

Reproducible Research in Computational Biology

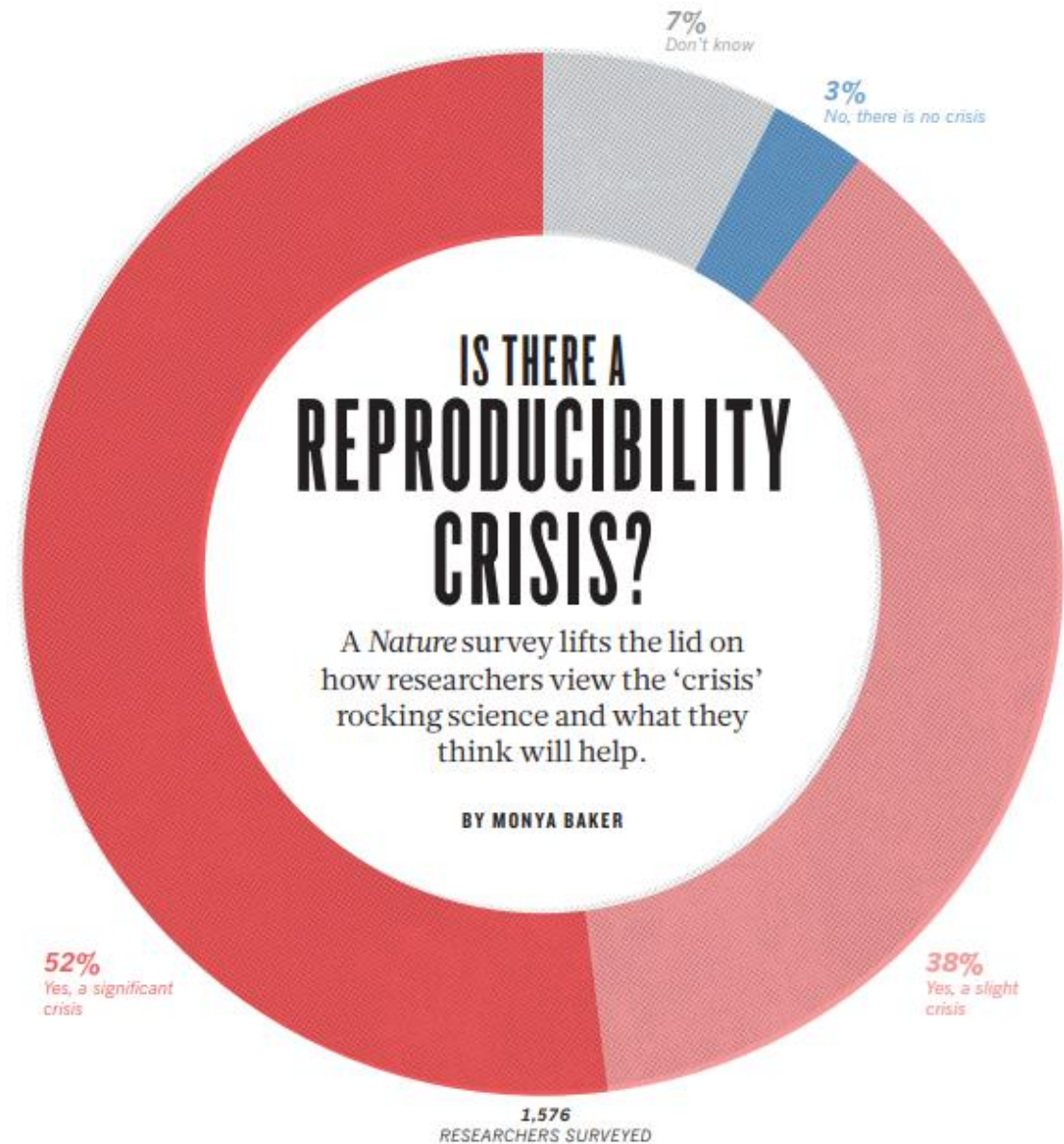
Best Practices @ Institut Curie Bioinformatics Core Facility

Nicolas Servant, PhD

29th September 2020



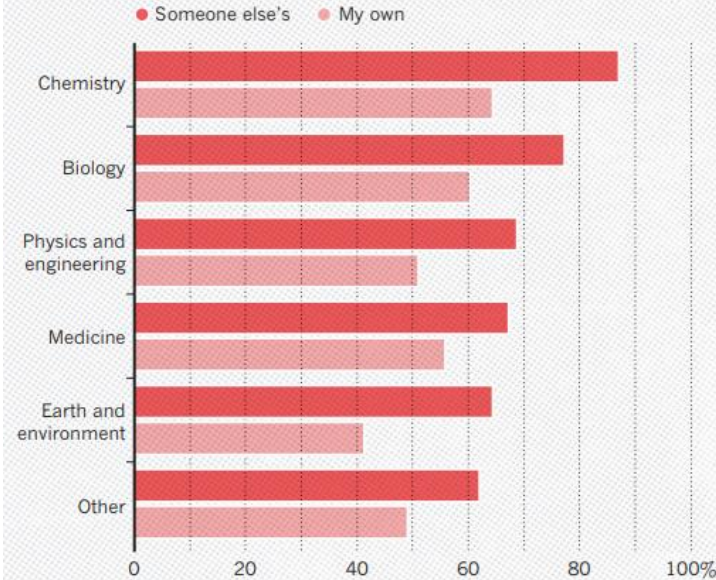
Reproducible Research



Reproducible Research

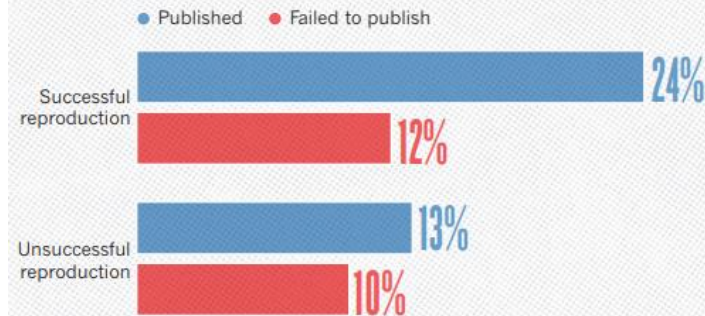
HAVE YOU FAILED TO REPRODUCE AN EXPERIMENT?

Most scientists have experienced failure to reproduce results.



HAVE YOU EVER TRIED TO PUBLISH A REPRODUCTION ATTEMPT?

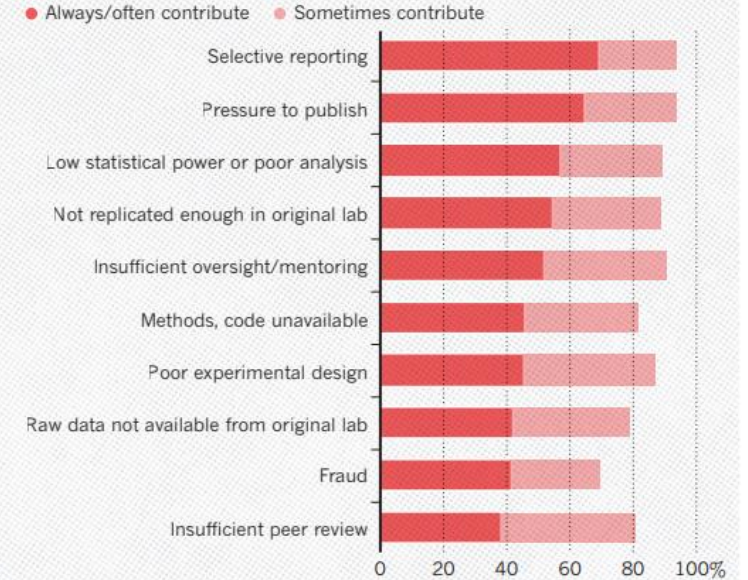
Although only a small proportion of respondents tried to publish replication attempts, many had their papers accepted.



Number of respondents from each discipline:
Biology 703, Chemistry 106, Earth and environmental 95, Medicine 203, Physics and engineering 236, Other 233

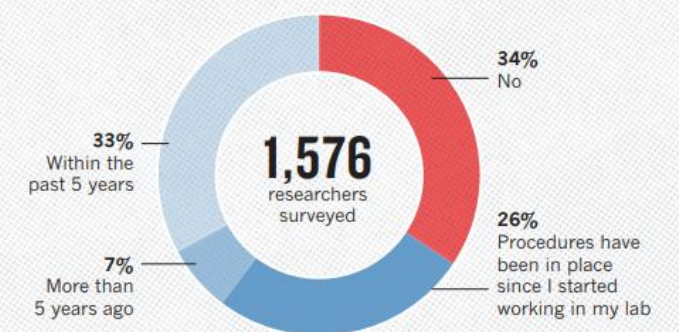
WHAT FACTORS CONTRIBUTE TO IRREPRODUCIBLE RESEARCH?

Many top-rated factors relate to intense competition and time pressure.



HAVE YOU ESTABLISHED PROCEDURES FOR REPRODUCIBILITY?

Among the most popular strategies was having different lab members redo experiments.



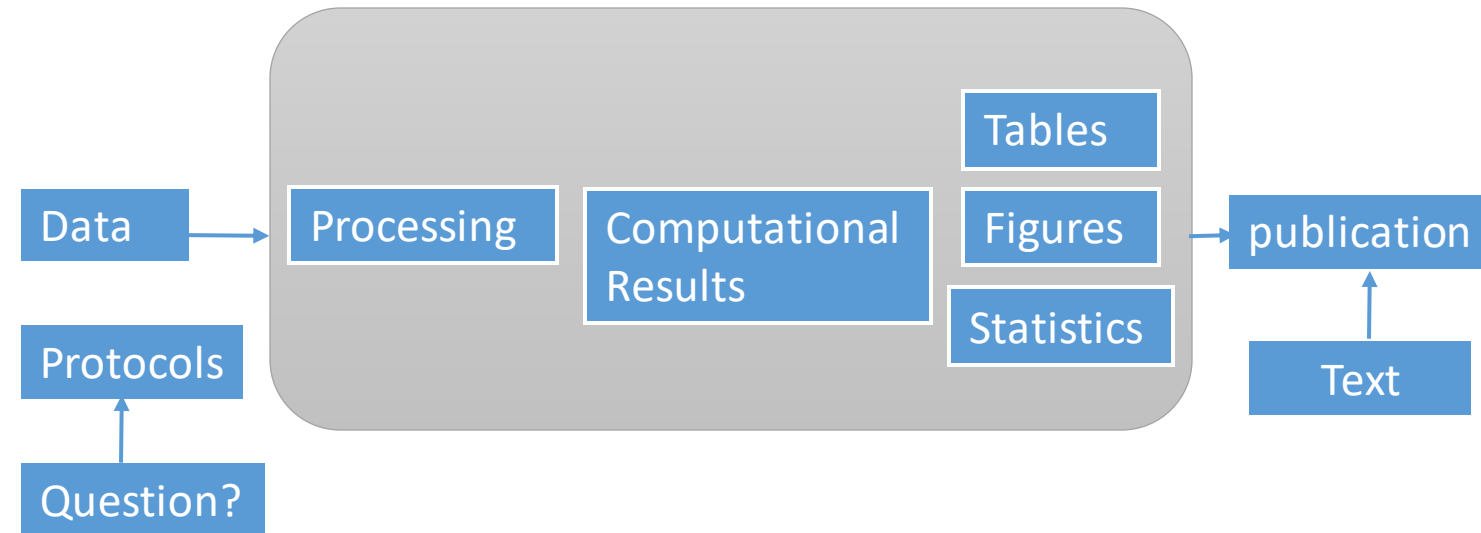
What I report (or think it is useful to report)



What I skipped reporting
(data, pipeline, recipes,
codes, etc.)

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

Reproducibility 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

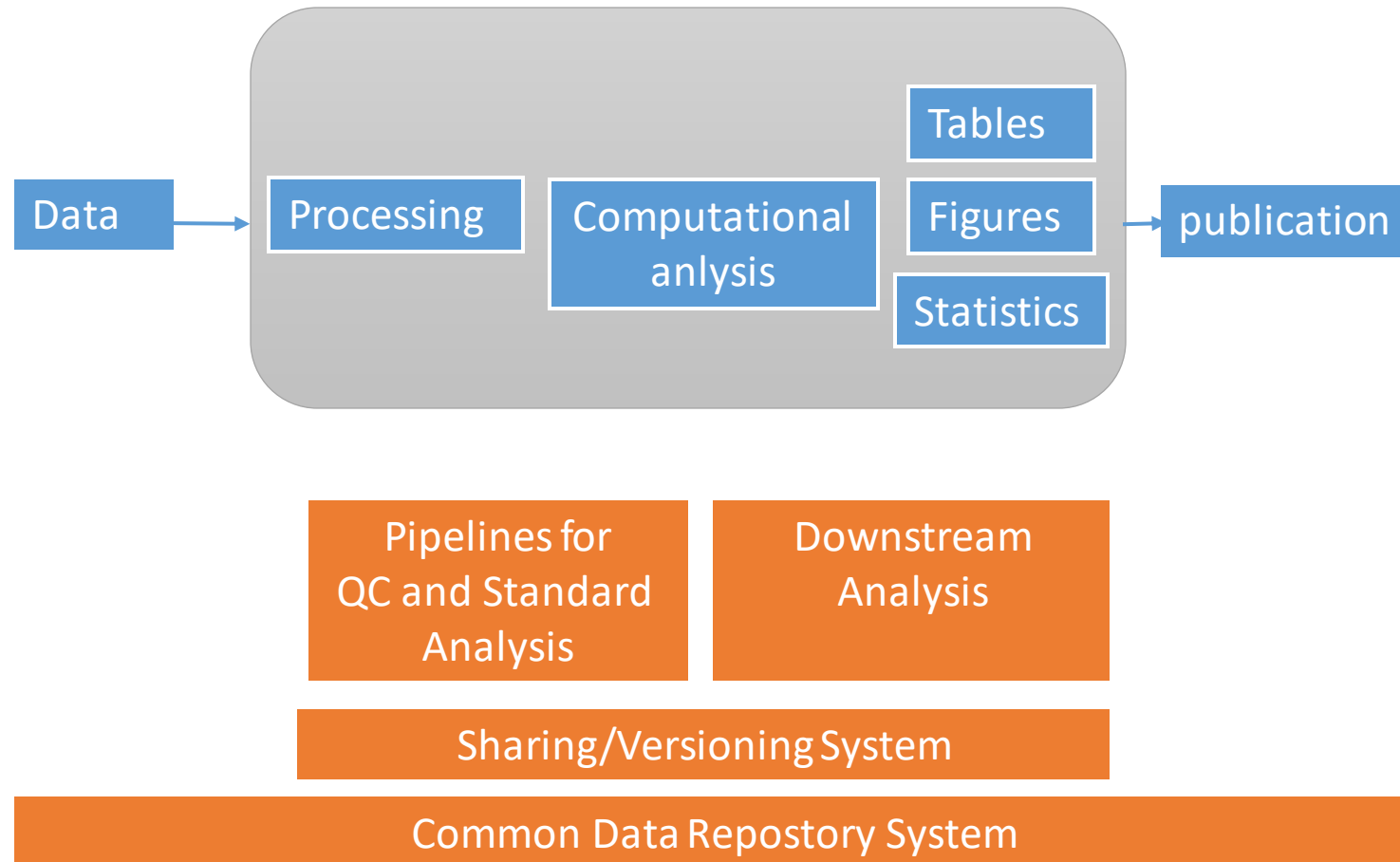
What does that mean in practice ?



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 analysts / production engineers / computer scientists
- Promote code review and common pipelines for every bioinformatician



A community effort to collect a curated set of analysis pipelines built using Nextflow.



- 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

The logo for Nextflow, featuring the word "next" in a green, lowercase, sans-serif font, followed by "flow" in a black, lowercase, sans-serif font. The "x" in "next" is stylized with a green circular arrow around it.

Versioning

Reproducibility requires codes versioning

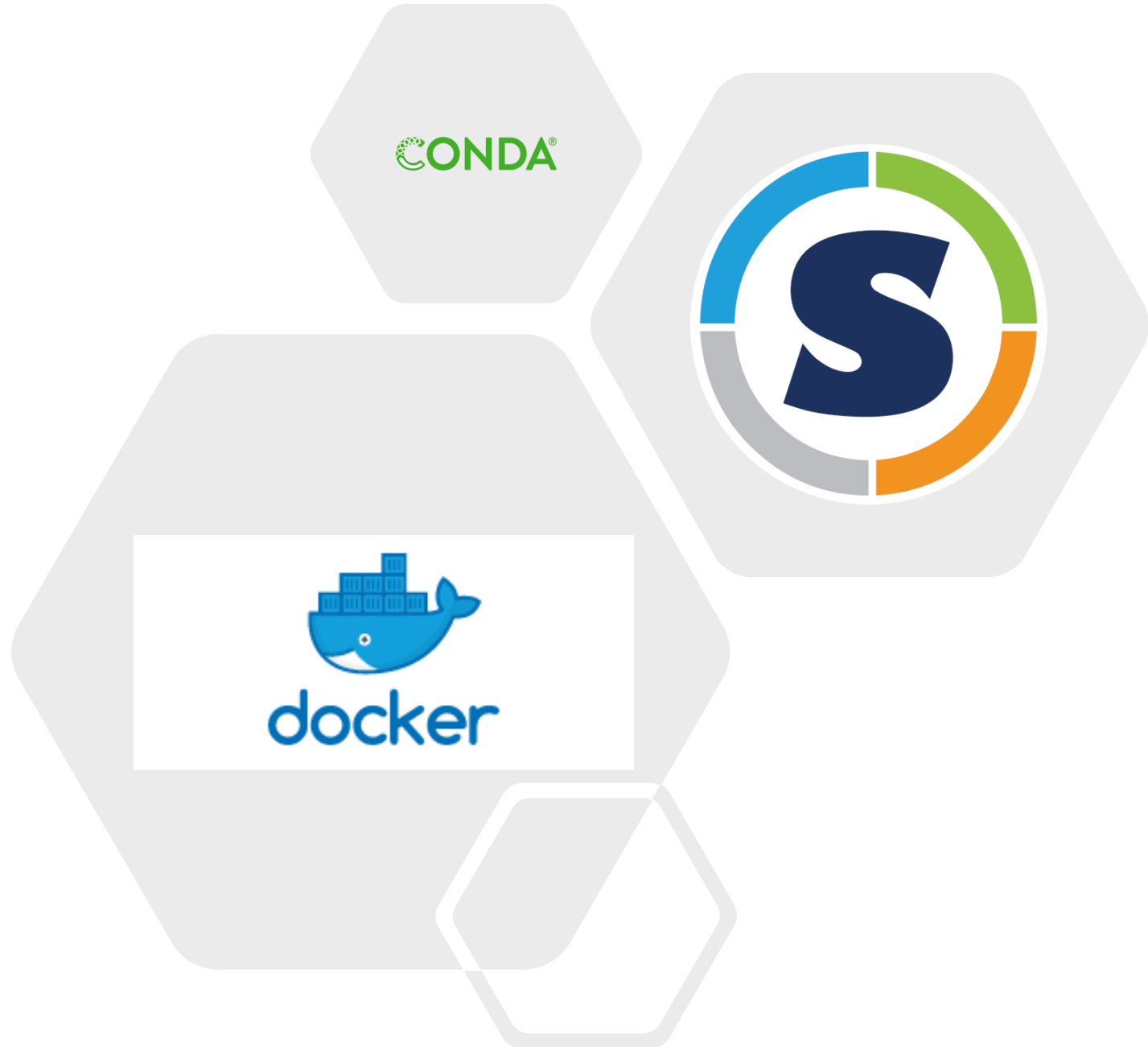
Git main features :

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

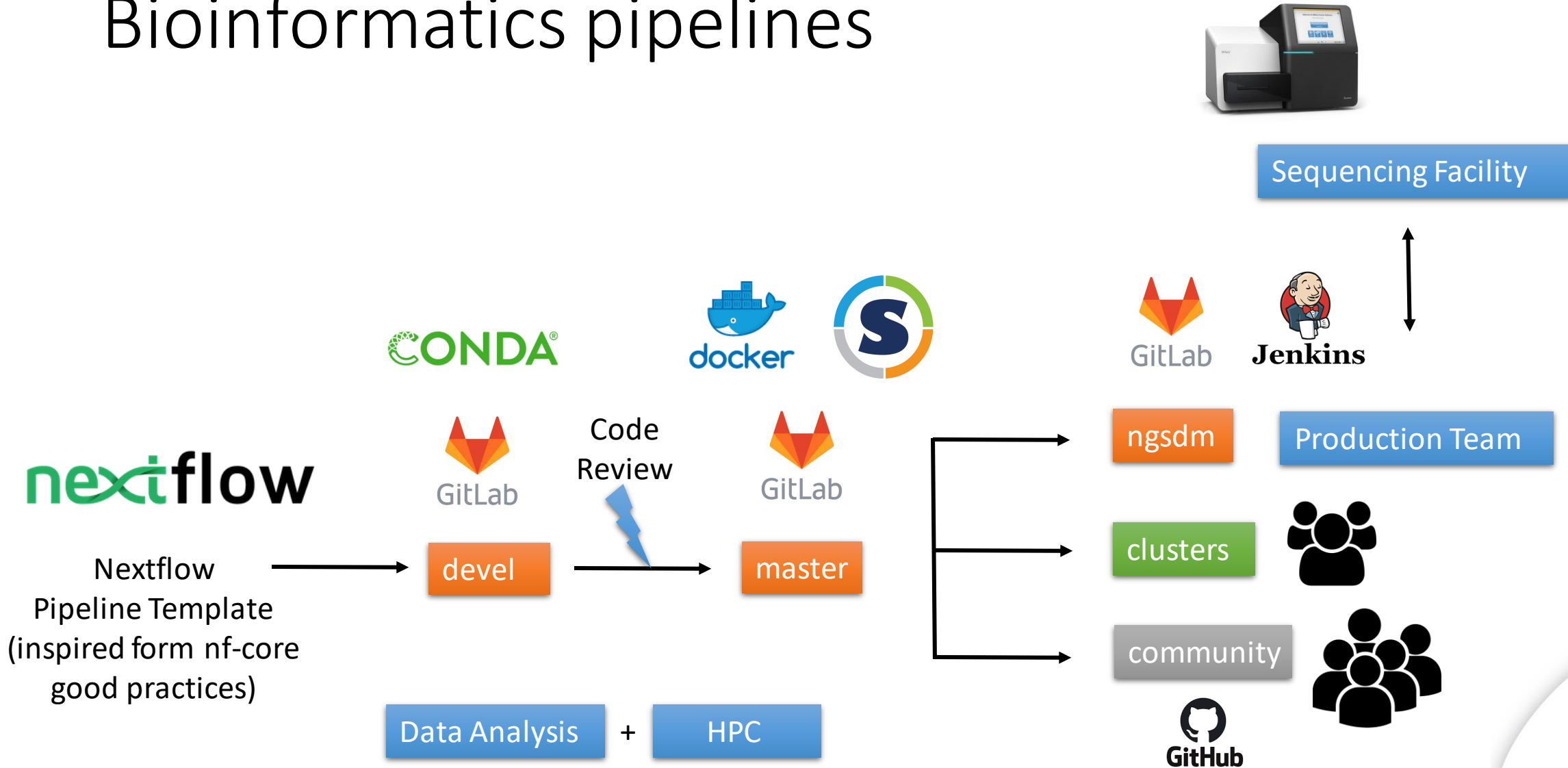


Containers

- 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 ?



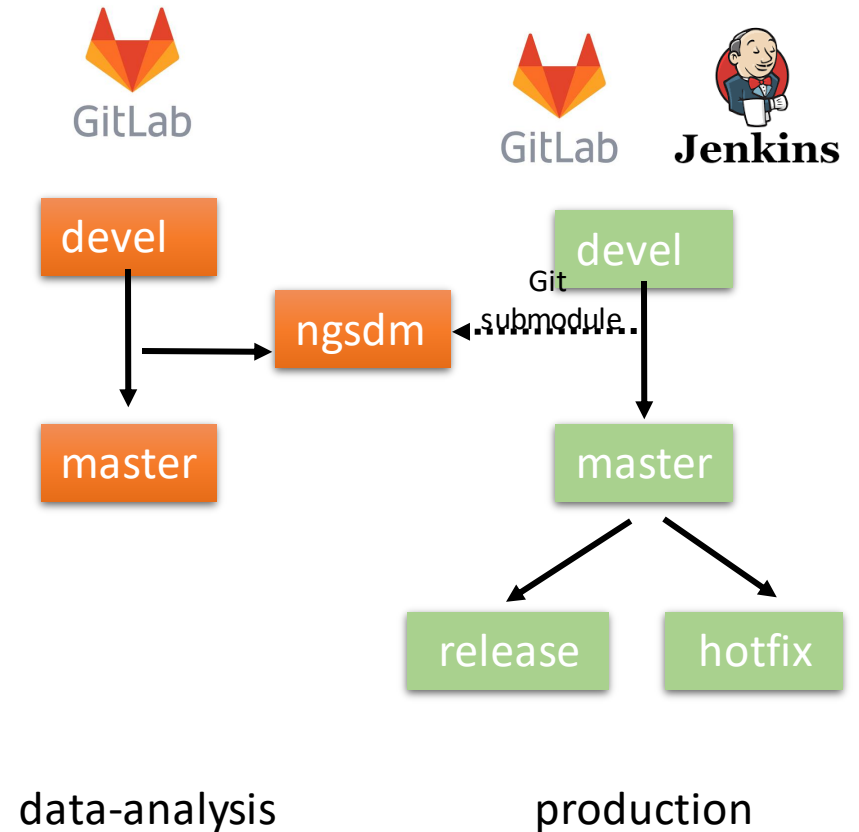
Bioinformatics pipelines



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

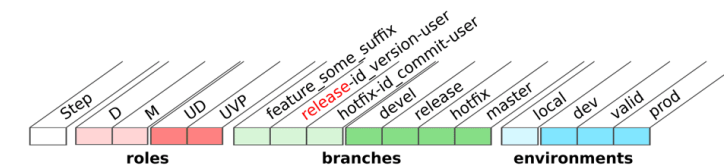


Production environment

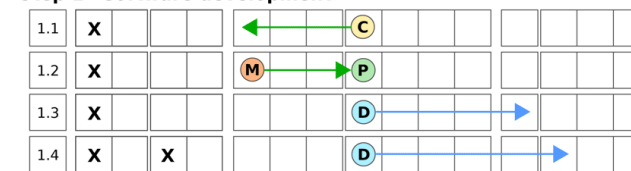
biogitflow: development workflow protocols for
bioinformatics pipelines with git and GitLab [version 1;
peer review: awaiting peer review]

✉ Choumouss Kamoun^{1-4*}, Julien Roméjon^{1-4*}, Henri de Soyres¹⁻⁴, Apolline Gallois¹⁻⁴, Elodie Girard¹⁻⁴, ✉ Philippe Hupé ¹⁻⁵

 create a branch  push  merge  tag  deploy  operational testing
→ action on the deployment environment → action on the git repository

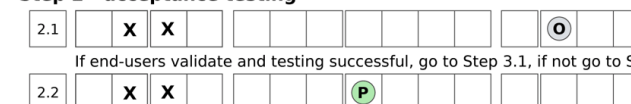


Step 1 - software development



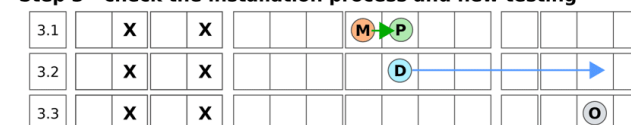
If testing successful go to Step 2, if not go to Step 1.1

Step 2 - acceptance testing

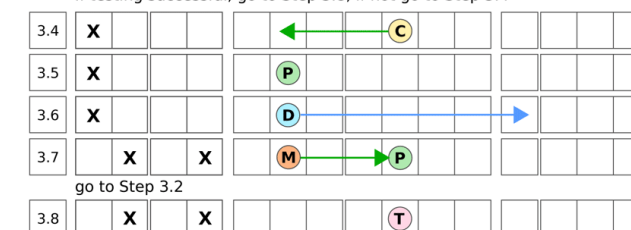


If end-users validate and testing successful, go to Step 3.1, if not go to Step 1.1

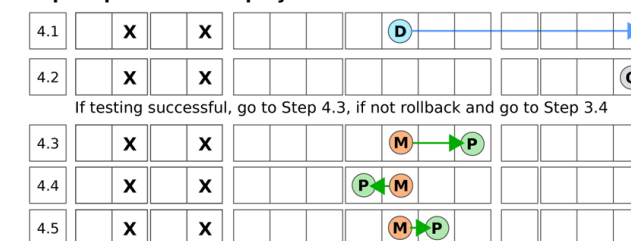
Step 3 - check the installation process and new testing



If testing successful, go to Step 3.8, if not go to Step 3.4



Step 4 - production deployment



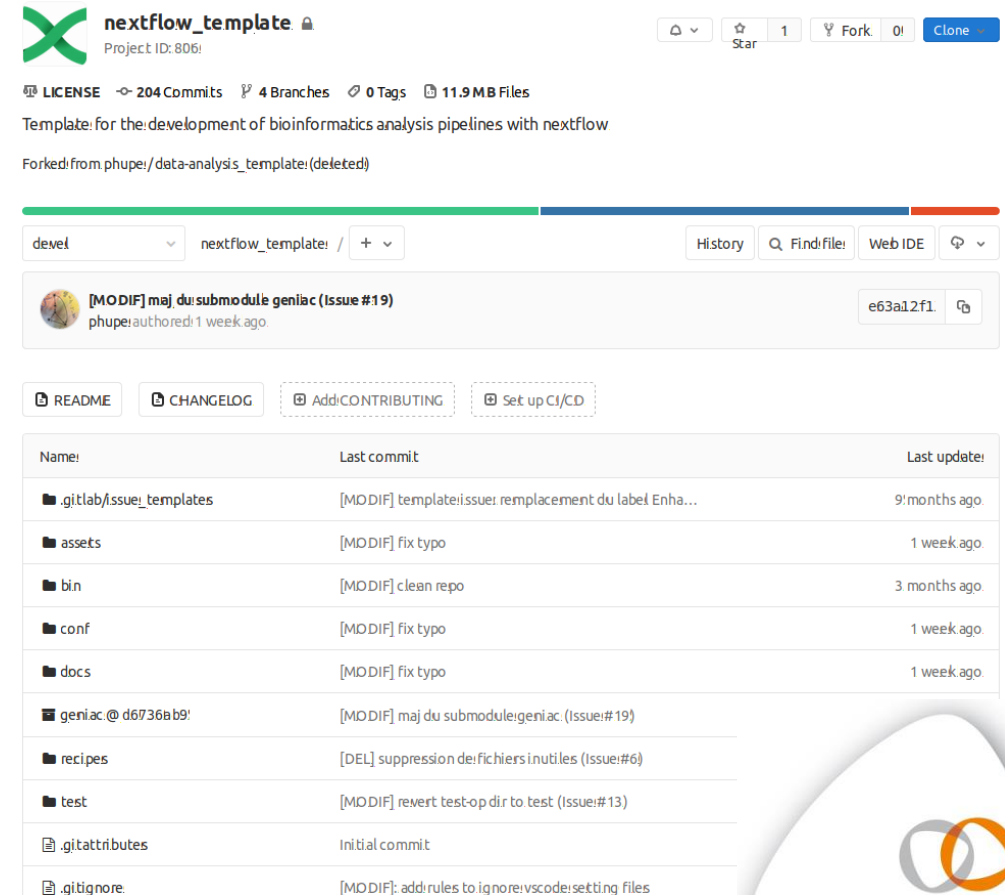
* using GitLab merge request ** only CHANGELOG is modified


Guidelines for Nextflow pipelines






Define a common Nextflow pipeline template with :

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

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



nextflow_template  Project ID: 8061



 LICENSE  204 Commits  4 Branches  0 Tags  11.9 MB Files

Template: for the development of bioinformatics analysis pipelines with nextflow





Forked from: phupei/data-analysis_template (deleted)











devel nextflow_template / +

History Find file Web IDE

 [MODIF] maj du submodule geniac (Issue #19) e63a12f1 

phupei authored 1 week ago

 README  CHANGELOG  Add CONTRIBUTING  Set up CI/CD

Name	Last commit	Last update
 .gitlab/issue_templates	[MODIF] template issue: remplacement du label Enha...	9 months ago
 assets	[MODIF] fix typo	1 week ago
 bin	[MODIF] clean repo	3 months ago
 conf	[MODIF] fix typo	1 week ago
 docs	[MODIF] fix typo	1 week ago
 geniac@d6736bb9	[MODIF] maj du submodule geniac (Issue #19)	
 recipes	[DEL] suppression de fichiers inutiles (Issue #6)	
 test	[MODIF] revert test-op dir to test (Issue #13)	
 .gitattributes	Initial commit	
 .gitignore	[MODIF] add rules to ignore vscode setting files	

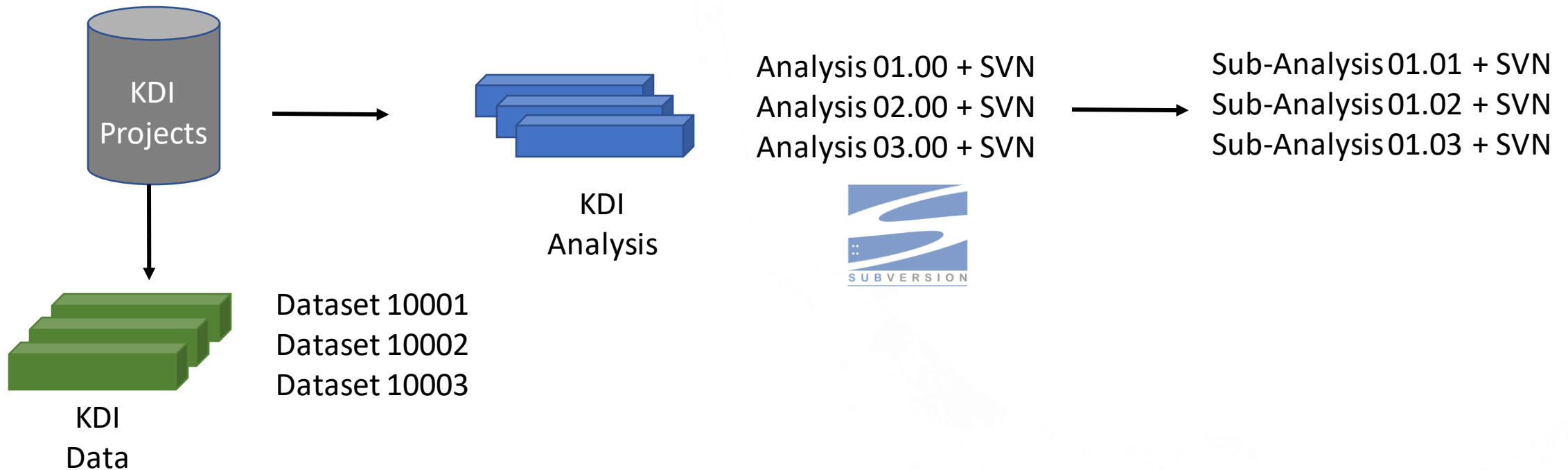
List of available Nextflow pipelines

<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 / with(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

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 dependencies 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 management + containers
- Continuous integration

Downstream analysis

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

nextflow

CONDA®



GitLab



GitHub





Many Thanks
