Data Science & Research Reproducibility

Sang-Yun Oh

Department of Statistics and Applied Probability

UC SANTA BARBARA

What is Computational Reproducibility?

- Reviewable Research
  The descriptions of the research methods can be independently assessed

- Replicable Research
  Data, code, and/or software tools are made available to duplicate the research result (may not be public)

- Confirmable Research
  The main conclusions can be obtained independently using the description of algorithms and methodology provided in the publication.

(Stodden et al., 2013)

What is Computational Reproducibility?

- Auditable Research
  Sufficient records, including data and software, have been archived (potentially privately) so that the research can be defended later if necessary or differences between independent confirmations resolved

- Open or Reproducible Research
  Auditable research made openly available, so that one may (a) fully audit the computational procedure, (b) replicate and also independently reproduce the results of the research, and (c) extend the results or apply the method to new problems.

(Stodden et al., 2013)

Computational Reproducibility

An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures.

Donoho, 1998

Home > European Journal for Philosophy of Science > Article

Epistemic issues in computational reproducibility: software as the elephant in the room

Paper in Historical and Social Studies of Science | Published: 17 April 2021 | **11**, Article number: 38 (2021)

Download PDF

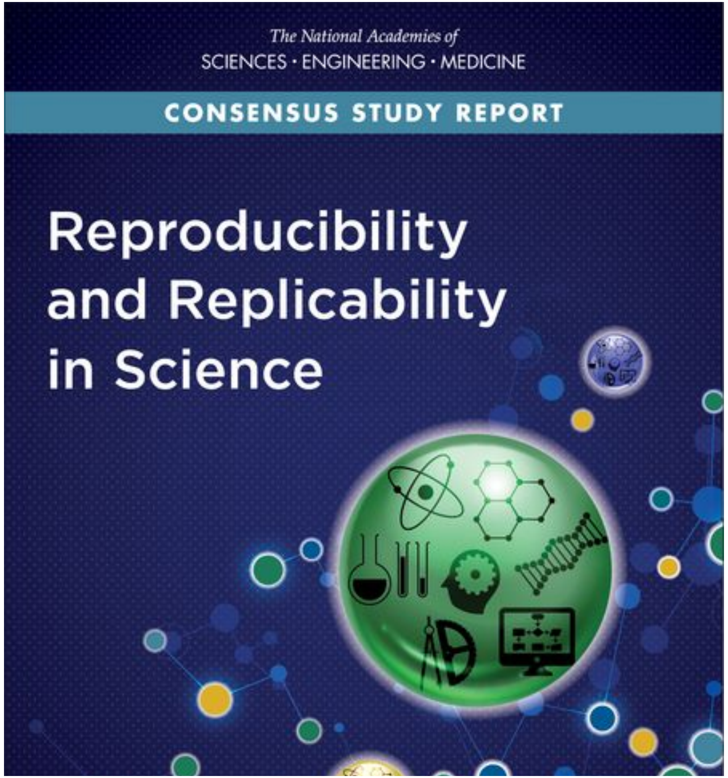
Access provided by California Digital Library TA

European Journal for Philosophy of Science

Aims and scope →

Submit manuscript →

How bad is the Reproducibility Problem?



Committee on Reproducibility and Replicability in Science

Board on Behavioral, Cognitive, and Sensory Sciences
Committee on National Statistics
Division of Behavioral and Social Sciences and Education

Nuclear and Radiation Studies Board
Division on Earth and Life Studies

Board on Mathematical Sciences and Analytics
Committee on Applied and Theoretical Statistics
Division on Engineering and Physical Sciences

Board on Research Data and Information
Committee on Science, Engineering, Medicine, and Public Policy
Policy and Global Affairs

ENVIRONMENTAL RESEARCH
LETTERS

PERSPECTIVE • OPEN ACCESS

The critical need to foster computational reproducibility

Robert Reinecke[5,1,2], Tim Trautmann[3], Thorsten Wagener[1] and Katja Schüler[4]

Published 25 March 2022 • © 2022 The Author(s). Published by IOP Publishing Ltd

Environmental Research Letters, Volume 17, Number 4

Citation Robert Reinecke *et al* 2022 *Environ. Res. Lett.* **17** 041005

DOI 10.1088/1748-9326/ac5cf8

Article PDF   Article ePub

Figures ▾   References ▾   Open science ▾

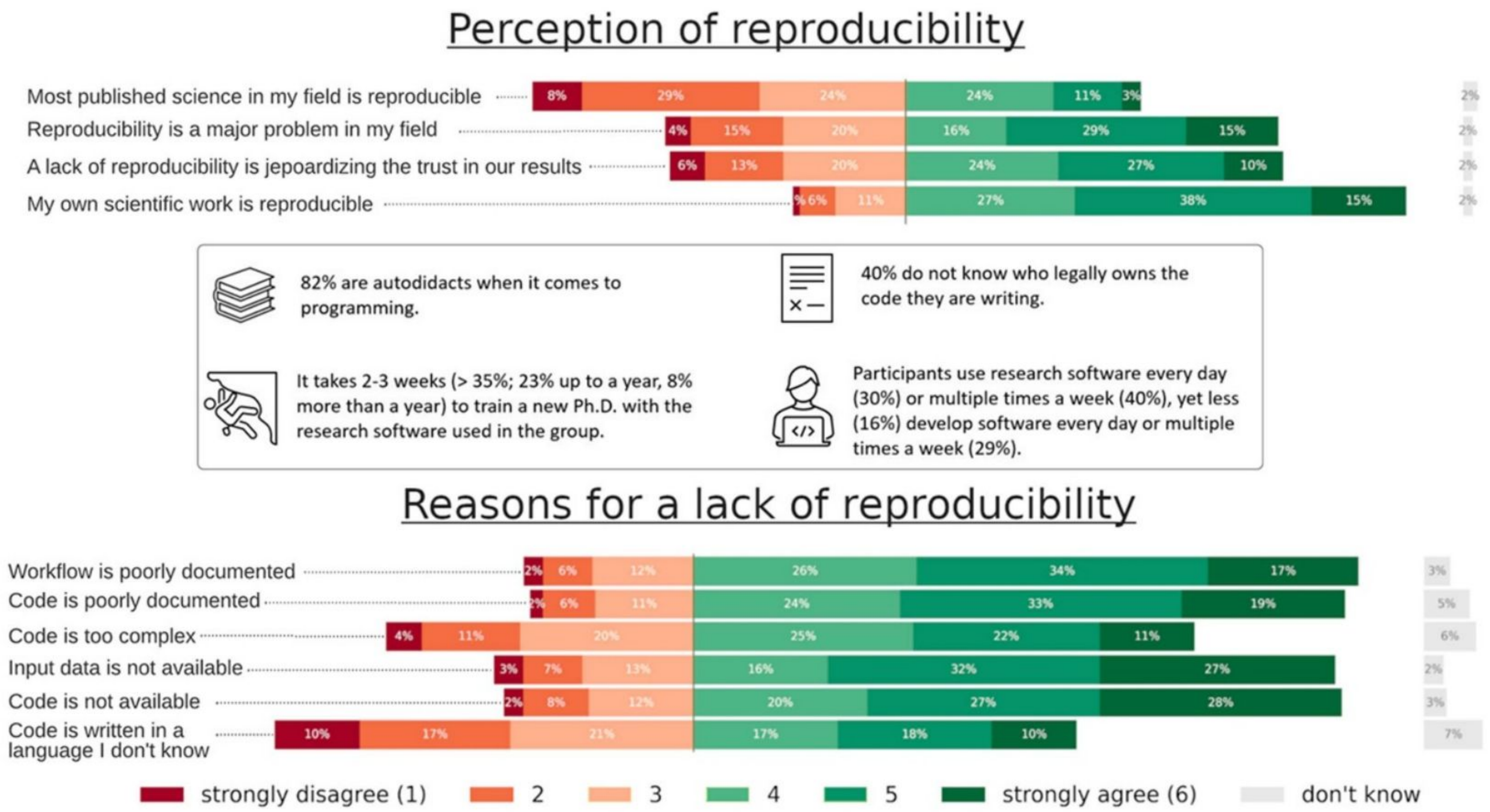


Figure 1. Polling results ($n = 265$) of perception and lack of reproducibility (bar charts) and summary of figure S4 on how the community develops and uses code.

Solutions

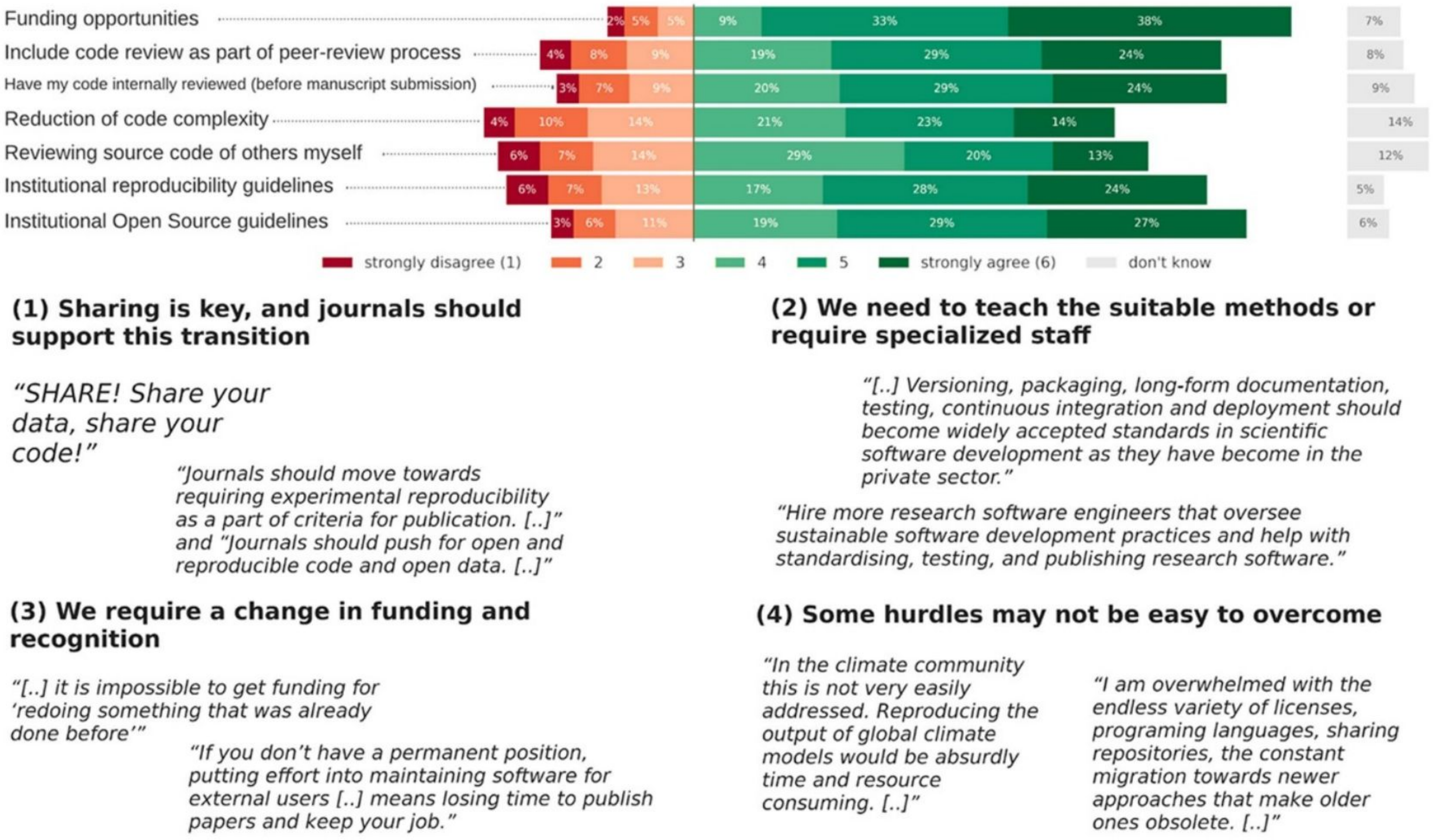


Figure 2. Polling results ($n = 265$) on proposed solutions to increase reproducibility and voices from the community on fostering reproducibility.

ARBARA

Ingredients for Computational Reproducibility

- Data (real or simulated)
- Algorithm implementation
- Analysis pipeline
  Preprocessing, analysis, post-processing, external validation, etc.
- Generated report (figures and tables)

Ingredients for Computational Reproducibility

- Data (real or simulated)
- Algorithm implementation
- Analysis pipeline
  Preprocessing, analysis, post-processing, external validation, etc.
- Generated report (figures and tables)
- Computational environment
  Operating system, R/Python versions, package versions

PLOS COMPUTATIONAL BIOLOGY

EDITORIAL

Ten Simple Rules for Reproducible Computational Research

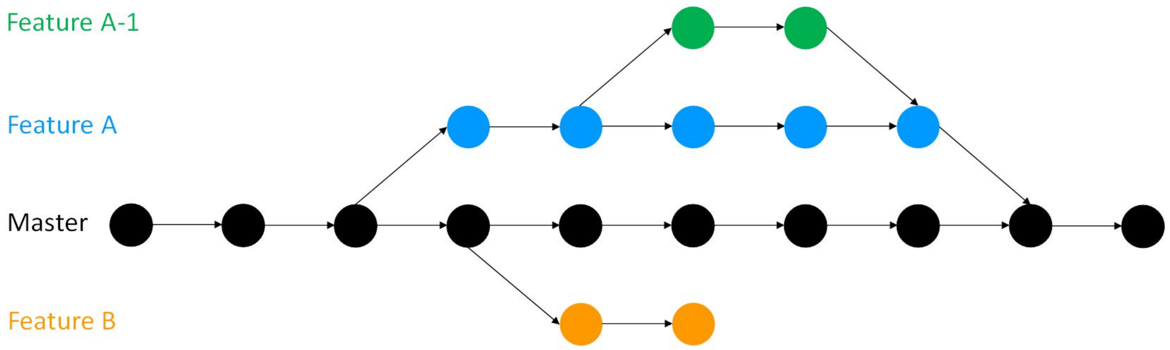
Geir Kjetil Sandve, Anton Nekrutenko, James Taylor, Eivind Hovig

Published: October 24, 2013 • https://doi.org/10.1371/journal.pcbi.1003285

10 Simple Rules

- Rule 1: For Every Result, Keep Track of How It Was Produced
- Rule 2: Avoid Manual Data Manipulation Steps
- Rule 3: Archive the Exact Versions of All Extemal Programs Used
- Rule 4: Version Control All Custom Scripts
- Rule 5: Record All Intermediate Results, When Possible in Standardized Formats
- Rule 6: For Analyses That Include Randomness, Note Underlying Random Seeds
- Rule 7: Always Store Raw Data behind Plots
- Rule 8: Generate Hierarchical Analysis Output, Allowing Layers of Increasing Detail to Be Inspected
- Rule 9: Connect Textual Statements to Underlying Results
- Rule 10: Provide Public Access to Scripts, Runs, and Results

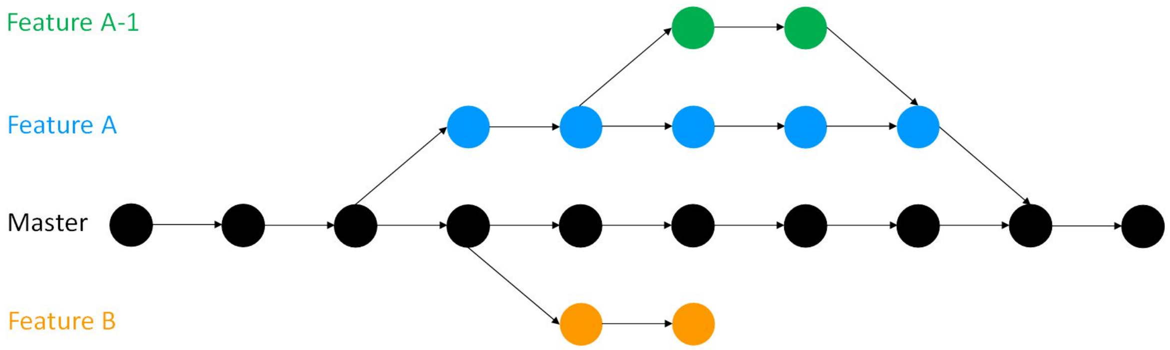
Version Control Everything and Test End-to-End



[The Turing Way]

- Computational environment
- Algorithm implementation
- Analysis pipeline

Version Control Everything and Test End-to-End



The Turing Way

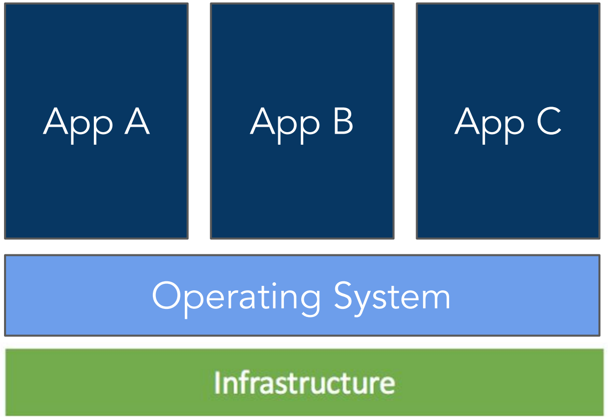
- Computational environment (scriptable software installations)
- Algorithm implementation (R/Python/other source code)
- Analysis pipeline (glue scripts: Make, Python, etc.)

Reproducing Computational Systems

		Interaction style	
		Graphical	Command line
What is reproduced?	Software and versions	**Binder**	**Conda**
	Entire system	**Virtual Machines**	**Containers**

The Turing Way

Personal Computers



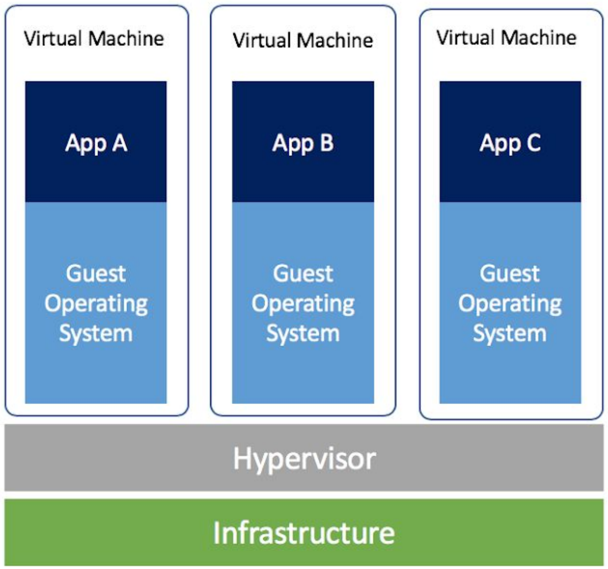
1. Physical hardware infrastructure
2. Download and Install OS
3. Install applications (scriptable in Linux)
4. Run application
5. Problems: interaction between Apps, compatibility with OS and infrastructure

Package Management System (Conda)

- Conda is open source cross-platform package management system
- Keep track of packages and dependencies
- Multiple environments can coexist
- Package availability depends on platform:

  https://anaconda.org/search?q=gcc

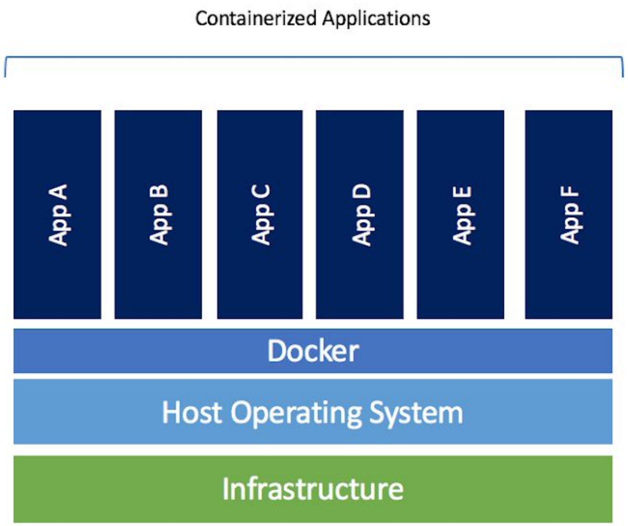
Virtual Machines (VM)



Software emulation of a physical computer

1. One OS + One App = No problems?
2. How to communicate between VMs?
3. How to share storage between VMs?
4. Replicated OS is wasteful

Containers (Docker, Podman, Singularity)



Docker isolates applications under one OS

1. Install Docker
2. Download or build application image
3. Run application in a container
4. One container instance is similar to VM
5. Storage can be shared through Host OS

Image Source

Containers (Docker) vs. VM

- Containers are lighter than VMs

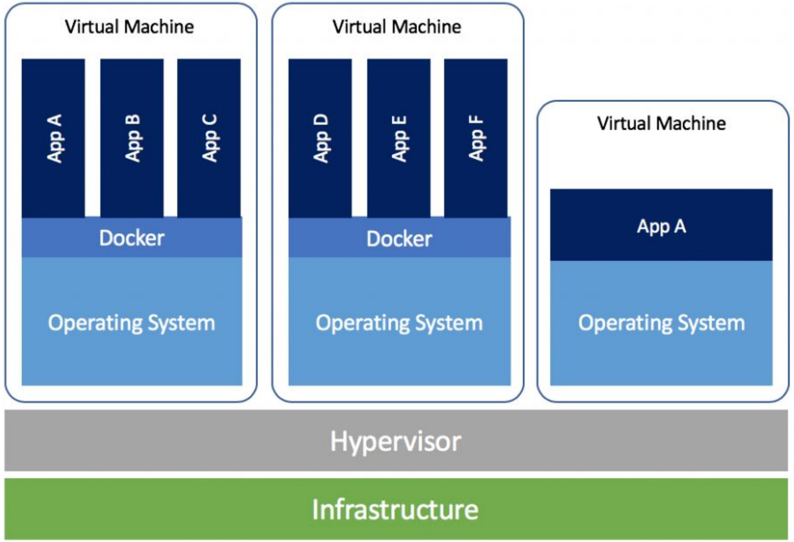
  VM emulate a full computer and installs a full OS

- Container is OS dependent, VM is OS independent

  Not an issue for centrally managed environments

- VM and Containerized applications can co-exist

Everyday Reproducible Research Computing



Setup 1

- One docker app for one project/student: e.g., project, teaching, grad student

Setup 2

- Each grad student gets one VM instance
- Self-manage multiple docker apps e.g. 1 for research 1 for teaching

Image Source

Binder (not reliable as of 2023)

- Binder runs on Jupyter framework
- Launches a Docker image around a repository
- Docker image is created from repo2docker
- https://mybinder.org runs Docker image for free
- Many types of environments are possible

Reproducibility Platforms

- CodeOcean [not free]
  Jupyterlab with additional features, hardware, and collaboration
  Nature Publishing, EBSCO

- WholeTale [free for now]
  NSF funded platform development
  Similar to CodeOcean

- CodeOcean and WholeTale exports environment specifications

- Many are built on Docker and Jupyter framework

Questions?

Jupyter Hub for Data Science Courses at UCSB

- Currently running for 10 courses per quarter at UCSB
    - ~5 classes using R / Rstudio
    - ~5 classes using Python Notebooks
- Serve approximately 1000+ students per quarter
- Students can access to their from any web browser
- Can easily give computer exams