Code management for a reproducible science with Git, GitFlow, GitHub and Zenodo

Nicolas Barrier[1]
nicolas.barrier@ird.fr

Criscely Luján[1,2]
criscely.lujan@ird.fr

[1]IRD, UMR-MARBEC

[2]Université Paris-Sud

2021-06-23

Introduction

To reproduce an ecosystem modelling study, you will need:

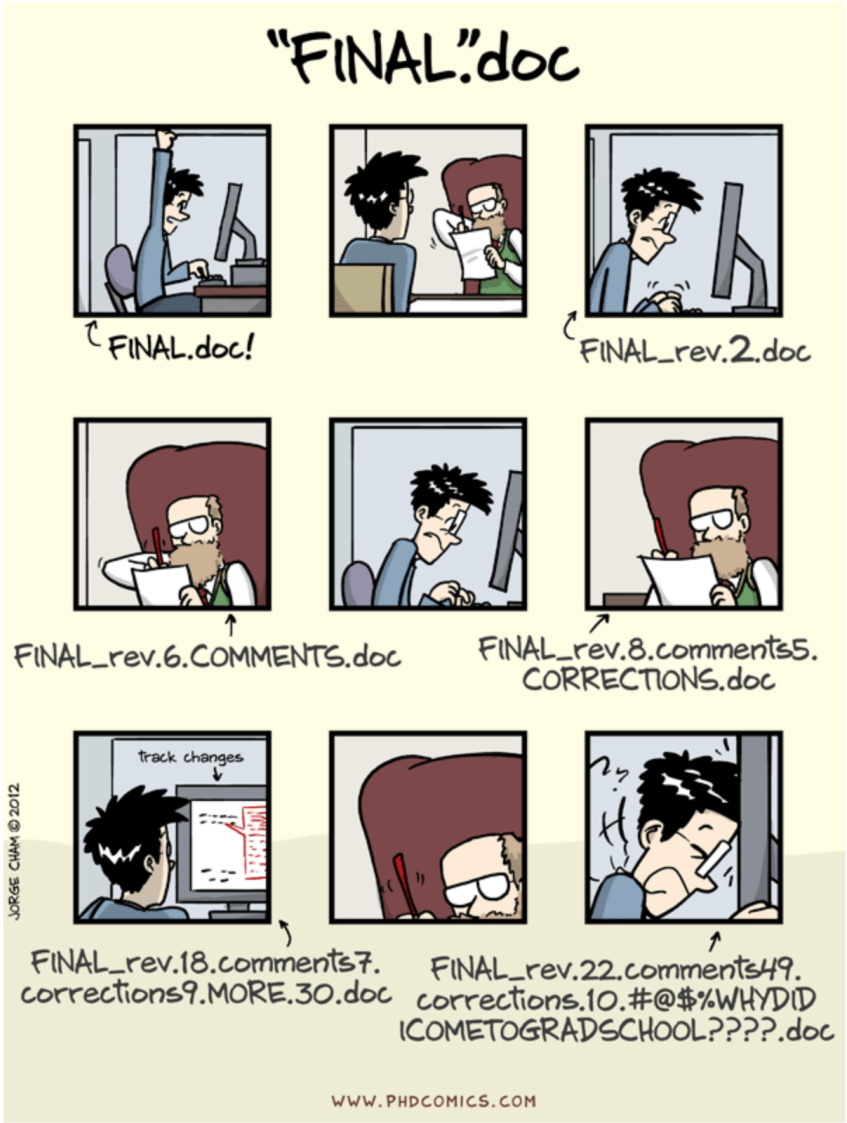
- The model parameters (time-step, mortality rates)
- The forcing files (temperature, oxygen, plankton, etc.)
- **The numerical model** (source or executable)
- **The analysis scripts** (R, Python, Matlab)
- The external libraries used (graphical libraries, mathematical ones, etc.)

Therefore, a proper management of codes (model + scripts) is necessary.

This can be achieved by version control

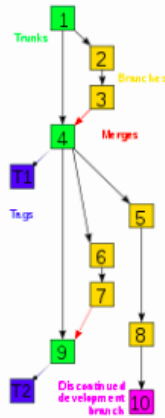
Version control

- Tracks changes (*commits*) over time
- Saves snapshots of a project (*tags*)
- Creates derivates of a project (*branches*)
- Facilitates collaboration among multiple users



Version control softwares

V·T·E	**Version control software**	[hide]
Years, where available, indicate the date of first stable release. Systems with names *in italics* are no longer maintained or have planned end-of-life dates.		
Local only	**Free/open-source**	RCS (1982) · SCCS (1972)
	Proprietary	PVCS (1985) · *QVCS* (1991)
Client–server	**Free/open-source**	CVS (1986, 1990 in C) · CVSNT (1998) · QVCS Enterprise (1998) · Subversion (2000)
	Proprietary	AccuRev SCM (2002) · ClearCase (1992) · *CMVC* (1994) · Dimensions CM (1980s) · *DSEE* (1984) · Endevor (1980s) · Integrity (2001) · Panvalet (1970s) · Perforce Helix (1995) · SCLM (1980s?) · Software Change Manager (1970s) · StarTeam (1995) · Surround SCM (2002) · Synergy (1990) · Team Concert (2008) · Team Foundation Server (2005) · Visual Studio Team Services (2014) · Vault (2003) · *Visual SourceSafe* (1994)
Distributed	**Free/open-source**	*ArX* (2003) · BitKeeper (2000) · *Codeville* (2005) · Darcs (2002) · *DCVS* (2002) · Fossil (2007) · Git (2005) · *GNU arch* (2001) · GNU Bazaar (2005) · Mercurial (2005) · Monotone (2003) · Pijul (2015) · *SVK* (2003) · Veracity (2010)
	Proprietary	*TeamWare* (1990s?) · Code Co-op (1997) · Plastic SCM (2006) · Team Foundation Server (2013) · Visual Studio Team Services (2014)
Concepts	Baseline · Branch · Changeset · Commit · Data comparison · Delta compression · Fork (Gated commit) · Interleaved deltas · Merge · Repository · Tag · Trunk	
Category · Comparison · List		



Source: https://en.wikipedia.org/wiki/List_of_version-control_software

- **Popular and successful**
  - Active development + multiple extensions
- **Tracks any type of files**
  - Works best with ASCII files (`.html`, `.tex`, `.R`, `.c`, `.f90`)
  - Large binary files with Git-LFS (`.nc`, `.Rdata`, `.mat`, `.csv`)
- **Branching**
  - Light branch management
  - Smarter merges
  - Easy code management with Git-Flow

GitHub

Several web-based hosting services are available:

- **GitHub**, Gitlab
- Institutional: GitLab Bioinfo, GitLab UM, SourceSup, Ifremer Forge

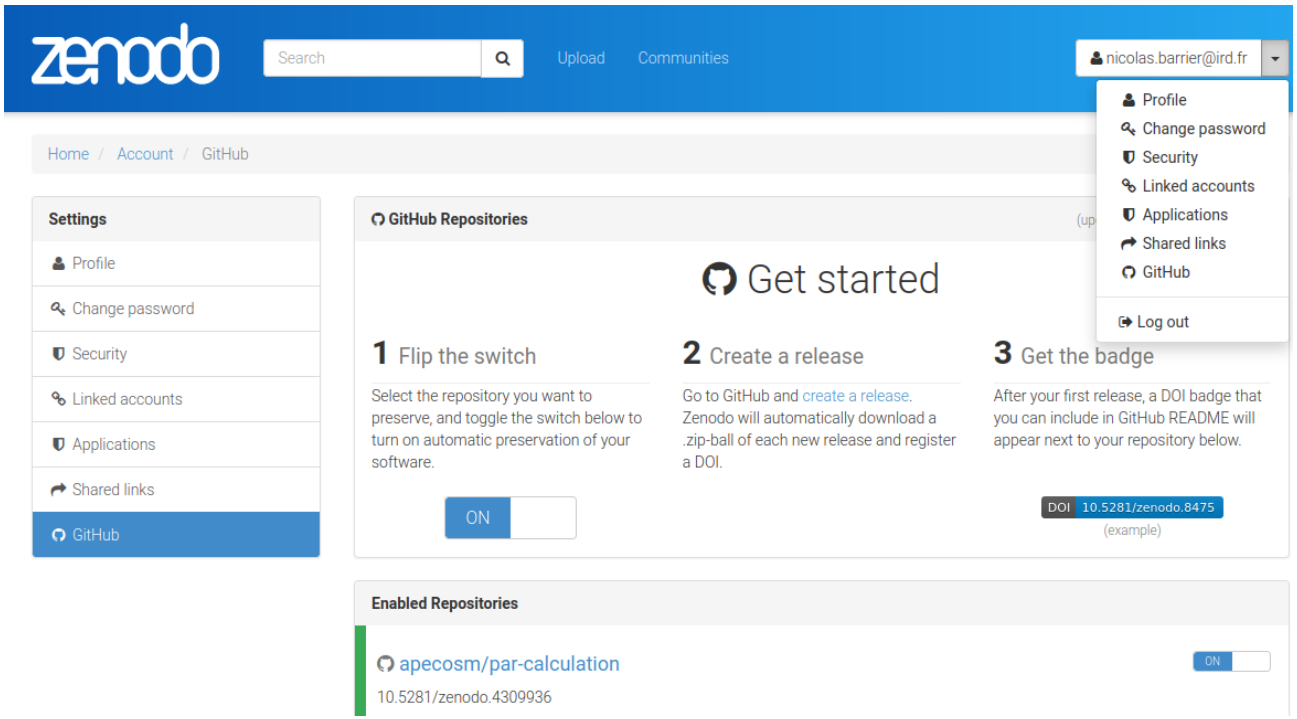
Why use  ?

- Popular (largest host of source code in the world)
- Facilitates worldwide collaboration
- Free (and improved accounts for education)
- Works with Git-LFS
- Continuous integration (GitHub Actions)
- Association with **Zenodo**

Zenodo

Zenodo = *a catch-all repository for EC funded research*, based on FAIR principes (**F**indable, **A**ccessible, **I**nteroperable, **R**eusable).

Automatic generation of DOIs when a new release is drafted.



Good practices

Main branch (`main`/`master`)

- Not updated too often
- Each change is identified (`tag` and `release`)

Development branch (`develop`).

- Independent developments in separated branches

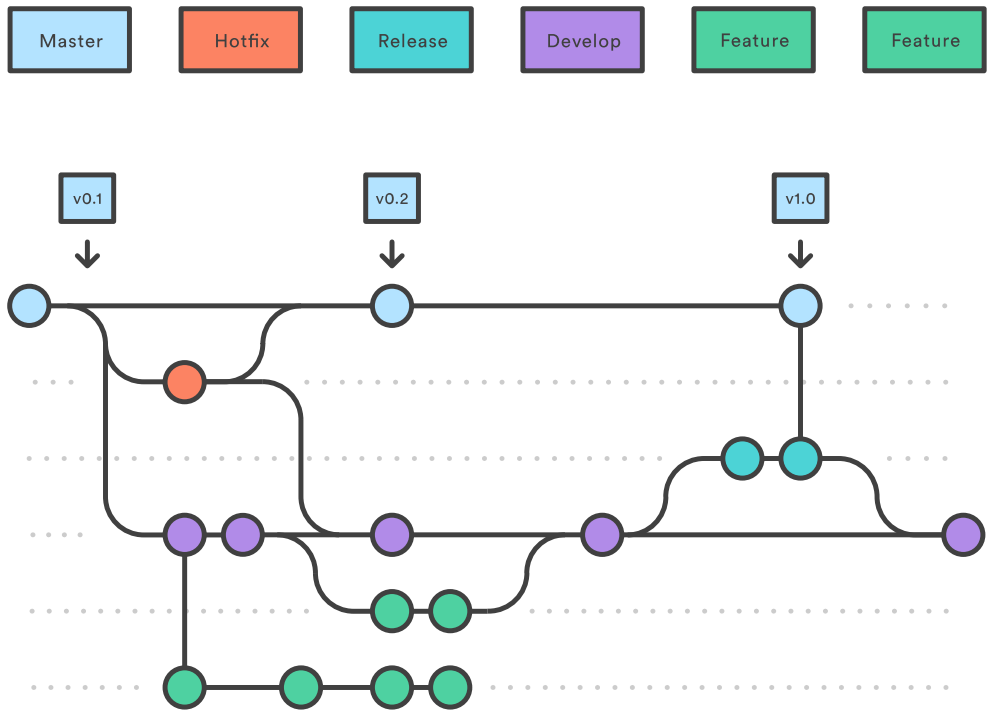
Bug fix:

- In `master` with a new version number
- In `develop`

Seems complicated? Not at all, this is GitFlow!

Git extension to manipulate these branches.

```
git flow TYPE start NAME
git flow TYPE publish NAME
git flow TYPE finish NAME
TYPE=branch type (feature, release, hotfix)
NAME=branch name (feature name, version number)
```



Example: Ichthyop

For an example, visit https://github.com/ichthyop/ichthyop

ICHTHYOP