Swiss Institute of Bioinformatics

INTRODUCTION TO SEQUENCING DATA ANALYSIS

Reproducible Computational Research

Deepak Tanwar
Frédéric Burdet

April 23-25, 2025
Adapted from previous year courses

elixir SWITZERLAND

Learning Objectives

Understand the difference between replicable and reproducible research.

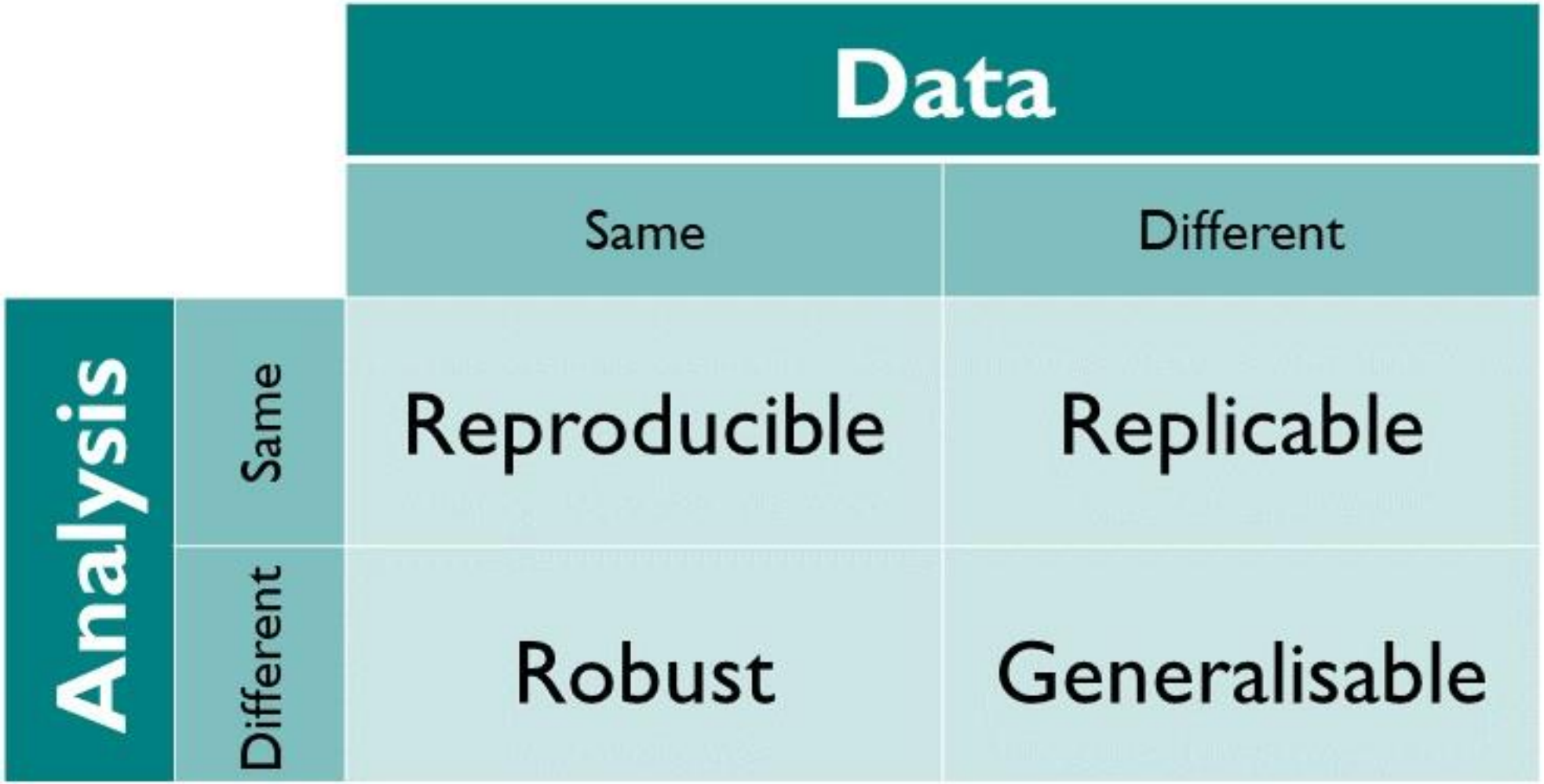
Learn about the reproducibility crisis and its impact on science.

Get introduced to literate programming for clearer, more reproducible code.

Apply simple rules to make your own code reproducible and easy to understand.

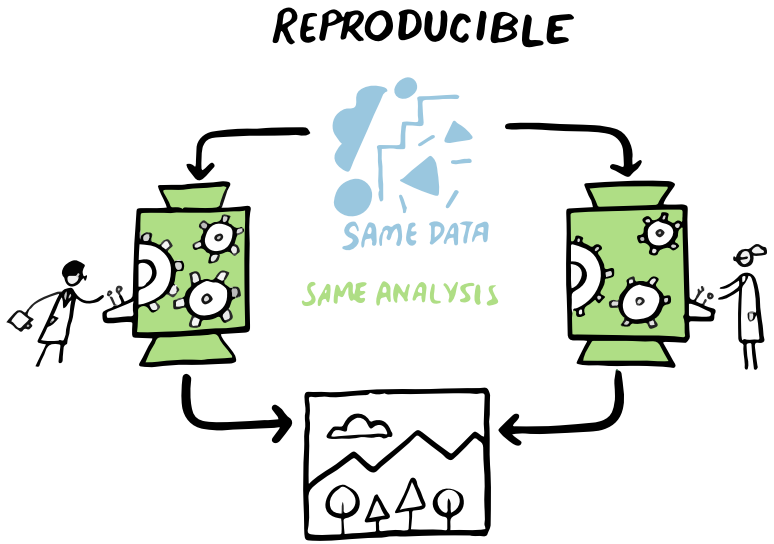
Computational reproducibility is the ability to **recreate**, **exactly same**, an earlier **research/ analysis** given the **same data** and **code**.

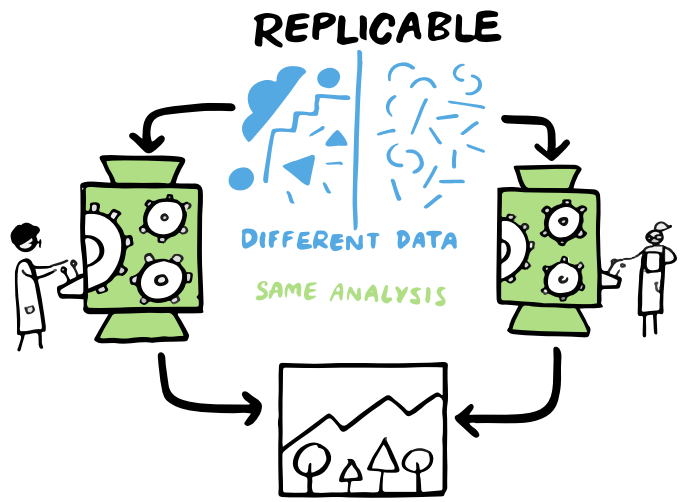
Replication vs Reproducibility

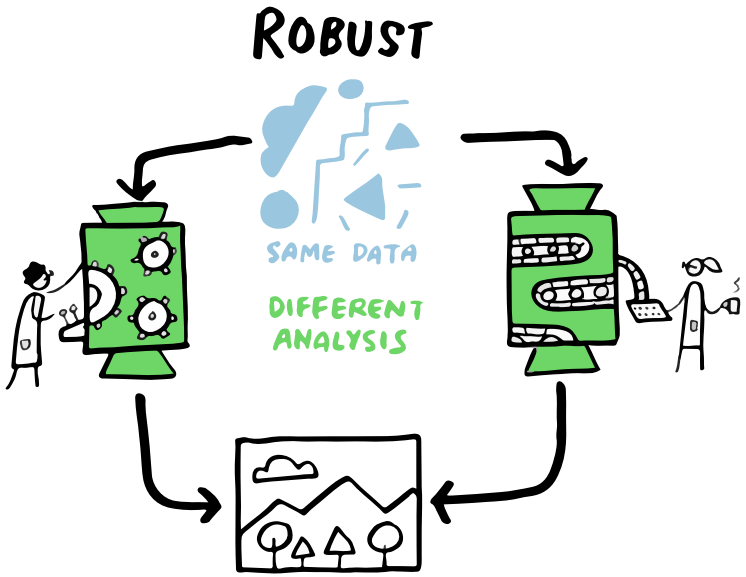


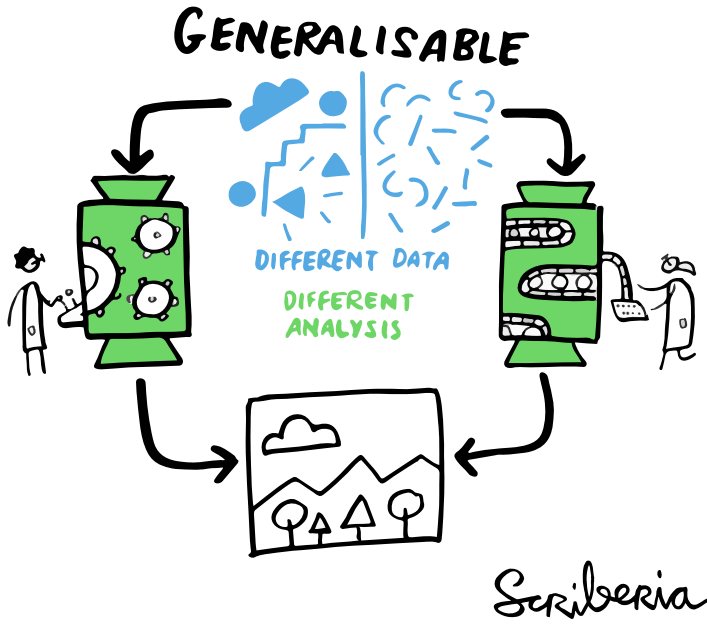
Claerbout, J. F., & Karrenbach, M. (1992). Electronic documents give reproducible research a new meaning. In *SEG Technical Program Expanded Abstracts*. https://doi.org/10.1190/1.1822162

Cartoon depiction of terms

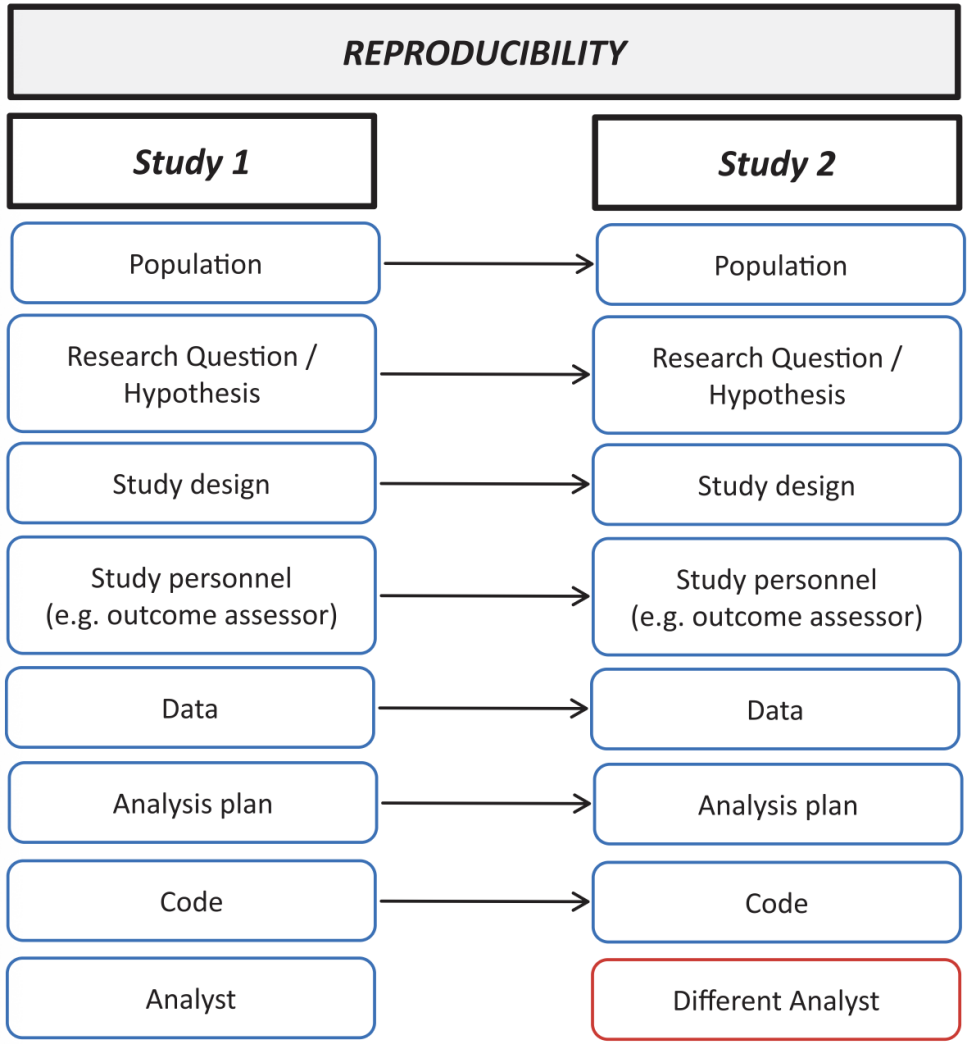




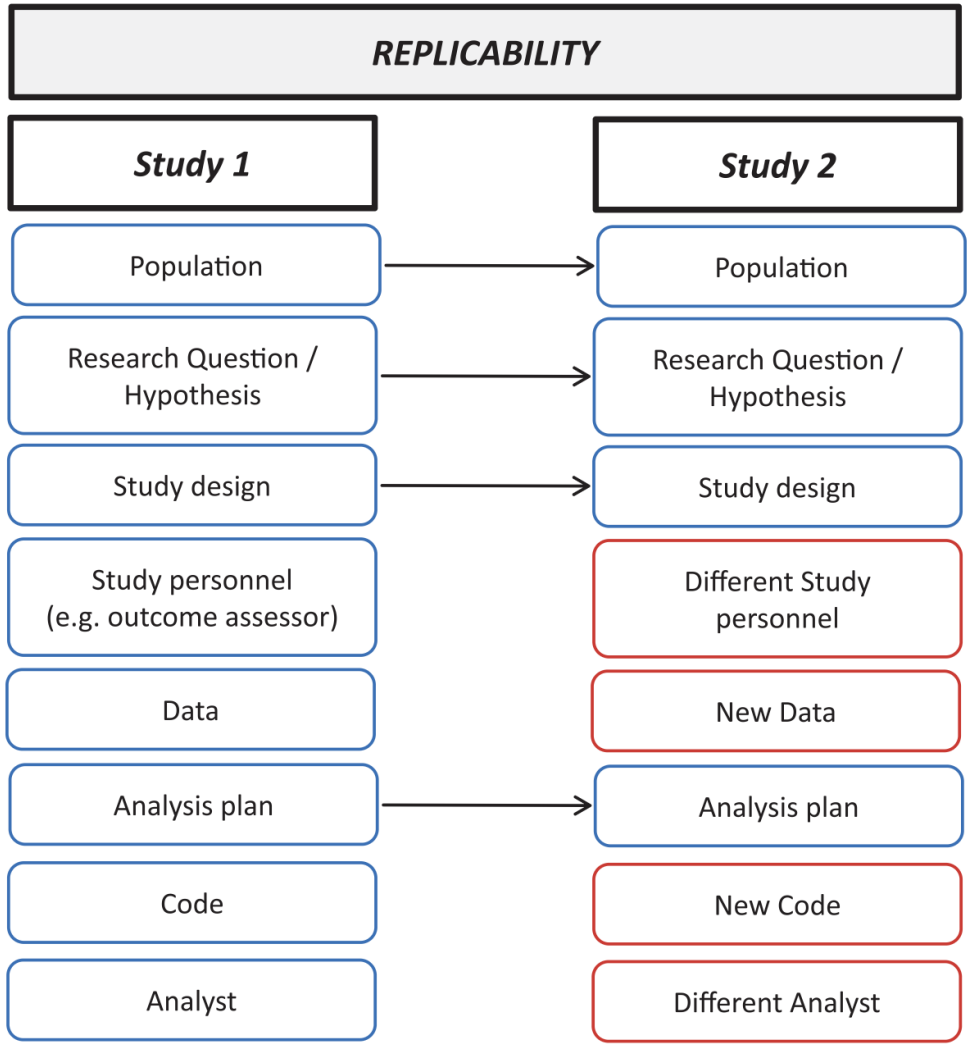




Replicability vs Reproducibility



Study 2 has successfully ***reproduced*** Study 1 if the estimates from both studies are consistent



Study 2 has successfully ***replicated*** Study 1 if the estimates from both studies are consistent

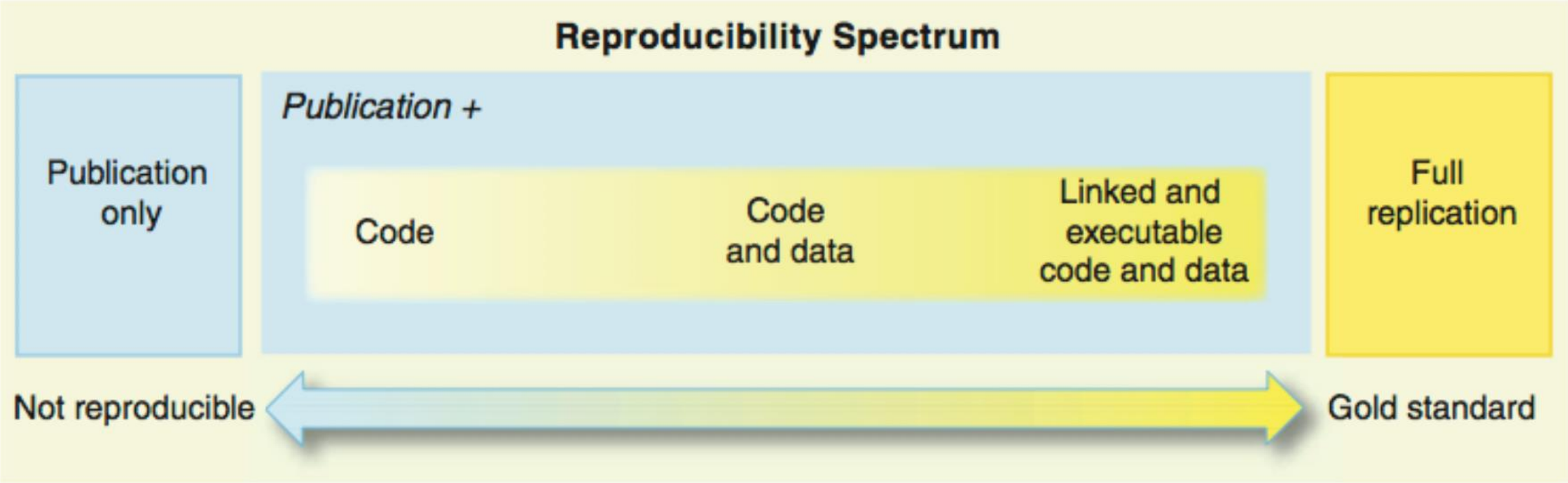
10.1097/j.pain.0000000000001254

Quiz 1

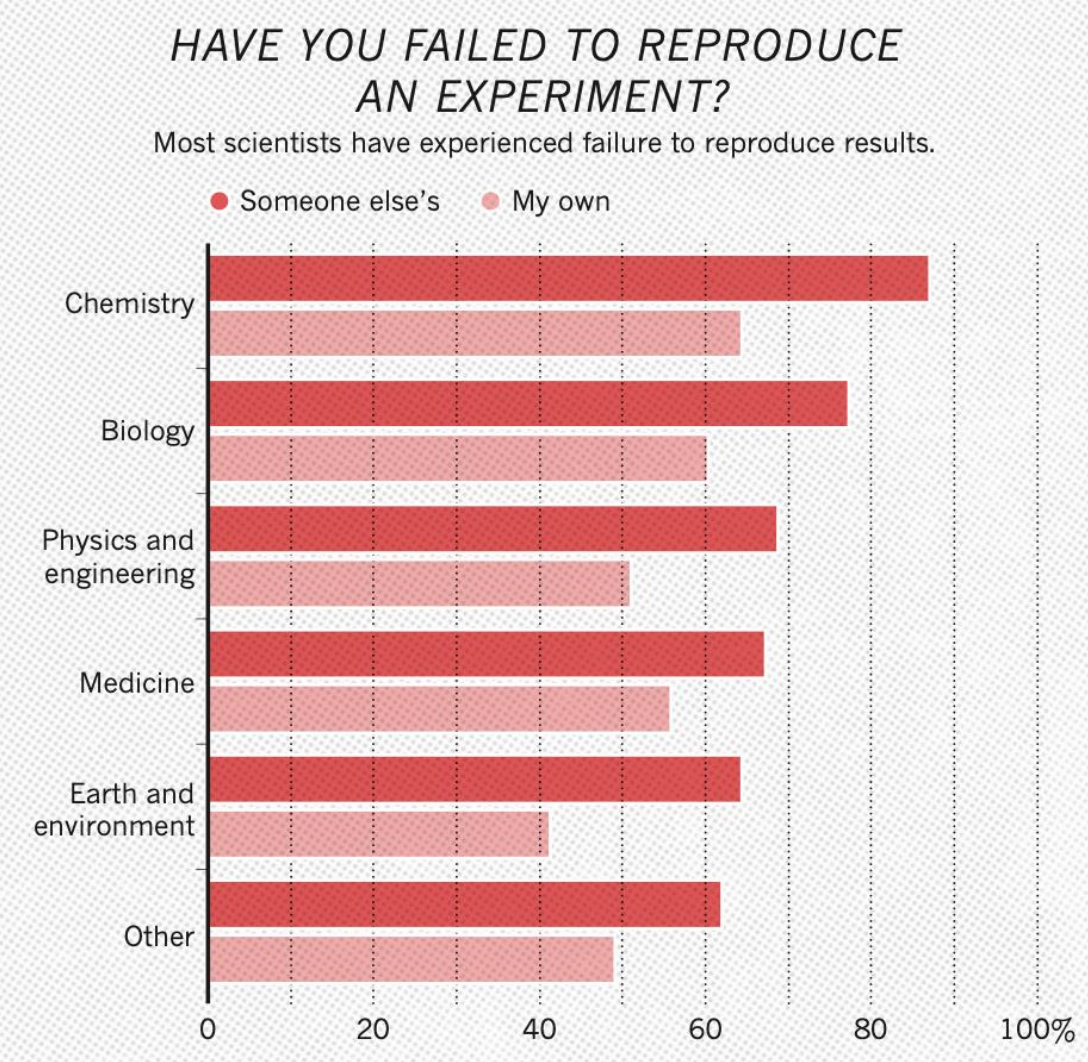
What best describes *computational reproducibility*?

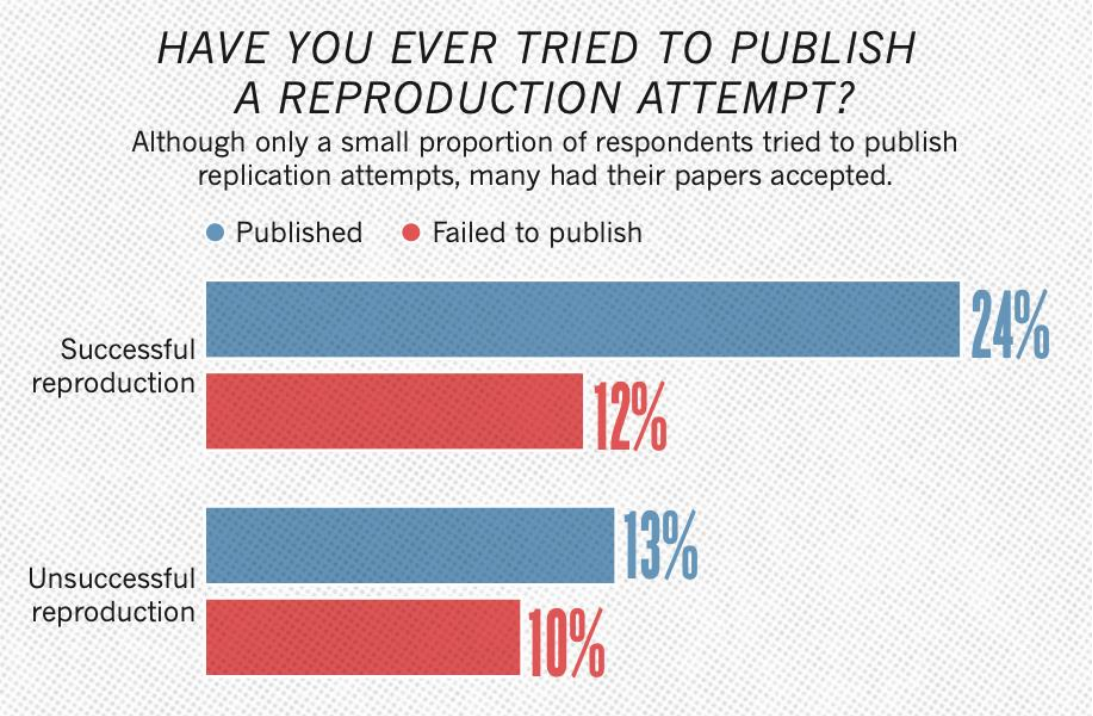
A) Running a new experiment to confirm previous results
B) Repeating the same lab protocol on a different sample
C) Re-creating the same analysis using the same code and data
D) Rewriting the entire codebase to improve performance

Reproducibility spectrum



The reproducibility crisis in science





https://www.nature.com/articles/533452a

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

Quiz 2

What is one key issue highlighted in the reproducibility crisis?

A) There are too few publications in top-tier journals
B) Many studies cannot be reproduced by independent labs
C) All computational results are automatically reproducible
D) Academic research never uses animal models

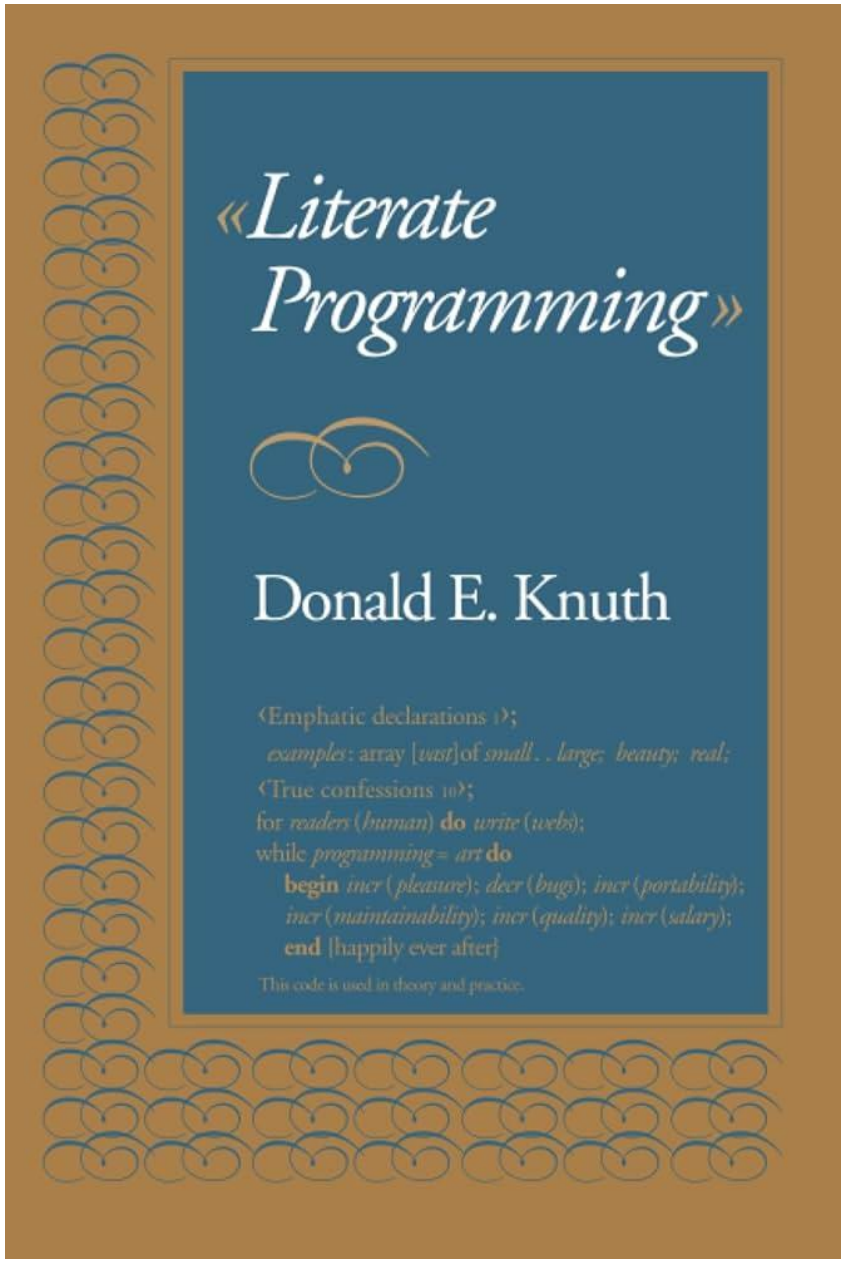
Academic Bias and biotech failures

The unspoken rule is that at least 50% of the studies published even in top tier academic journals – Science, Nature, Cell, PNAS, etc… – can't be repeated with the same conclusions by an industrial lab. In particular, key animal models often don't reproduce.

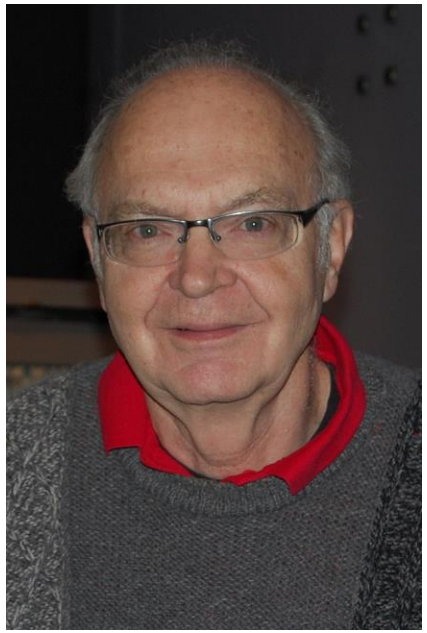
https://lifescivc.com/2011/03/academic-bias-biotech-failures/

A perfectly typeset text example

Literate programming



Literate programming is a programming methodology that combines a programming language with a documentation language, making programs more robust, more portable, and more easily maintained than programs written only in a high-level language.



Quiz 3

What is the goal of literate programming?

A) To teach programming to literature students
B) To improve hardware compatibility of software
C) To combine code and documentation for clarity
D) To automate writing of scientific papers

Tools for literate programming in R

Tools for literate programming in LaTeX

Sweave – Integrates R code into LaTeX (older tool).

knitr – Improved version of Sweave, works with R and LaTeX.

Pweave – Python + LaTeX (like knitr, but for Python).

noweb – Original language-independent tool for literate programming.

Tools for literate programming in Multiple-languages

quarto®

Org Mode

Your life in plain text

A *GNU Emacs* major mode for keeping notes, authoring documents, computational notebooks, literate programming, maintaining to-do lists, planning projects, and more — in a fast and effective plain text system.

GNU ELPA org 9.7.28 | report bug | feedback | source code | sr.ht Project

Libera.Chat | Matrix

Quiz 4

Which of the following is not a tool for literate programming?

A) Jupyter Notebook

B) knitr

C) Sweave

D) FastQC

Why reproducible?

1. For yourself!
    a. Adjusting your analysis
    b. Sharing your analysis
    c. Find out what the heck you did > 2 weeks ago

2. Because the academic community requires it..
    1. Many journals require accompanied code
    2. Proposals often require a data management plan

5 simple rules to get started

1. Execute the commands from a script
2. Number scripts based on their order of execution
3. Give your scripts a descriptive and active name
4. Make your scripts specific
5. Directories and variables at the beginning of the script

Rule 1

Execute the commands from a script to be able to trace back your steps

All output **files** and **directories** created from within a script

Adjusting your analysis becomes possible

Makes your analysis **portable**. It can be run:

- On a different computer
- By your colleague

Rule 2

Number scripts based on their order of execution (e.g. 01_download_reads.sh)

Easily trace the order of execution

Seperates **main** scripts from **secondary** scripts (i.e. scripts called by another script)

Example:
01_download_reads.sh
02_run_fastqc.sh
03_trim_reads.sh
04_run_fastqc_trimmed.sh

Rule 3

Give your scripts a **descriptive and active names**

Makes it easier to identify the script of interest

Example:

01_download_reads.sh
02_run_fastqc.sh
03_trim_reads.sh
04_run_fastqc_trimmed.sh

Rule 4

Make your scripts **specific**, i.e. do not combine many different commands in the same script

Makes your scripts **modular** (i.e. you can use it for other analyses)

Makes **job submission** more efficient

Turn your script in a **pipeline** later

Rule 5

Refer to **directories and variables at the beginning** of the script

Directories and variables need to be **changed** often
No need to **search** through the whole script to change them
Debugging is easier

Example:

```
#!/usr/bin/env bash
TRIMMED_DIR=~/workdir/trimmed_data
READS_DIR=~/workdir/reads
mkdir -p $TRIMMED_DIR
```

Quiz 5

Which of the following is not one of the five simple rules for reproducible code?

A) Execute commands from a script
B) Use modular and specific scripts
C) Hard-code all file paths
D) Define variables at the top of scripts

Further steps


Version control (git, GitHub, GitLab)

Pipelines

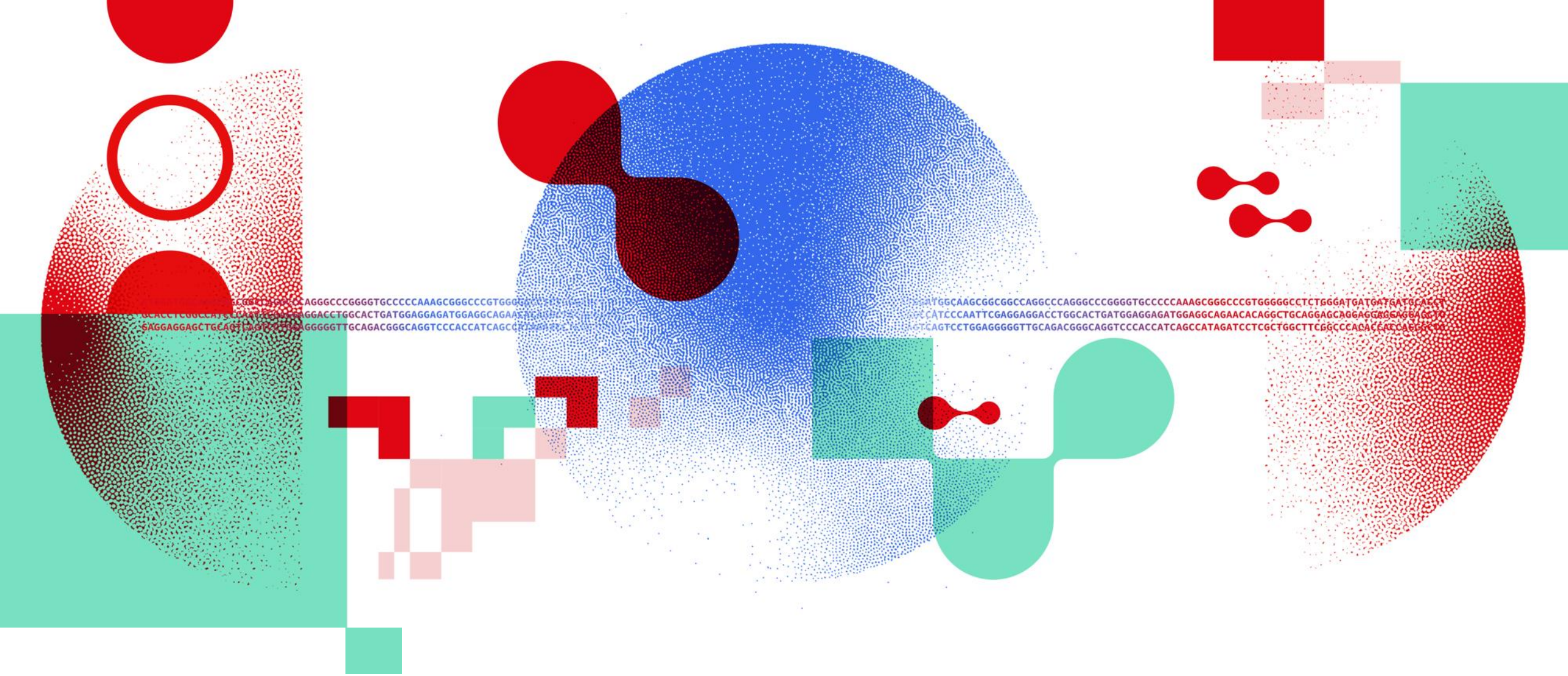
Notebooks


Further readings

1. Sandve, G.K. *et al.* (2013b) 'Ten simple rules for reproducible computational research,' *PLoS Computational Biology*, 9(10), p. e1003285. https://doi.org/10.1371/journal.pcbi.1003285.
2. Heise, V. *et al.* (2023) 'Ten simple rules for implementing open and reproducible research practices after attending a training course,' *PLoS Computational Biology*, 19(1), p. e1010750. https://doi.org/10.1371/journal.pcbi.1010750.
3. Rule, A. *et al.* (2018b) 'Ten simple rules for reproducible research in Jupyter notebooks,' *arXiv (Cornell University)* [Preprint]. https://doi.org/10.48550/arxiv.1810.08055.

Summary

1. **Reproducibility is essential** for trust and progress in science.
2. **Reproducibility ≠ Replicability** – know the difference.
3. The **reproducibility crisis** affects even top-tier research.
4. **Literate programming** improves transparency by combining code and explanation.
5. Follow **simple coding rules** to make your work reproducible.
6. Think ahead: **version control, documentation, and sharing** are key for reproducible research.



Thank you

DATA SCIENTISTS FOR LIFE

sib.swiss