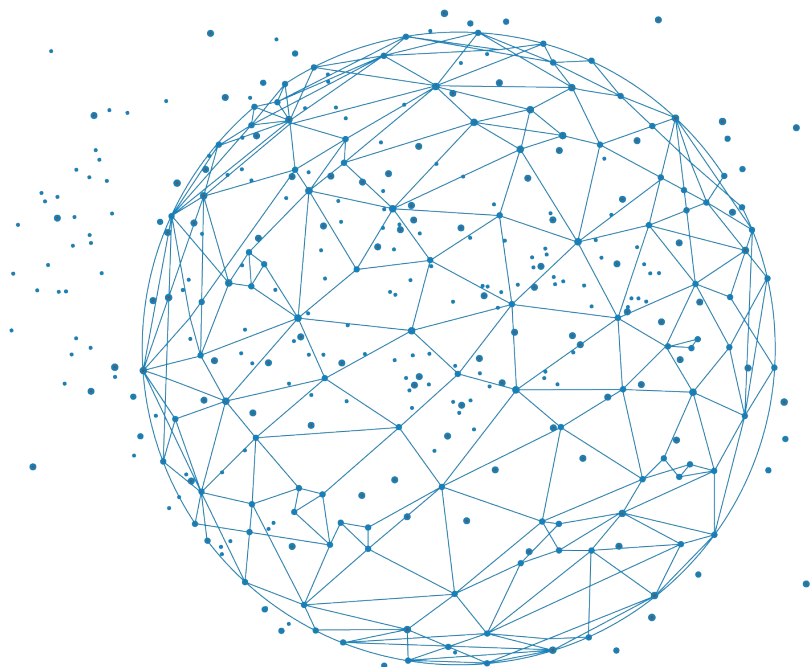




FAIR Bioinfo 2024 - Strasbourg



FAIR Bioinfo 2024 Conclusion

Equipe pédagogique : V. Cognat, T. Denecker, J. Seiler
Helpers : L. Bouri, A. Simon

BiGEst





Introduction to FAIR & Open Science

Notebooks with Jupyter

Getting started with conda

Getting started with Git and Github

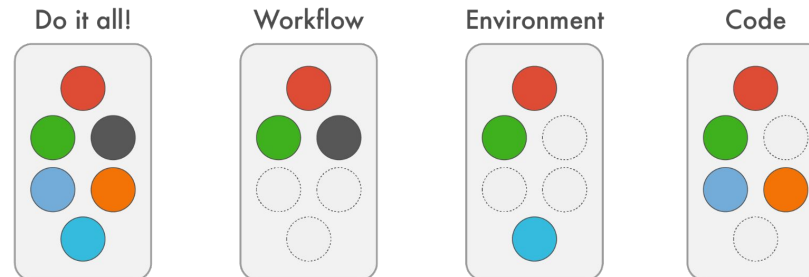
Distributing your pipeline with Docker and Apptainer

Creating a Snakemake workflow

IFB calculation cluster (SLURM)

GitHub Pages and Zenodo

How to make a computational research project reproducible using several different tools



https://nbis-reproducible-research.readthedocs.io/en/course_2104/introduction/

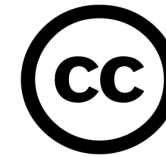


F 
Findable

A 
Accessible

I 
Interoperable

R 
Reusable





F: Software, and its associated metadata, is easy for both humans and machines to find.

F1. Software is assigned a globally unique and persistent identifier.

F1.1. Components of the software representing levels of granularity are assigned distinct identifiers.

F1.2. Different versions of the software are assigned distinct identifiers.

F2. Software is described with rich metadata.

F3. Metadata clearly and explicitly include the identifier of the software they describe.

F4. Metadata are FAIR, searchable and indexable.



Barker, M., Chue Hong, N.P., Katz, D.S. et al. Introducing the FAIR Principles for research software. Sci Data 9, 622 (2022). <https://doi.org/10.1038/s41597-022-01710-x>



A: Software, and its metadata, is retrievable via standardised protocols.

A1. Software is retrievable by its identifier using a standardised communications protocol.



A1.1. The protocol is open, free, and universally implementable.



A1.2. The protocol allows for an authentication and authorization procedure, where necessary.



A2. Metadata are accessible, even when the software is no longer available.



Barker, M., Chue Hong, N.P., Katz, D.S. et al. Introducing the FAIR Principles for research software. *Sci Data* 9, 622 (2022). <https://doi.org/10.1038/s41597-022-01710-x>



I: Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards.

I1. Software reads, writes and exchanges data in a way that meets domain-relevant community standards.

I2. Software includes qualified references to other objects.



Barker, M., Chue Hong, N.P., Katz, D.S. et al. Introducing the FAIR Principles for research software. Sci Data 9, 622 (2022). <https://doi.org/10.1038/s41597-022-01710-x>



R: Software is both usable (can be executed) and reusable (can be understood, modified, built upon, or incorporated into other software).

R1. Software is described with a plurality of accurate and relevant attributes.

R1.1. Software is given a clear and accessible license.

R1.2. Software is associated with detailed provenance.

R2. Software includes qualified references to other software.

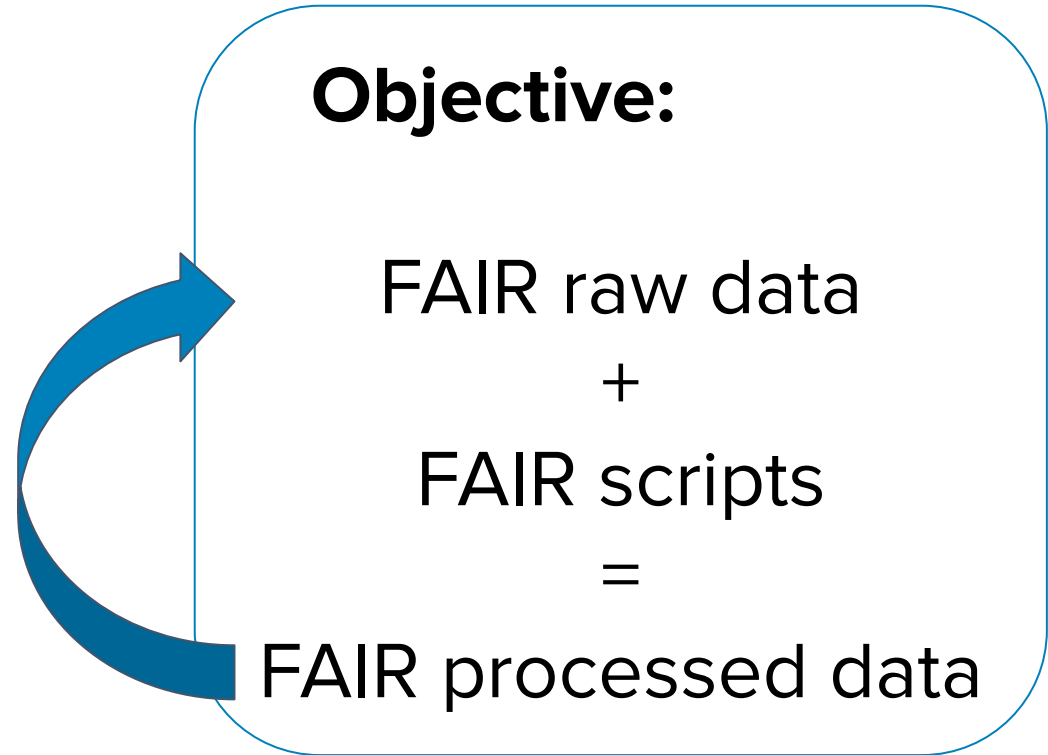
R3. Software meets domain-relevant community standards.



Barker, M., Chue Hong, N.P., Katz, D.S. et al. Introducing the FAIR Principles for research software. Sci Data 9, 622 (2022). <https://doi.org/10.1038/s41597-022-01710-x>



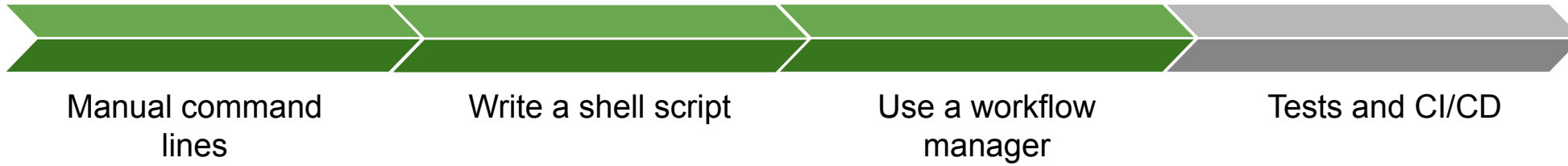
A virtuous
cycle



Is it possible to go further?



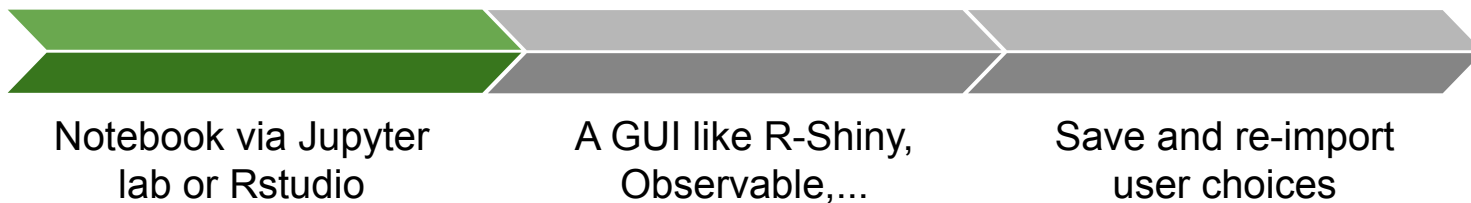
Automation



Softwares



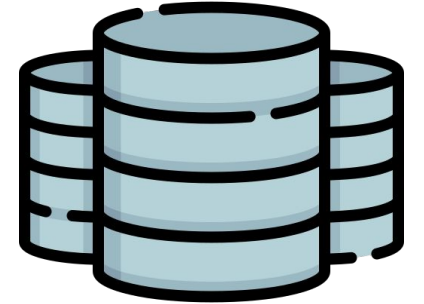
Interactive user analysis





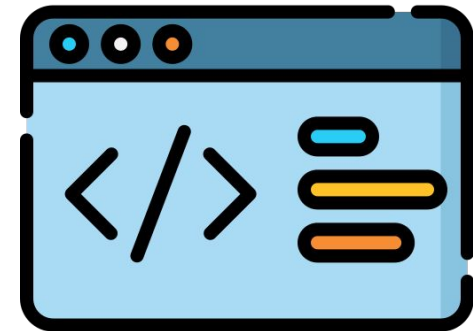
Data

- What if the data is not 100% FAIR?
- How do you manage large volumes of data?
- How do you manage data and metadata updates?



Code

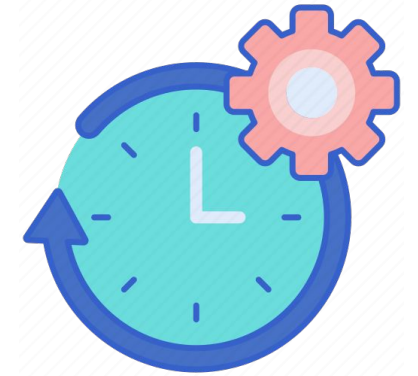
- How can we be sure that the code will always be accessible?
- Is it acceptable if adaptations have to be made? At what point?
- Should all the code be provided (for value enhancement, creation of a start-up, etc.)?





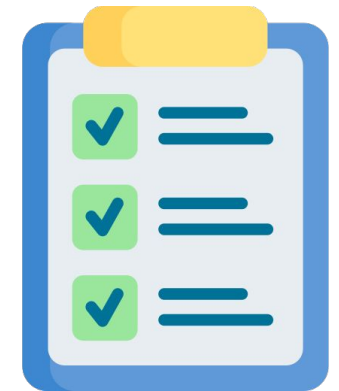
Competence and sensitivity

- Willingness but technical inability to do so
- "Why do it, there's no point".
- Too time-consuming



Coverage

- Should everything be reproducible?
- When should it be done?
- At the beginning? But what if it doesn't work?
- At the end?





Reproducibility to the exact bit?

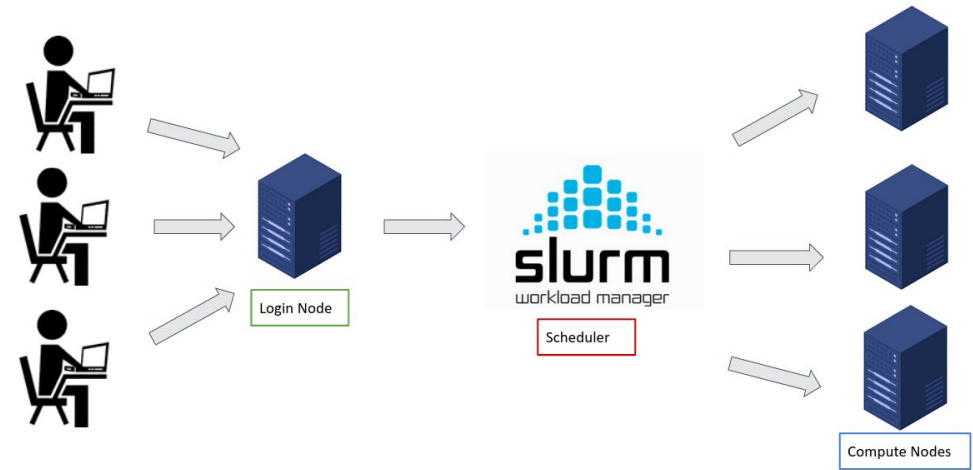
Container uses some resources of the support machine

⇒ version control of the env : Nix, Guix



HPC and parallelization?

Loss of computational order,
multithreading, identical hardware?







IFB training courses on FAIR

1. FAIR principles for the management of research data in the life sciences (**FAIRdata**)
2. FAIR principles in a Bioinformatics project (**FAIRbioinfo**)



<https://moodle.france-bioinformatique.fr/>



INSTITUT FRANÇAIS DE BIOINFORMATIQUE

<https://www.france-bioinformatique.fr/formations>



Our helpers

Alix and Laurent

IFB Core Cluster taskforce

Julien, Gildas, and all those who provide in the shadows

Organisations

BiGEst, IGBMC, IBMP, IFB