new **branch** of history

Commit new file

saving first draft of my recipe 

Add an optional extended description...

Commit directly to the `master` branch.

Create a **new branch** for this commit and start a pull request. [Learn more about pull requests.](#)

Commit new file **Cancel**

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner *



bast

Repository name *



/ recipe 



Great repository names are short and memorable. Need inspiration? How about [solid-couscous](#)?

Description (optional)

A collection of my cooking recipes.



Public 

Anyone on the internet can see this repository. You choose who can commit.



Private 

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

Initialize this repository with a README

This will let you immediately clone the repository to your computer. 

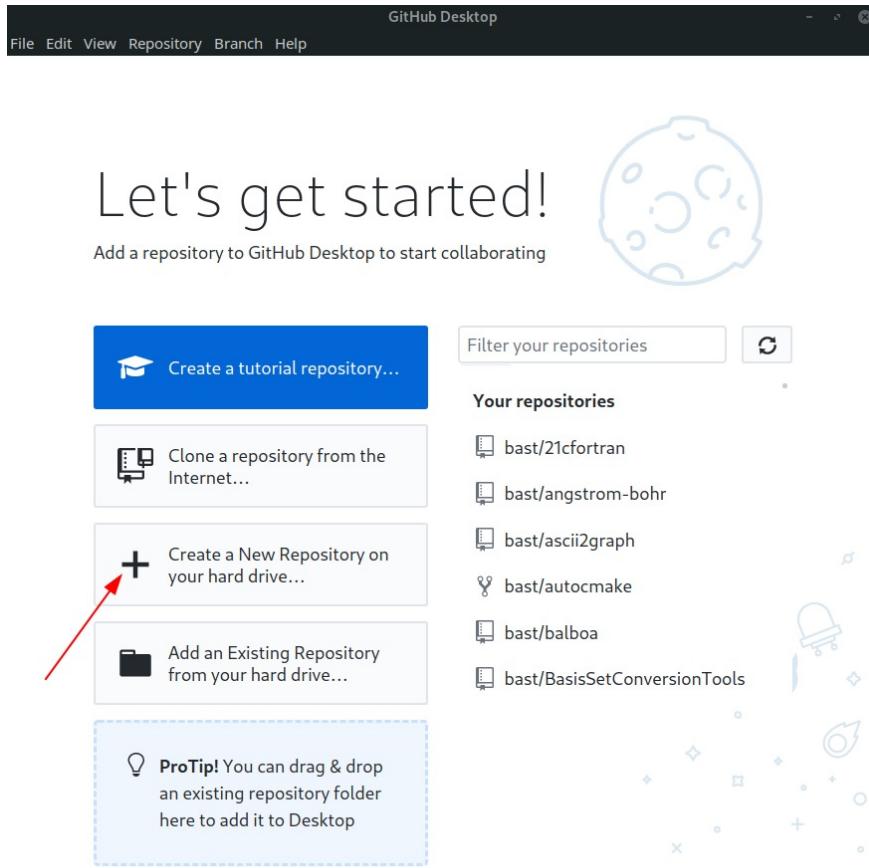
Add .gitignore: None ▾

Add a license: Creative Commons ... ▾

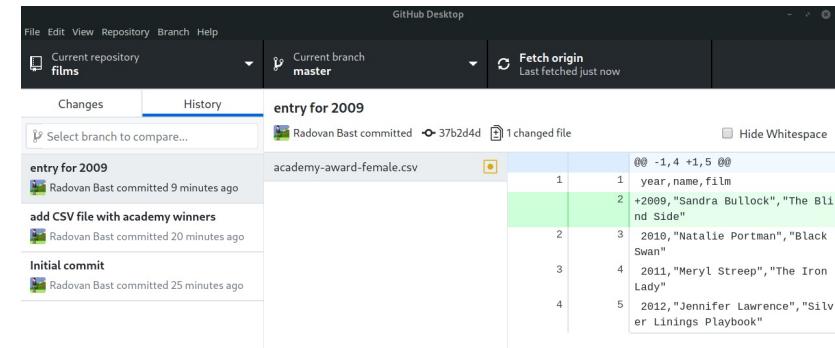


Create repository

Versioning on a local PC



- Git, **GitHub Desktop**, Tower, ...
- **Initialize** a repository for storing versions on your own computer
- **Add** files from your working directory to the **staging area**
- **Push** changes to the web
- **Pull** changes from the web



The screenshot shows the GitHub Desktop application with the "History" tab selected. The current repository is "films" and the current branch is "master". The fetch origin was last fetched just now. The commit log for the "entry for 2009" branch is displayed:

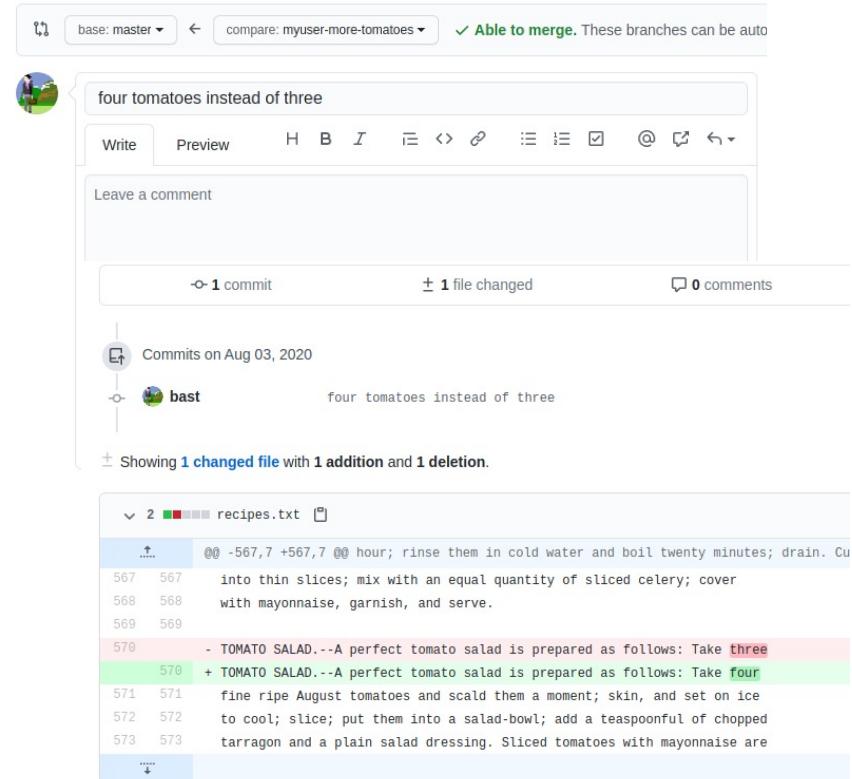
Commit	Author	Date	Message
1	Radovan Bast	9 minutes ago	academy-award-female.csv
2	Radovan Bast	20 minutes ago	+2009, "Sandra Bullock", "The Blind Side"
3	Radovan Bast	25 minutes ago	2010, "Natalie Portman", "Black Swan"
4	Radovan Bast	25 minutes ago	2011, "Meryl Streep", "The Iron Lady"
5	Radovan Bast	25 minutes ago	2012, "Jennifer Lawrence", "Silver Linings Playbook"

Collaborative versioning

- Curate a “**master**” branch of your project’s version history
- **Request** changes from one branch to be pulled into another on the web
Merge changes from one branch into another
- **Resolve** conflicting changes

Open a pull request

The change you just made was written to a new branch named `myuser-more-tomatoes`. Create a pull request below.



The screenshot shows a pull request interface with the following details:

- Base:** master
- Compare:** myuser-more-tomatoes
- Status:** Able to merge. These branches can be auto-merged.
- Commit Message:** four tomatoes instead of three
- Author:** bast
- Date:** Commits on Aug 03, 2020
- Changes:** 1 commit, 1 file changed, 0 comments
- File Diff:** recipes.txt
 - Line 567: -567,7 +567,7 @ hour; rinse them in cold water and boil twenty minutes; drain. Cut into thin slices; mix with an equal quantity of sliced celery; cover with mayonnaise, garnish, and serve.
 - Line 570: -TOMATO SALAD--A perfect tomato salad is prepared as follows: Take **three** +TOMATO SALAD--A perfect tomato salad is prepared as follows: Take **four**
 - Line 571: fine ripe August tomatoes and scald them a moment; skin, and set on ice
 - Line 572: to cool; slice; put them into a salad-bowl; add a teaspoonful of chopped
 - Line 573: tarragon and a plain salad dressing. Sliced tomatoes with mayonnaise are

- Not a replacement for back-ups
- Write informative commit messages
- Establish conventions for using and naming branches
- Best out-of-the box experience with text-based files (lines)
- Use tags / hashes to reference a specific revision
- Use data archives to preserve important revisions, e.g., Zenodo or Figshare

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
AAAAAAA	3 HOURS AGO
ADKFJSLKDFJSDKLJFJ	3 HOURS AGO
MY HANDS ARE TYPING WORDS	2 HOURS AGO
HAAAAAAAAANDS	2 HOURS AGO

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

Git Commit by xkcd CC-BY-NC 2.5, <https://xkcd.com/1296/>