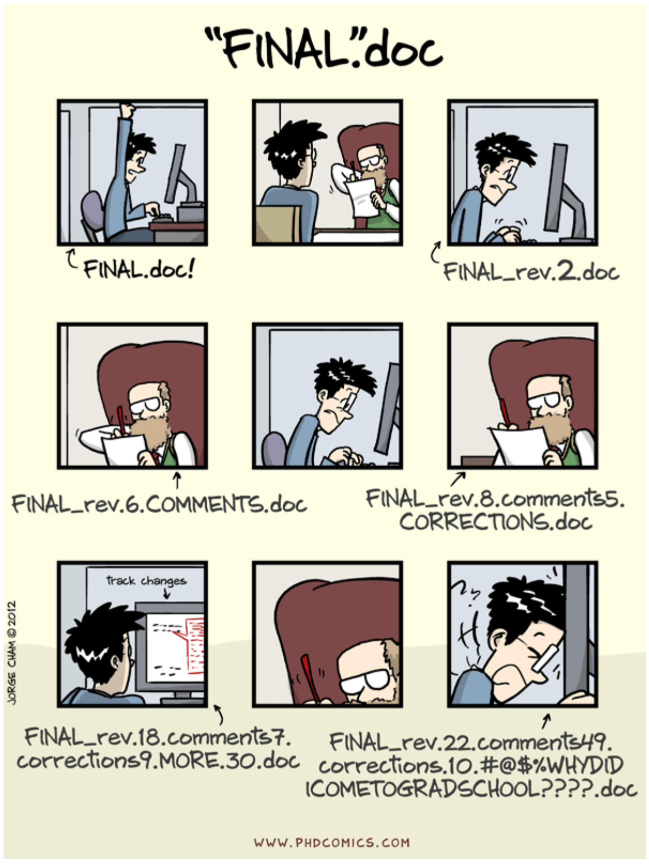
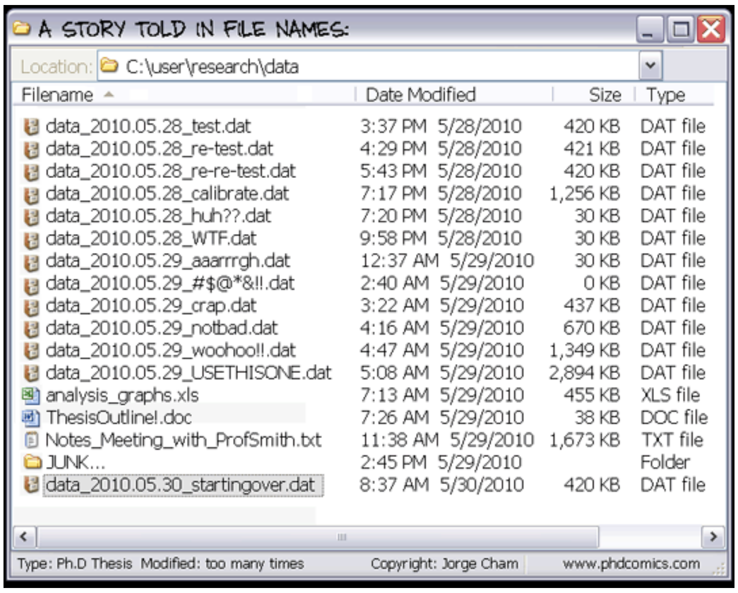
# Hints and suggestions on *Version Control*

# for the (text) files of your projects

Michele GIUGLIANO

## Does this look familiar?




A STORY TOLD IN FILE NAMES:

Location: C:\user\research\data

| Filename | Date Modified | Size | Type |
|---|---|---|---|
| data_2010.05.28_test.dat | 3:37 PM 5/28/2010 | 420 KB | DAT file |
| data_2010.05.28_re-test.dat | 4:29 PM 5/28/2010 | 421 KB | DAT file |
| data_2010.05.28_re-re-test.dat | 5:43 PM 5/28/2010 | 420 KB | DAT file |
| data_2010.05.28_calibrate.dat | 7:17 PM 5/28/2010 | 1,256 KB | DAT file |
| data_2010.05.28_huh??.dat | 7:20 PM 5/28/2010 | 30 KB | DAT file |
| data_2010.05.28_WTF.dat | 9:58 PM 5/28/2010 | 30 KB | DAT file |
| data_2010.05.29_aaarrrgh.dat | 12:37 AM 5/29/2010 | 30 KB | DAT file |
| data_2010.05.29_#$@*&!!.dat | 2:40 AM 5/29/2010 | 0 KB | DAT file |
| data_2010.05.29_crap.dat | 3:22 AM 5/29/2010 | 437 KB | DAT file |
| data_2010.05.29_notbad.dat | 4:16 AM 5/29/2010 | 670 KB | DAT file |
| data_2010.05.29_woohoo!!.dat | 4:47 AM 5/29/2010 | 1,349 KB | DAT file |
| data_2010.05.29_USETHISONE.dat | 5:08 AM 5/29/2010 | 2,894 KB | DAT file |
| analysis_graphs.xls | 7:13 AM 5/29/2010 | 455 KB | XLS file |
| ThesisOutline!.doc | 7:26 AM 5/29/2010 | 38 KB | DOC file |
| Notes_Meeting_with_ProfSmith.txt | 11:38 AM 5/29/2010 | 1,673 KB | TXT file |
| JUNK... | 2:45 PM 5/29/2010 | | Folder |
| data_2010.05.30_startingover.dat | 8:37 AM 5/30/2010 | 420 KB | DAT file |

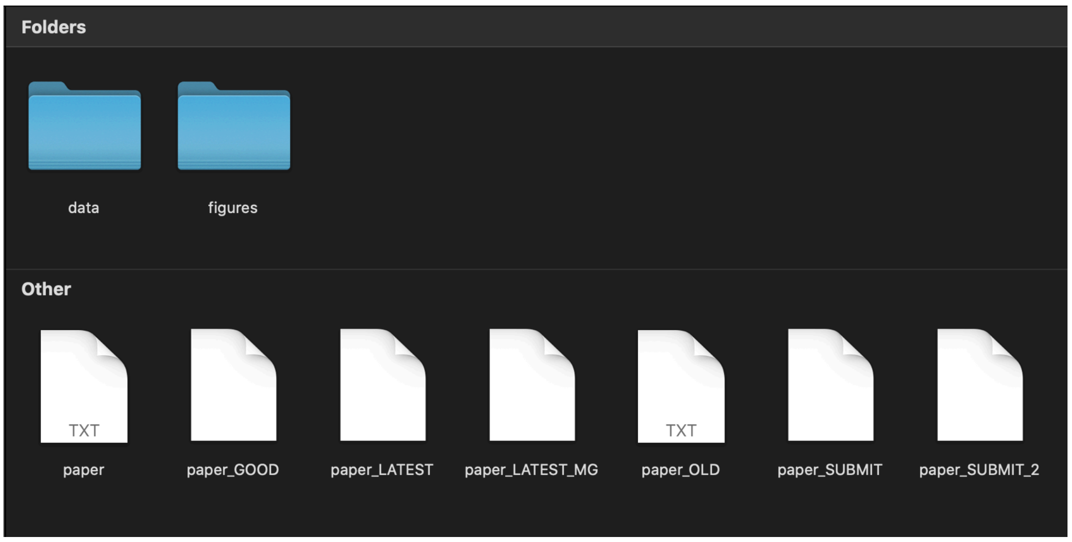
Type: Ph.D Thesis Modified: too many times Copyright: Jorge Cham www.phdcomics.com

# (Ideal) Digital Lab Journaling

- Every "project" has its own, *ad hoc*, directory on your PC: **MyProjectA/***
- **MyProjectA/** contains several (appropriate) subdirectories, e.g.
  - MyProjectA/literature/*
  - MyProjectA/rawData/* (organised appropriately, e.g. per day of exp.?)
  - MyProjectA/matlabCode/*
  - MyProjectA/analyzedData/* (with appropriate subfolders)
  - MyProjectA/figures/* (including scripts to generate each figure from the analysed data)
  - MyProjectA/text/*
  - MyProjectA/text/manuscript/*
- There is a corresponding (dedicated) **notebook** on your own *personal KMS* (*Knowledge Management System* - Evernote? Microsoft OneNote? Apple Notes? Google Documents? Google Keep? Dropbox Paper? Bear? Notion?)
- They both have automated (incremental) daily/weekly backups on at least two external hard drives - each kept at a separate place - and also sync to the cloud (DropBox? OneDrive? GDrive? Box?)

# What is Version Control?

- It is a *support technology*, precious for whoever works with any type of files (especially text files, analysis code, etc. …)
- It promotes reproducible research (logging every step)
- It helps avoiding the **mess** below…



## What is Version Control?

- A system (central or distributed) that keeps track of all changes to files, and/or of *simultaneous* changes (multiple contributors),
- and that allows to recall (any) past versions of the same files,
- to display what changed from one version to the next
- and to easily reconcile/merge concurrent modifications.
- Impractical and imprecise, if done “manually” by
    1. renaming files with clever strategies (.old, .orig, .bak, …), adding versions (paper_v1, paper_v2,…), or dates (paper_Feb14_2020, paper_Feb20_2020,…). But… what if your hard disk crashes?
    2. storing files on DropBox to backup/share/collaborate with others. But… ever occurred to you a “sync conflict”?

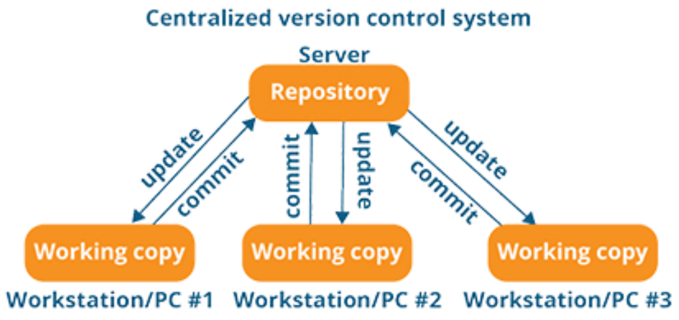
## Why Version Control?

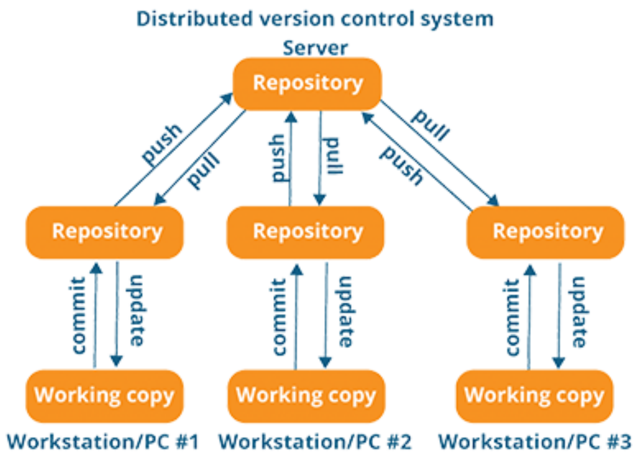
- Enforce discipline without efforts, keeping files tidy and organised
- Backup + restore, in case of a accidental deletions or edits
- Archive subsequent versions
- Maintain historical info on all versions (who did what & when)
- Update/introduce/explore new changes safely, leaving the “current” working) version untouched
- Sync to more than one computer
- Collaborate simultaneously, in a small or a large team

## Better than <u>solely</u> using DropBox & co.

- Cloud syncing offers no fine-grained version control over **what**, **when**, and **who** changed each file…

## <u>Version Control</u>





## Distributed Version Control

**Git** by Linus Torvalds
(2005)

See

https://git-scm.com/
https://git-scm.com/doc

https://www.github.com
https://guides.github.com/activities/hello-world/

http://gitkraken.com
https://www.gitkraken.com/learn-git

## Best used for

- Text files (manuscripts, grant applications, analysis code, tabular and text data, lab book notes)

- Binary files (images, raw data, etc.) are ok but with some exception are difficult to “diff” (outline differences between version)

- Large (binary) files: can be “version controlled” too, with some special support called “LFS” (Large File System)

## Some jargon: definitions

- **repository (repo)** - database where latest files + past revisions are stored
- **local** - the local computer folder that hosts the repository
- **server / remote (repository)** - a remote computer that hosts the repository
- **client(s)** - the computer(s) connecting to the server
- **working copy** - your local directories and files
- **branch/master** - the location for the files in the repository
- **head** - the current (or latest) version of the database in the repository

## More jargon: basic actions

- **add** - place a file under version control
- **check in** (or **commit**) - send local changes to the repo and **annotate them**
- **check out** - obtain from the repo a specific working copy
- **ignore** - allow some files to exist in the working copy but not in the repo ("not under version control")
- **push** - upload the repo to a server
- **pull** - download the repo from a server
- **revert** - throw away the working copy and restore last version
- **update/sync** - update the working copy to the latest revisions

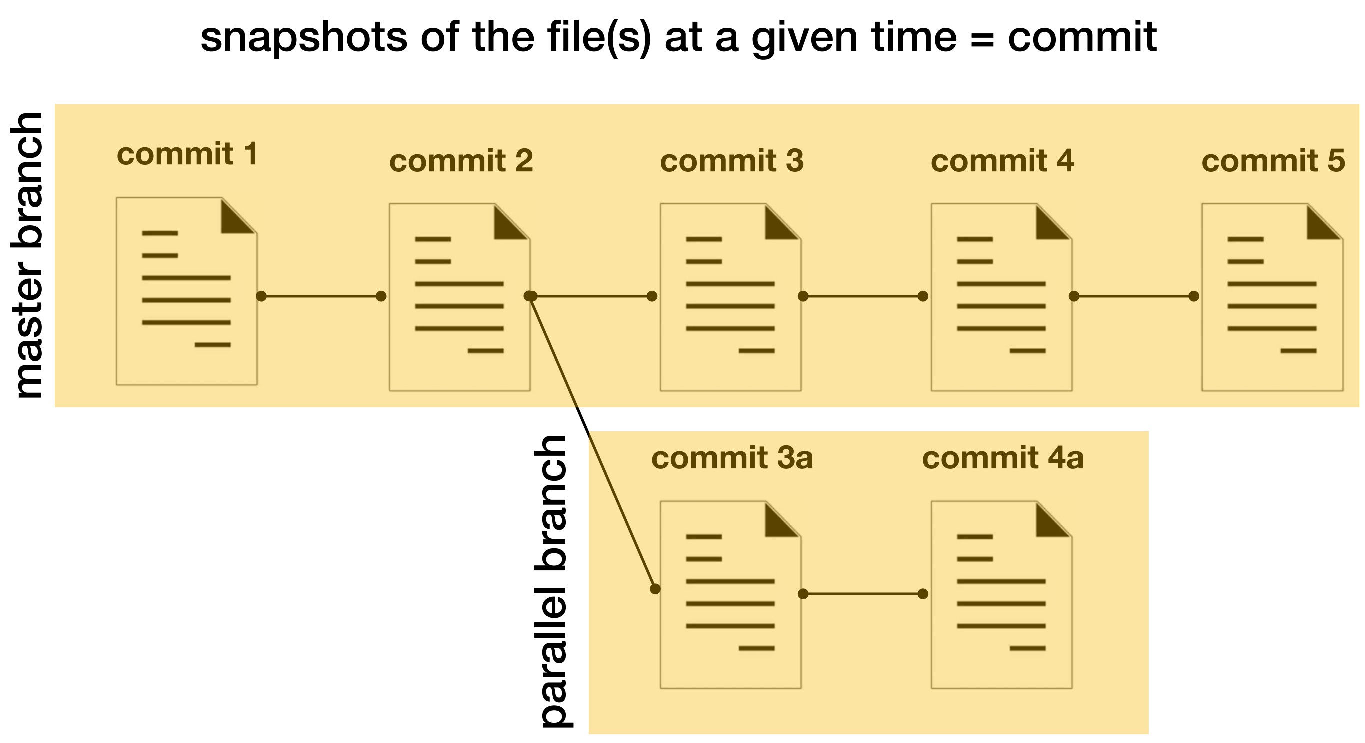
## Git is a *watchdog* for files (their creation/deletion/editing)

## It forces you to document/comment changes, once they are committed

http://www.konscience.de/2015/04/ksl002-digital-lab-journalling-with-git/

## Commit messages (e.g.) show “love” for your future self!

- *Initial commit*
- *Added the Results section*
- *Moved numerical results from figure captions into the main text*
- *Included revisions by my supervisor*
- *Incorporated the suggestions from Reviewer 1*
- *Rewritten Materials and Methods for increased clarity*
- *Corrected the horizontal scale bar in Figure 3*
- *Proof-read*
- *Added relevant citations in the Discussion*
- *Moved incubation time from the Results to the Methods*



## github.com and GitKraken (as a *soft way* into Git revision control)

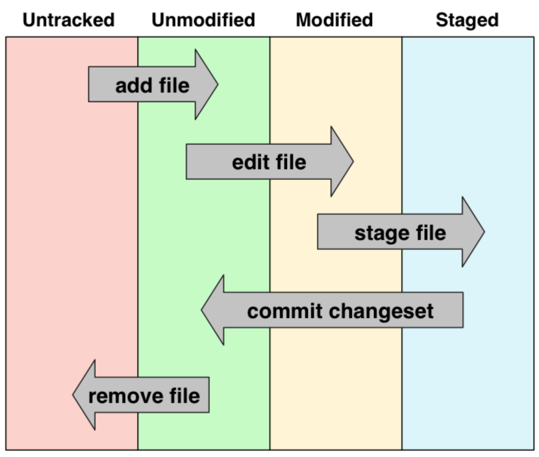
- Download **GitKraken** from http://gitkraken.com/download (Mac/Win/Linux)
- Create a **GitHub** account at https://github.com/join
- Apply for a **GitHub** "**education**" account:
    - navigate to https://education.github.com
    - select "**GitHub student development pack**"
    - Click on "Get the Pack" button and then on "Get Student Benefits"
    - Submit the requested info, i.e. **your @sissa.it email** address, etc.

## Demo with GitKraken

## Typical git workflow 1

"work" = change / edit / modify / create new / delete

- **work** —> stage —> commit —> **work** —> stage —> ….

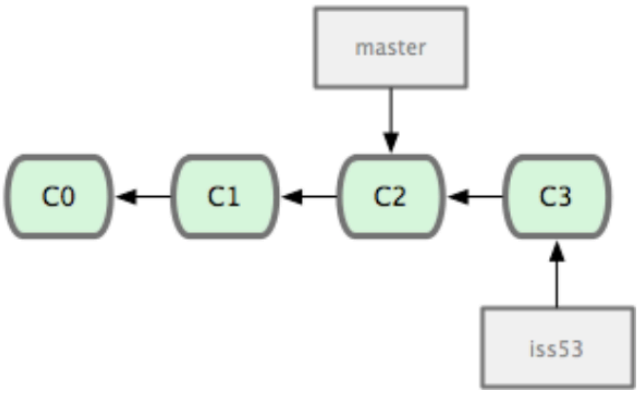


always work on **master**

## Typical git workflow 2

“work” = change / edit / modify / create new / delete

- branch —> **work** —> stage —> commit —> merge it into the master and delete the branch—> branch —> **work** —> …



never work on **master**

## Additional pointers

- https://scfbm.biomedcentral.com/articles/10.1186/1751-0473-8-7
- https://uc3.cdlib.org/2014/05/05/github-a-primer-for-researchers/
- https://mollygibson.github.io/2014-08-11-wustl/lessons/git-notebook/git-for-scientists.slides.html#/
- https://marciovm.com/i-want-a-github-of-science
- http://www.konscience.de/2015/04/ksl002-digital-lab-journalling-with-git/

- https://software-carpentry.org/lessons/dashboard/ https://youtu.be/PEoULFdSCRU
- https://guides.github.com/activities/hello-world/
- https://git-scm.com/book/en/v2
- https://www.git-tower.com/learn/git/ebook/en/desktop-gui/introduction
- http://ndpsoftware.com/git-cheatsheet.html