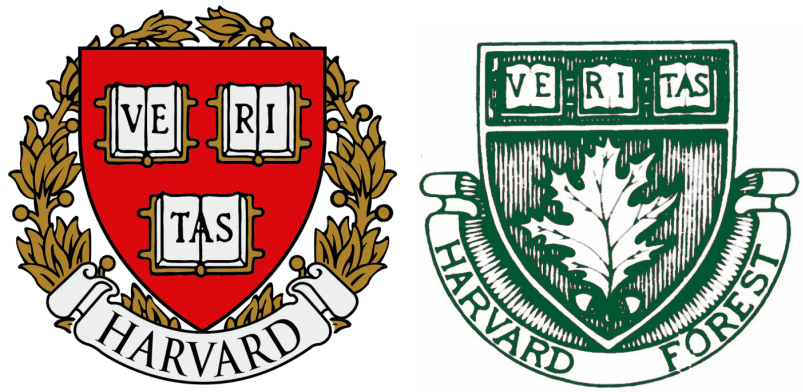
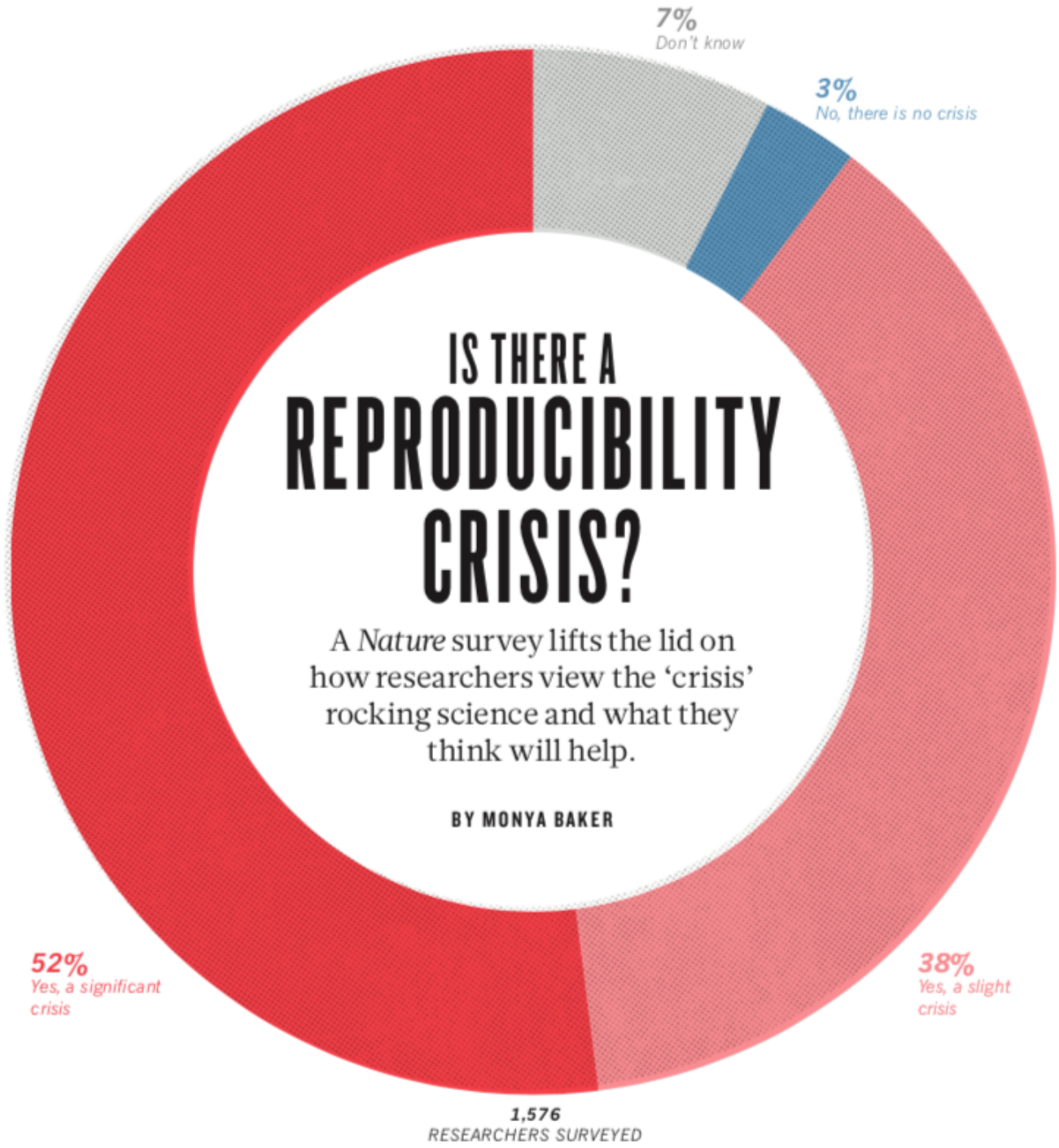
Opportunity in the Reproducibility Crisis

Computational tools to improve scientific benefaction

Matthew K. Lau, PhD

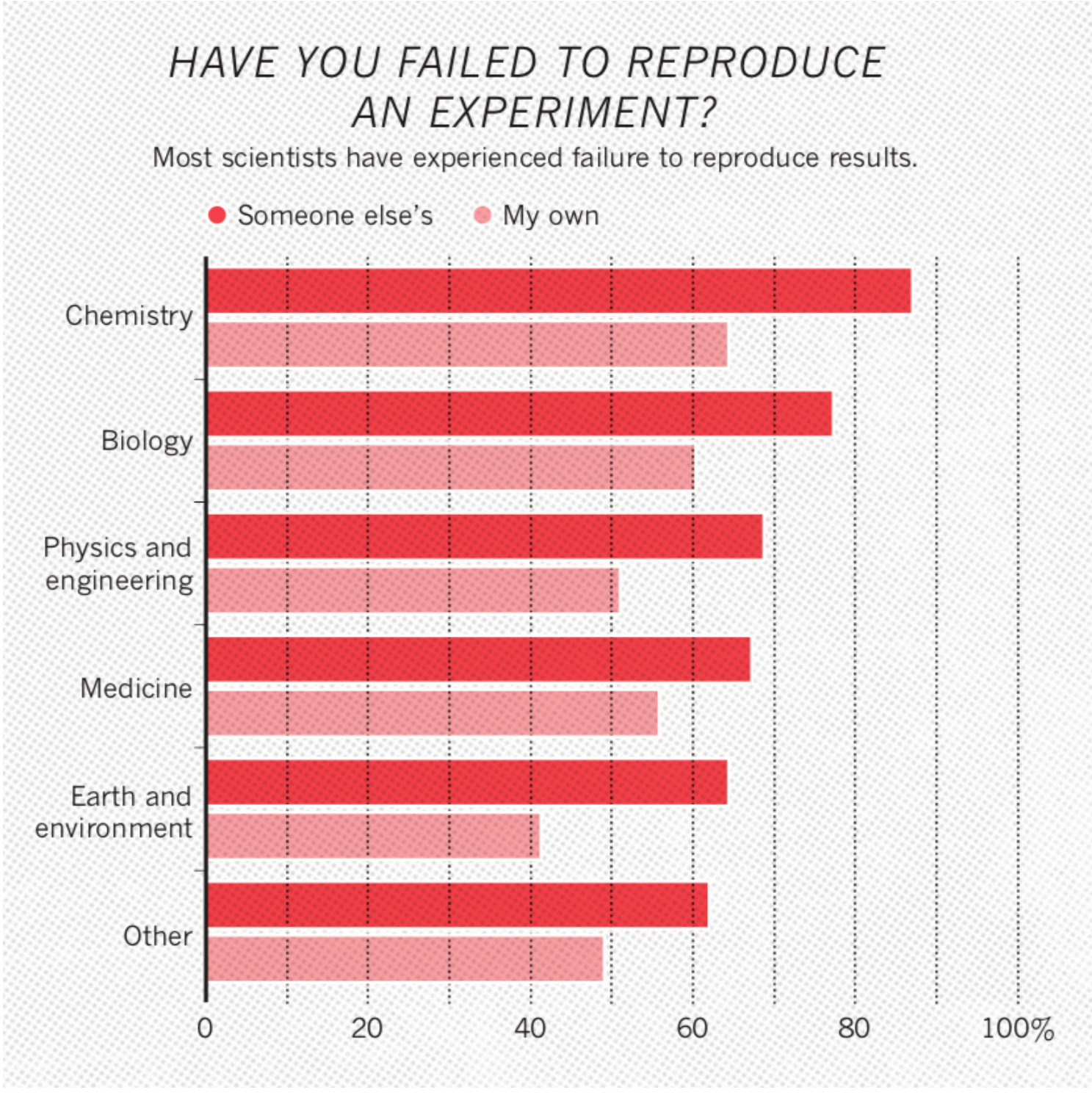




IS THERE A REPRODUCIBILITY CRISIS?

A *Nature* survey lifts the lid on how researchers view the 'crisis' rocking science and what they think will help.

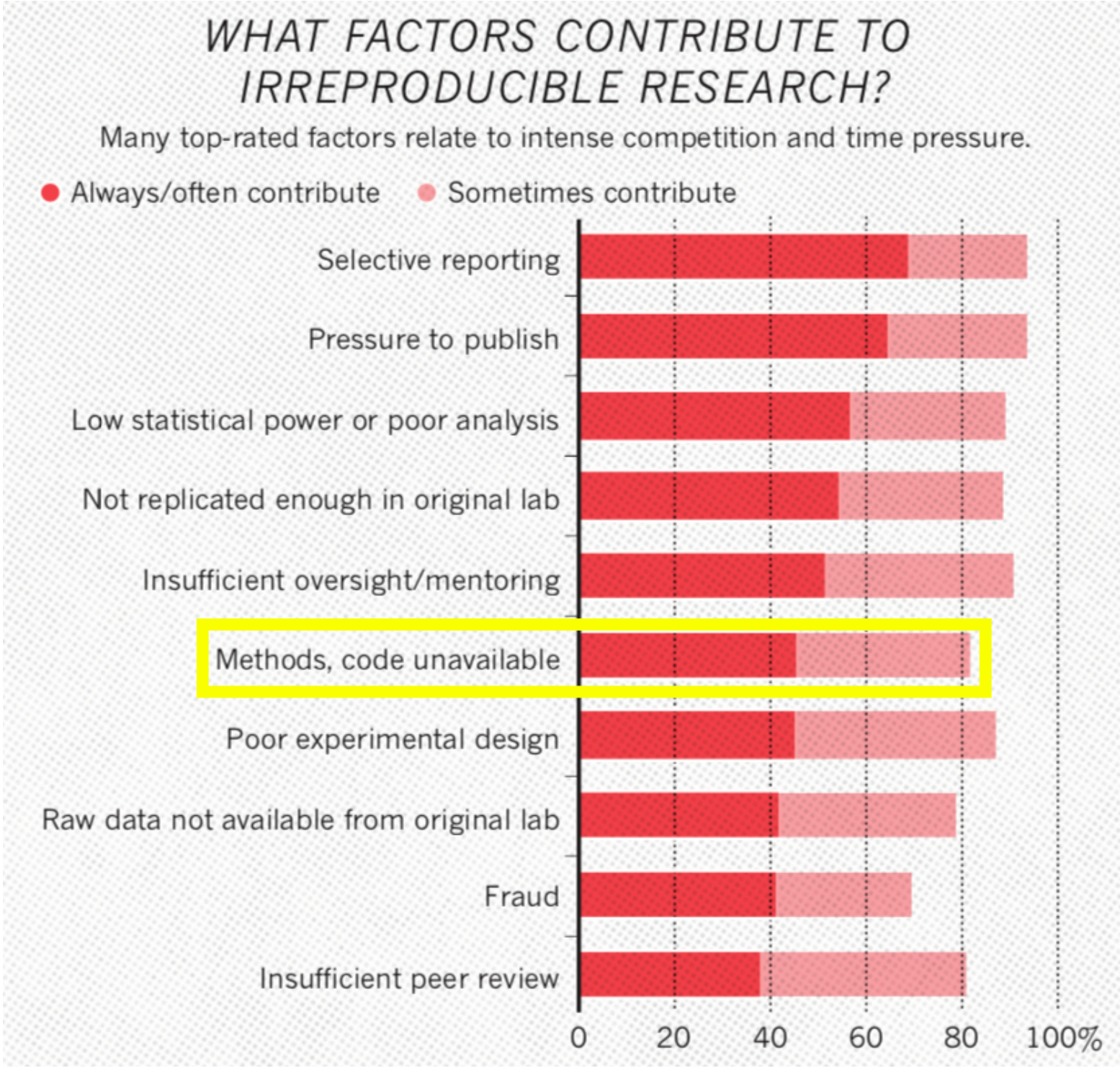
BY MONYA BAKER


HAVE YOU FAILED TO REPRODUCE AN EXPERIMENT?
Most scientists have experienced failure to reproduce results.
Someone else's
My own
Chemistry
Biology
Physics and engineering
Medicine
Earth and environment
Other
0
20
40
60
80
100%

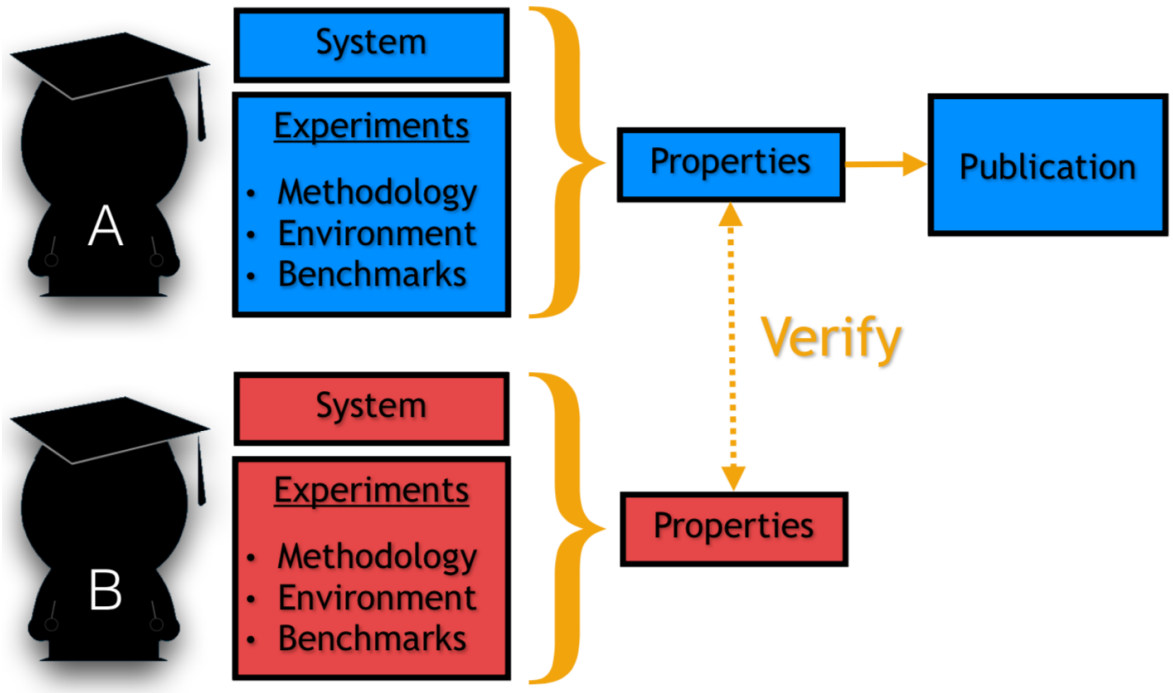
The Chinese use two brush strokes to write the word “crisis.” One brush stroke stands for danger; the other for opportunity. In a crisis, be aware of the danger – but recognize the opportunity.

John F. Kennedy

Speech in Indianapolis (April 12, 1959)

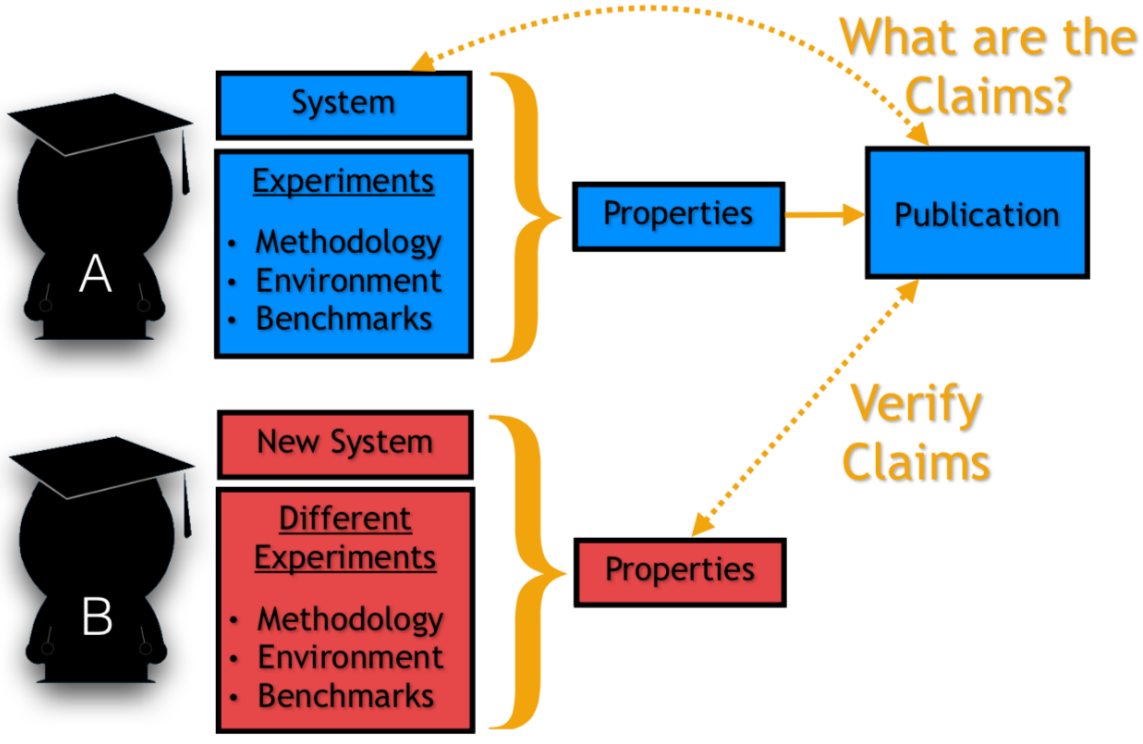

WHAT FACTORS CONTRIBUTE TO IRREPRODUCIBLE RESEARCH?
Many top-rated factors relate to intense competition and time pressure.
Always/often contribute
Sometimes contribute
Selective reporting
Pressure to publish
Low statistical power or poor analysis
Not replicated enough in original lab
Insufficient oversight/mentoring
Methods, code unavailable
Poor experimental design
Raw data not available from original lab
Fraud
Insufficient peer review
0
20
40
60
80
100%

Opportunity: Benefaction not just reproducibility



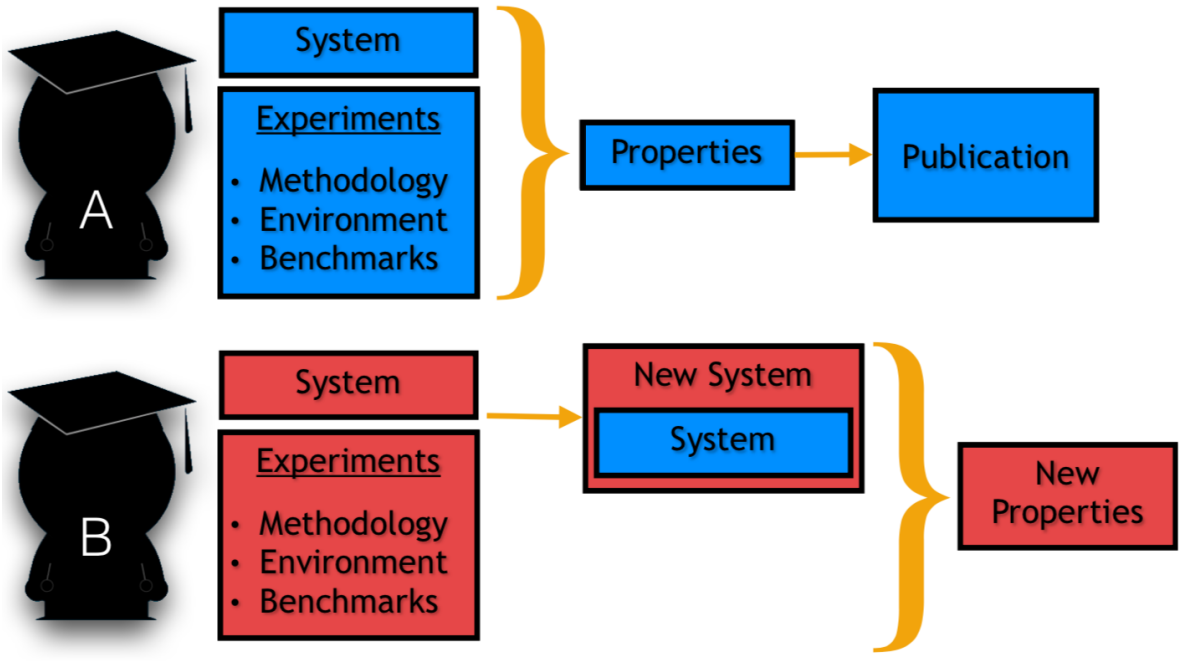
Colberg et al. 2015 Comm ACM

Opportunity: Benefaction not just reproducibility



Colberg et al. 2015 Comm ACM

Opportunity: Benefaction not just reproducibility



Colberg et al. 2015 Comm ACM

Benefaction: Computational Best Practices

- Open data (FAIR, Dataverse, Data Dryad)
- Open software (R, python, Julia)
- Project architecture (README, src, doc, data, bin, results)
- Notebooks (Rmarkdown, Jupyter)
- Version control (git, subversion)

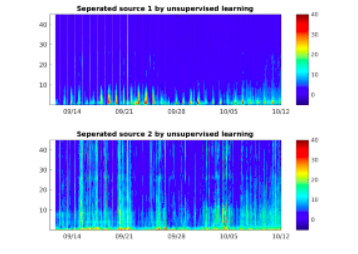
Benefaction: Capsules & Continuous Integration

CODE OCEAN BETA | EXPLORE | HELP | + UPLOAD YOUR CODE | LOG IN | SIGN UP

Search keyword, research field, title, author, DOI, etc.

More Capsules

ENGINEERING Nov 2018



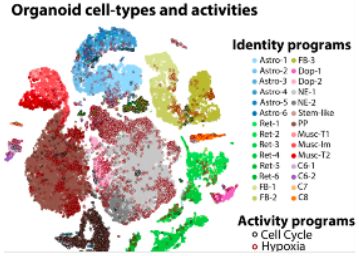
Tzu-Hao Lin

Demonstration of soundscape separation by using the Soundscape_Viewer

A demonstration of using...

Scientific Reports, 2017

BIOINFORMATICS Nov 2018



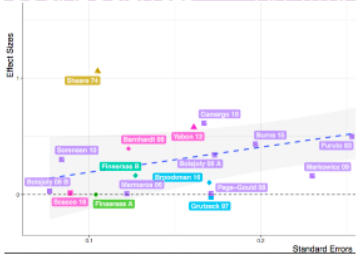
Dylan Kotliar, Adrian Veres, M. ...

Identifying Gene Expression Programs of Cell-type Identity and Cellular Activity with...

This capsule is meant to...

bioRxiv, 2018

SOCIAL SCIENCES Oct 2018



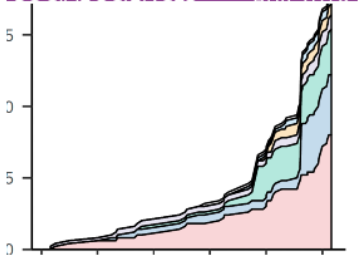
Betsy Levy Paluck, Seth Ariel Gr...

The contact hypothesis re-evaluated: code and data

This code reproduces the...

Behavioural Public Policy, 2018

SOCIAL SCIENCES Aug 2018



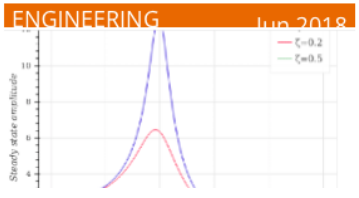
Tom Hardwicke, John Ioannidis

Mapping the Universe of Registered Reports

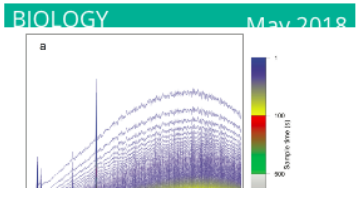
The Registered Reports format...

Nature Human Behavi... , 2018

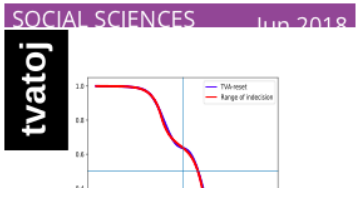
ENGINEERING Jun 2018



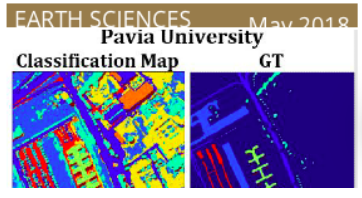
BIOLOGY May 2018



SOCIAL SCIENCES Jun 2018



EARTH SCIENCES May 2018



Benefaction: Capsules & Continuous Integration

[](travis.png)

Scientific culture is shifiting

IDEAS IN ECOLOGY AND EVOLUTION 8: 55–57, 2015
doi:10.4033/iee.2015.8.9.c

Received 13 May 2015; Accepted 8 June 2015

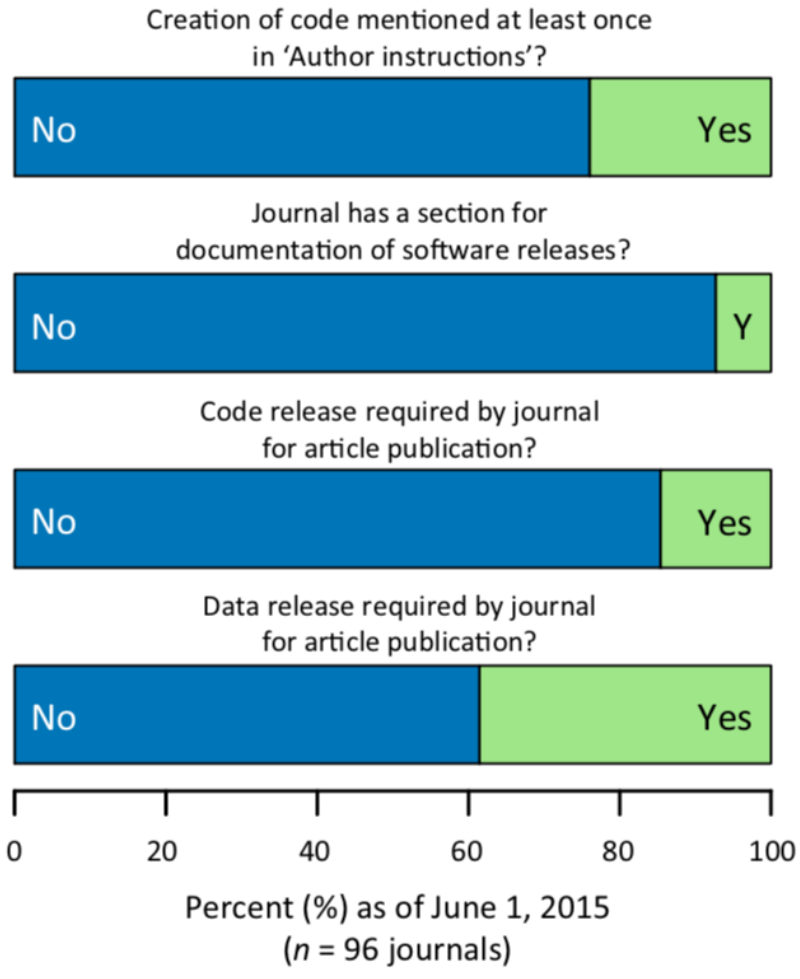
Commentary

Some thoughts on best publishing practices for scientific software

Ethan P. White

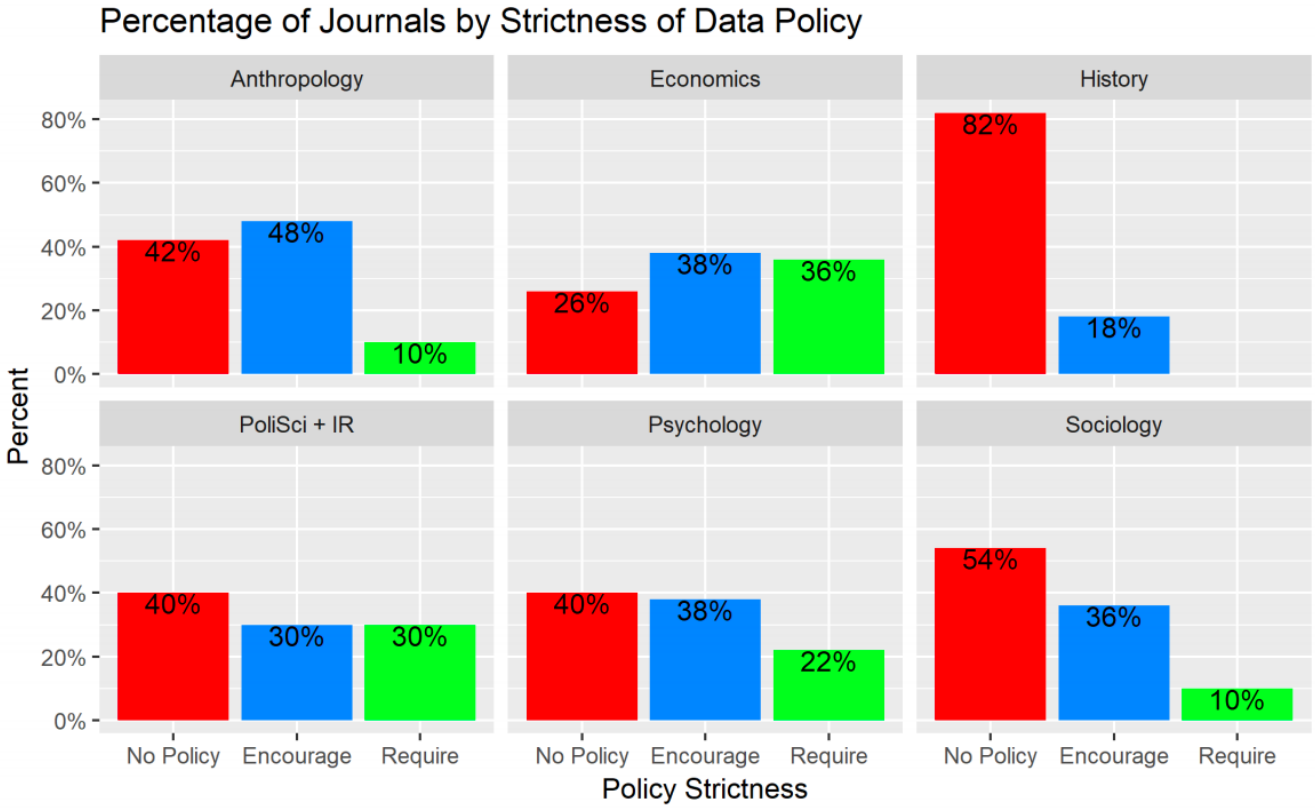
Ethan P. White (ethan@weecology.org), Department of Wildlife Ecology and Conservation, University of Florida, Gainesville, FL 32611-0430 and *Department of Biology, Utah State University, Logan, UT 84322*

Scientific culture is shifiting



Meeslan, Heer and White 2016 Trends in Eco Evo

Scientific culture is shifiting



Crosas et al. 2018 SocArXiv

Journal Policy Impacts

PNAS PNAS PNAS

An empirical analysis of journal policy effectiveness for computational reproducibility

Victoria Stodden[a,1], **Jennifer Seiler**[b], **and Zhaokun Ma**[b]

[a]School of Information Sciences, University of Illinois at Urbana–Champaign, Champaign, IL 61820; and [b]Department of Statistics, Columbia University, New York, NY 10027

Edited by David B. Allison, Indiana University Bloomington, Bloomington, IN, and accepted by Editorial Board Member Susan T. Fiske January 9, 2018 (received for review July 11, 2017)

A key component of scientific communication is sufficient information for other researchers in the field to reproduce published findings. For computational and data-enabled research, this has often been interpreted to mean making available the raw data from which results were generated, the computer code that generated the findings, and any additional information needed such as workflows and input parameters. Many journals are revising author guidelines to include data and code availability. This work evaluates the effectiveness of journal policy that requires the data and code necessary for reproducibility be made available postpublication by the authors upon request. We assess the effectiveness of such a policy by (*i*) requesting data and code from authors and (*ii*) attempting replication of the published findings. We chose a random sample of 204 scientific papers published in the journal *Science* after the implementation of their policy in February 2011. We found that we were able to obtain artifacts from 44% of our sample and were able to reproduce the findings for 26%. We find this policy—author remission of data and code postpublication upon request—an improvement over no policy, but currently insufficient for reproducibility.

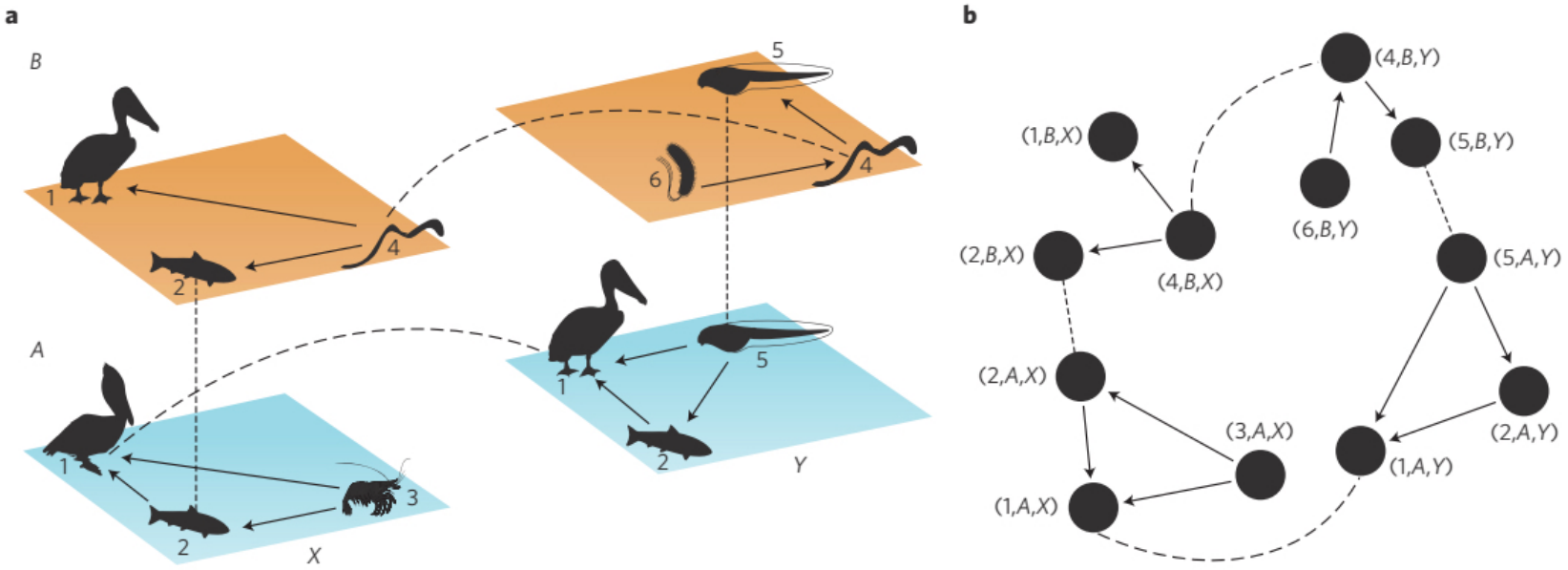
reproducible research | data access | code access | reproducibility policy | open science

putational reproducibility of published results. We use a survey instrument to test the availability of data and code for articles published in *Science* in 2011–2012. We then use the scientific communication standards from the 2012 Institute for Computational and Experimental Research in Mathematics (ICERM) workshop report to evaluate the reproducibility of articles for which artifacts were made available (11). We then assess the impact of the policy change directly, by examining articles published in *Science* in 2009–2010 and comparing artifact ability to our postpolicy sample from 2011–2012. Finally, we discuss possible improvements to journal policies for enabling reproducible computational research in light of our results.

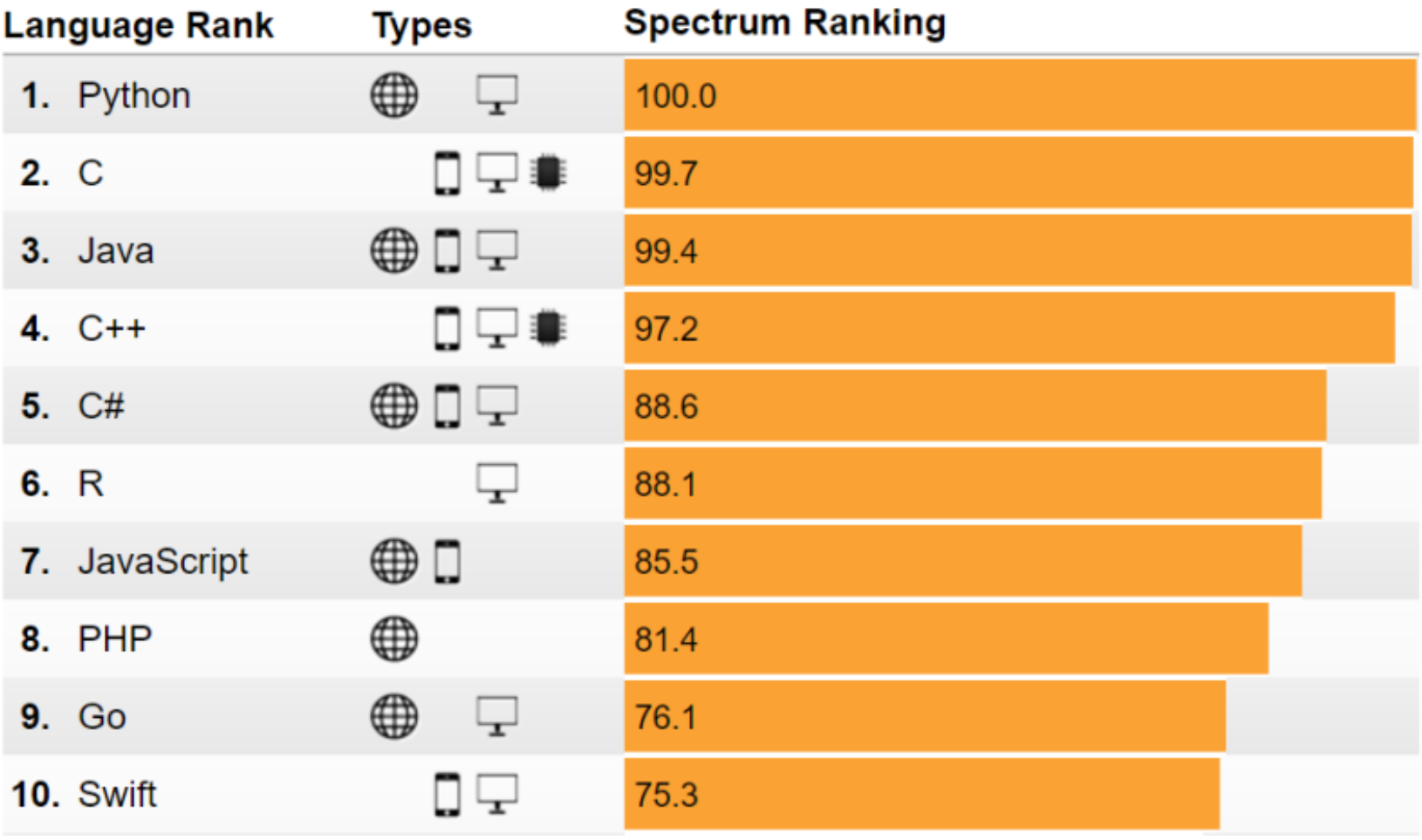
Results

We emailed corresponding authors in our sample to request the data and code associated with their articles and attempted to replicate the findings from a randomly chosen subset of the articles for which we received artifacts. We estimate the artifact recovery rate to be 44% with a 95% bootstrap confidence interval of the proportion $[0.36, 0.50]$, and we estimate the replication rate to be 26% with a 95% bootstrap confidence interval $[0.20, 0.32]$.


PANIC


a
B
A
1
2
3
4
5
6
X
Y
b
(4,B,Y)
(1,B,X)
(5,B,Y)
(6,B,Y)
(2,B,X)
(4,B,X)
(5,A,Y)
(2,A,X)
(2,A,Y)
(3,A,X)
(1,A,Y)
(1,A,X)

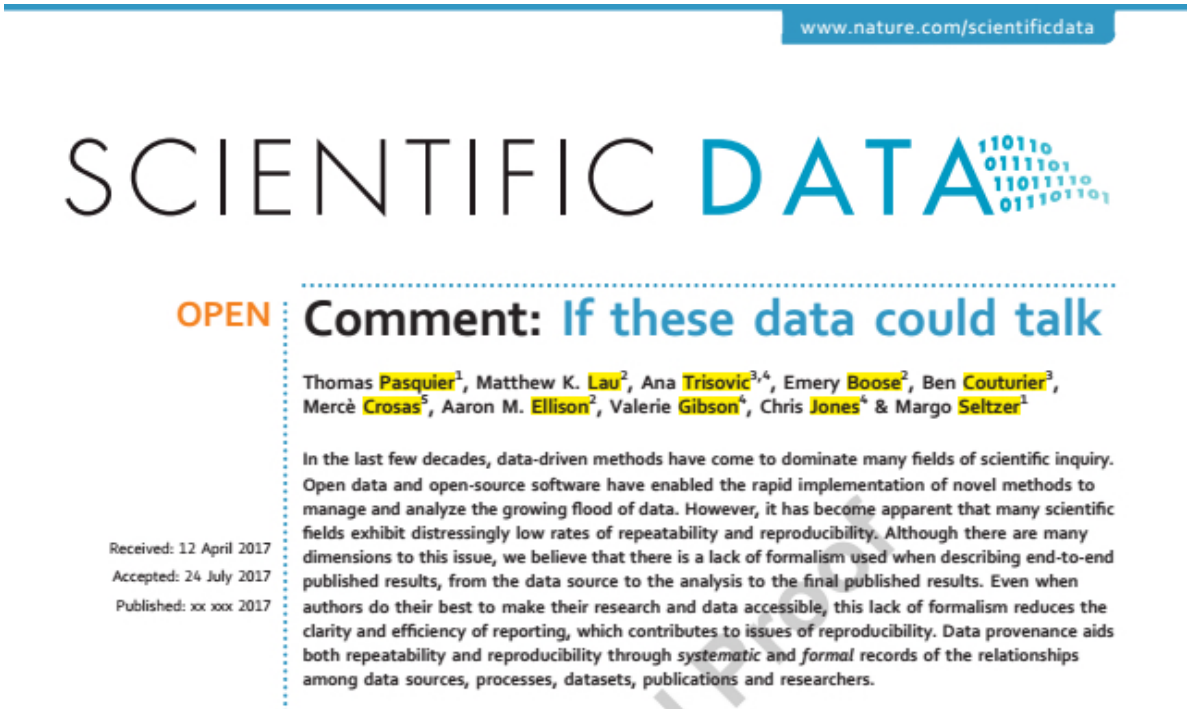
Reality = An Engineering Gap



Reality = The Engineering Gap

Reality = The Engineering Gap

Reality = The Engineering Gap



www.nature.com/scientificdata

SCIENTIFIC DATA

OPEN

Comment: If these data could talk

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

Received: 12 April 2017
Accepted: 24 July 2017
Published: xx xxx 2017

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.

Reality = The Engineering Gap

Most scientists don't want to produce software, they want to do their science.

encapsulator

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

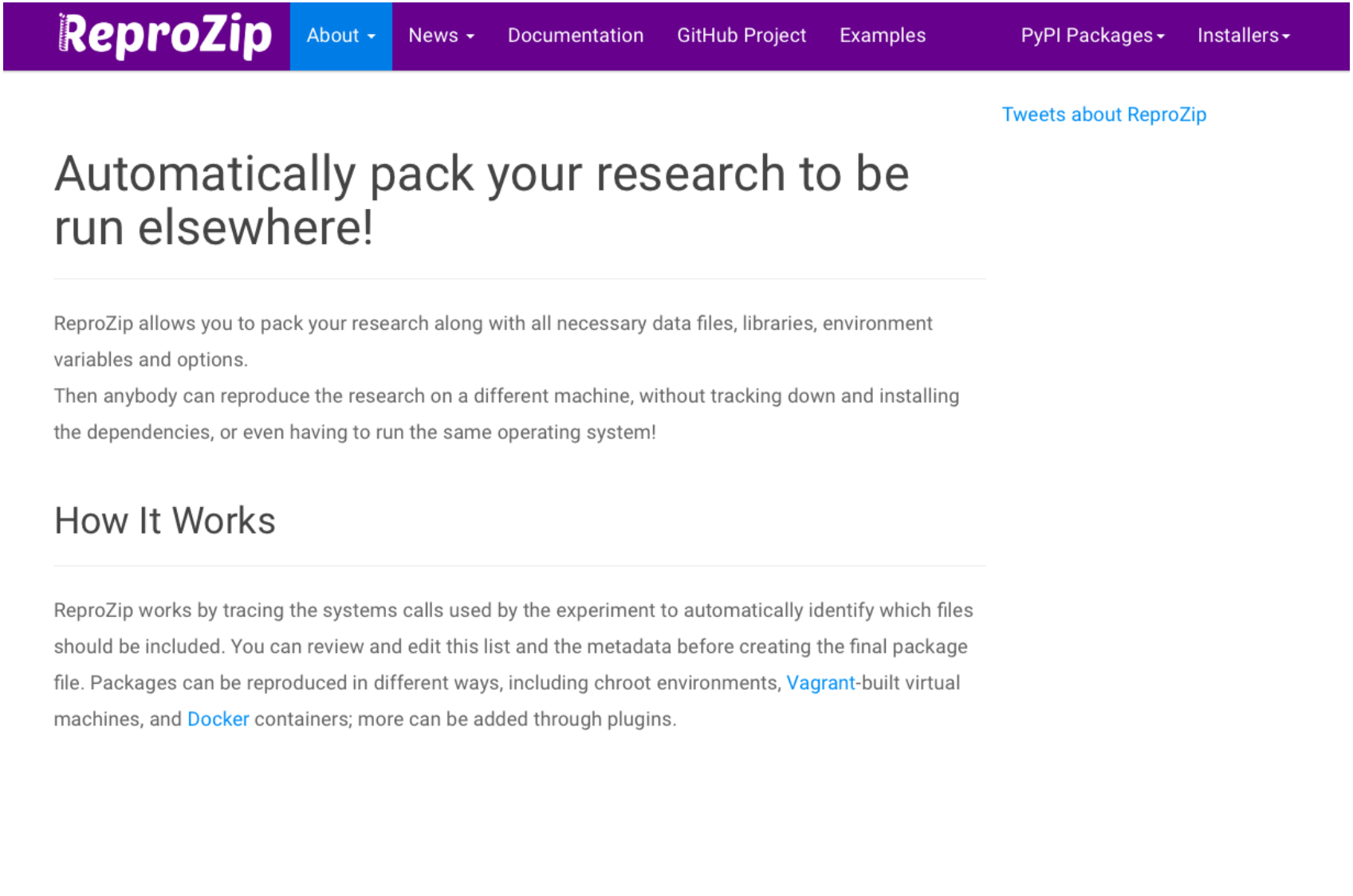
Emery R. Boose, Mercè Crosas, Aaron M. Ellison, and Margo Seltzer
Harvard University

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.[1] 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.[2] However, multiple publications point out these approaches' shortcomings,[3,4] 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,[5] 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 keep-

encapsulator



encapsulator

Goal: Simplify computational reproducibility

1. Create a data "capsule" with code, data and environment

encapsulator

Goal: Simplify computational reproducibility

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

encapsulator

Goal: Simplify computational reproducibility

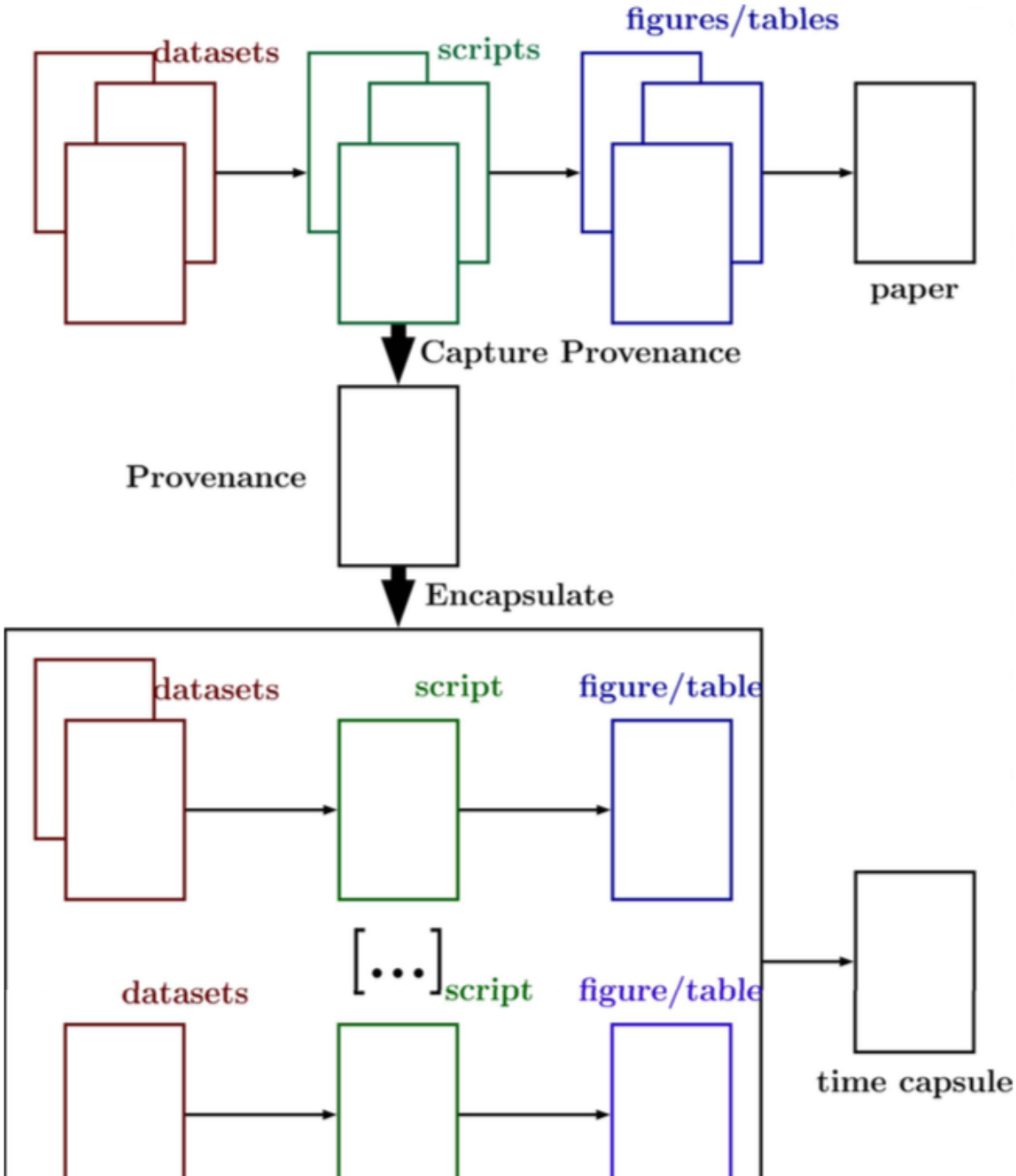
1. Capsule = all necessary software and data
2. Cleaned = organize files, remove non-essential code and re-format

encapsulator

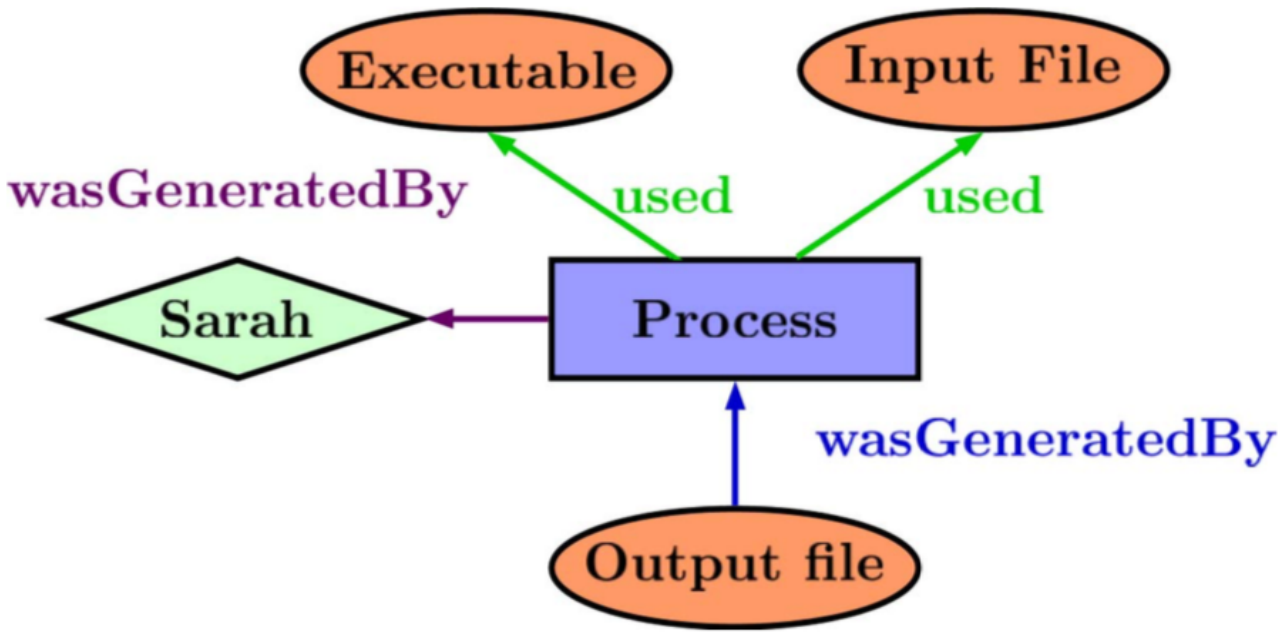
Basic Usage (current paradigm):

1. Code as usual in your normal environment while recording provenance
2. Run encapsulator from the console
3. List desired results
4. Product = Capsule containing essential code and data with a virtual machine

encapsulator



What is data provenance?



encapsulator

Example: Messycode.R

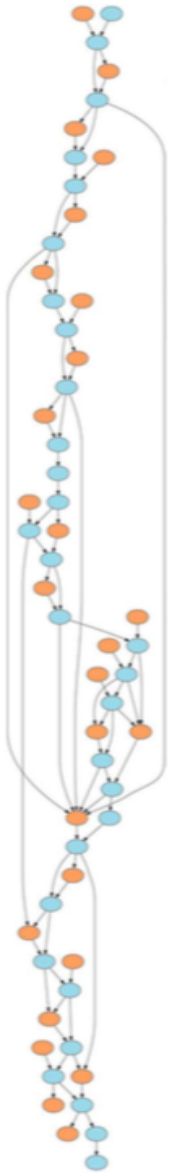


Figure 4. Provenance graph corresponding to a small R script (approximately 60 lines of code).

encapsulator

Example: Messycode.R

- near stream-of-consciousness coding that follows a train of thought in script development,
- output to console that is not written to disk,
- intermediate objects that are abandoned,
- library and new data calls throughout the script,
- output written to disk but not used in final documents,
- code that is not modularized,
- code that is syntactically correct but not particularly comprehensible.

encapsulator

Example: Messycode.R

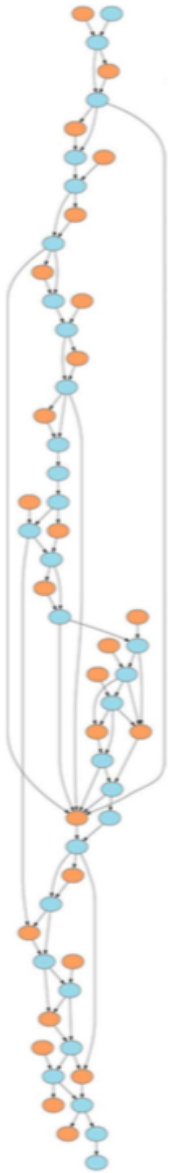


Figure 4. Provenance graph corresponding to a small R script (approximately 60 lines of code).

encapsulator

Example: Messycode.R

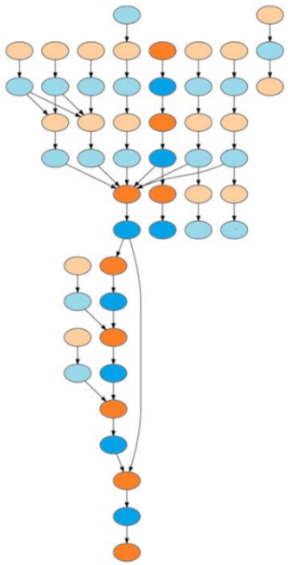


Figure 5. Data dependency transformation of the provenance graph shown in Figure 4.

Conclusion

- Capsules = sharing (data, code, metadata)
- Continuous Integration = development * sharing * discovery
- Support the transition to using CI and capsules

Thanks you

Contact Info:

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

Github: MKLau

Slack: MKLau

Much of this work was supported by NSF SSI-1450277 (End-to-End Provenance) and ACI-1448123 (Citation++). More details are available at https://projects.iq.harvard.edu/provenance-at-harvard