Opportunity in the Reproducibility Crisis

Computational tools to improve scientific benefaction

Matthew K. Lau, PhD

Overview

- Crisis: rampant irreproducibility
- Opportunity: better science

Goal

- Easier methods for bridging the software engineering gap
- Capsules and provenance
- mkSci = continuous integration is the future of science

Benefaction

- Trisovic et al 2018

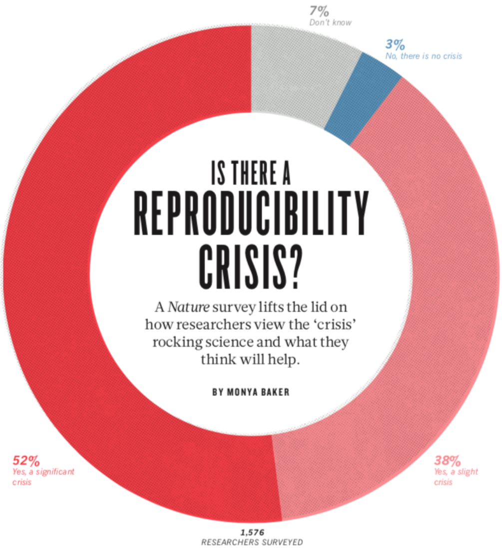
Reproducibility: Computation

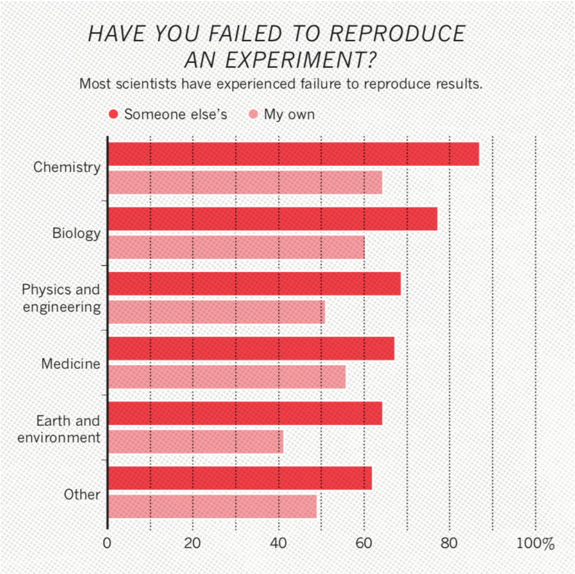
- Project architecture (README, src, doc, data, bin, results)
- Notebooks (Rmarkdown, Jupyter)
- Version control (git, subversion)
- Capsule Databases/Engines (CodeOcean, Globus)
- Continuous Integration (at the beginning)
- Scalability?

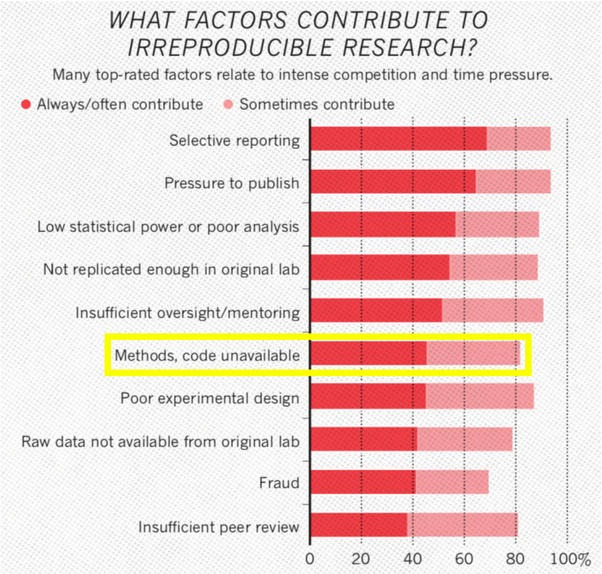
IS THERE A REPRODUCIBILITY CRISIS?

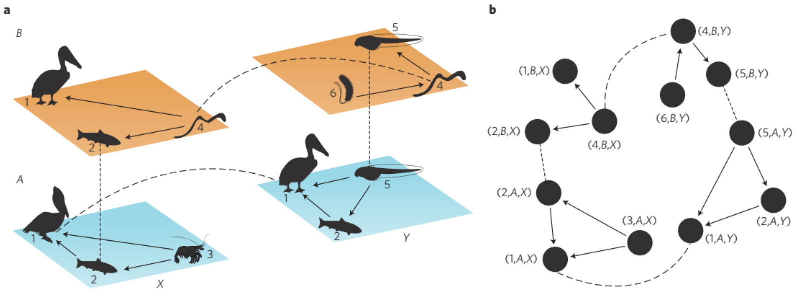
A *Nature* survey lifts the lid on how researchers view the 'crisis' rocking science and what they think will help.

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%


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%



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)
(3,A,X)
(2,A,Y)
(1,A,Y)
(1,A,X)

Motivation: Code in Ecology

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

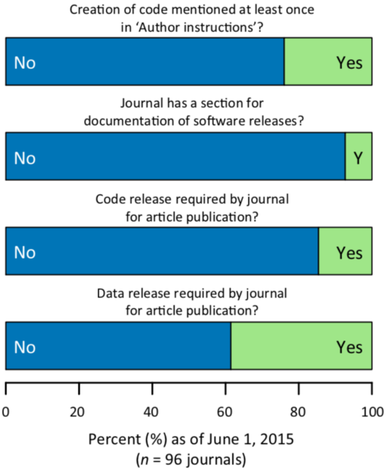
Commentary

Some thoughts on best publishing practices for scientific software

Ethan P. White

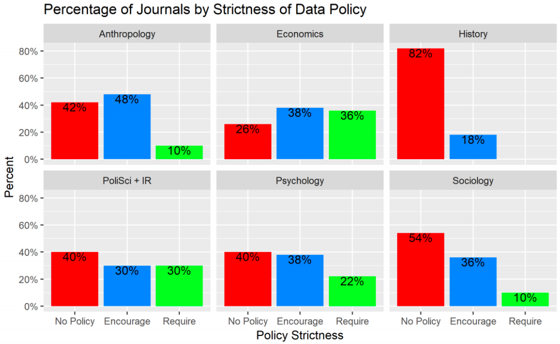
Ethan P. White (ethan@weecology.org), Department of Wildlife Ecology and Conservation, University of Florida, Gainesville, FL 32611-0430 and *Department of Biology, Utah State University, Logan, UT 84322*

Motivation: Ecology Journal Policies



Meeslan, Heer and White 2016 Trends in Eco Evo

Motivation: Social Science Journal Policies



Crosas et al. 2018 SocArXiv

Motivation: Journal Policy Impacts

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 $[0.20, 0.32]$.

Motivation: Journal Policy Impacts

Table 1. Responses to emailed requests (*n* = 180)

Type of response	Count	Percent, %
Did not share data or code:		
Contact another person	20	11
Asked for reasons	20	11
Refusal to share	12	7
Directed back to supplement	6	3
Unfulfilled promise to follow up	5	3
Impossible to share	3	2
Shared data and code	65	36
Email bounced	3	2
No response	46	26

Motivation: Journal Policy Impacts

Table 2. ICERM implementation criteria for articles deemed likely to reproduce (*n* = 56)

ICERM criteria	Percent compliant, %
A precise statement of assertions to be made in the paper.	100
Full statement (or valid summary) of experimental results.	100
Salient details of data reduction & statistical analysis methods.	91
Necessary run parameters were given.	86
A statement of the computational approach, and why it constitutes a rigorous test of the hypothesized assertions.	8
Complete statements of, or references to, every algorithm used, and salient details of auxiliary software (both research and commercial software) used in the computation.	80
Discussion of the adequacy of parameters such as precision level and grid resolution.	79
Proper citation of all code and data used, including that generated by the authors.	79
Availability of computer code, input and output data, with some reasonable level of documentation.	77
Avenues of exploration examined throughout development, including information about negative findings.	68
Instructions for repeating computational experiments described in the article.	63
Precise functions were given, with settings.	41
Salient details of the test environment, including hardware, system software, and number of processors used.	13

Stodden et al. 2018 PNAS

Goal: Repeatability/Reproducibility

metadata + data + code + results + contact

Goal: Repeatability/Reproducibility

BestPractices(metadata + data + code + results + contact)

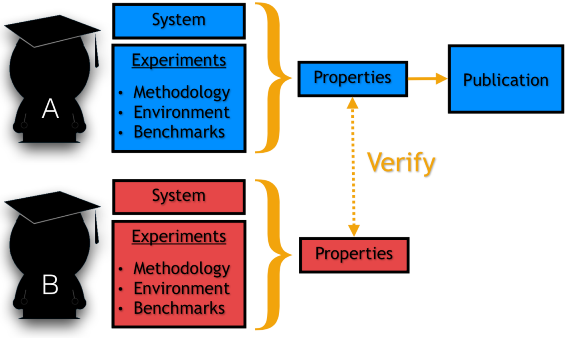
Goal: Repeatability/Reproducibility

BestPractices(metadata * data * code * results * contact)

Opportunity: Benefaction not just reproducibility

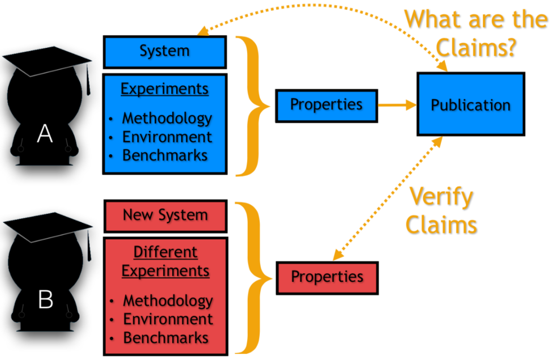
$$Synthesis = f(benefaction)$$

Opportunity: Benefaction not just reproducibility



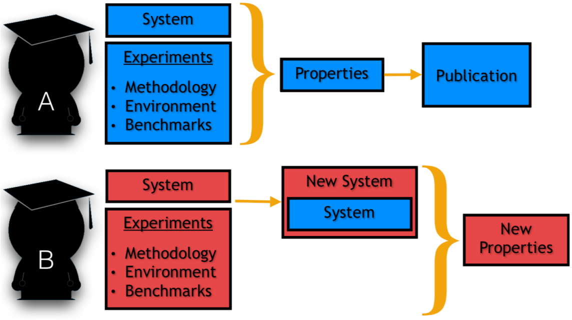
Colberg et al. 2015 Comm ACM

Opportunity: Benefaction not just reproducibility



Colberg et al. 2015 Comm ACM

Opportunity: Benefaction not just reproducibility



Colberg et al. 2015 Comm ACM

Tools: Research Pipeline

Thought	Data Collection	Data Processing	Analysis	Reporting
Meta-Data	*Meta-Data* + *Provenance*	*Provenance* + *Versioning*	*Versioning* + *Provenance*	*Lit Prog* + *Versioning*

Reality: Common Ground

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: Common Ground

Figure 1

Reality: Common Ground

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

Reality: Common Ground

- *Most scientists don't want to produce software, they want to do science.*
- *Let's automate as much of the process as we can to lower activation energy, decrease error rates and increase sharing.*

Tools: 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 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

Figure 2: IEEE: Computing in Science & Engineering 2018

Tools: Encapsulator

Goal: Simplify computational reproducibility

1. Create a data “capsule” with code, data and environment

Tools: Encapsulator

Goal: Simplify computational reproducibility

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

Tools: Encapsulator

Goal: Simplify computational reproducibility

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

Tools: 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

What is data provenance?

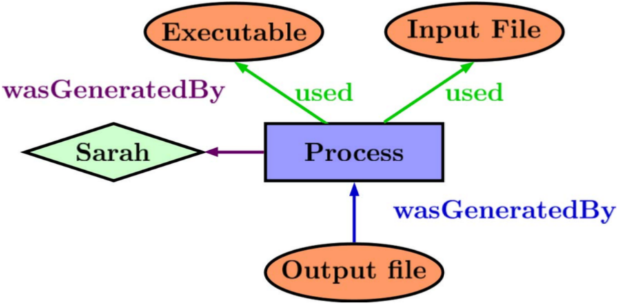


Figure 3

encapsulator(A Kit of Parts): Capsule creation

- Virtual Machine (encapsulator)
- Docker (containR)
- Literate computing notebook (Jupyter)
- Compressed (Reprozip)
- Capsule database (Code Ocean)

Conclusion: The next great challenge is synthesis

** Software should not limit science **

Conclusion: The next great challenge is synthesis

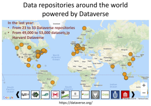


Figure 4

Questions and Discussion:

Possible discussion topics:

Figure 5

1. What checks are in place to verify and link dataverses?
2. Can provenance production become a part of the checking system?
3. What are the pros and cons of automated checking/verification and/or cleaning/encapsulation of dataverses?
4. I'm focused on R's wild-wild-west, but how does this translate to other languages?

Contact Info:

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

Github: MKLau

Tools: Overview

	Dataverse	Code Ocean	Zenodo	Figshare	Dryad	PANGAEA	GitHub & Bitbucket	Supplementary Material
Meta Data	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Data Hosting	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Code Hosting	Yes	Yes	Yes	No	No	No	Yes	Yes
Versioning	No?	No?	Yes	No	No	No	Yes	No
Capsules	No	Yes	No	No	No	No	No	No
Assigns DOI	Yes	Yes	Yes	Yes	Yes	Yes	No	No
License	Flexible	Flexible	Flexible	MIT	CC0	CC-BY	Flexible	None
Cost	None	Possible	None	None	Possible	None	None	None

Adapted from Mislan, Heer & White 2016 Trends in Ecol Evol