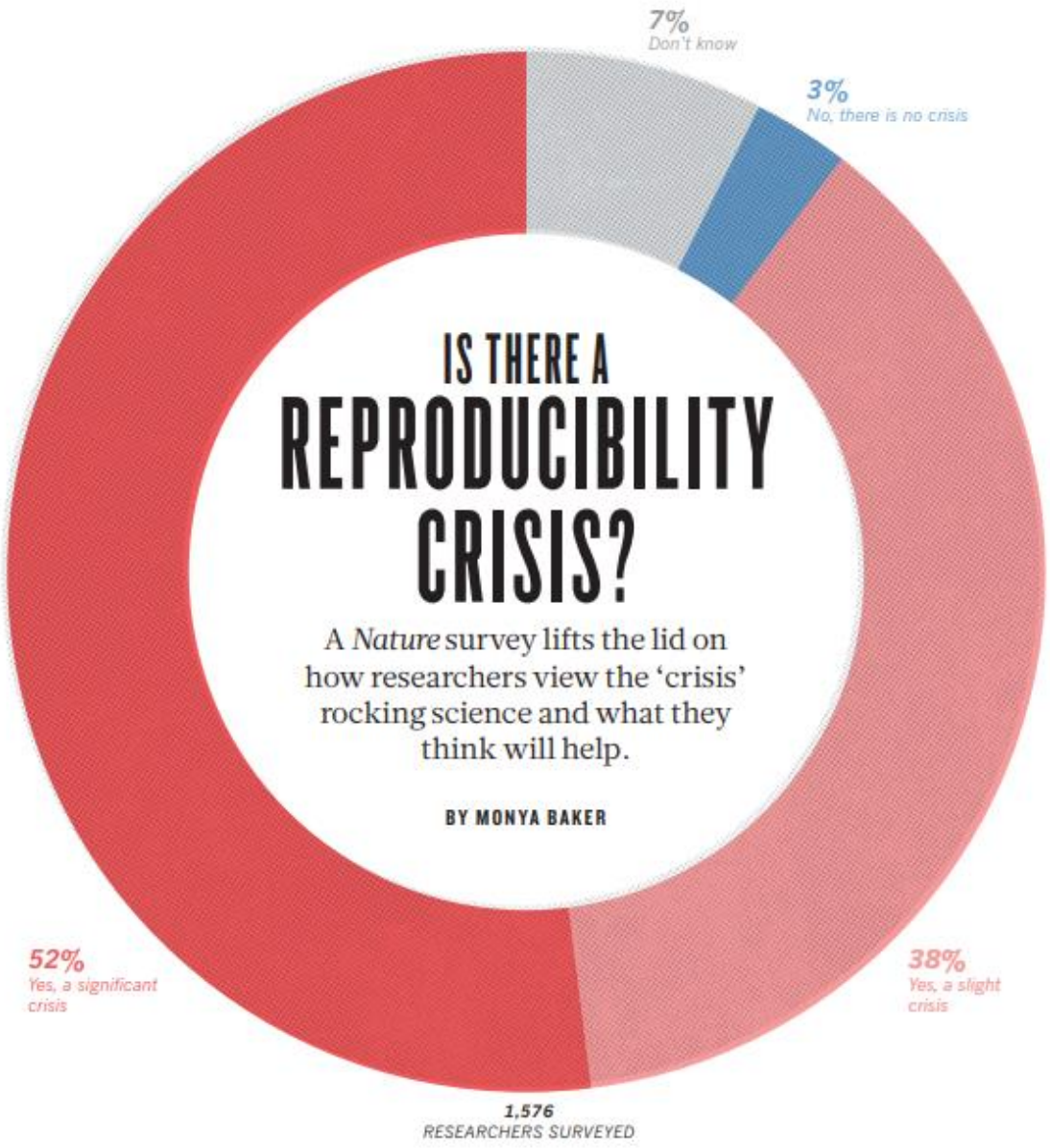
# Reproducible Research in Computational Biology

Best Practices @ Institut Curie Bioinformatics Core Facility

Nicolas Servant, PhD

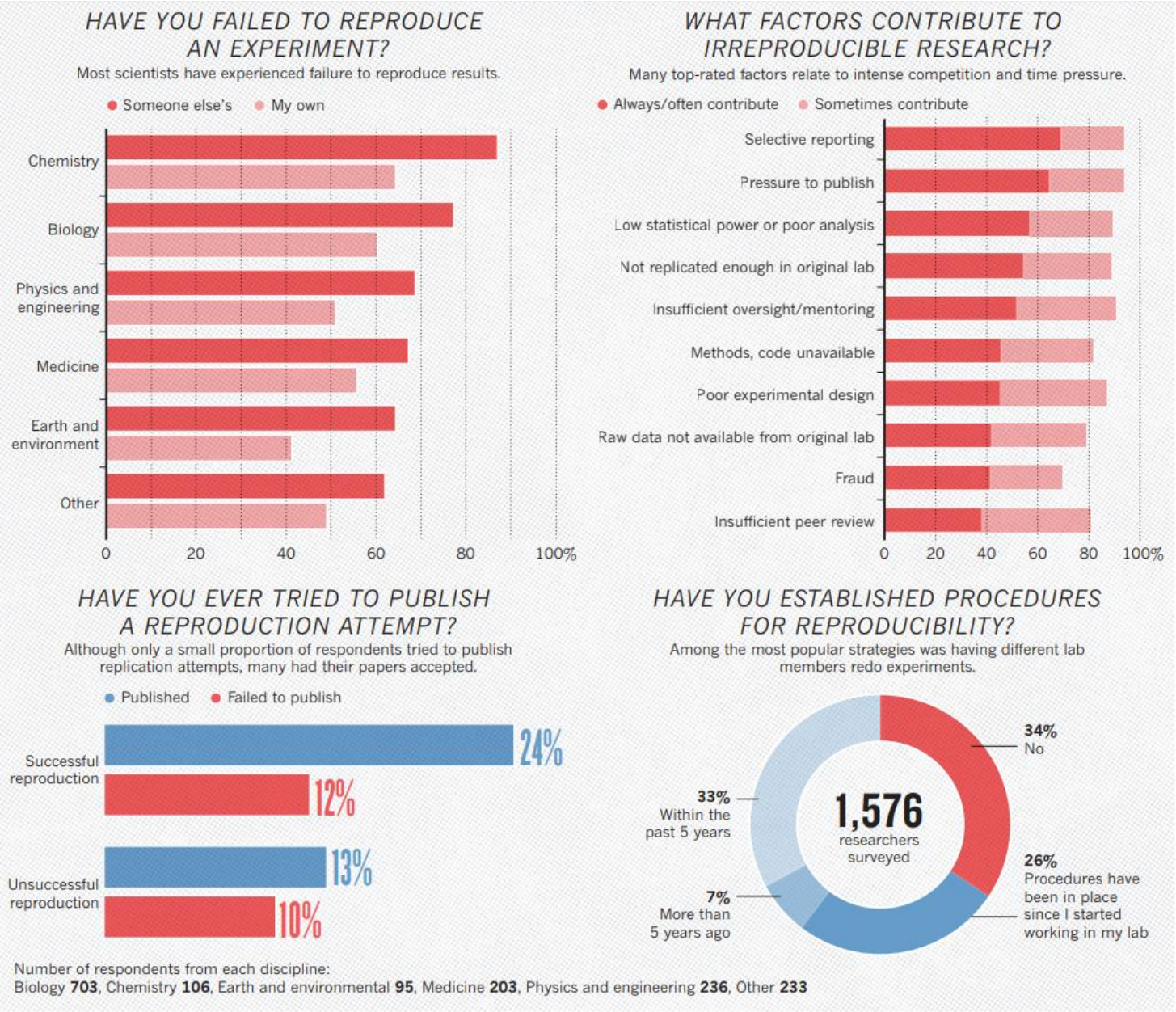
29th September 2020

# Reproducible Research




Baker M. Nature. 2016

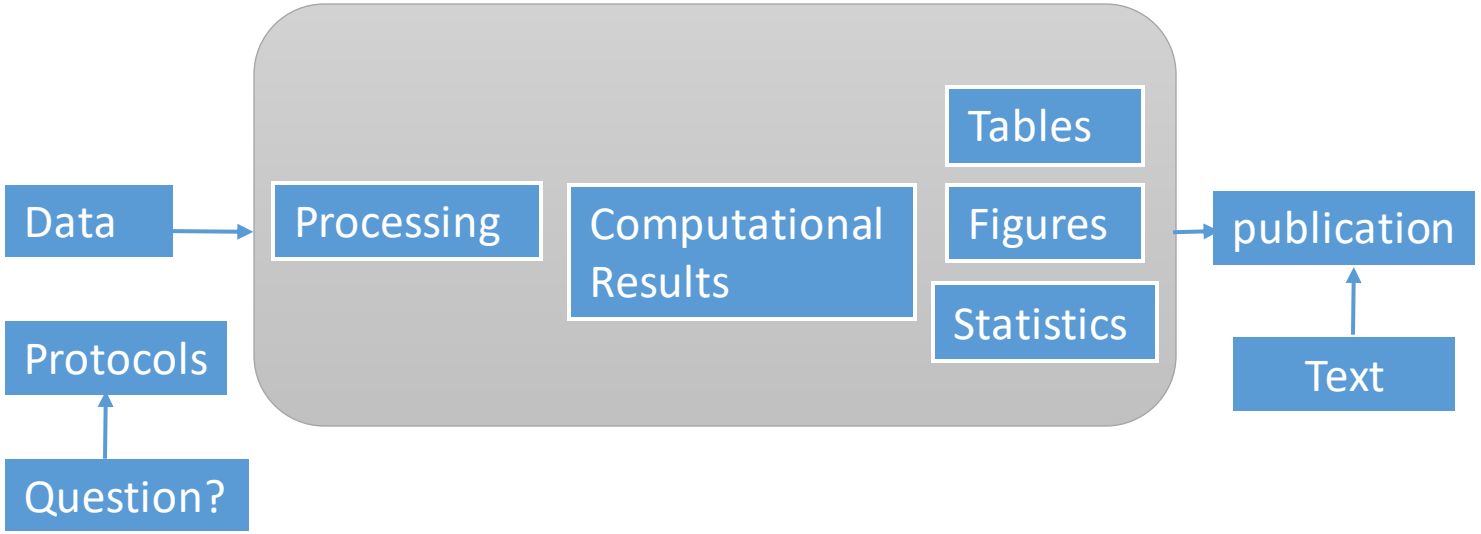
# Reproducible Research




Baker M. Nature. 2016


# What is Reproducibility ?

**Reproducibility is the practice of distributing anything required to reproduce a scientific result**



# Requirements for reproducibility

Reproducibility means :

- Sharing raw (and processed) Data
- Sharing a complete set of instructions explaining all steps used in the processing and analyzing the data
- Sharing all codes and computational environments to create the table/figures/etc.

The codes must be :

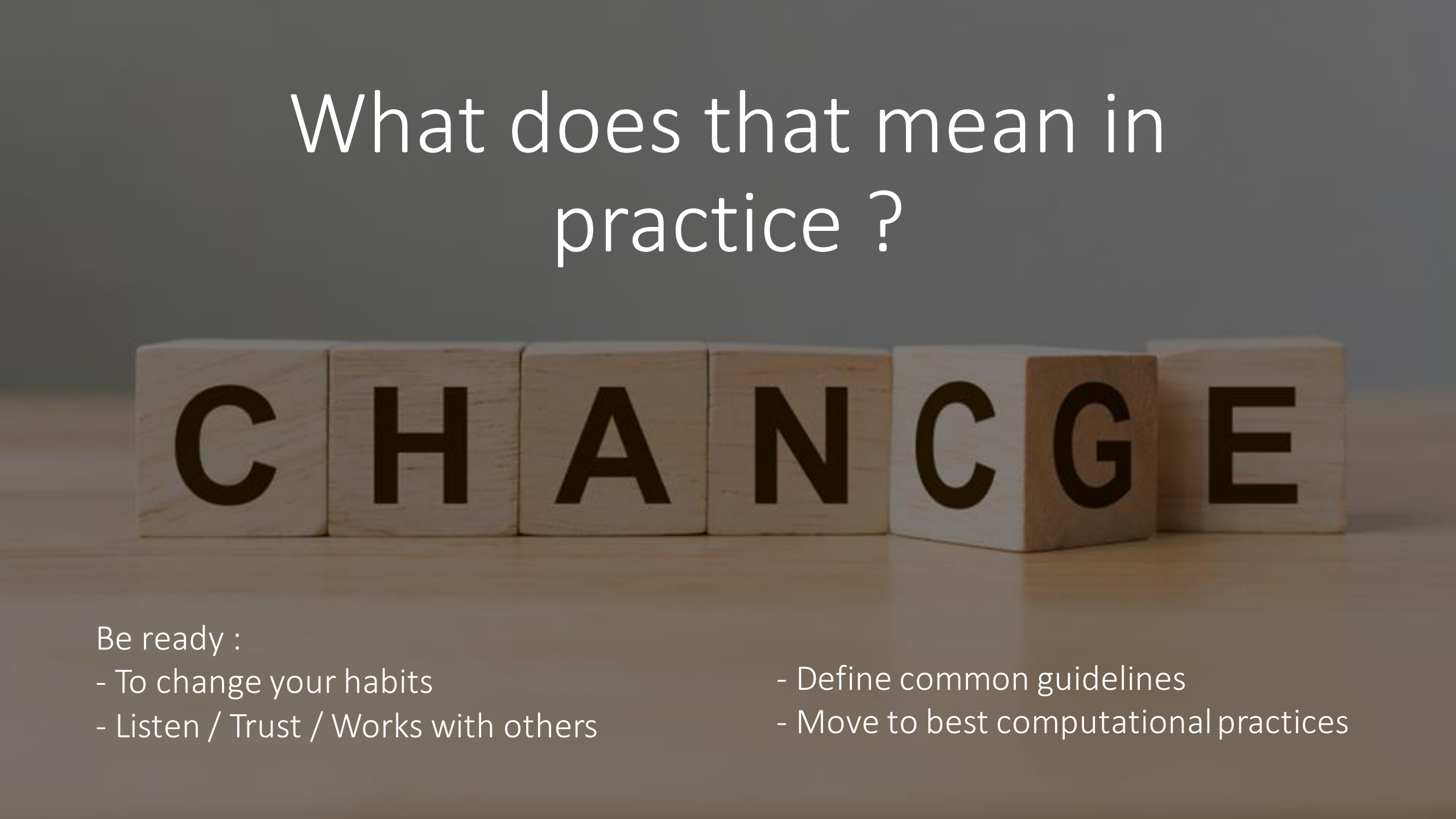
- Written following good programming practices
- Reviewed
- Versioned

# Benefits of Reproducibility

Reproduciblity can

- Increase the likelihood that your research is correct
- Makes it easier to check your research
- Makes it easier to share with others
- Help to be more efficient with reusable code and instruction
- Is a good opportunity to define common working practices

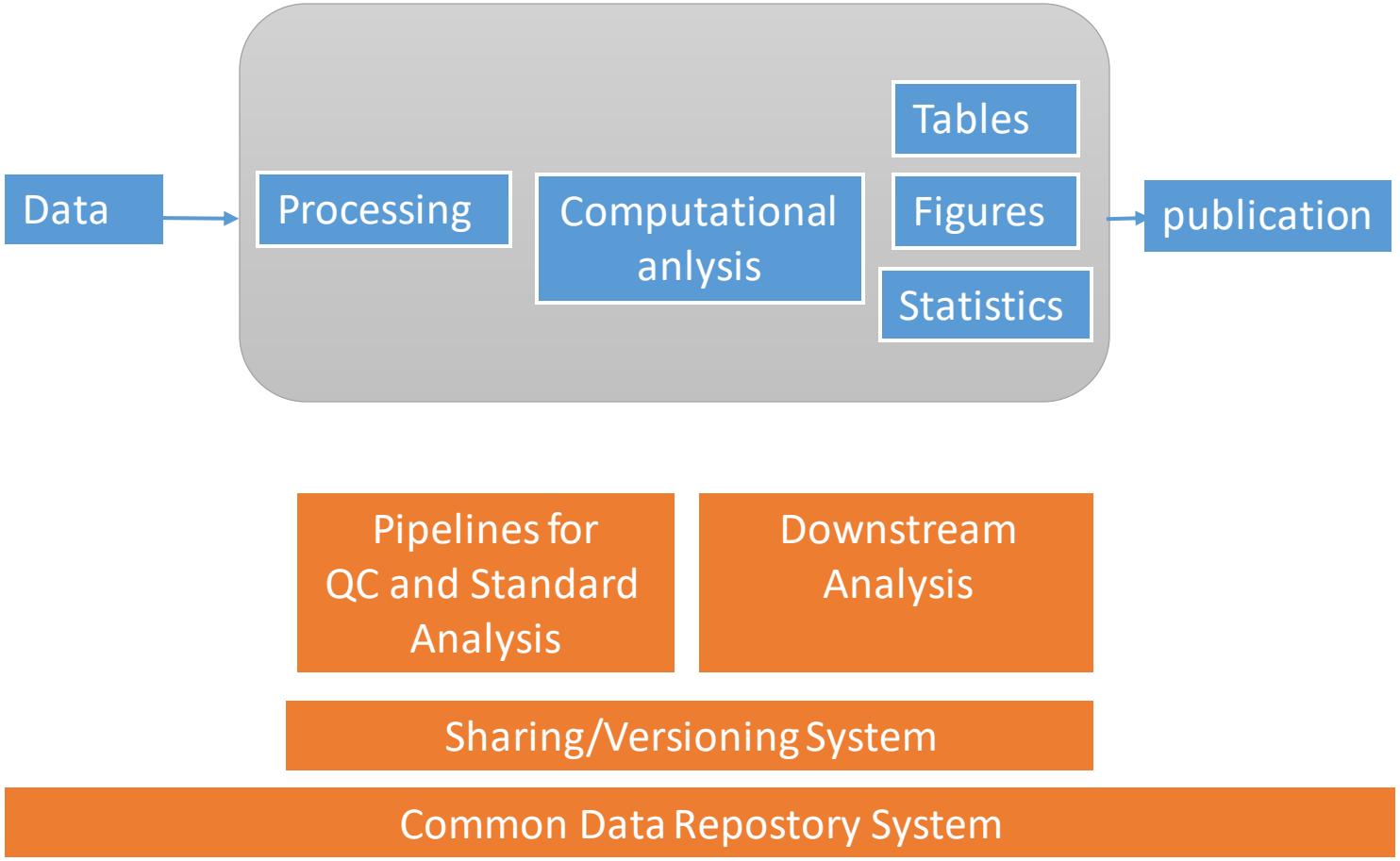
# What does that mean in practice ?




Be ready :

- To change your habits
- Listen / Trust / Works with others
- Define common guidelines
- Move to best computational practices

# Reproducibility : How ? At which level ?

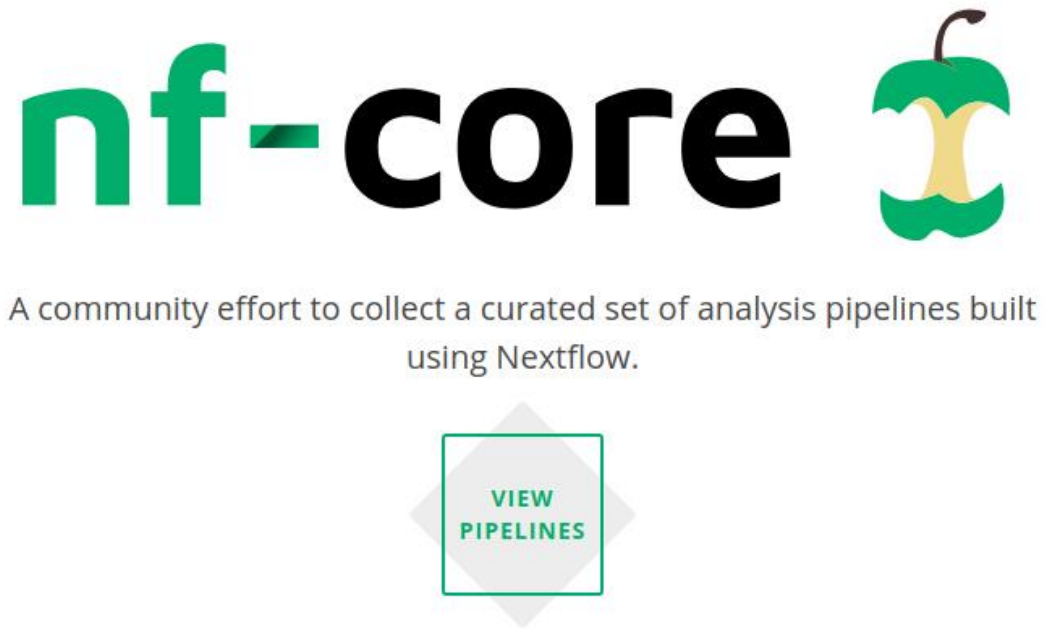


# Bioinformatics pipelines

Pipelines which can be automatically run by users from NGS raw data, and by the production team on all NGS data from the sequencing facility

- Define a common way to write bioinformatics pipelines *for all*
- Define a working organization for data analysists / production engineers / computer scientists
- Promote code review and common pipelines for every bioinformatician

institutCurie




- A community effort
- with 35 Worldwide Institutions
- > 100 developers
- 22 released pipelines
- 17 in development
-

- Nextflow based
- One pipeline template
- Coding best-practices
- Code reviewing
- Containers (conda, docker, singularity)
- Continuous integration

# Workflow Management System

**Why Nextflow ?**

- nf-core community
- Compatible with Docker/Singularity/Conda
- Highly portable

**Using nf-core pipelines ?**

- An amazing resource for the community
- High-quality of the codes
- Slack/discussion/guidelines

**But**

- Do not always answer our internal constraints and requirements

nextflow

# Versioning

Reproducibility requires codes versioning

Git main features :

- Fast, easy to use
- Mature GUI for project managment
- Keep track of code changes
- An efficient branch system
- Promote collaboratif project

# Containers

- Ease software installation
- Reproducible: allow the control of tools version
- Portable: run software on almost all infrastructures
- Conda is easy to use even for a data-analyst
- Singularity vs Docker ?

# Bioinformatics pipelines

Sequencing Facility

CONDA

docker

GitLab

Jenkins

nextflow

GitLab

Code Review

GitLab

Nextflow Pipeline Template (inspired form nf-core good practices)

devel

master

ngsdm

Production Team

clusters

community

GitHub

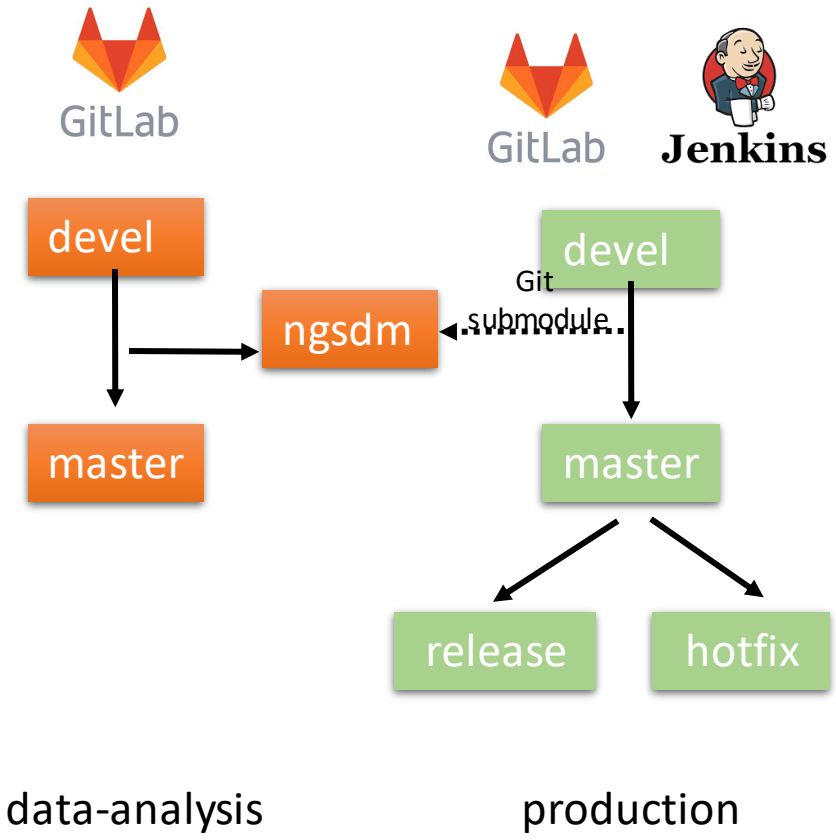
Data Analysis + HPC

institutCurie

# Production environment

Different environment for analysis and production :

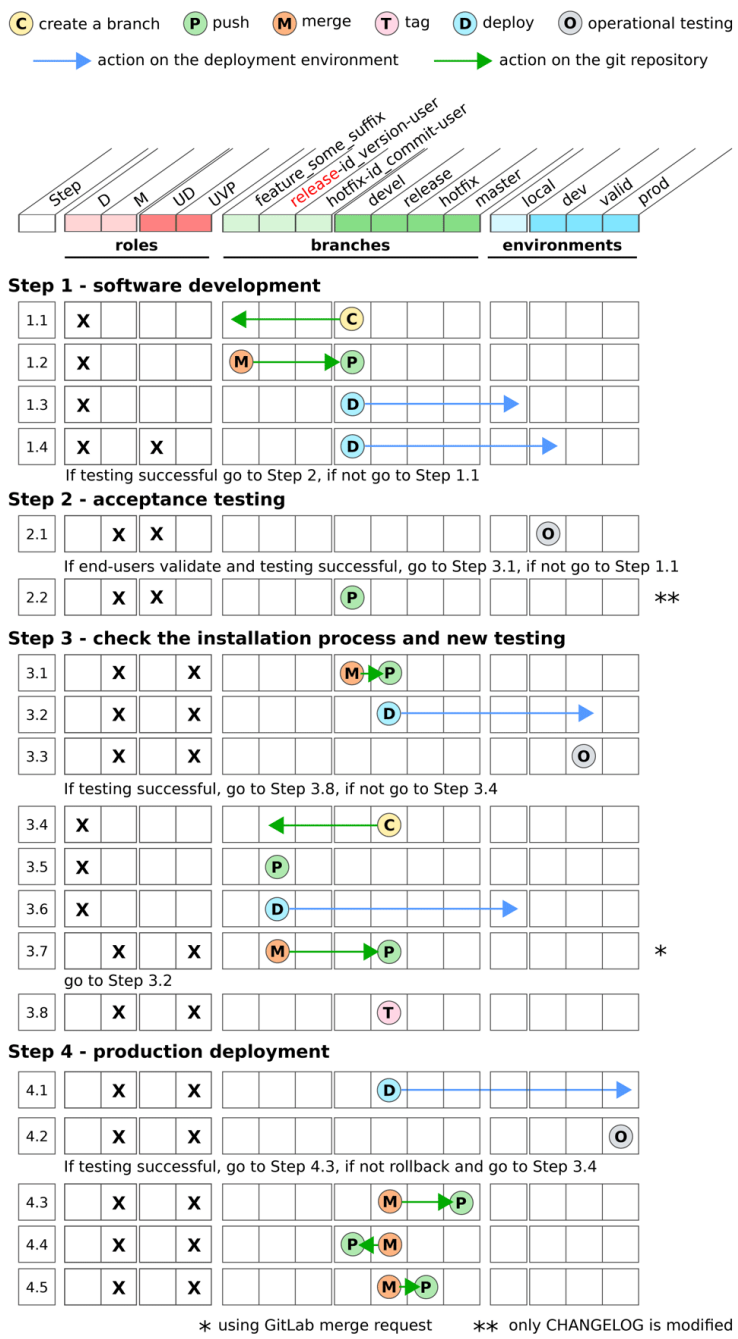
- Data analysis pipelines require more flexibility
- Data analysis pipelines are frequently updated
- Production requires more tracability (dev/valid/prod)
- Production pipelines are less frequently updated
- Production means accreditation in some cases



# Production environment

***biogitflow*: development workflow protocols for bioinformatics pipelines with git and GitLab [version 1; peer review: awaiting peer review]**

Choumouss Kamoun[1-4*], Julien Roméjon[1-4*], Henri de Soyres[1-4], Apolline Gallois[1-4], Elodie Girard[1-4], Philippe Hupé [1-5]

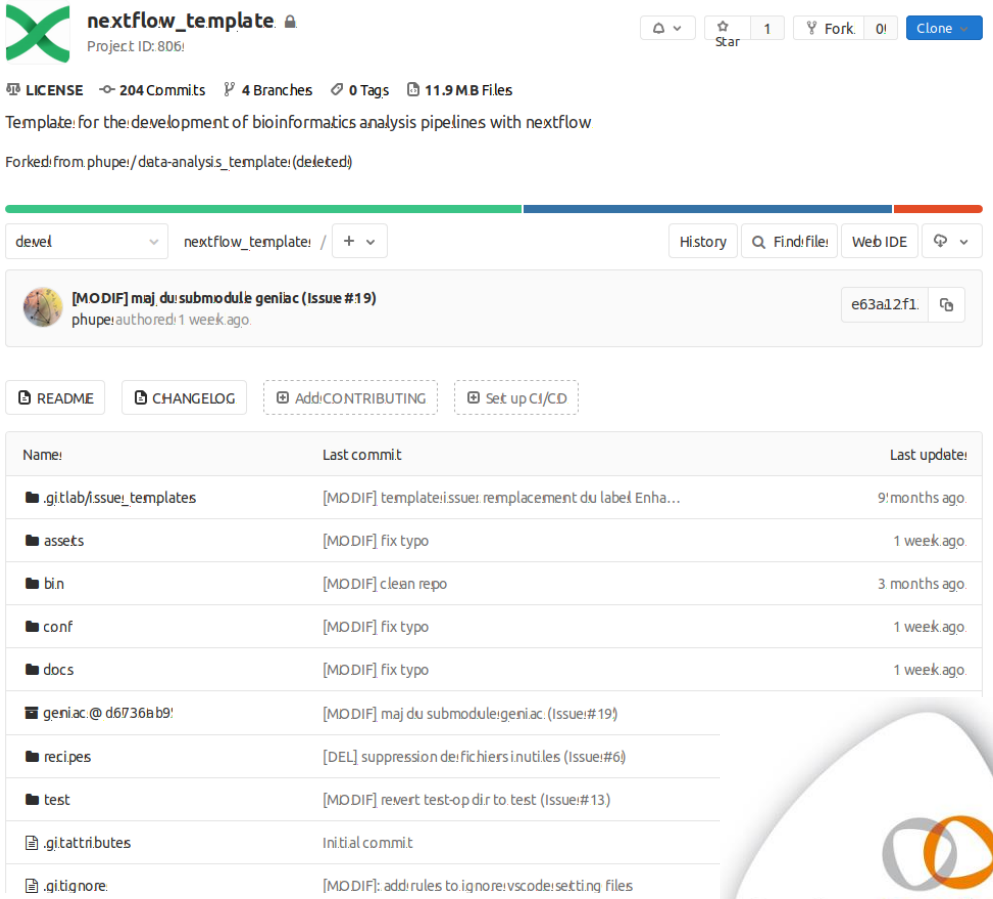


# Guidelines for Nextflow pipelines

Define a common Nexflow pipeline template with :

- Common organisation (folder, scripts)
- Good coding practices
- *Nf-geniac* compatibility

https://github.com/bioinfo-pf-curie/geniac-template



# List of available Nextflow pipelines

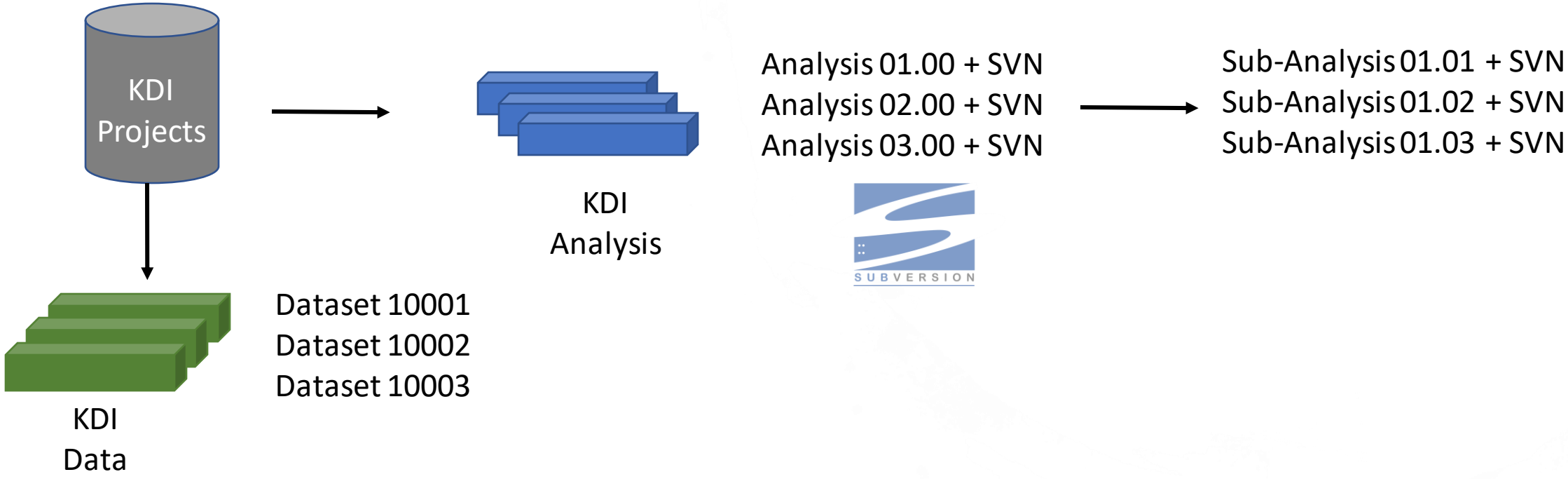
https://bioinfo-pf-curie.github.io/topics/nf

- **Raw-qc** – quality controls of raw data / trimming
- **RNA-seq** - quality controls and gene expression analysis
- **ChIP-seq** - quality controls / up to peak calling / spike-in normalisation / whith(out) inputs
- **ATAC-seq** - quality controls / detection of open chromatin regions / Tn5 insertion sites
- **Allele-specific mapping** – both RNA-seq and ChIP-seq analysis
- **Hi-C** – quality controls / TADs, compartments calling

institutCurie

# Moving to downstream analysis

First challenge : keep track of all analysis performed on a project



# Moving to downstream analysis

- Most of downstream analysis are performed in R
- Dealing with reproducibility is even harder with R
- Managing R packages with conda can be challenging
- Use of the **renv** package

# Renv

- New version of the *Packrat* package
- Available at https://rstudio.github.io/renv/articles/renv.html

- Isolate your packages dependies from the core R source
- Use a receipe file (.lock) to store all packages and versions used for your project
- Handles packages from CRAN, BioC, github, gitlab, bitbucket

# Renv

1. renv::init() - Initialize a new project-local environment with a private R library
2. Install your packages, work as usually on your local environment project
3. renv::snapshot() - Save the state of the project library to a lockfile (renv.lock)
4. renv::restore() - Restore your local environment defined in the renv.lock file

# Reproducibility : changing your habits ?

## Pros

- Reproducibility: A step forward
- A real gain in terms of efficiency in pipeline development
- Save time for routine analysis
- Everyone is able to understand/use/debug the pipelines
- Successfully applied for clinical bioinformatics (accreditation)

## Cons

- Defining good practices for everyone (analyst, HPC, etc.) is not always easy
- Reproducibility has a cost

# Take-home Messages

Manage Data and Analysis related to each project

**Bioinformatics pipelines**

- Good practices for coding template
- Good practices for versioning
- Workflow managment + containers
- Continuous integration

**Downstream analysis**

- Renv is a promising approach
- Version code and publish them for publication

# Many Thanks