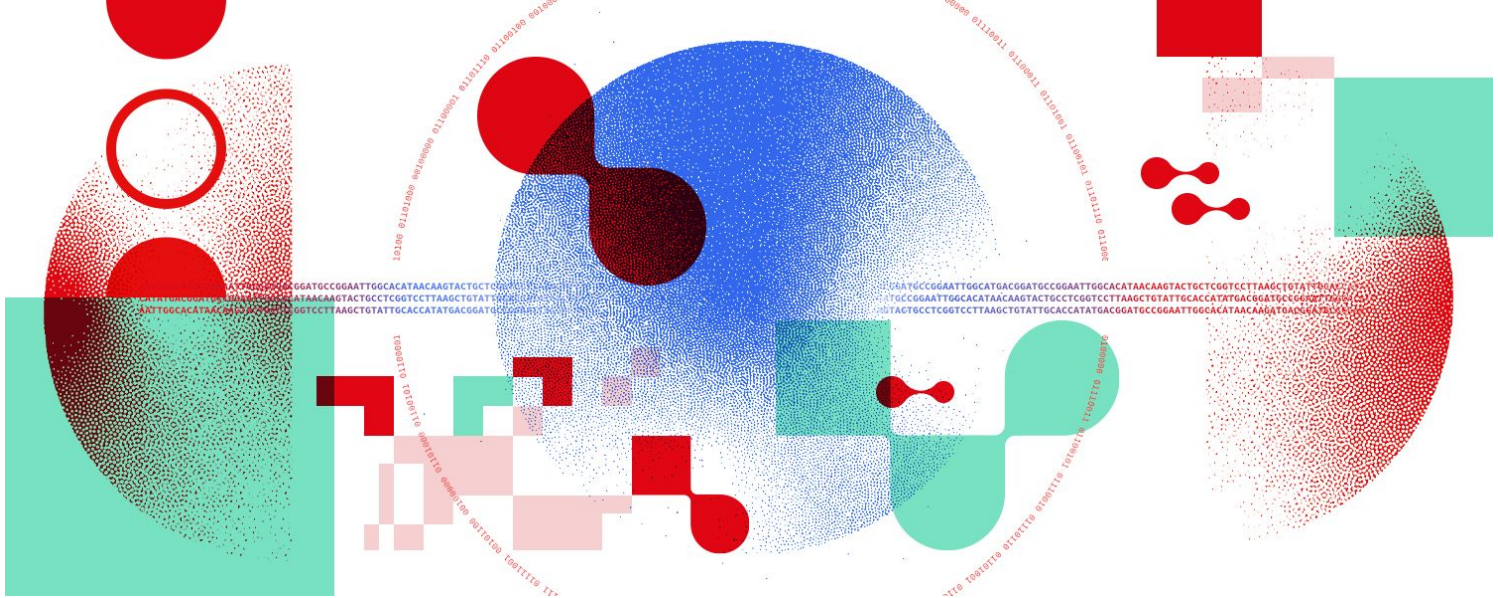


Snakemake for reproducible research

Introduction to Snakemake

Centre hospitalier universitaire vaudois

Empa
Materials Science and Technology

Antonin Thiébaut & Rafael Riudavets Puig
antonin.thiebaut@chuv.ch
Rafael.RiudavetsPuig@empa.ch

SIB
Swiss Institute of Bioinformatics

Reproducibility

- Question 1

What is reproducibility?

- Replicability vs repeatability vs reproducibility
- "Reproducibility is more or less the ability to draw similar conclusions from replicates studies"
  - Diaba-Nuhoho, P., Amponsah-Offeh, M., *BMC Research Notes* (2021), https://doi.org/10.1186/s13104-021-05875-3
- Key component of the scientific method, "cornerstone of science"

Reproducibility crisis

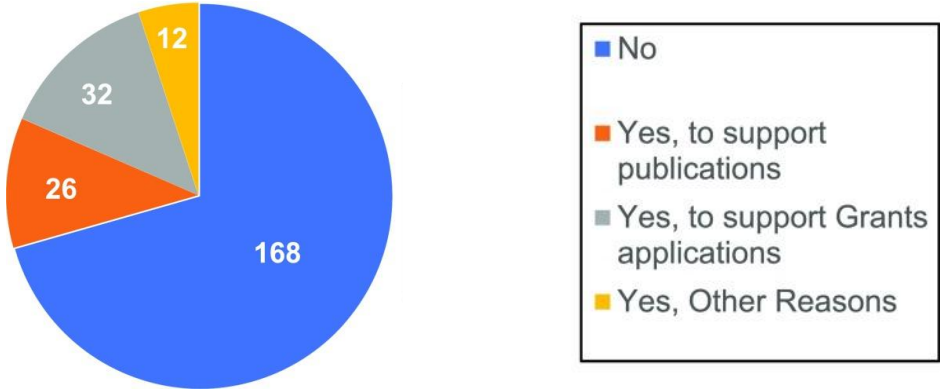
- Question 2

Is there a reproducibility crisis?

- Baker, M., *Nature* (2016), https://doi.org/10.1038/533452a:
  - ~1600 researchers: 52% significant crisis, 38% slight crisis (90% in total)

- Alfredo Sánchez-Tójar, Universität Bielefeld:
  - Publication bias in ecology: https://www.youtube.com/watch?v=wdhzLrPUJJY
  - 83 articles of 3 fields: ~30% of partial replication, **0% of true replication**

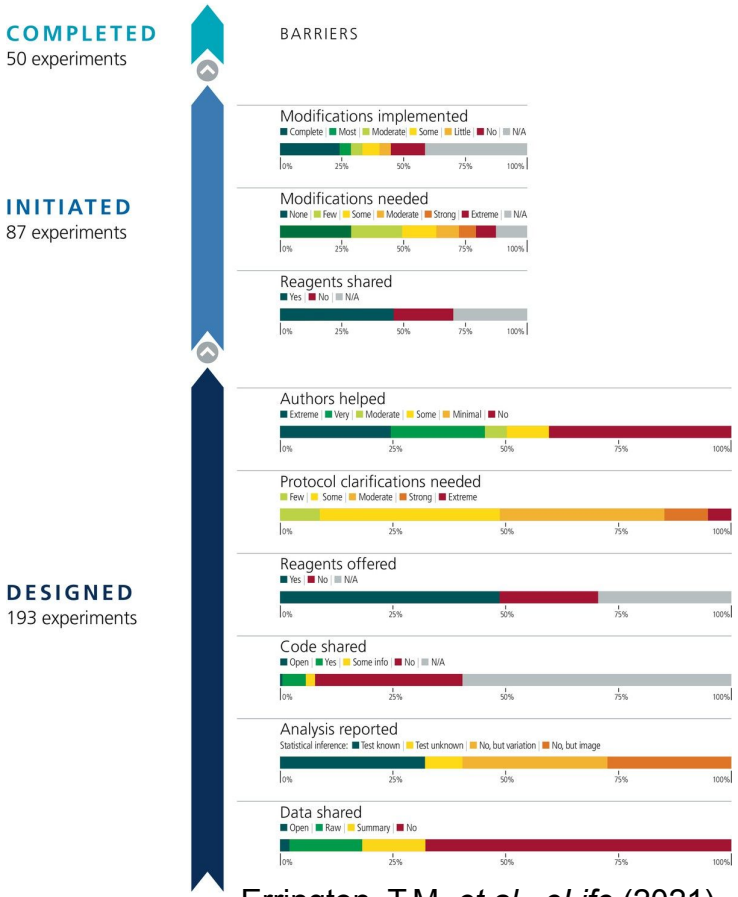
- Knudtson, K. L., *J Biomol Tech.* (2019), https://doi.org/10.7171%2Fjbt.19-3003-001
  - Has your core's rigor and reproducibility practice statement ever been requested?



Why is that?

Why is that?

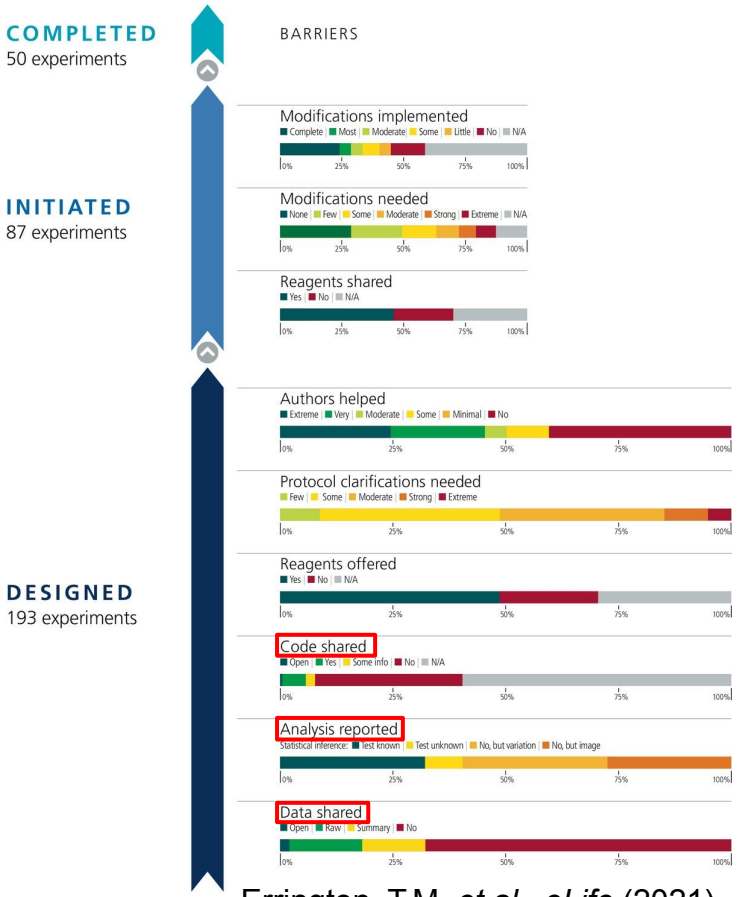
- Absence of knowledge/infrastructure
- Questionable research practices and fraud
- Statistical issues
  - Low statistical power
  - Statistical heterogeneity
- Publication system in science
  - Publication bias (non-significant results/unoriginal replications not published)
  - "Publish or perish"
  - Standards of reporting, open-access



Errington, T.M. *et al.*, *eLife* (2021), https://doi.org/10.7554/eLife.67995

Why is that?

- Absence of knowledge/infrastructure
- Questionable research practices and fraud
- Statistical issues
  - Low statistical power
  - Statistical heterogeneity
- Publication system in science
  - Publication bias (non-significant results/unoriginal replications not published)
  - "Publish or perish"
  - Standards of reporting, open-access



Errington, T.M. *et al.*, *eLife* (2021), https://doi.org/10.7554/eLife.67995

Workflow Management Systems (WMS)

- Question 3

What do WMS bring?

- WMS can can solve several hidden reproducibility issues:
  - Entirely:
    - OS
    - Version
    - Language
    - Readability
    - Availability
  - Partially (at least):
    - File format
    - Metadata
    - Parameters/options

What is the idea behind WMS?

- Provide reproducible and scalable infrastructures to set-up, perform, and monitor a defined sequence of tasks

What is the idea behind WMS?

- Provide reproducible and scalable infrastructures to set-up, perform, and monitor a defined sequence of tasks

	Nextflow	Snakemake	Galaxy
Language (How to code the worklow?)	Groovy (~ Java)	Extension of Python	Java + Python

What is the idea behind WMS?

- Provide reproducible and scalable infrastructures to set-up, perform, and monitor a defined sequence of tasks

	Nextflow	Snakemake	Galaxy
Language (How to code the worklow?)	Groovy (~ Java)	Extension of Python	Java + Python
Execution system (How are tasks organised and run?)	Join processes manually	Automatically resolve dependencies from last outputs to first inputs	Web-based GUI, manual task organisation but no need for programming knowledge

What is the idea behind WMS?

- Provide reproducible and scalable infrastructures to set-up, perform, and monitor a defined sequence of tasks

	Nextflow	Snakemake	Galaxy
Language (How to code the worklow?)	Groovy (~ Java)	Extension of Python	Java + Python
Execution system (How are tasks organised and run?)	Join processes manually	Automatically resolve dependencies from last outputs to first inputs	Web-based GUI, manual task organisation but no need for programming knowledge

Python, you said?

- Question 4

Some interesting features of Snakemake

- **Python-based**: concise and readable = user-friendly

Some interesting features of Snakemake

- **Python-based**: concise and readable = user-friendly
- Easily deployable/executable **locally** or **remotely** (computation clusters and clouds)

Some interesting features of Snakemake

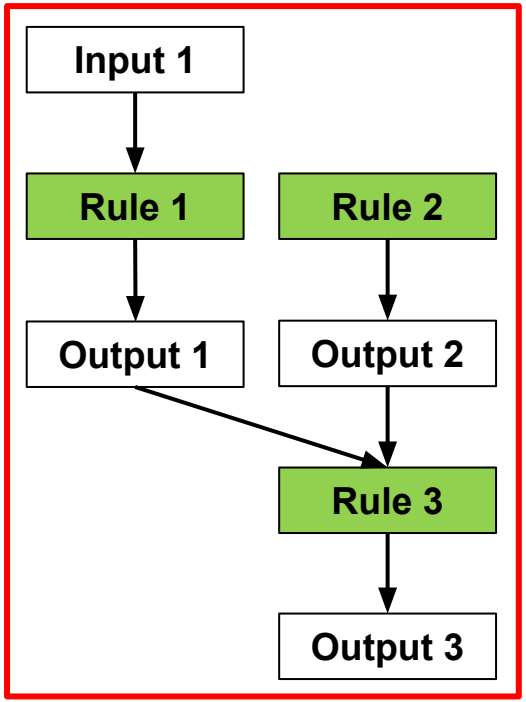
- **Python-based**: concise and readable = user-friendly
- Easily deployable/executable **locally** or **remotely** (computation clusters and clouds)
- Integrated package management via **conda/mamba** (package manager) and **apptainer** (container manager)

Some interesting features of Snakemake

- **Python-based**: concise and readable = user-friendly
- Easily deployable/executable **locally** or **remotely** (computation clusters and clouds)
- Integrated package management via **conda/mamba** (package manager) and **apptainer** (container manager)
- Once you have downloaded a workflow, it is easy to:
  - Run Snakemake in a **strictly controlled environment** (OS, software, versions, parameters…)
  - Efficiently and automatically reproduce analyses, results and figures

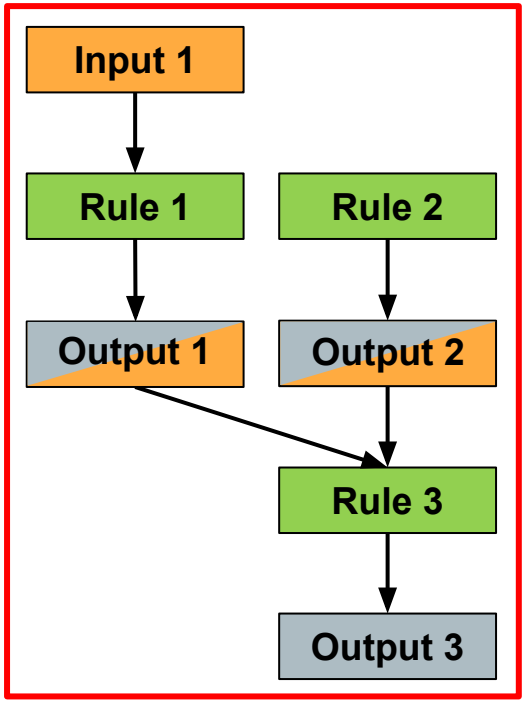
How does Snakemake work?

- Workflow:
  - Network of dependent **rules**



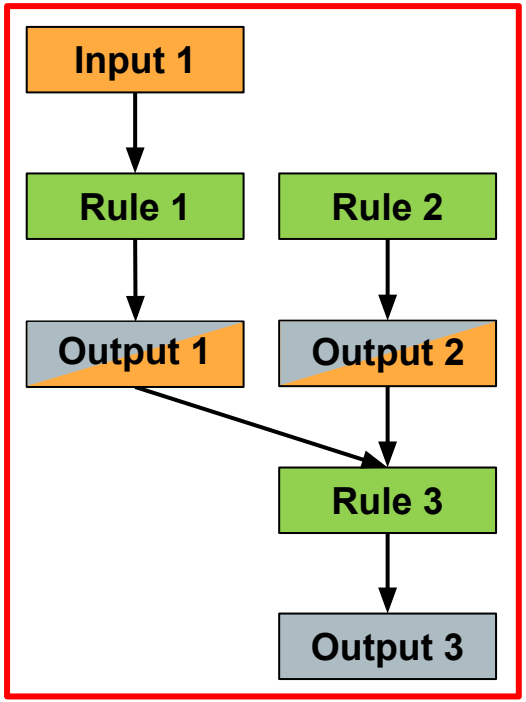
How does Snakemake work?

- Workflow:
  - Network of dependent **rules**
- **Rule**:
  - Smallest part of a workflow
  - **Set of instructions** to create one or more **output(s)** from zero or more **input(s)**



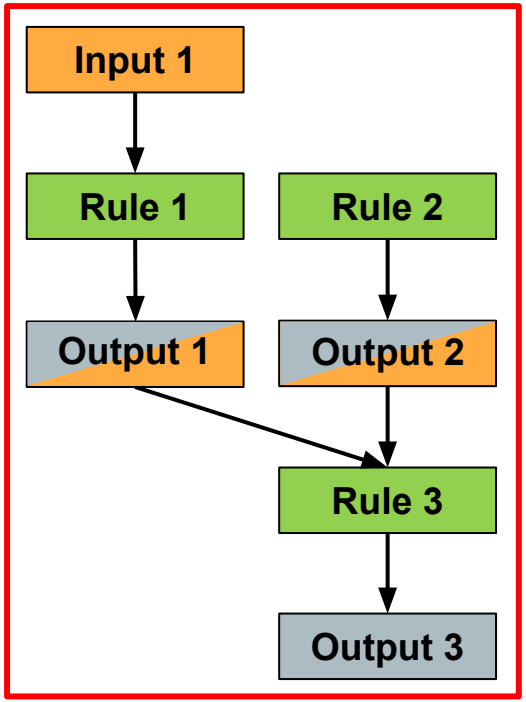
How does Snakemake work?

- Workflow:
  - Network of dependent **rules**
- **Rule**:
  - Smallest part of a workflow
  - **Set of instructions** to create one or more **output(s)** from zero or more **input(s)**
- Job:
  - **Execution** of a **rule** with **specific** input(s)/output(s)
  - Success conditions:
    - **No error**
    - **All expected outputs present**



How does Snakemake work?

- Workflow:
  - Network of dependent **rules**
- **Rule**:
  - Smallest part of a workflow
  - **Set of instructions** to create one or more **output(s)** from zero or more **input(s)**
- Job:
  - **Execution** of a **rule** with **specific** input(s)/output(s)
  - Success conditions:
    - **No error**
    - **All expected outputs present**
- Directed Acyclic Graph (DAG) determined from the required outputs.



Why is a DAG useful?

- Skip parts of the DAG to avoid recomputing ➡ Save time and resources (CPU, memory, energy, money)
- Change/add inputs to existing analyses without re-running everything
- Resume running a workflow that failed part-way

What does Snakemake really look like?

```
rule rename_file:
    input:
        'data/file.txt'
    output:
        'results/renamed_file.txt'
    shell:
        'mv data/file.txt results/file_renamed.txt'
```

Snakemake lingo cheatsheet

- Snakemake keyword
- Rule name (user-defined)
- Snakemake directives
- Directives values:
  - Object
  - String (file path)
  - Instruction (command)
  - Numeric values (seen later)
- Wildcards
- Placeholders

```
rule rename_file:
    input:
        'data/{file}.txt'
    output:
        'results/{file}.txt'
    shell:
        'mv {input} {output}'
```

Exercises

- Throughout the day:
  - Develop a simple RNAseq analysis workflow, from reads (fastq files) to Differentially Expressed Genes (DEG)

- For this session:
  - Understand the structure of a Snakemake workflow
  - Create your first rules and Snakefile
  - Chain rules together
  - Run your first workflow