

Snakemake for reproducible research

Introduction to Snakemake



UNIL | Université de Lausanne

Antonin Thiébaud

antonin.thiebaut@unil.ch



Swiss Institute of
Bioinformatics

Reproducibility

- Question 1

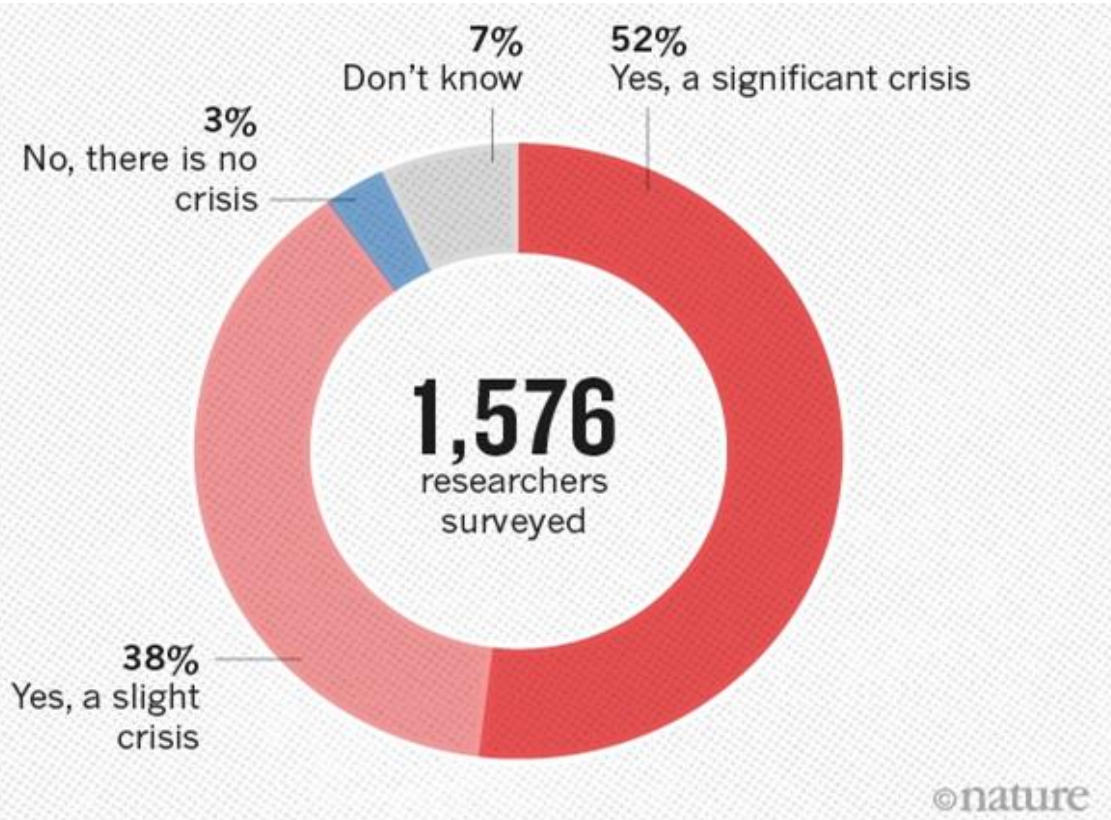
What is reproducibility?

- 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”

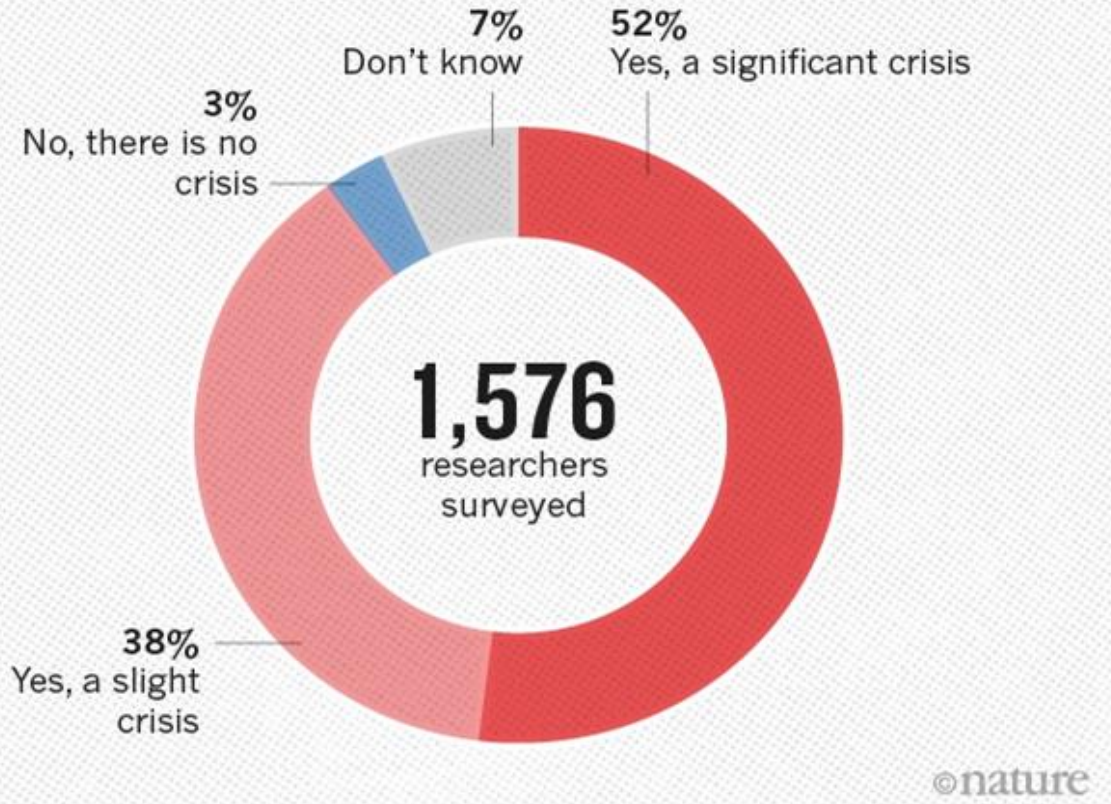
Reproducibility crisis

- Question 2

Is there a reproducibility crisis?



Is there a reproducibility crisis?



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

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

COMPLETED

50 experiments

INITIATED

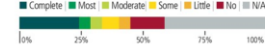
87 experiments

DESIGNED

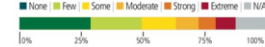
193 experiments

BARRIERS

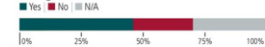
Modifications implemented



Modifications needed



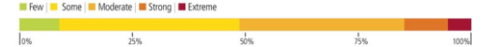
Reagents shared



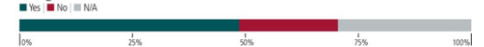
Authors helped



Protocol clarifications needed



Reagents offered



Code shared



Analysis reported



Data shared



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

COMPLETED

50 experiments

INITIATED

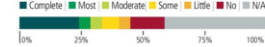
87 experiments

DESIGNED

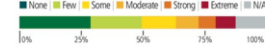
193 experiments

BARRIERS

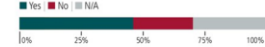
Modifications implemented



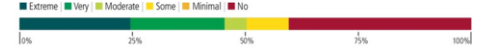
Modifications needed



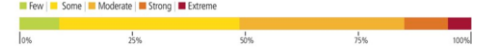
Reagents shared



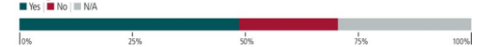
Authors helped



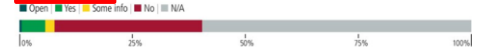
Protocol clarifications needed



Reagents offered



Code shared



Analysis reported



Data shared

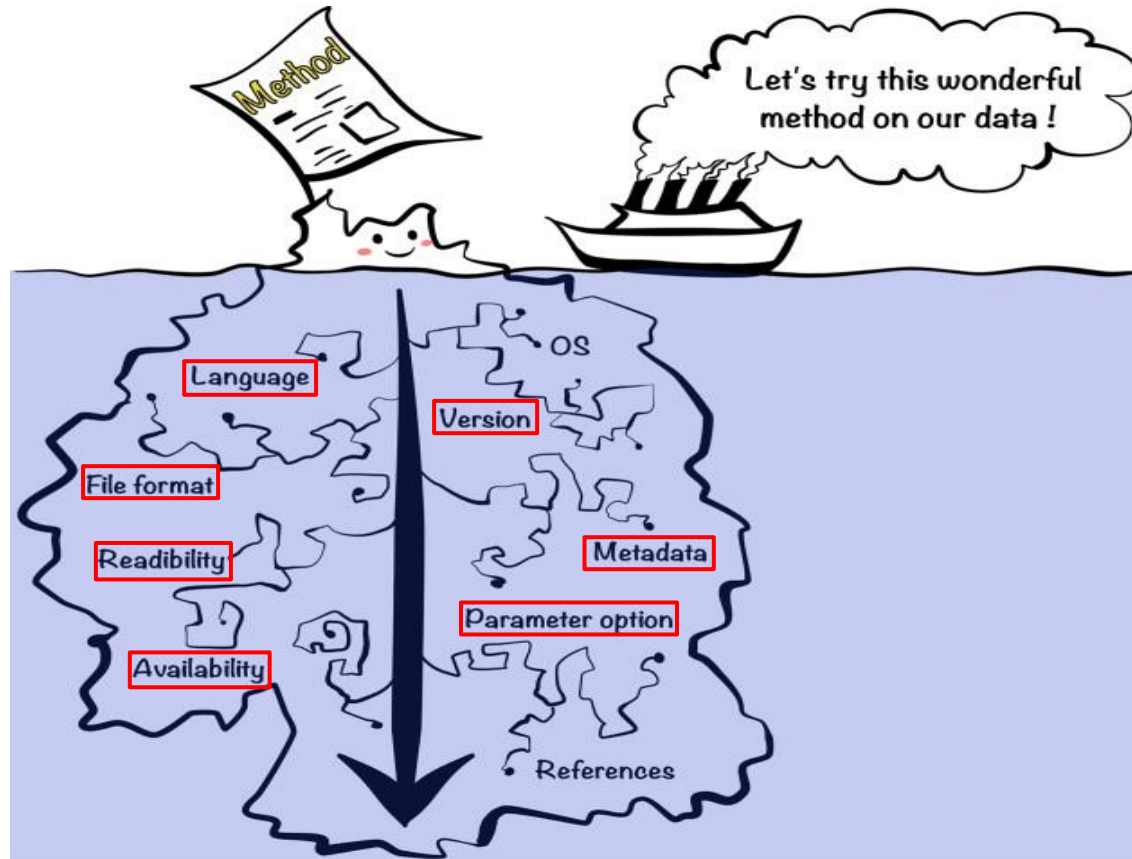


Errington, T.M. *et al.*, *eLife* (2021),
<https://doi.org/10.7554/eLife.67995>

Workflow Management Systems (WMS)

- Question 3

What do WMS bring?



What is the idea behind WMS?

- Implement reproducible, portable, and scalable data analyses

What is the idea behind WMS?

- Implement reproducible, portable, and scalable data analyses
- Two parts:
 - Workflow definition language \Rightarrow implement the workflow
 - Workflow execution system \Rightarrow organise the jobs and run the workflow in variable environments

What is the idea behind WMS?

- Implement reproducible, portable, and scalable data analyses
- Two parts:
 - Workflow definition language ⇒ implement the workflow
 - Workflow execution system ⇒ organise the jobs and run the workflow in variable environments
- Multiple systems exist. Most popular ones are:
 - [Nextflow](#): “top-down” approach, implemented in Groovy (~ Java)
 - [Snakemake](#): “bottom-up” approach resolving dependencies, implemented in Python
 - [Galaxy](#): web-based GUI to make computational biology available to people without programming knowledge, implemented in... Python and Java!

What is the idea behind WMS?

- Implement reproducible, portable, and scalable data analyses
- Two parts:
 - Workflow definition language ⇒ implement the workflow
 - Workflow execution system ⇒ organise the jobs and run the workflow in variable environments
- Multiple systems exist. Most popular ones are:
 - [Nextflow](#): “top-down” approach, implemented in Groovy (~ Java)
 - [Snakemake](#): “bottom-up” approach resolving dependencies, implemented in Python
 - [Galaxy](#): web-based GUI to make computational biology available to people without programming knowledge, implemented in... Python and Java!

Python, you said?

- Question 4

Overview of Snakemake's general features



- User-friendly language: overlay of **Python**

Overview of Snakemake's general features



- User-friendly language: overlay of **Python**
- Can be easily executed on local machines, HPCs, and clouds

Overview of Snakemake's general features



- User-friendly language: overlay of **Python**
- Can be easily executed on local machines, HPCs, and clouds
- Handles dependencies with **conda/mamba** (package manager) and **docker/singularity** (containers)

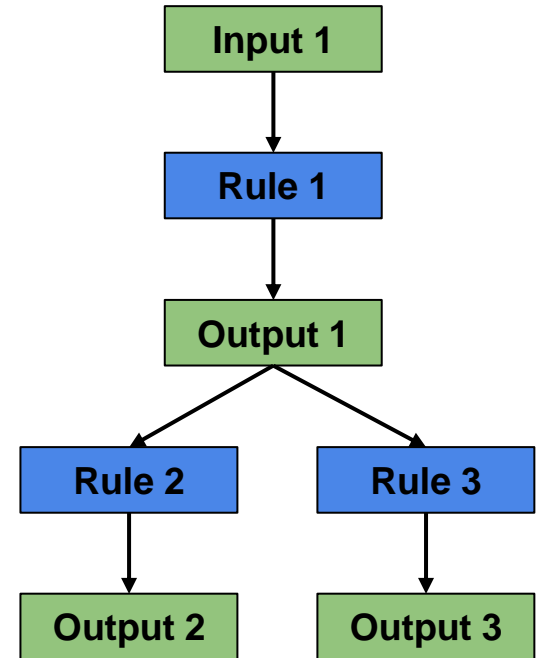
Overview of Snakemake's general features



- User-friendly language: overlay of **Python**
- Can be easily executed on local machines, HPCs, and clouds
- Handles dependencies with **conda/mamba** (package manager) and **docker/singularity** (containers)
- With Snakemake, conda, and docker installed, you can:
 - Download a workflow (e.g. from a Github or Gitlab repository)
 - Run Snakemake in a controlled environments (software, versions, parameters, OS...)
 - Automatically and efficiently reproduce all the analyses and results

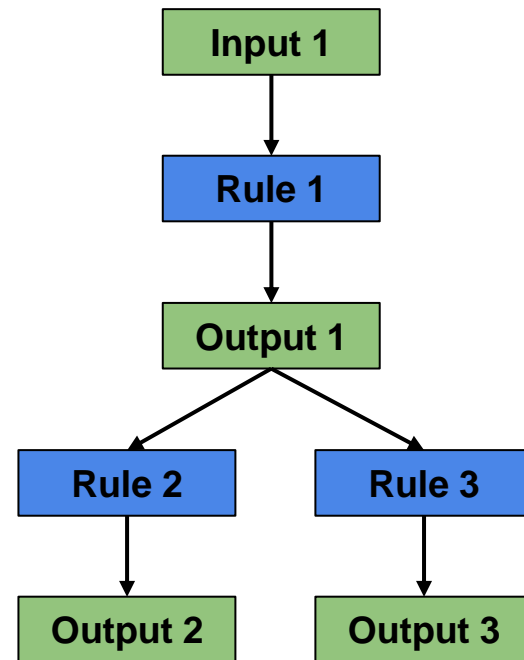
How does Snakemake work?

- Workflow:
 - Collection of **interdependent rules** to generate specific **outputs**



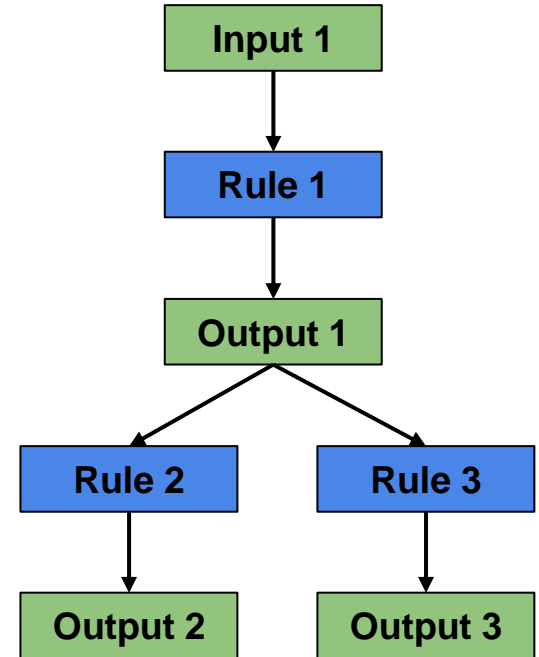
How does Snakemake work?

- Workflow:
 - Collection of **interdependent rules** to generate specific **outputs**
- Rule:
 - Basic workflow unit
 - **Template (recipe)** to produce an **output** (1 or more files)
 - *Can* use an **input**
 - Generates **jobs** when executed



How does Snakemake work?

- **Workflow:**
 - Collection of **interdependent rules** to generate specific **outputs**
- **Rule:**
 - Basic workflow unit
 - **Template (recipe)** to produce an **output** (1 or more files)
 - *Can use an **input***
 - Generates **jobs** when executed
- **Job:**
 - Single **execution** of a **rule** (apply the recipe to specific data)
 - Successful if **all outputs are present and no error**



What does Snakemake really look like?

```
rule first_step:
    input:
        'results/first_step.txt'
    output:
        'results/second_step.txt'
    shell:
        'cp results/first_step.txt results/second_step.txt'
```


Exercises

- Throughout the day:
 - Develop a simple RNAseq analysis workflow, from reads (fastq files) to Differentially Expressed Genes (DEG)
- For now:
 - Understand the structure of a Snakemake workflow
 - Create your first rules and Snakefile
 - Chain rules together
 - Run your first workflow

