seminar: reproducibility and performance of privacy-enhancing technologies (repet)

INTRODUCTION TO REPRODUCIBILITY

Quentin Guilloteau

18 October 2024

University of Basel

outline

How does Academia work?

the publication process

1. Researchers work on a research question
2. They write a paper (*i.e.*, a document to present their work/findings)
3. They submit the paper to a conference or journal
4. Other researchers will read the paper and decide if it is worth publishing
5. The more prestigious the conference or journal, the more "success" for the researchers
6. The more publications, the more "success" for the researchers

let's imagine for a minute that we are...

... Mathematicians

- We prove a new theorem
- the paper contains the proof
- the proof will forever be part of the paper
- you do not need a special paper or pen to re-do the proof

let's imagine for a minute that we are...

... Mathematicians

- We prove a new theorem
- the paper contains the proof
- the proof will forever be part of the paper
- you do not need a special paper or pen to re-do the proof

... Computer Scientists

- We write some program
- the paper does not contain a link to the code???
- the link to the code could break in the future
- you need some obscure software dependencies to re-run the program

let's imagine for a minute that we are...

... Mathematicians

- We prove a new theorem
- the paper contains the proof
- the proof will forever be part of the paper
- you do not need a special paper or pen to re-do the proof

... Computer Scientists

- We write some program
- the paper does not contain a link to the code???
- the link to the code could break in the future
- you need some obscure software dependencies to re-run the program

$\hookrightarrow$ Curry–Howard correspondence $\simeq$ "*program* = *proof*"

what if someone wants the source code?

- Collberg et al. [1] (2015)
- looked at 600 papers and try to get the code and reproduce the results
- and contacted the authors
- funny responses from the authors ("My dog ate my homework")

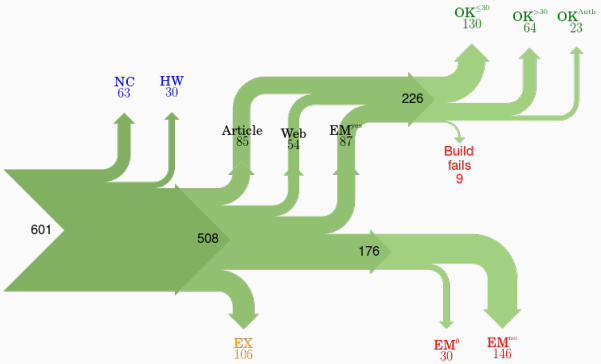


Figure 1: Collberg et al.

collberg et al. – some answers

"I am just hoping that it is a stable working version of the code, and matches the implementation we finally used for the paper. Unfortunately, I have lost some data when my laptop was stolen last year. The bad news is that the code is not commented and/or clean. So, I cannot really guarantee that you will enjoy playing with it."

collberg et al. – some answers

"I am just hoping that it is a stable working version of the code, and matches the implementation we finally used for the paper. Unfortunately, I have lost some data when my laptop was stolen last year. The bad news is that the code is not commented and/or clean. So, I cannot really guarantee that you will enjoy playing with it."

"[STUDENT] was a graduate student in our program but he left a while back so I am responding instead. For the paper we used a prototype that included many moving pieces that only [STUDENT] knew how to operate and we did not have the time to integrate them in a ready-to-share implementation before he left"

collberg et al. – some answers

"I am just hoping that it is a stable working version of the code, and matches the implementation we finally used for the paper. Unfortunately, I have lost some data when my laptop was stolen last year. The bad news is that the code is not commented and/or clean. So, I cannot really guarantee that you will enjoy playing with it."

"[STUDENT] was a graduate student in our program but he left a while back so I am responding instead. For the paper we used a prototype that included many moving pieces that only [STUDENT] knew how to operate and we did not have the time to integrate them in a ready-to-share implementation before he left"

"Most importantly, I do not have the bandwidth to help anyone come up to speed on this stuff."

The Reproducibility Crisis

Reproducibility Crisis

- Baker [2], [3] raised awareness about the issue
- since then, all the fields of science are getting concerned/interested in reproducibility

What is "Reproducibility"?

- It depends on who you ask .... No consensus between sciences
- In computer science: ACM gave some definitions [4]
  - *Repeatability* (Same team, same experimental setup)
  - *Reproducibility* (Different team, same experimental setup)
  - *Replicability* (Different team, different experimental setup)
- some sciences use also the term "Robustness"

what is the goal of reproducibility?

- Should not all research work be reproducible?
- Is it a proof of the correctness of the work?
- Restore trust in science?
- or an effort so that future researchers can use the work?
  - *"Standing on the shoulders of giants"*
  - Science is a self-correcting and iterative process
  - the objective: precise introduction of Variation [5], [6]

Open Science and FAIR

- **F**indable, **A**ccessible, **I**nteroperable, **R**eusable
- Open Science: one key concept is "*Transparency*"

what about in computer science?

"Computers are deterministic, hence no reproducibility issues" – a fool

- young science
- blur between what are the tools and what are the objects of study
- studying the side effects of decisions made by peers
- computers are physical machines
- Experiments are *"quick"* and *"free"* $\rightarrow$ leads to some quick and dirty work
- Programming errors, mistakes (Excel [7]), difficult to control
  - floating point associativity ($round(a + round(b + c)) \neq round(round(a + b) + c)$)
  - order of compilation flags has an impact on the performance [8]
  - the size of the environment variable has an impact on performance [8]

artifact evaluation process (since $\simeq$ 2015)

- Conferences and journals are now having a voluntary "Artifact Evaluation" process to make sure that the accepted papers are "reproducible"
- Authors send their code, scripts, data and a description of how to use it
- then reviewers will try to rerun experiments
- Authors get badges to reward their effort based on reviews

The goals:

- validate results
- restore trust
- promote artifact sharing

Figure 2: ACM badges

artifact description appendix

Appendix: Artifact Description/Artifact Evaluation

Artifact Description (AD)

I. Overview of Contributions and Artifacts

A. Paper's Main Contributions

Provide a list of all main contributions of the paper.

C_1 This is the 1st contribution.
C_2 This is the 2nd contribution.
C_3 This is the 3rd contribution.

B. Computational Artifacts

List the computational artifacts related to this paper along with their respective DOIs. Note that all computational artifacts may be archived under a single DOI.

A_1 https://doi.org/YYYY/zenodo.0XXXXX
A_2 https://doi.org/ZZ.YYYY/zenodo.1XXXXX
A_3 https://doi.org/ZZ.YYYY/zenodo.2XXXXX

Provide a table with the relevant computational artifacts, highlight their relation to the contributions (from above) and point to the elements in the paper that are reproducible by each artifact, e.g., which figures or tables were generated with the artifact.

Artifact ID	Contributions Supported	Related Paper Elements
A_1	C_1	Tables 1-2 Figure 3
A_2	C_2	Tables 2-3 Figures 1-2
..		

II. Artifact Identification

Provide the following six subsections for each computational artifact A_i.

A. Computational Artifact A_1

Relation To Contributions

Briefly explain the relationship between the artifact and contributions.

Expected Results

Provide a higher level description of what outcome to expect from the corresponding experiments. Provide an explanation of how the results substantiate the main contributions.

Algorithm A should be faster than Algorithms C and B in all GPU scenarios.

Expected Reproduction Time (in Minutes)

Estimate the time required to reproduce the artifact, providing separate estimates for the individual steps: Artifact Setup, Artifact Execution, and Artifact Analysis.

The expected computational time of this artifact on GPU X is 20 min.

Artifact Setup (incl. Inputs)

Hardware: Specify the hardware requirements and dependencies (e.g., a specific interconnect or GPU type is required).

Software: Introduce all required software packages, including the computational artifact. For each software package, specify the version and provide the URL.

Datasets / Inputs: Describe the datasets required by the artifact. Indicate whether the datasets can be generated, including instructions, or if they are available for download, providing the corresponding URL.

Installation and Deployment: Detail the requirements for compiling, deploying, and executing the experiments, including necessary compilers and their versions.

Artifact Execution

Provide an abstract description of the experiment workflow of the artifact. It is important to identify the main tasks (processes) and how they depend on each other.

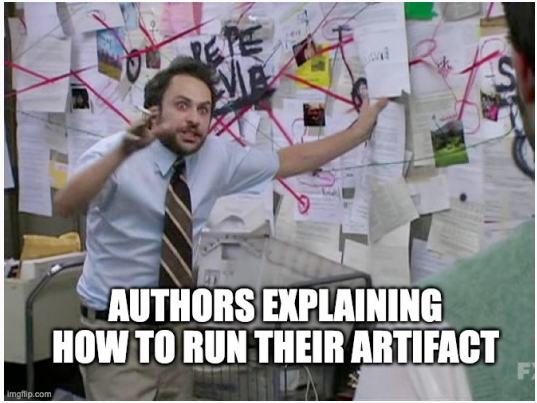
A workflow may consist of three tasks: T_1, T_2, and T_3. The task T_1 may generate a specific dataset. This dataset is then used as input by a computational task T_2, and the output of T_2 is processed by another task T_3, which produces the final results (e.g., plots, tables, etc.). State the individual tasks T_i and provide their dependencies, e.g., $T_1 \rightarrow T_2 \rightarrow T_3$.

Provide details on the experimental parameters. How and why were parameters set to a specific value (if relevant for the reproduction of an artifact), e.g., size of dataset, number of data points, input sizes, etc. Additionally, include details on statistical parameters, like the number of repetitions.

Artifact Analysis (incl. Outputs)

B. Computational Artifact A_2

Provide the same type of information as done for Computational Artifact A_1.



is the source code enough?

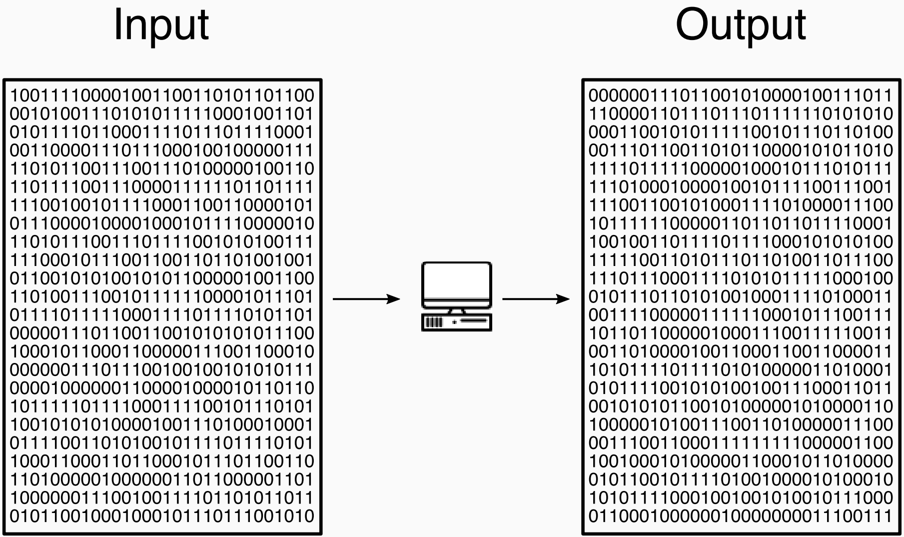
From one compiler to another:

- will the application's behavior be the same?
- will the application's performance be the same?
- will the application's 3^{rd} party libraries be compatible?

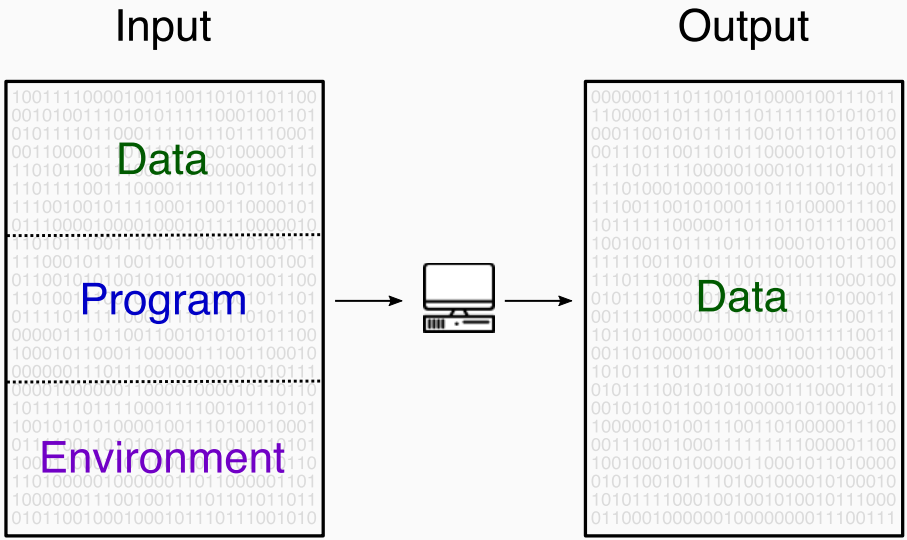
From one machine to another:

- will the required kernel's features be available?
- will the required architecture features be available?

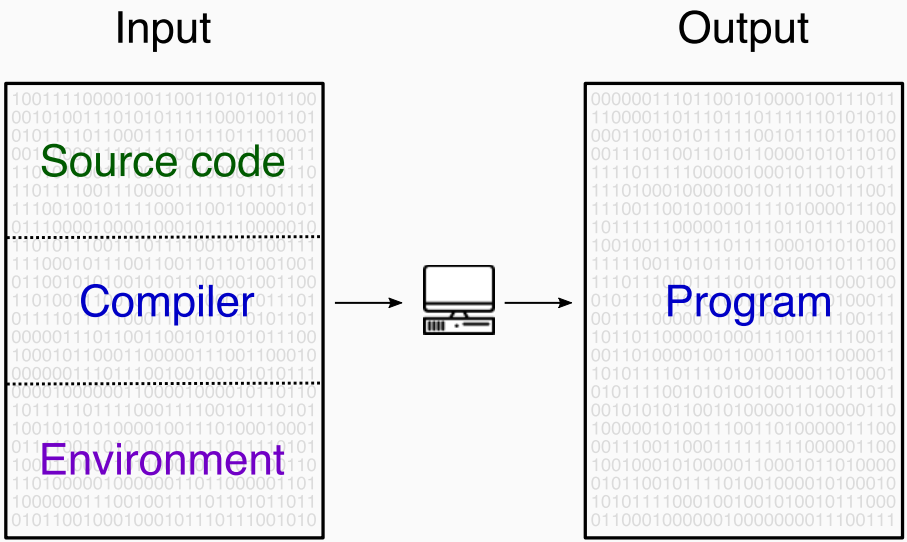
what is a computation? (from konrad hinsen's slides)



what is a computation? (from konrad hinsen's slides)



what is a computation? (from konrad hinsen's slides)



Reproducibility and Methodology

are those two experiments the “same”?

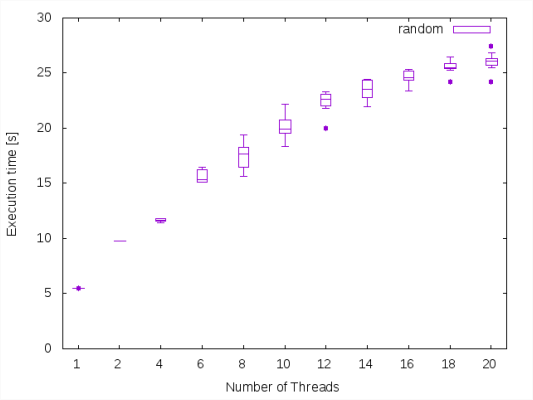


Figure 3: Initial Experiment

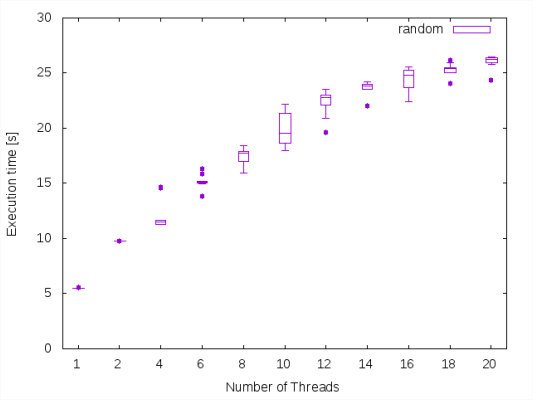


Figure 4: Reproduction

↪ Is this a successful reproduction?

noise in measurements

Computers are physical machines!

- Disks
- Cosmic rays
- Temperature

Resources are shared

- Sharing network
- Sharing CPU
- Sharing RAM

Controlling everything is difficult

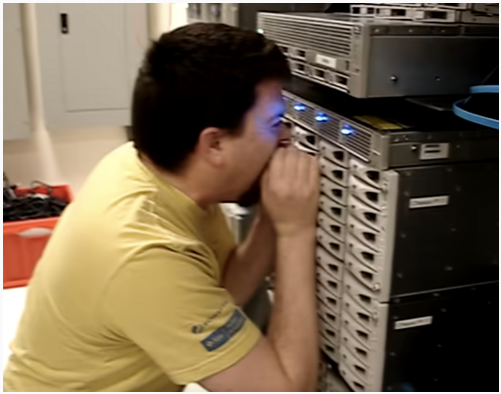


Figure 5: "Shouting in the Datacenter" (https://www.youtube.com/watch?v=tDacjrSCeq4)

bit-wise reproducibility

Ultimate proof = bit-wise reproducibility

Good for simulations, “simple” measures

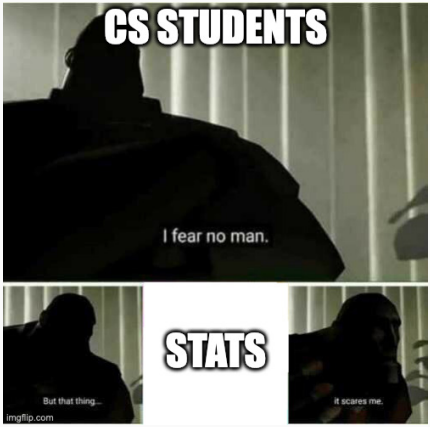
However, not always possible (measure of performance)

How to protect ourselves from experimental noise?

statistics to the rescue

- Proper design of experiments to reduce noise/incertitude
- Statistical tests for significance of results
- But no-one is educated enough...
- In CS: Data Analysis $\simeq$ computing average [9]

But...
the choice of the statistical methods is also subject to reproducibility issues



pre-registration and registered reports

Goal: explicit protocol, variables, methods, and avoid statistical manipulation

Pre-registration

1. Authors submit their protocol to a journal
2. Review of the protocol (if protocol accepted, then the final paper will be accepted no matter the conclusion of the work)
3. Authors start collecting data
4. Authors write the final paper
5. Review of the final paper

Registered Reports
Authors upload on pre-print servers their protocol before collecting data, no review

pre-registration and registered reports

Goal: explicit protocol, variables, methods, and avoid statistical manipulation

Pre-registration

1. Authors submit their protocol to a journal
2. Review of the protocol (if protocol accepted, then the final paper will be accepted

More careful, but slower, research process

3. Authors start collecting data
4. Authors write the final paper
5. Review of the final paper

Registered Reports
Authors upload on pre-print servers their protocol before collecting data, no review

can you find the plotting mistakes?

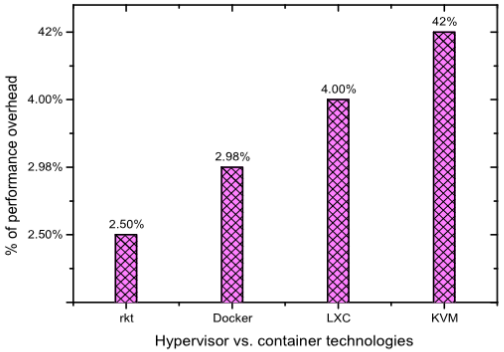


Fig. 3 Graphical evaluation of hypervisor versus container technologies for CPU-intensive benchmarks

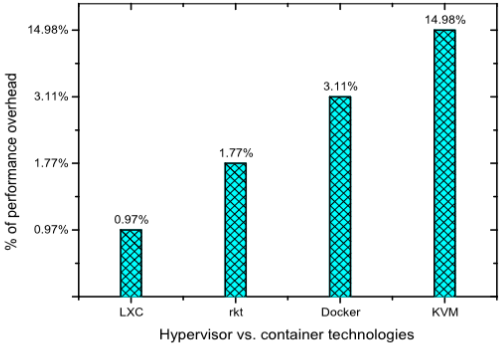
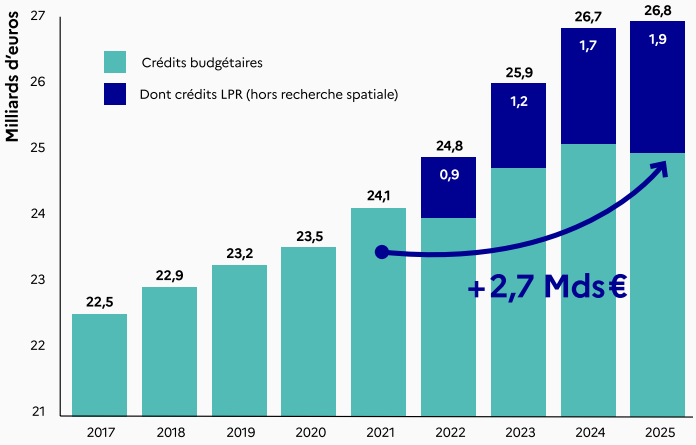


Fig. 4 Graphical evaluation of hypervisor versus container technologies for memory-intensive benchmarks

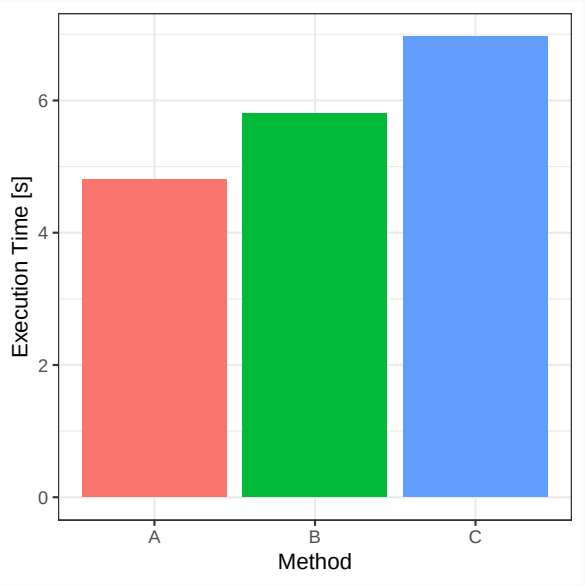
can you find the plotting mistakes?

Crédits du ministère de l'Enseignement supérieur et de la Recherche*

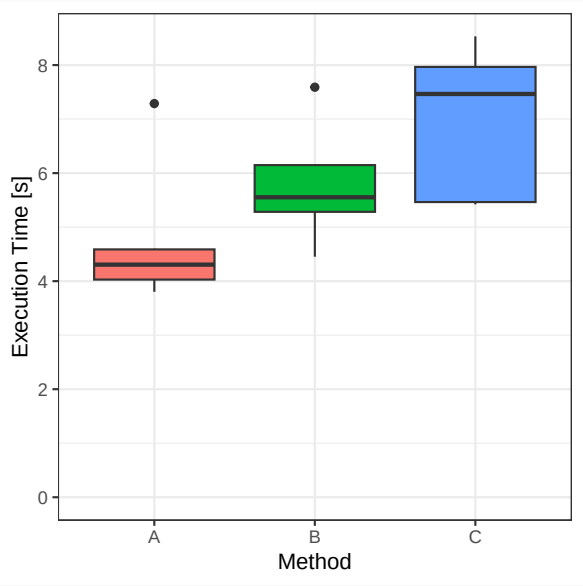


*Crédits ouverts en loi de finances sur les programmes budgétaires 150, 231 et 172 (courant), hors CVEC.

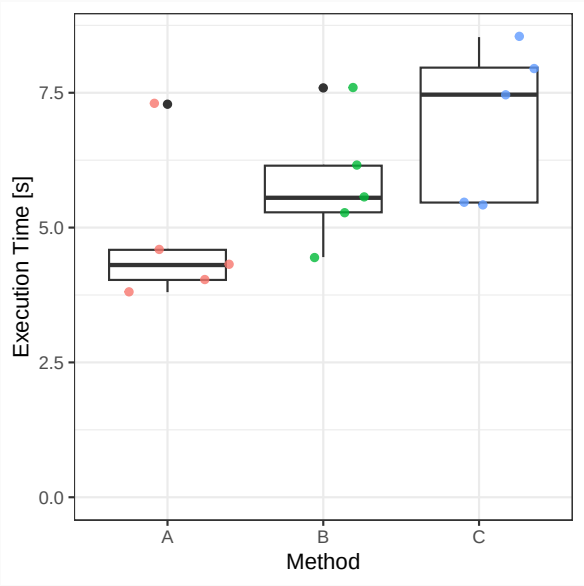
can you find the plotting mistakes?



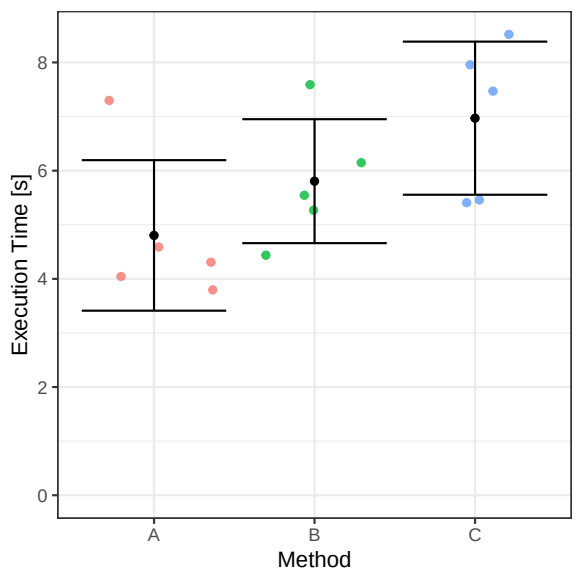
can you find the plotting mistakes?



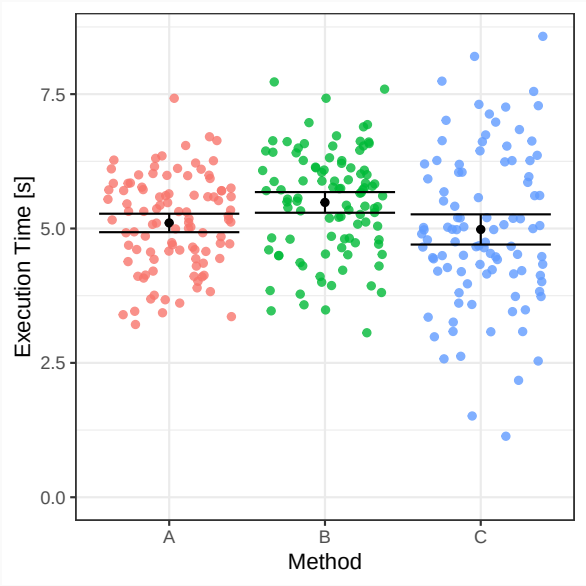
can you find the plotting mistakes?



can you find the plotting mistakes?



can you find the plotting mistakes?



Reproducibility and Longevity

longevity

Reproducibility not only for today, but also for tomorrow!

- Would you expect a math proof to disappear off a paper?
- Think about the people that will try to use your work
- What do we need?
  - Access to the artifact
  - Same software environment
  - Same hardware (?!)

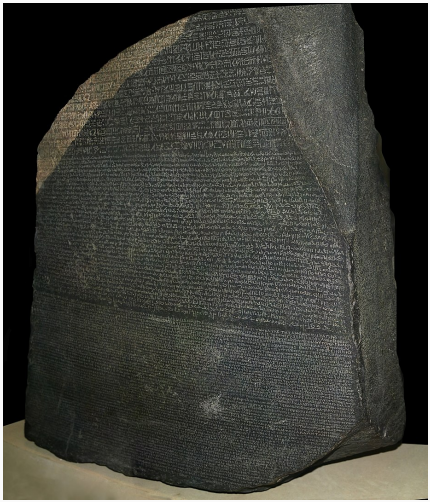


Figure 6: Rosetta Stone

how to ensure long term reproducibility?

Availability

- Zenodo [10] for data (CERN)
- Software Heritage [11], [12] for source code (UNESCO)

Software Environment

- Functional Package Managers: Nix [13], Guix [14]

Experimental Platform

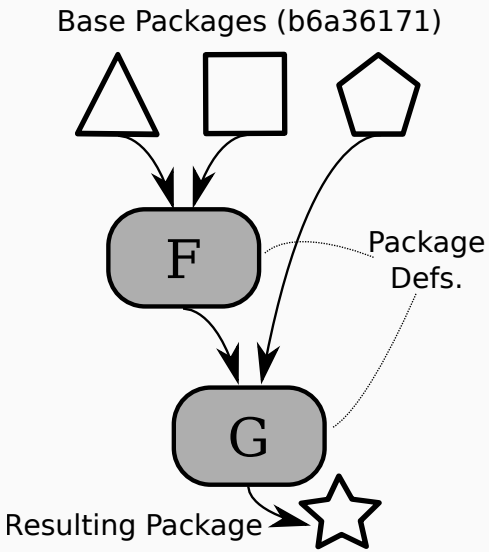
- Testbeds: shared computational platforms funded by public actors [15] (Chameleon, CloudLab, Grid'5000)

deeper dive: software environment

- Depend on an uncontrollable external state (`apt update`, `apt install`)
- Check the content of downloaded objects (`curl`, `wget`)
  - The object behind the same URL can change through time
  - Take the cryptographic hash of the object (`sha256sum`, `md5sum`, etc.)
- Pay attention to the commit for Git repositories (`git clone https://github.com/me/myrepo`)
  - By default: latest commit of main branch
  - Commit can be deleted... (Software Heritage)
- Classical Package Managers put everything in `/usr/bin`, `/usr/lib` $\rightsquigarrow$ potential conflicts

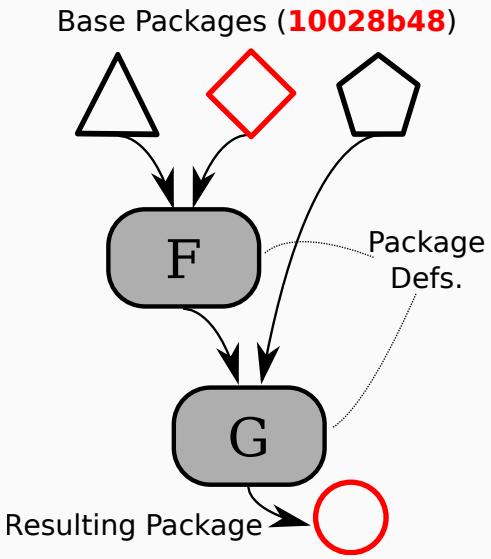
functional packages managers

- Nix [13], Guix [14]: reproducible by design!
- packages = functions
  - inputs = dependencies
  - body = commands to build the package
  - base packages defined in Git
- sandbox, no side effect
- `/nix/store/hash(inputs)-my-pkg`
- Store immutable, read-only
- precise definition of `$PATH`
- can build: container, VM, etc.



functional packages managers

- Nix [13], Guix [14]: reproducible by design!
- packages = functions
  - inputs = dependencies
  - body = commands to build the package
  - base packages defined in Git
- sandbox, no side effect
- `/nix/store/hash(inputs)-my-pkg`
- Store immutable, read-only
- precise definition of `$PATH`
- can build: container, VM, etc.



Conclusion

conclusion

Reproducibility is a result of good scientific methodology and practices

To keep in mind

- Ask yourselves "What could go wrong?", "What is not being controlled?"
- Automate as much as possible (while keeping some flexibility)
- Educate yourself in Statistics / Design of Experiments / Data Analysis (e.g., Statistical Dances, Benchmarking Crimes, Bad plotting practices)
- Sensibilize your peers, your advisors, etc.
- Use Git
- Do not be shy to share everything (code, data, metadata)

questions for you

What do you think about:

- the reproducibility of AI?
- the reproducibility of proprietary tools?
- the environmental cost of reproducibility?

Seminar Logistics

important dates

Sep 26	You: Topic preferences due
Sep 27	Lecturers: Topic assignments and presentation dates sent out
Oct 25	You: Agree on plan for programming project
Oct 30	You: Draft presentation slides
Nov 1	You: Initial version of report due (anonymized, for peer review)
Nov 8	You: Peer review due
Nov 21	You: Final version of report due (graded)
Dec 12	You: Project implementation due

Next Meeting

Oct 25: Reproducibility Hackathon (12:30-18:00)

- Remember to bring your laptop charger

Q&A

[1] C. Collberg, T. Proebsting, and A. M. Warren, **"Repeatability and Benefaction in Computer Systems Research - A Study and a Modest Proposal,"** en, p. 68, 2015.

[2] M. Baker, **"Reproducibility crisis,"** *Nature*, vol. 533, no. 26, pp. 353–66, 2016.

[3] M. Baker, **"1,500 scientists lift the lid on reproducibility,"** *Nature*, vol. 533, no. 7604, pp. 452–454, May 2016. DOI: 10.1038/533452a. [Online]. Available: http://www.nature.com/doifinder/10.1038/533452a (visited on 05/03/2019).

[4] ACM, **Artefact review badging,** https://www.acm.org/publications/policies/artifact-review-badging, Accessed: 2023-04-04.

[5] D. G. Feitelson, **"From Repeatability to Reproducibility and Corroboration,"** en, *ACM SIGOPS Operating Systems Review*, vol. 49, no. 1, pp. 3–11, Jan. 2015. DOI: 10.1145/2723872.2723875. [Online]. Available: https://dl.acm.org/doi/10.1145/2723872.2723875 (visited on 05/21/2020).

[6] M. Mercier, A. Faure, and O. Richard, **"Considering the development workflow to achieve reproducibility with variation,"** in *SC 2018-Workshop: ResCuE-HPC*, 2018, pp. 1–5.

[7] D. Lewis, **"Autocorrect errors in excel still creating genomics headache.,"** *Nature*, 2021.

[8] T. Mytkowicz, A. Diwan, M. Hauswirth, and P. F. Sweeney, **"Producing wrong data without doing anything obviously wrong!"** *ACM Sigplan Notices*, vol. 44, no. 3, pp. 265–276, 2009.

[9] S. Hunold, **"A survey on reproducibility in parallel computing,"** *arXiv preprint arXiv:1511.04217*, 2015.

[10] zenodo, **Zenodo,** https://zenodo.org/, Accessed: 2023-03-30.

[11] S. Heritage, **Software heritage,** https://www.softwareheritage.org/, Accessed: 2023-03-30.

[12] R. Di Cosmo and S. Zacchiroli, **"Software heritage: Why and how to preserve software source code,"** in *iPRES 2017-14th International Conference on Digital Preservation*, 2017, pp. 1–10.

[13] E. Dolstra, M. de Jonge, and E. Visser, **"Nix: A Safe and Policy-Free System for Software Deployment,"** en, p. 14, 2004.

[14] L. Courtès, **"Functional Package Management with Guix,"** en, *arXiv:1305.4584 [cs]*, May 2013. [Online]. Available: http://arxiv.org/abs/1305.4584 (visited on 06/13/2020).

[15] L. Nussbaum, **"Testbeds support for reproducible research,"** in *Proceedings of the reproducibility workshop*, 2017, pp. 24–26.