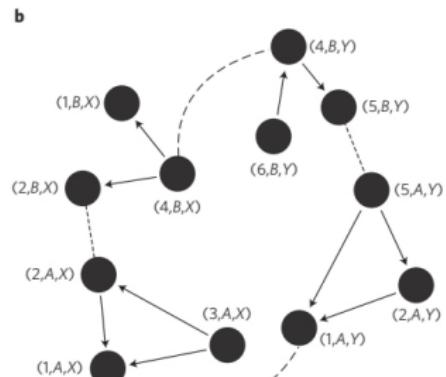
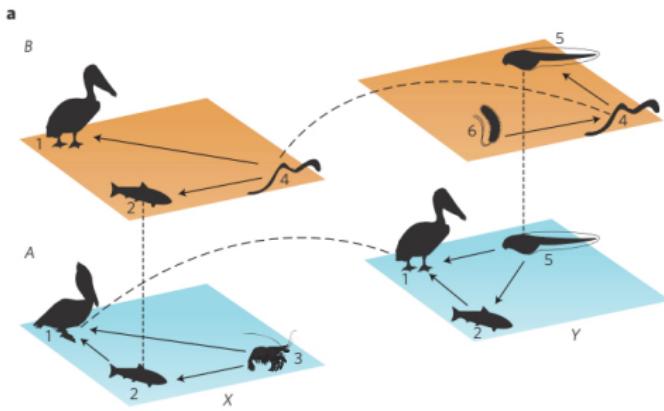


# Opportunity in the reproducibility crisis: Computational tools to improve scientific benefaction

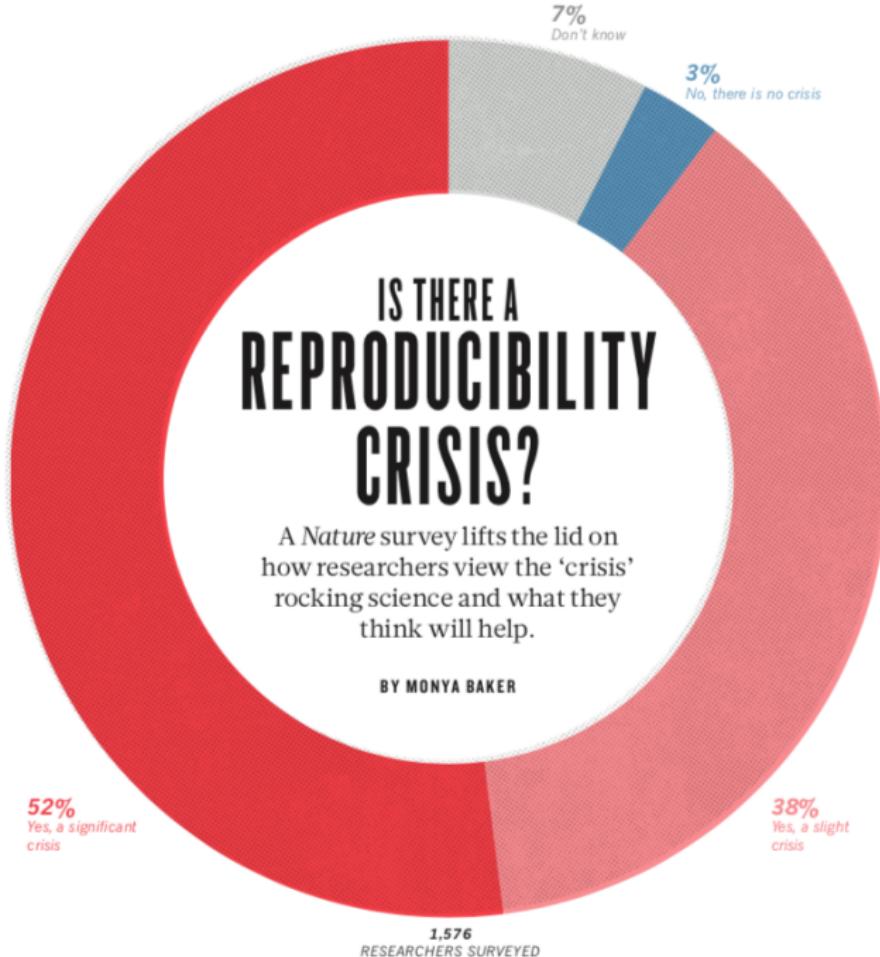


The next great challenge in synthesis



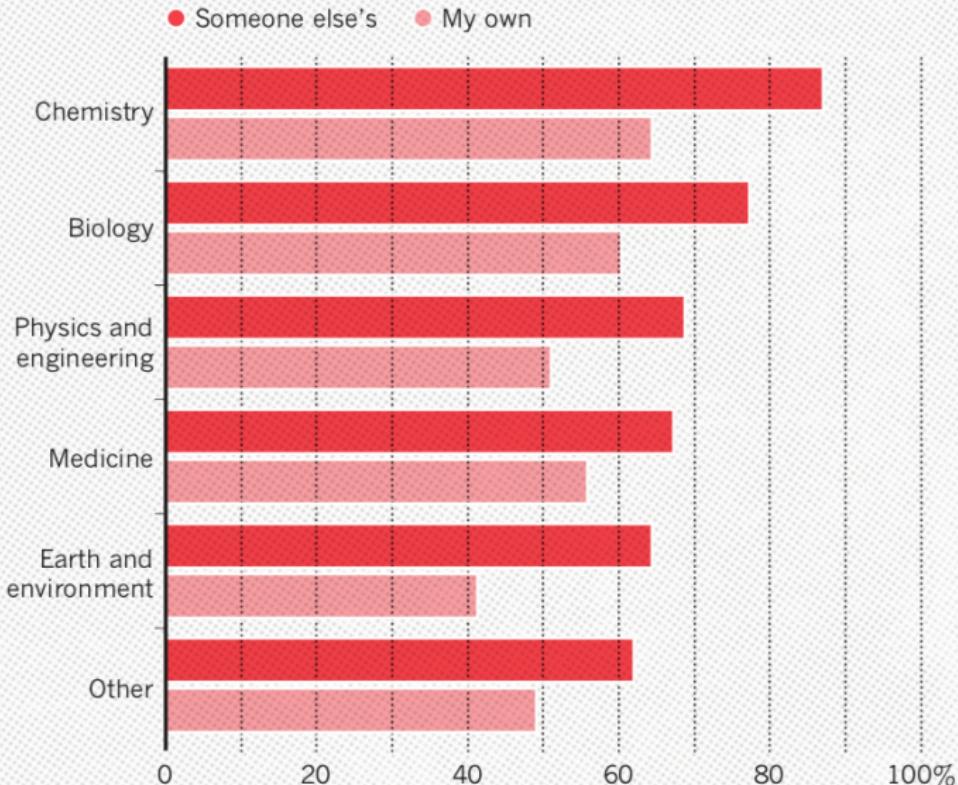
“Software should not limit science.”





# HAVE YOU FAILED TO REPRODUCE AN EXPERIMENT?

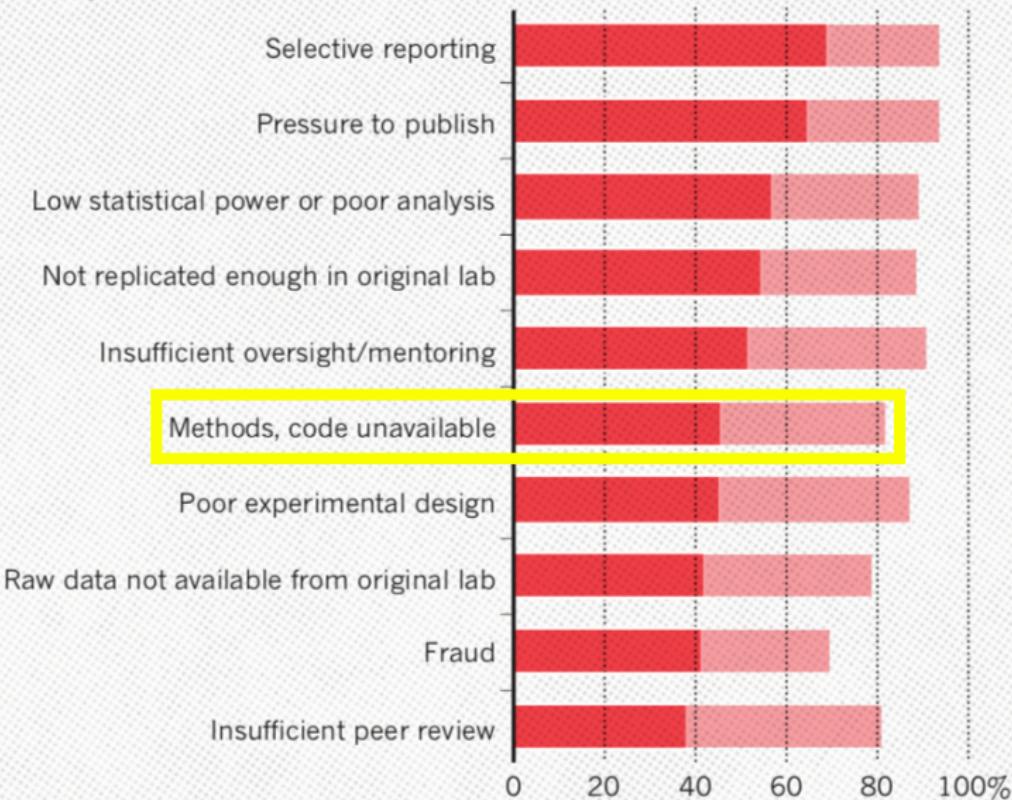
Most scientists have experienced failure to reproduce results.



# WHAT FACTORS CONTRIBUTE TO IRREPRODUCIBLE RESEARCH?

Many top-rated factors relate to intense competition and time pressure.

- Always/often contribute
- Sometimes contribute







# SCIENTIFIC DATA

110110  
0111101  
11011110  
011101101

OPEN

## Comment: If these data could talk

Thomas Pasquier<sup>1</sup>, Matthew K. Lau<sup>2</sup>, Ana Trisovic<sup>3,4</sup>, Emery Boose<sup>2</sup>, Ben Couturier<sup>3</sup>, Mercè Crosas<sup>5</sup>, Aaron M. Ellison<sup>2</sup>, Valerie Gibson<sup>4</sup>, Chris Jones<sup>4</sup> & Margo Seltzer<sup>1</sup>

In the last few decades, data-driven methods have come to dominate many fields of scientific inquiry. Open data and open-source software have enabled the rapid implementation of novel methods to manage and analyze the growing flood of data. However, it has become apparent that many scientific fields exhibit distressingly low rates of repeatability and reproducibility. Although there are many dimensions to this issue, we believe that there is a lack of formalism used when describing end-to-end published results, from the data source to the analysis to the final published results. Even when authors do their best to make their research and data accessible, this lack of formalism reduces the clarity and efficiency of reporting, which contributes to issues of reproducibility. Data provenance aids both repeatability and reproducibility through *systematic* and *formal* records of the relationships among data sources, processes, datasets, publications and researchers.

Received: 12 April 2017

Accepted: 24 July 2017

Published: xx xxx 2017





# Sharing and Preserving Computational Analyses for Posterity with *encapsulator*

**Thomas Pasquier**  
University of Cambridge

**Matthew K. Lau and Xueyuan Han**  
Harvard University

**Elizabeth Fong and Barbara S. Lerner**  
Mount Holyoke College

**Emery R. Boose, Mercè Crosas, Aaron M. Ellison, and Margo Seltzer**  
Harvard University

**Editors:** Lorena A. Barba, labarba@gwu.edu; George K. Thiruvathukal, gkt@cs.luc.edu

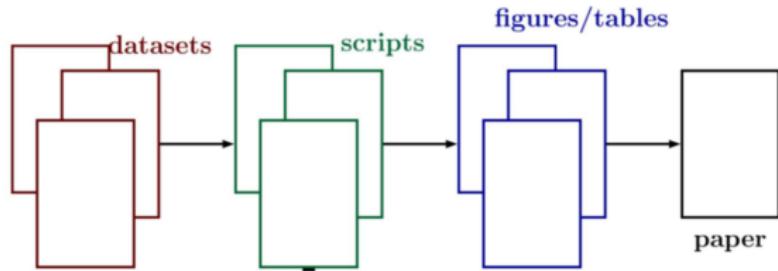
Reproducibility has become a recurring topic of discussion in many scientific disciplines.<sup>1</sup> Although it might be expected that some studies will be difficult to reproduce, recent conversations highlight important aspects of the scientific endeavor that could be improved to facilitate reproducibility. Open data and open source software are two important parts of a concerted effort to achieve reproducibility.<sup>2</sup> However, multiple publications point out these approaches' shortcomings,<sup>3,4</sup> such as the identification of dependencies, poor documentation of the installation processes, "code rot," failure to capture dynamic inputs, and technical barriers.

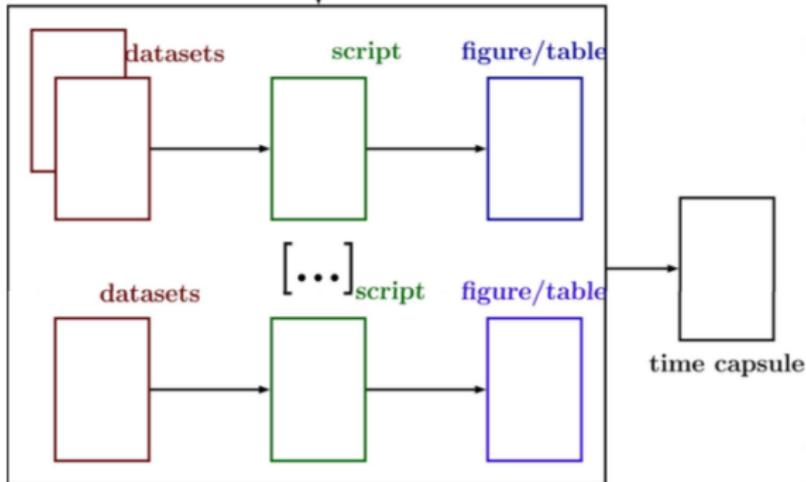
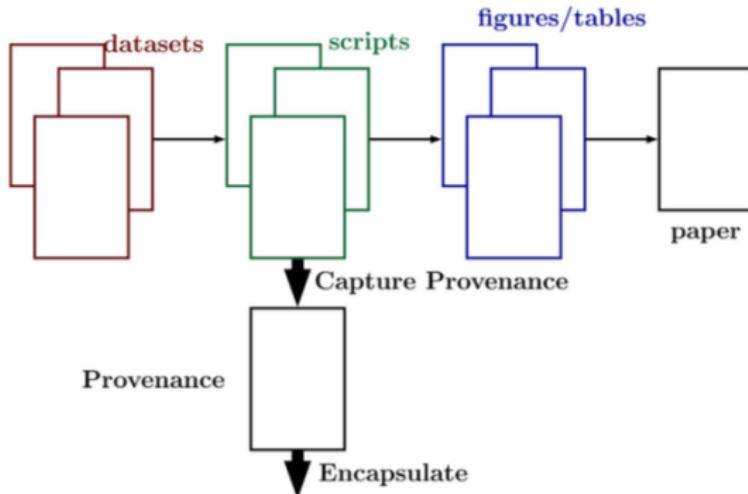
In prior work,<sup>5</sup> we pointed out that open data and open source software alone are insufficient to ensure reproducibility, as they do not capture information about the computational execution, that is, the "process" and context that produced the results using the data and code. In keeping with the "open" culture, we defined open process as the practice of both sharing the source and the input data and providing a description of the entire computational

# Encapsulator

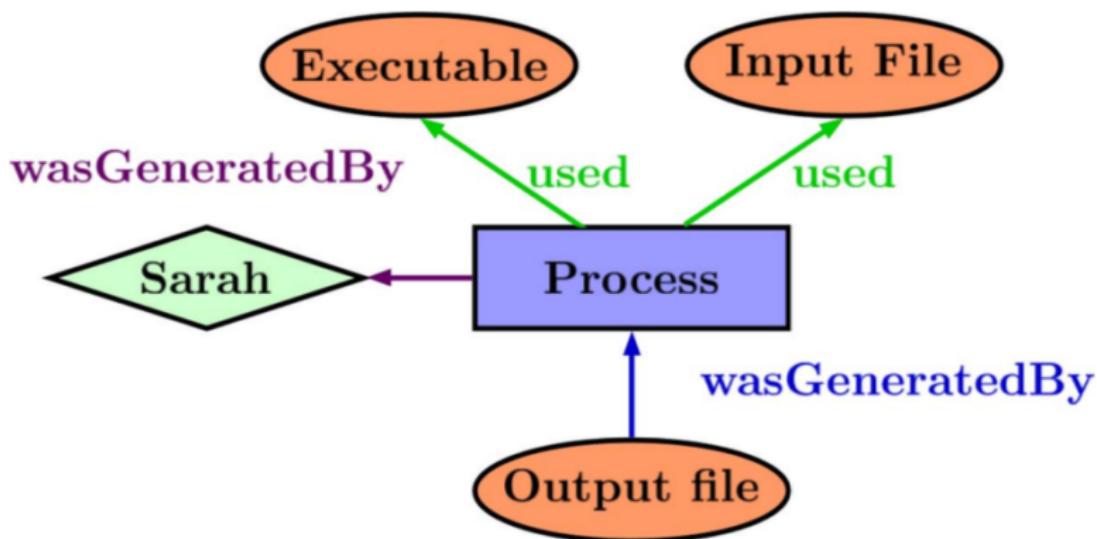
Goal: Simplify computational reproducibility

1. Create a data “capsule” with code, data and environment
2. Increase transparency with “cleaned” code and workspace

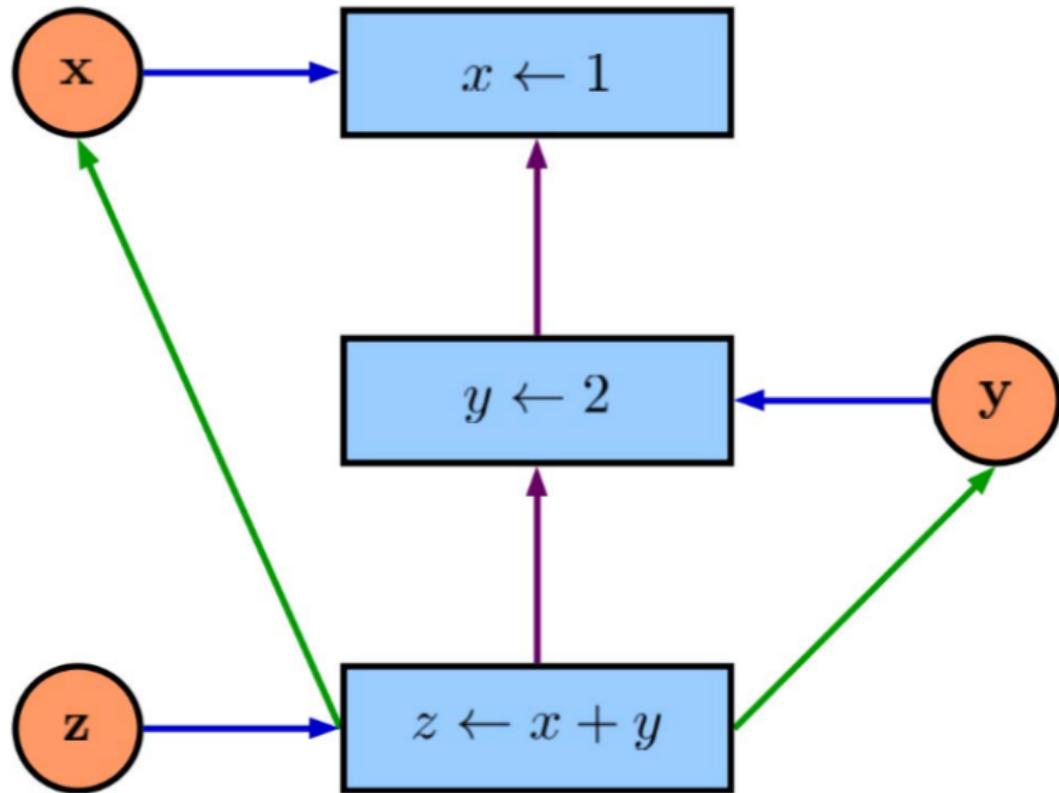




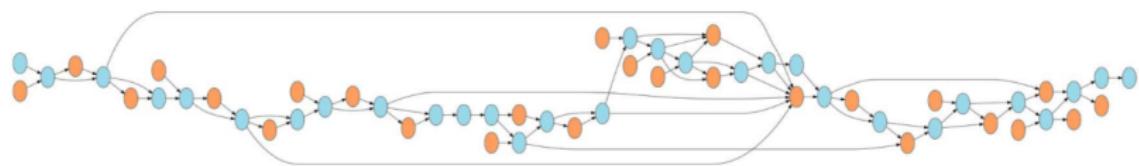
## What is data provenance?

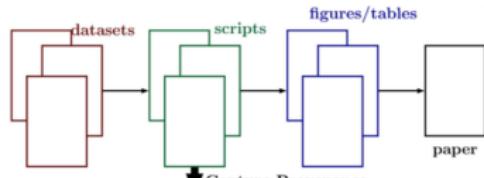


## What is data provenance?



Prov. Huh. What is it good for?

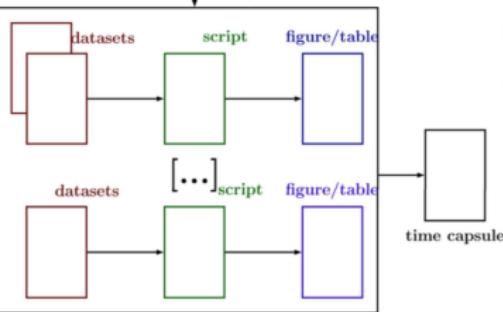


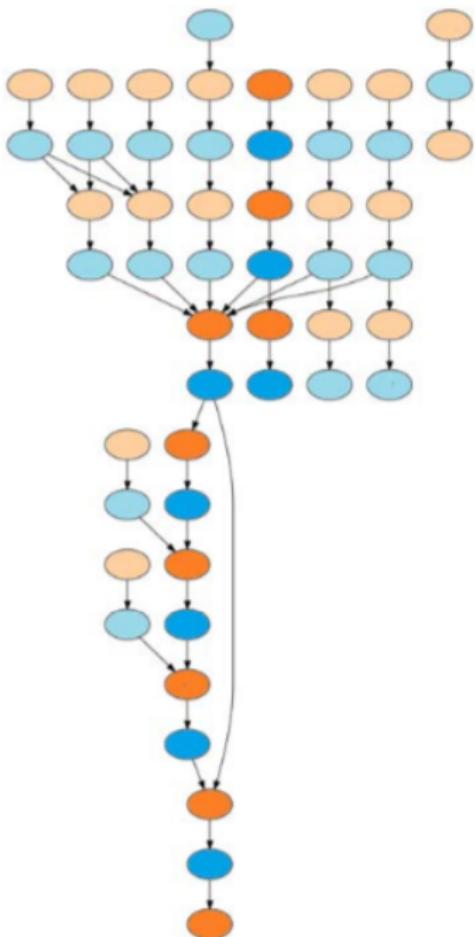


Capture Provenance

Provenance

Encapsulate





# Challenges

- ▶ Out-of-tree Libraries
- ▶ Nondeterminism
- ▶ Capsule OS (*Linux, VirtualBox*)
- ▶ Language Support (*R*)

## Future Directions

- ▶ Test Dataverse products
- ▶ Integration with IDEs (*RStudio*)
- ▶ Container Support (e.g. *Docker*)
- ▶ Domain-Specific Environments

Email: ***matthewklau@fas.harvard.edu*** Github: **MKLau**

*This work was supported by NSF grants SSI-1450277 (End-to-End Provenance) and ACI-1448123 (Citation++).*

*More details about those projects are available at  
<https://projects.iq.harvard.edu/provenance-at-harvard>.*