
Versioning of data and code using Git

Introduction to Data Management Practices course

NBIS DM Team

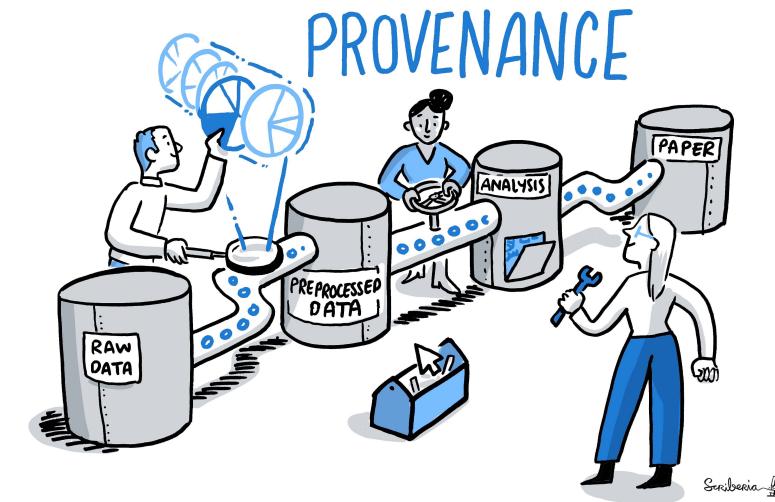
data@nbis.se

<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

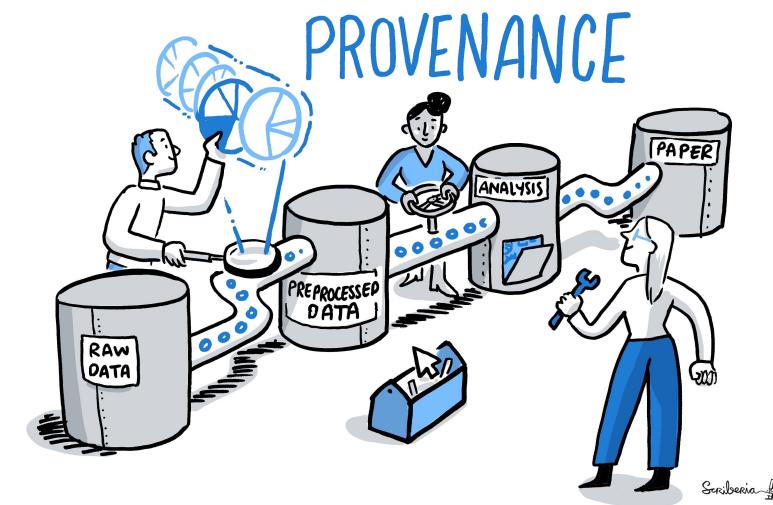
Raw (experiment) data – produce, collect, license, get access, ...

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

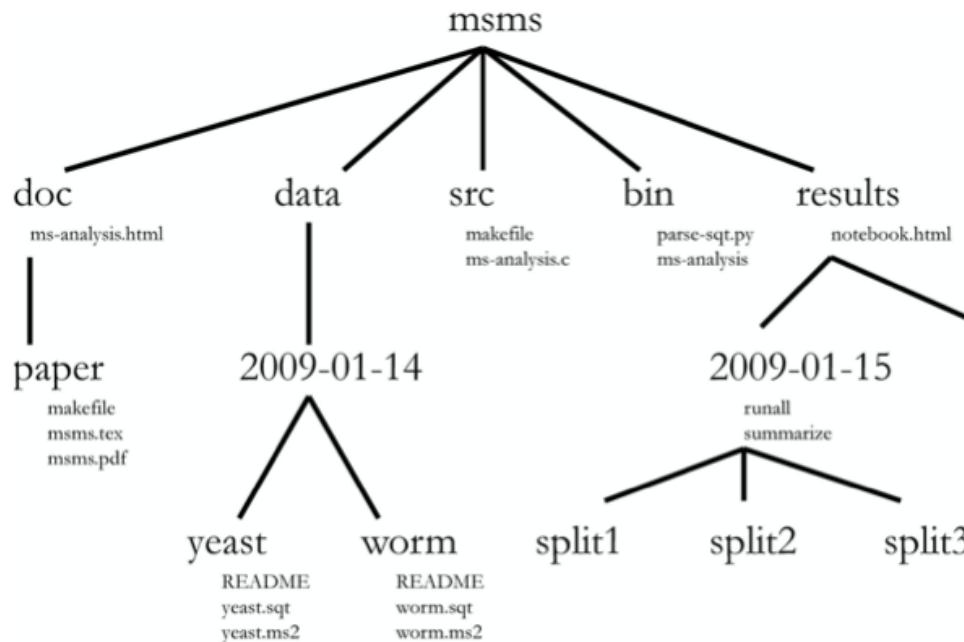


- **Keep original (raw) versions** of data files, or keep documentation that allows the reconstruction of original files
- **Track the location of files** if they are stored in a variety of locations
- Establish **terms and conditions of data use** within the project team and beyond
- Keep a ‘master 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

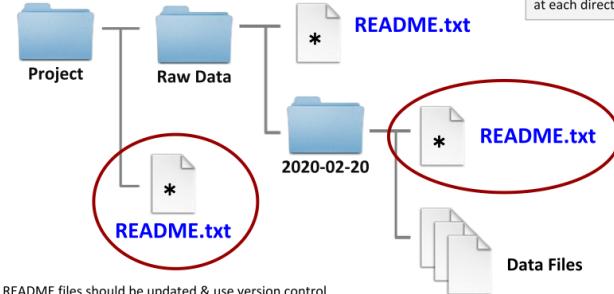
- When working on different (local) workstations, e.g. laptop at home and the desktop in the office:
 - **always make sure that you are working on the most current version, for example with the help of versioning software or guidelines**
 - make sure that the most **current version is always backed up 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

- 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

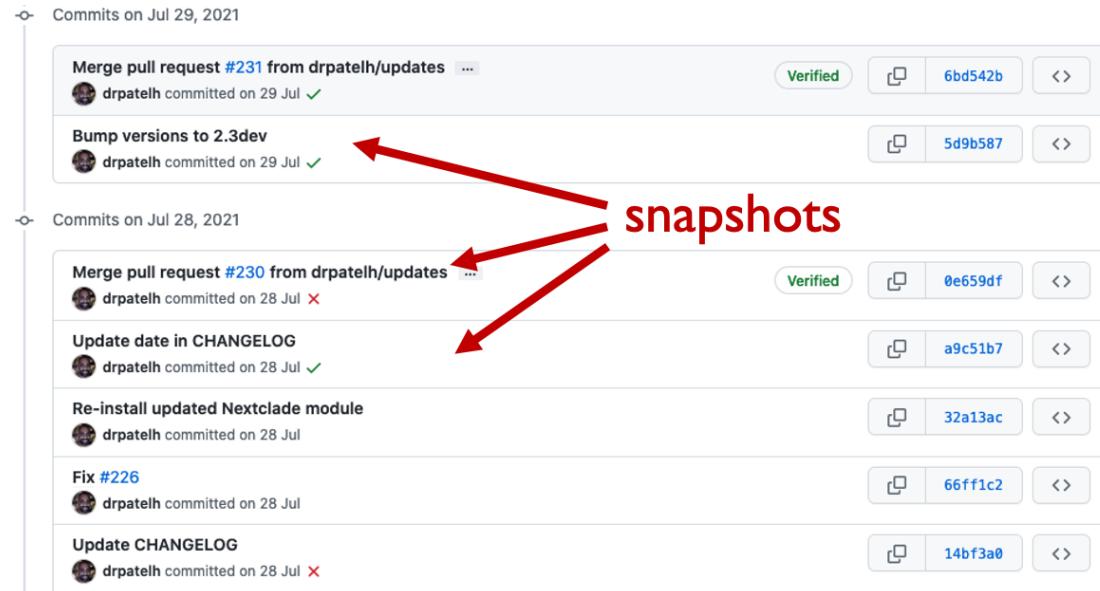
Duplicate / rename files and folders
Changes.txt

- 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

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
FINAL_rev.20.comments10

10:35 AM - 1 Feb 2017

1,197 Retweets 1,750 Likes

Follow

 Nicola Gigante
@gignico

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

<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 analysis recipe

Add an optional extended description...

Commit directly to the `main` 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 • **Repository name** • 
wna-se / analysis-recipe 

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

Description (optional)
A collection of my project's analysis recipies

 **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.

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

Add a README file
This is where you can write a long description for your project. [Learn more.](#)

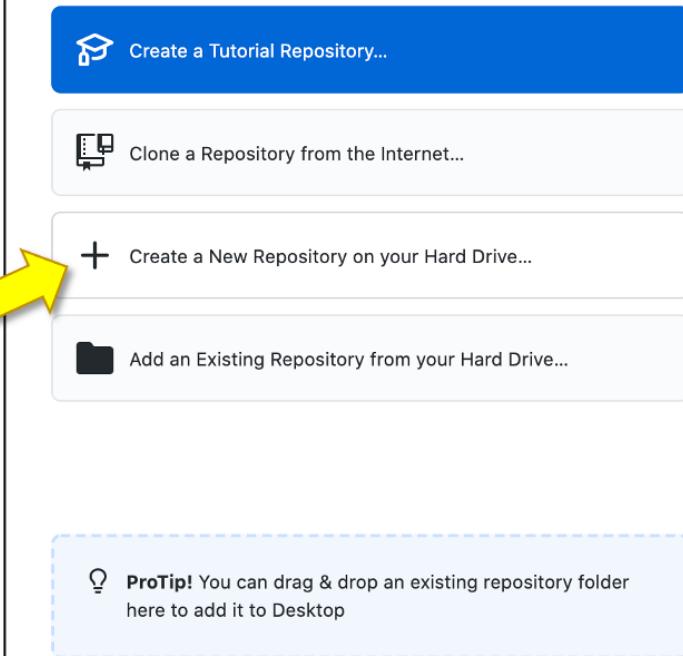
Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: None ▾

Let's get started!

Add a repository to GitHub Desktop to start collaborating



 Create a Tutorial Repository...

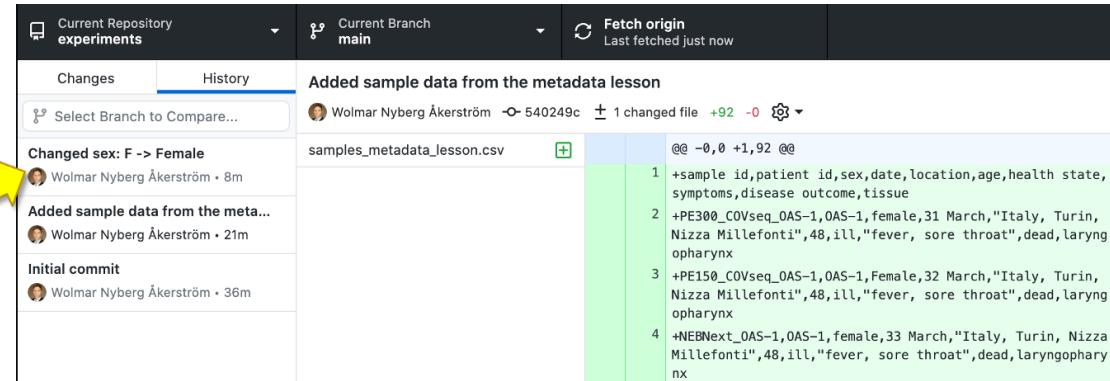
 Clone a Repository from the Internet...

 Create a New Repository on your Hard Drive...

 Add an Existing Repository from your Hard Drive...

 **ProTip!** You can drag & drop an existing repository folder here to add it to Desktop

- **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



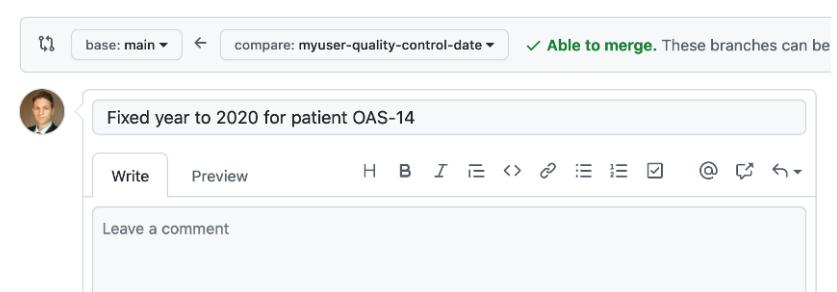
Current Repository experiments		Current Branch main	Fetch origin Last fetched just now
 Changes	 History	 Select Branch to Compare...	
<p>Added sample data from the metadata lesson</p> <p> Wolmar Nyberg Åkerström · 540249c ± 1 changed file +92 -0 ⚙️</p> <p>Changed sex: F -> Female</p> <p> Wolmar Nyberg Åkerström · 8m</p> <p>Added sample data from the meta...</p> <p> Wolmar Nyberg Åkerström · 21m</p> <p>Initial commit</p> <p> Wolmar Nyberg Åkerström · 36m</p>			
<pre>samples_metadata_lesson.csv</pre> <pre>@@ -0,0 +1,92 @@ 1 +sample id,patient id,sex,date,location,age,health state, 2 ,symptoms,disease outcome,tissue 2 +PE300_COVseq_OAS-1,OAS-1,female,31 March,"Italy, Turin, 2 Nizza Millefonti",48,ill,"fever, sore throat",dead,laryng 2 opharynx 3 +PE150_COVseq_OAS-1,OAS-1,Female,32 March,"Italy, Turin, 3 Nizza Millefonti",48,ill,"fever, sore throat",dead,laryng 3 opharynx 4 +NEBNnext_OAS-1,OAS-1,female,33 March,"Italy, Turin, Nizza 4 Millefonti",48,ill,"fever, sore throat",dead,laryngophary 4 nx</pre>			

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-quality-control-date`. Create a pull req changes.



Fixed year to 2020 for patient OAS-14

main (#1)

wna-se committed 11 hours ago Verified

commit 6c51c64d94aa01f12974b7ecbe83f0431ba73a49

samples_metadata_lesson.csv

```

15 PE150_COVseq_OAS-13,OAS-13,female,31/3/2020,"Italy, Turin, Torino",83,ill,"fatigue, loss of taste",dead,laryngopharynx
16 NEBNext_OAS-13,OAS-13,female,31/3/2020,"Italy, Turin, Torino",83,ill,"fatigue, loss of taste",dead,laryngopharynx
17 PE300_COVseq_OAS-14,OAS-14,Male,4/1/2020,"Italy, Turin, Campidoglio",21,ill,fever,dead,laryngopharynx
18 - PE150_COVseq_OAS-14,OAS-14,M,4/1/2021,"Italy, Turin, Campidoglio",21,ill,fever,dead,laryngopharynx
19 - NEBNext_OAS-14,OAS-14,M,4/1/2022,"Italy, Turin, Campidoglio",21,ill,fever,dead,laryngopharynx
20 PE300_COVseq_OAS-15,OAS-15,Female,1/4/2020,"Italy, Turin, Turin",44,healthy,N/A,healthy,lung
21 PE150_COVseq_OAS-15,OAS-15,female,1/4/2020,"Italy, Turin, Turin",44,healthy,N/A,healthy,lung
22 NEBNext_OAS-15,OAS-15,female,1/4/2020,"Italy, Turin, Turin",44,healthy,N/A,healthy,lung

```

- 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/>