Reproducibility

Instituto de computação uff

Vanessa Braganholo

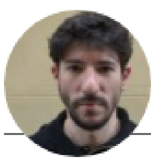
REPRODUCIBILITY:
WHY DOES IT MATTER?

EL PAÍS — INTERNACIONAL

PESQUISAS CIENTÍFICAS ›

Ciência vive uma epidemia de estudos inúteis

Cientistas dos EUA, Reino Unido e Holanda denunciam que a pesquisa está perdendo parte de sua credibilidade



NUÑO DOMÍNGUEZ

19 JAN 2017 - 16:09 BRST

Há séculos, não bastava a Newton e Galileu realizarem descobrimentos capazes de mudar a história. Deveriam também repetir suas experiências diante de todos os seus colegas, e esses, por sua vez, as repetiam por sua conta antes de ficarem completamente convencidos. Esse princípio de reprodutibilidade foi fundamental para o avanço da ciência desde então. Na atualidade, essa garantia essencial está se perdendo, e coloca em dúvida a validade de muitos estudos em quase todas as disciplinas.

nature

Explore content | About the journal | Publish with us | Subscribe

nature > news > article

NEWS | 25 April 2025

Huge reproducibility project fails to validate dozens of biomedical studies

Unique reproducibility effort in Brazil focuses on common methods rather than a single field — and prompts call for reform.

By Rodrigo de Oliveira Andrade

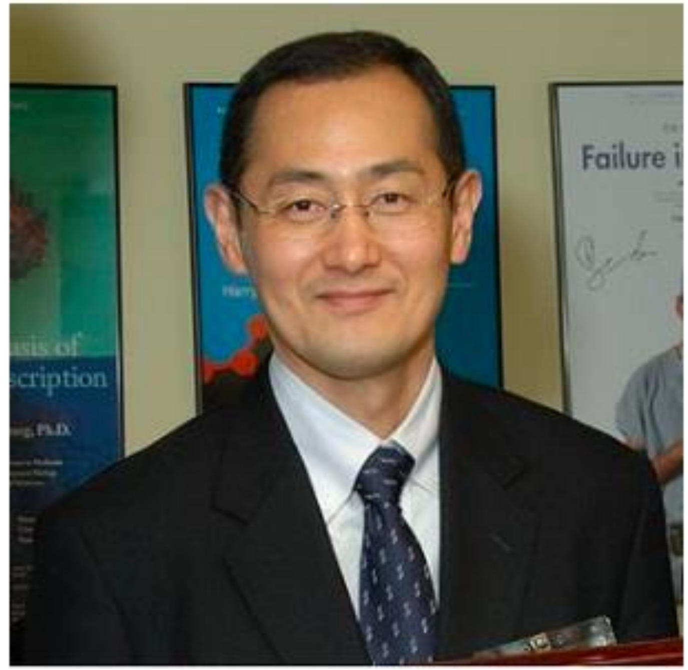
Fonte: https://www.nature.com/articles/d41586-025-01266-x

Instituto de Computação

Nature Methods has retracted a 2017 paper suggesting a common gene editing technique may cause widespread collateral damage to the genome.

The notice has a long backstory: After the paper was published, it immediately drew an outcry from critics (including representatives from companies who sell the tool, whose stock fell after publication). Some critics argued that the authors, led by Vinit B. Mahajan at Stanford University, hadn't employed sufficient controls, so they couldn't be sure that the observed mutations stemmed from the tool, rather than normal background variation between mice. Only months after the paper appeared, the journal issued an expression of concern about the article. In a new preprint posted on BioRxiv on Monday, the authors concede that their critics may be right.

Kyoto University has "punitively dismissed" a researcher found guilty of falsifying nearly all of the figures in a 2017 stem cell paper.



Shinya Yamanaka

According to an announcement Wednesday, the university fired the paper's corresponding author, Kohei Yamamizu, after determining he had fabricated and falsified data in all but one figure in the 2017 *Stem Cell Reports* paper. The findings of the investigation, which were announced in January, found that Yamamizu, who worked at the Center for iPS Cell Research and Application (CiRA), was the only person responsible for the manipulation.

But CiRA's director, Shinya Yamanaka—who shared a Nobel Prize for his work in stem cell biology—has taken responsibility for the incident as well. In an official statement, Yamanaka said he felt "a strong responsibility for not having prevented research misconduct at our institute:"

2) *Novel Mechanism of Inhibition of Dendritic Cells Maturation by Mesenchymal Stem Cells via Interleukin-10 and the JAK1/STAT3 Signaling Pathway:*

> *Following publication of this article [1], concerns were raised regarding the presented data.*
>
> *In Figure 5, the P-JAK1 and STAT3 Western blot panels are duplicates.*
>
> *Four pairs of panels are duplicated in Figure 7:*
>
> *7A panels CD86 and OX62.*
>
> *7B panels CD86 and OX62.*
>
> *7A panel CD11b/c and 7C panel CD11b/c.*
>
> *7A panel MHC-II and 7B panel CD80.*

(…) continues on the next slide

In view of the concerns regarding the reliability of the results and the absence of the raw data images, the authors and PLOS ONE Editors retract this article. The authors wish to apologize to readers.

Common to theses cases

- They were all peer reviewed papers
- Most of the problems were found by scientists **trying to reproduce the research**
- Comments are usually sent to the editors, or published on the Web
  - PubPeer
  - BioRxiv

PUBPEER
The online Journal club

DOI, PMID, arXiv ID, keyword, author, etc.

LOGIN CREATE ACCOUNT

Home

About PubPeer

The PubPeer Foundation

The **PubPeer** Foundation is a California-registered public-benefit corporation with 501(c)(3) nonprofit status in the United States. The overarching goal of the Foundation is to improve the quality of scientific research by enabling innovative approaches for community interaction. The bylaws of the Foundation establish pubpeer.com as a service run for the benefit of its readers and commenters, who create its content. Our current focus is maintaining and developing the **PubPeer** online platform for post-publication peer review.

The PubPeer database contains all articles. Search results return articles with comments.

Mario Schietroma

To leave the first comment on a specific article, paste a unique identifier such as a **DOI**, **PubMed ID**, or **arXiv ID** into the search bar.

Search publications for: **Mario Schietroma**

ERRATUM

2 months ago

RETRACTED: High-concentration supplemental perioperative oxygen and surgical site infection following elective colorectal surgery for rectal cancer: a prospective, randomized, double-blind, controlled, single-site trial

Mario Schietroma, Emanuela M. Cecilia, Federico Sista, Francesco Carlei, Beatrice Pessia, Gianfranco Amicucci

The American Journal of Surgery (2014) 1 comment

8 months ago

Dexamethasone for the prevention of recurrent laryngeal nerve palsy and other complications after thyroid surgery: a randomized double-blind placebo-controlled trial

Mario Schietroma, Emanuela Marina Cecilia, Francesco Carlei, Federico Sista, Giuseppe De Santis, Laura Lancione, Gianfranco Amicucci

JAMA Otolaryngology–Head & Neck Surgery (2013) 1 comment

Home / Publications

Dexamethasone for the prevention of recurrent laryngeal nerve palsy and other complications after thyroid surgery: a randomized double-blind placebo-controlled trial

JAMA Otolaryngology–Head & Neck Surgery (2013) - 1 Comment

doi: 10.1001/jamaoto.2013.2821 issn: 2168-6181 pubmed: 23681030 issn: 2168-619X

Mario Schietroma, Emanuela Marina Cecilia, Francesco Carlei, Federico Sista, Giuseppe De Santis, Laura Lancione, Gianfranco Amicucci

#1 Polyommatus Arasbarani commented 9 months ago

2017 expression of concern. http://jamanetwork.com/journals/jamaotolaryngology/fullarticle/2645374

report permalink Reply

WHAT IS REPRODUCIBILITY?

What is Reproducibility?

- There is no consensus
- Scientists use slightly different definitions for reproducibility
- We will adopt one that seems to be well accepted


Dagstuhl Seminar 16041 , January 2016
Reproducibility of Data-Oriented Experiments in e-Science
http://www.dagstuhl.de/en/program/calendar/semhp/?semnr=16041
BY NC ND
© SCHLOSS DAGSTUHL – LZI GMBH
licensed under Creative Commons License CC BY-NC-ND

Definition of Reproducible Experiment in Computational Science

- An experiment composed by a sequence of steps ***S*** that has been developed at time ***T***, on environment (hardware and OS) ***E***, and on data ***D*** **is reproducible if** it can be executed with a sequence of steps ***S'*** (different or the same as ***S***) at time ***T' > T***, on environment ***E'*** (different or the same as ***E***), and on data ***D'*** (different or the same as ***D***) with **consistent results** ***(R and R' consistent)***

FREIRE, J.; CHIRIGATI, F. Provenance and the Different Flavors of Computational Reproducibility. **IEEE Data Engineering Bulletin**. V. 41:15-26, 2018.

Definition of Reproducible Experiment in Computational Science

- This definition includes both **exact reproducibility** and **approximate reproducibility**
- **Exact Reproducibility** (a.k.a. **repeatability**): requires reproducing **the exact same result**
  - $S' = S$ and $E' = E$ and $D' = D \Rightarrow R = R'$
- **Approximate Reproducibility**: involves producing **similar results** as the original ones
  - $S' \neq S$ or $E' \neq E$ or $D' \neq D \Rightarrow R \sim R'$

FREIRE, J.; CHIRIGATI, F. Provenance and the Different Flavors of Computational Reproducibility. **IEEE Data Engineering Bulletin**. V. 41:15-26, 2018.

Reproduce x Replicate

- **Reproduce**: to execute **the exact same experiment** (**same code, same data**) in a different environment

- **Replicate**: **independent investigators** address a scientific hypothesis and build up evidence for or against it (**different code, different data**)

PENG, R. Reproducible Research in Computational Science. **Science**. V. 443:1226-1227, 2011.

Replication: not easy!

- Depending on the type of the experiment, and the resources it requires, **replication may be nearly impossible**
  - May require lots of computing power
  - May require access to big telescopes
  - May require access to a particle accelerator
  - May require decades of following up subjects (e.g. drug tests)
  - …

PENG, R. Reproducible Research in Computational Science. **Science**. V. 443:1226-1227, 2011.

Reproducibility in Computational Science

"An attainable **minimum standard** for assessing the value of scientific claims, particularly **when full independent replication of a study is not feasible**"

"A result is said to be **reproducible** if another researcher can take the original **code** and **input data**, execute it, and re-obtain the same result."

PENG, R. Reproducible Research in Computational Science. **Science**. V. 443:1226-1227, 2011.

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Reproduce

Repeat

Re-run

Replicate

Reuse

The R* brouhaha

- For a program to contribute to science, it should be rerunnable (R^1), repeatable (R^2), reproducible (R^3), reusable (R^4), and replicable (R^5)

Reproduce

Repeat

Re-run

Replicate

Reuse

GOBLE, C. What is reproducibility? The Rbrouhaha, In:First International Workshop on Reproducible Open Science (Hannover), 2016.
BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

R¹ - Rerunnable

- A **rerunnable** code is one that can be run again when needed
  - It becomes intrinsically difficult as code ages
  - It implies we need knowledge of the original environment (***E***), access to the code (***S***) and data (***D***)
  - ***S'= S*** and ***E' ~ E*** and ***D'= D***
  - Note that nothing is said about the **result**

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Example: Random Walk (R^0)

LISTING 0: Random walk (R^0) raw code, archive

```
import random

x = 0
for i in xrange(10):
    step = random.choice([-1,+1])
    x += step
    print x,
```

Output

```
-1, 0, -1, 0, -1, 0, -1, 0, 1, 2
# with the steps being -1,+1,-1,+1,-1,+1,-1,+1,+1,+1
```

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Example: Random Walk (R^0)

LISTING 0: Random walk (R^0) raw code, archive

```
import random

x = 0
for i in xrange(10):
    step = random.choice([-1,+1])
    x += step
    print x,
```

*Environment info is **unknown**.*
Does it work on any Python version?

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Example: Random Walk (R^0)

LISTING 0: Random walk (R^0) raw code, archive

```
import random

x = 0
for i in xrange(10):
    step = random.choice([-1,+1])
    x += step
    print x,
```

xrange and print are deprecated in Python 3

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Example: Rerunnable Random Walk (R^1)

LISTING 1: Re-runnable random walk (R^1) raw code, archive

```
# Tested with Python 3
import random

x = 0
walk = []
for i in range(10):
    step = random.choice([-1,+1])
    x += step
    walk.append(x)

print(walk)
```

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Example: Rerunnable Random Walk (R^1)

LISTING 1: Re-runnable random walk (R^1) raw code, archive

```
# Tested with Python 3
import random

x = 0
walk = []
for i in range(10):
    step = random.choice([-1,+1])
    x += step
    walk.append(x)

print(walk)
```

Environment info

Scientist is responsible for keeping this info

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Repeatable (R^2)

- A **repeatable** code is one that can be rerun and that produces the same result on **successive runs**
  - Program needs to be **deterministic**
  - Control the initialization of pseudo-random number generators
  - **Results of the successive executions** need to be available (so it is possible to compare with current results)
  - ***S'= S*** and ***E'= E*** and ***D'= D*** *and* ***R = R'***

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Repeatable (R^2)

- Nothing is said about the results of **past executions** (made a long time ago)
- All we care here is that we execute the code several times, sucessively, and the same result is produced every time.

Example: Repeatable Random Walk (R^2)

LISTING 2: Re-runnable, repeatable random walk (R^2) raw code, archive

```
# Tested with Python 3
import random

random.seed(1) # RNG initialization

x = 0
walk = []
for i in range(10):
    step = random.choice([-1,+1])
    x += step
    walk.append(x)

print(walk)
# Saving output to disk
with open('results-R2.txt', 'w') as fd:
    fd.write(str(walk))
```

Example: Repeatable Random Walk (R^2)

LISTING 2: Re-runnable, repeatable random walk (R^2) raw code, archive

```
# Tested with Python 3
import random

random.seed(1) # RNG initialization

x = 0
walk = []
for i in range(10):
    step = random.choice([-1,+1])
    x += step
    walk.append(x)

print(walk)
# Saving output to disk
with open('results-R2.txt', 'w') as fd:
    fd.write(str(walk))
```

Random seed initialization

Save output to allow comparing different runs

(again scientist is responsible for recording provenance)

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Initialization of Random Seeds

- Verifying that the qualitative aspects of the results and the conclusions that are made are **not tied to a specific initialization of the pseudo-random generator** is an integral part of any scientific undertaking in computational Science
- This is usually done by **repeating the simulations multiple times with different seeds**

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Reproducible (R^3)

- A result is said to be **reproducible** if **another researcher** can take the original code and input data, execute it, and re-obtain the **same (compatible) result**
- A Repeatable program **will not necessarily** produce the same results all the time over **different execution environments**

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Example: Repeatable Random Walk (R^2)

LISTING 2: Re-runnable, repeatable random walk (R^2) raw code, archive

```
# Tested with Python 3
import random

random.seed(1) # RNG init

x = 0
walk = []
for i in range(10):
    step = random.choice
    x += step
    walk.append(x)

print(walk)
# Saving output to disk
with open('results-R2.txt', 'w') as fd:
    fd.write(str(walk))
```

*Due to a **change** that occurred in the **pseudo-random number generator** between **Python 3.2** and **Python 3.3**, executing this code in Python 3.3 **will NOT** generate the same results when compared to the Python 3.2 execution*

Repeatable Random Walk Example is not reproducible

- Executed with Python 2.7–3.2, the code will produce the sequence

  −1, 0, 1, 0, −1, −2, −1, 0, −1, −2

- But with Python 3.3–3.6, it will produce

  −1, −2, −1, −2, −1, 0, 1, 2, 1, 0

- With future versions of the language, it may change still

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Reproducibility (R^3)

- Executability (R^1) and determinism (R^2) are necessary but not sufficient for reproducibility
- The **exact execution environment** used to produce the results must also be specified, so people can try to use a **similar environment** when trying to reproduce the experiment
- ***S'= S*** and ***E' ~ E*** and ***D'= D and R ~ R'***

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Reproducibility (R^3)

- Having environment info is not enough
  - In our example, **should the code change** after the production of the results, someone provided with the last version of the code **will not be able to know which seed was used** to produce the results
  - **Result files should come alongside their context**, i.e., an exhaustive list of the parameters used as well as a precise description of the execution environment
  - The **code itself is part of that context**: the version of the code must be recorded

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Example: Reproducible Random Walk (R^3)

LISTING 3: Re-runnable, repeatable, reproducible random walk (R^3)

raw code, archive

```
# Copyright (c) 2017 N.P. Rougier and F.C.Y. Benureau
# Release under the BSD 2-clause license
# Tested with 64-bit CPython 3.6.2 / macOS 10.12.6
import sys, subprocess, datetime, random

def compute_walk():
    x = 0
    walk = []
    for i in range(10):
        if random.uniform(-1, +1) > 0:
            x += 1
        else:
            x -= 1
        walk.append(x)
    return walk
```

```
# If repository is dirty, don't run anything
if subprocess.call(("git", "diff-index",
                    "--quiet", "HEAD")):
    print("Repository is dirty, please commit first")
    sys.exit(1)

# Get git hash if any
hash_cmd = ("git", "rev-parse", "HEAD")
revision = subprocess.check_output(hash_cmd)

# Unit test
random.seed(42)
assert compute_walk() == [1,0,-1,-2,-1,0,1,0,-1,-2]

# Random walk for 10 steps
seed = 1
random.seed(seed)
walk = compute_walk()
```

Use git to keep track of code versions

```python
# If repository is dirty, don't run anything
if subprocess.call(("git", "diff-index",
                    "--quiet", "HEAD")):
    print("Repository is dirty, please commit first")
    sys.exit(1)

# Get git hash if any
hash_cmd = ("git", "rev-parse", "HEAD")
revision = subprocess.check_output(hash_cmd)

# Unit test
random.seed(42)
assert compute_walk() == [1,0,-1,-2,-1,0,1,0,-1,-2]

# Random walk for 10 steps
seed = 1
random.seed(seed)
walk = compute_walk()
```

Test for reproducibility

```
# Display & save results
print(walk)
results = {
    "data"      : walk,
    "seed"      : seed,
    "timestamp": str(datetime.datetime.utcnow()),
    "revision" : revision,
    "system"   : sys.version}
with open("results-R3.txt", "w") as fd:
    fd.write(str(results))
```

Record environment with output data

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Quick Recap

- **Reproducibility** implies **re-runnability** and **repeatability** and **availability**, yet imposes additional conditions
- Dependencies and platforms must be described as precisely and as specifically as possible
- Parameters values, the version of the code, and inputs should accompany the result files

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Reusability (R^4)

- Making your program **reusable** means it can be easily used, and **modified**, by you and other people, inside and outside your lab
- The easier it is to use your code, the lower the threshold is for other to study, modify and extend it
  - This implies it should be **well documented**!

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Reusability (R^4)

- Scientists constantly face the constraint of time
  - if a model is **available**, **documented**, and can be installed, run, and **understood** all in a few hours, it will be preferred over another that would require weeks to reach the same stage
- A **reproducible** and **reusable** code offers a platform both verifiable and easy-to-use, fostering the development of derivative works by other researchers on solid foundations
- Those derivative works **contribute to the impact of your original contribution (citations!!)**

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Reusability (R^4)

- Reusability is not as indispensable a requirement as re-runnability, repeatability, and reproducibility
- But it can contribute to strengthen reproducibility and re-runnability over the long-term

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Example: Reusable Random Walk (R^4)

LISTING 4: Re-runnable, repeatable, reproducible, reusable random walk (R^4)
raw code, archive

```
# Copyright (c) 2017 N.P. Rougier and F.C.Y. Benureau
# Release under the BSD 2-clause license
# Tested with 64-bit CPython 3.6.2 / macOS 10.12.6
import sys, subprocess, datetime, random

def compute_walk(count, x0=0, step=1, seed=0):
    """Random walk
       count: number of steps
       x0   : initial position (default 0)
       step : step size (default 1)
       seed : seed for the initialization of the
          random generator (default 0)
    """
```

```python
random.seed(seed)
x = x0
walk = []
for i in range(count):
    if random.uniform(-1, +1) > 0:
        x += 1
    else:
        x -= 1
    walk.append(x)
return walk
```

```python
def compute_results(count, x0=0, step=1, seed=0):
    """Compute a walk and return it with context"""
    # If repository is dirty, don't do anything
    if subprocess.call(("git", "diff-index",
                        "--quiet", "HEAD")):
        print("Repository is dirty, please commit")
        sys.exit(1)

    # Get git hash if any
    hash_cmd = ("git", "rev-parse", "HEAD")
    revision = subprocess.check_output(hash_cmd)

    # Compute results
    walk = compute_walk(count=count, x0=x0,
                        step=step, seed=seed)
```

```
return {
    "data"        : walk,
    "parameters": {"count": count, "x0": x0,
                   "step": step, "seed": seed},
    "timestamp" : str(datetime.datetime.utcnow()),
    "revision"  : revision,
    "system"    : sys.version}
```

```python
if __name__ == "__main__":
    # Unit test checking reproducibility
    # (will fail with Python<=3.2)
    assert (compute_walk(10, 0, 1, 42) ==
                 [1,0,-1,-2,-1,0,1,0,-1,-2])

    # Simulation parameters
    count, x0, seed = 10, 0, 1
    results = compute_results(count, x0=x0, seed=seed)

    # Save & display results
    with open("results-R4.txt", "w") as fd:
        fd.write(str(results))
    print(results["data"])
```

Tips for Producing Reusable Code

- **Avoid** hardcoded or magic numbers
- **Magic numbers** are those present directly in the source code (no name, no semantics)
- **Hardcoded values** are variables that cannot be changed through an argument or a parameter configuration file
- In the R^3 Random Walk example, the seed is hardcoded, and the number of steps is a magic number

Tips for Producing Reusable Code

- Code behavior **should not** be changed by commenting/uncommenting code
- Instead, it should be **explicitly set** through parameters that are accessible to the end user
- This improves reproducibility in two ways
  - it allows those conditions to be recorded as parameters in the result files, and
  - it allows to define separate scripts to run or configuration files to load to produce each of the figures of the published paper

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Replicability (R^5)

"the replication of important findings by **multiple independent investigators** is fundamental to the accumulation of scientific evidence"

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Replicability (R^5)

- **Replicability** is the implicit assumption that an article that does not provide the source code makes: that the description it provides of the algorithms is sufficiently precise and complete to re-obtain the results it presents
- Replicating implies **writing a new code** matching the conceptual description of the article, in order to obtain the same **(compatible) results**
- $S' \neq S$ ***and*** $(E' \neq E$ or $D' \neq D) \Rightarrow R \sim R'$

Replicability (R^5)

- Replication affords robustness to the results
  - should the original code contain an error, a different codebase creates the possibility that this error will not be repeated

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Replicability (R^5)

- Every paper is a mistake or a forgotten parameter away from irreplicability
- Replication efforts use the paper first, and then the reproducible code that comes along with it whenever the paper falls short of being precise enough

BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in **Neuroinformatics**. V.11, article 69, 2018.

Summary

Re-run (R^1)

- ***S'= S*** and
- ***E'~ E*** and
- ***D'= D***

Repeat (R^2)

- ***S'= S*** and
- ***E'= E*** and
- ***D'= D** and*
- ***R = R'***

Reproduce (R^3)

- ***S'= S*** and
- ***E' ~ E*** and
- ***D'= D and***
- ***R ~ R'***

Reuse (R^4)

- Document
- Avoid hardcoded or magic numbers
- Use parameters

Replicate (R^5)

- ***S'≠ S and***
- ***(E' ≠ E** or **D' ≠ D) and***
- ***R ~ R'***

Summary

Re-run (R^1)
- ***S'= S*** and
- ***E'~ E*** and
- ***D'= D***

Repeat (R^2)
- ***S'= S*** and
- ***E'~ E*** and
- ***D'= D** and*
- ***R = R'***

Reproduce (R^3)
- ***S'= S*** and
- ***E' ~ E*** and
- ***D'= D and***
- ***R ~ R'***

Reuse (R^4)
- Document
- Avoid hardcoded or magic numbers
- Use parameters

Replicate (R^5)
- ***S'≠ S and***
- ***(E' ≠ E** or **D' ≠ D) and***
- ***R ~ R'***

Minimum Scientific Standard

But we are not there yet…

- Reproducibility **is still not the norm** for computational experiments
- Scientists argue that it is **time-consuming** to create reproducible experiments
- **Usability** is an important requirement for a broader adoption of reproducibility
- "An independent user should be able to reproduce the results with a single mouse click"

FREIRE, J.; CHIRIGATI, F. Provenance and the Different Flavors of Computational Reproducibility. **IEEE Data Engineering Bulletin**. V. 41:15-26, 2018.

Making Reproducibility Easier

- Scientist should focus on research rather than making their code capture its own provenance
- There are several tools to easy reproducibility
  - noWorkflow, Sumatra, Reprozip, etc.
- Improvements still needed to make them “one mouse click away from reproducibility”

CAUSES OF NON-REPRODUCIBLE RESULTS

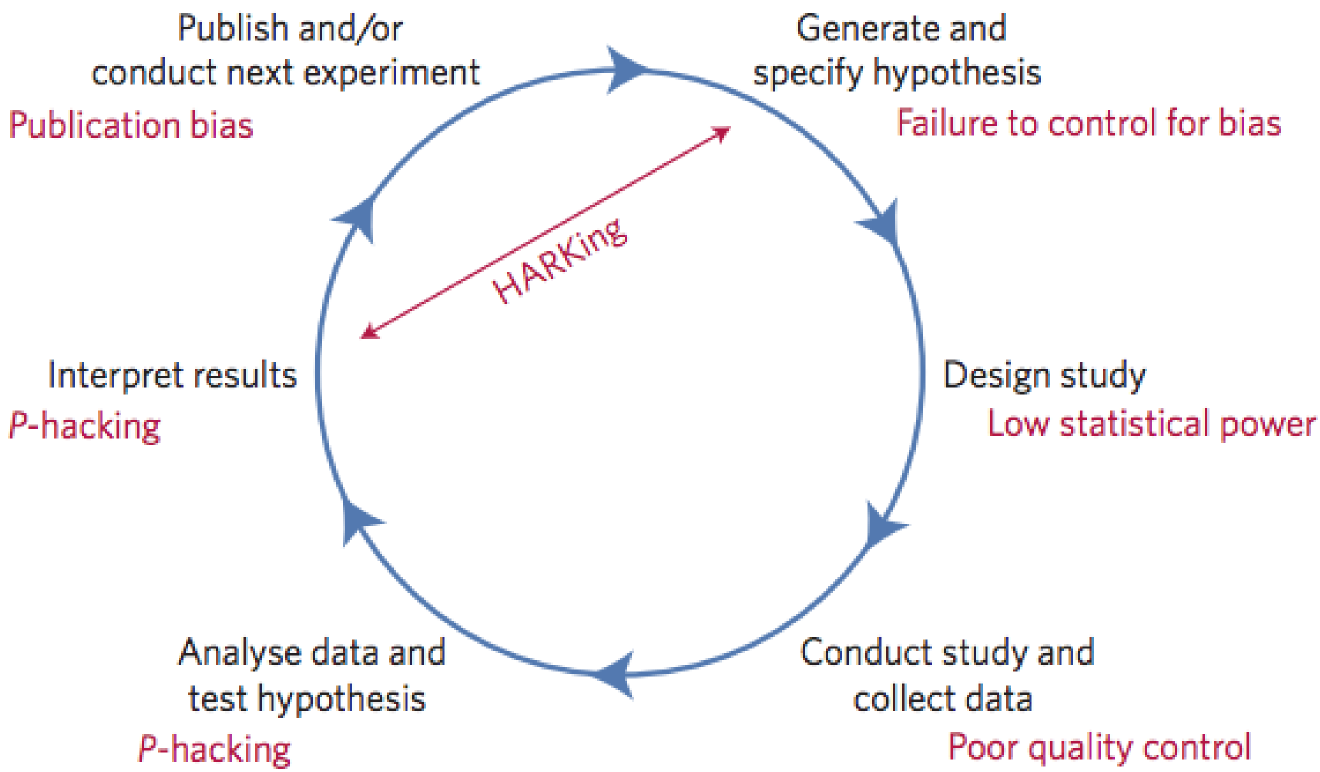


Figure 1 | Threats to reproducible science. An idealized version of the hypothetico-deductive model of the scientific method is shown. Various potential threats to this model exist (indicated in red), including lack of replication[5], hypothesizing after the results are known (HARKing)[7], poor study design, low statistical power[2], analytical flexibility[51], *P*-hacking[4], publication bias[3] and lack of data sharing[6]. Together these will serve to undermine the robustness of published research, and may also impact on the ability of science to self-correct.

p-hacking

While collecting and analyzing data, researchers have many decisions to make, including whether to collect more data, which outliers to exclude, which measure(s) to analyze, which covariates to use, and so on. If these decisions are not made in advance but rather are made as the data are being analyzed, then researchers may make them in ways that self-servingly increase their odds of publishing (Kunda, 1990). Thus, rather than placing entire studies in the file-drawer, researchers may file merely the subsets of analyses that produce nonsignificant results. We refer to such behavior as *p-hacking*.[1]

SIMONSOHN, U., NELSON, L., SIMMONS, J. P-Curve: A Key to the File-Drawer. Journal of Experimental Psychology: General. V. 143(2):534-547, 2014.

Table 1 | A manifesto for reproducible science.

Theme	Proposal	Examples of initiatives/potential solutions (extent of current adoption)	Stakeholder(s)
Methods	Protecting against cognitive biases	All of the initiatives listed below (* to ****) Blinding (**)	J, F
	Improving methodological training	Rigorous training in statistics and research methods for future researchers (*) Rigorous continuing education in statistics and methods for researchers (*)	I, F
	Independent methodological support	Involvement of methodologists in research (**) Independent oversight (*)	F
	Collaboration and team science	Multi-site studies/distributed data collection (*) Team-science consortia (*)	I, F
Reporting and dissemination	Promoting study pre-registration	Registered Reports (*) Open Science Framework (*)	J, F
	Improving the quality of reporting	Use of reporting checklists (**) Protocol checklists (*)	J
	Protecting against conflicts of interest	Disclosure of conflicts of interest (***) Exclusion/containment of financial and non-financial conflicts of interest (*)	J
Reproducibility	Encouraging transparency and open science	Open data, materials, software and so on (* to **) Pre-registration (**** for clinical trials, * for other studies)	J, F, R
Evaluation	Diversifying peer review	Preprints (* in biomedical/behavioural sciences, **** in physical sciences) Pre- and post-publication peer review, for example, Publons, PubMed Commons (*)	J
Incentives	Rewarding open and reproducible practices	Badges (*) Registered Reports (*) Transparency and Openness Promotion guidelines (*) Funding replication studies (*) Open science practices in hiring and promotion (*)	J, I, F

Estimated extent of current adoption: *, <5%; **, 5–30%; ***, 30–60%; ****, >60%. Abbreviations for key stakeholders: J, journals/publishers; F, funders; I, institutions; R, regulators.

MUNAFÒ, M. et al. A manifesto for reproducible science. Nature Human Behaviour. V. 1: article 21, 2017.

Table 1 | A manifesto for reproducible science.

Theme	Proposal	Examples of initiatives/potential solutions (extent of current adoption)	Stakeholder(s)
Methods	Protecting against cognitive biases	All of the initiatives listed below (* to ****) Blinding (**)	J, F
	Improving methodological training	Rigorous training in statistics and research methods for future researchers (*) Rigorous continuing education in statistics and methods for researchers (*)	I, F
	Independent methodological support	Involvement of methodologists in research (**) Independent oversight (*)	F
	Collaboration and team science	Multi-site studies/distributed data collection (*) Team-science consortia (*)	I, F
Reporting and dissemination	Promoting study pre-registration	Registered Reports (*) Open Science Framework (*)	J, F
	Improving the quality of reporting	Use of reporting checklists (**) Protocol checklists (*)	J
	Protecting against conflicts of interest	Disclosure of conflicts of interest (***) Exclusion/containment of financial and non-financial conflicts of interest (*)	J
Reproducibility	Encouraging transparency and open science	Open data, materials, software and so on (* to **) Pre-registration (**** for clinical trials, * for other studies)	J, F, R
Evaluation	Diversifying peer review	Preprints (* in biomedical/behavioural sciences, **** in physical sciences) Pre- and post-publication peer review, for example, Publons, PubMed Commons (*)	J
Incentives	Rewarding open and reproducible practices	Badges (*) Registered Reports (*) Transparency and Openness Promotion guidelines (*) Funding replication studies (*) Open science practices in hiring and promotion (*)	J, I, F

Estimated extent of current adoption: *, <5%; **, 5–30%; ***, 30–60%; ****, >60%. Abbreviations for key stakeholders: J, journals/publishers; F, funders; I, institutions; R, regulators.

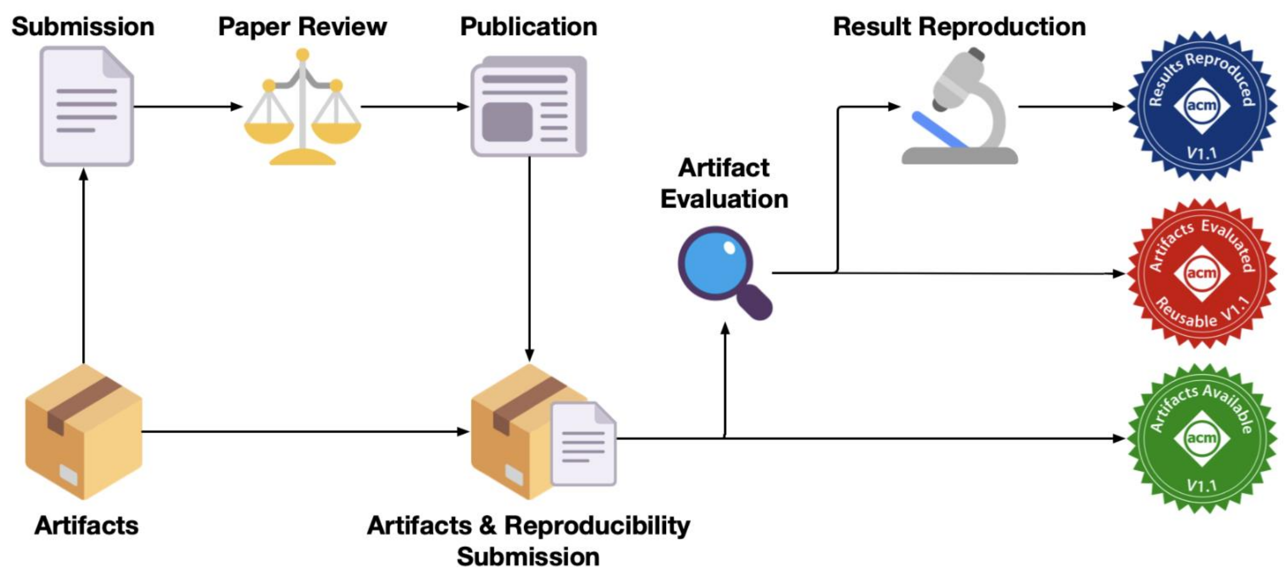
MUNAFÒ, M. et al. A manifesto for reproducible science. Nature Human Behaviour. V. 1: article 21, 2017.

Incentives

- ACM SIGMOD Most Reproducible Paper Award
- ACM SIGMOD Availability and Reproducibility

How it works?

On the high-level, the workflow of the SIGMOD Availability and Reproducibility process is shown below. For more details keep scrolling to the Process and Guidelines.



https://reproducibility.sigmod.org/reports.html

Badges and Reproducibility Reports

The reviewers submit their reports, which are published in this page. Following is the list of papers that passed the reproducibility test, the functionality test, and/or have made their artifacts available, along with the reproducibility reports (when applicable) and the badges awarded. The reproducibility reports are maintained here starting from reproduced papers of SIGMOD 2020.

To access all reproduced papers please click here.

ACM SIGMOD 2023 Badges and Reproducibility Results

Paper Title	Report	Badges Awarded
Polaris: Enabling Transaction Priority in Optimistic Concurrency Control	PDF	Results Reproduced V1.1, Artifacts Available V1.1, Artifacts Evaluated Reusable V1.1
SafeBound: A Practical System for Generating Cardinality Bounds	PDF	Results Reproduced V1.1, Artifacts Available V1.1, Artifacts Evaluated Reusable V1.1
Efficient Star-based Truss Maintenance on Dynamic Graphs	PDF	Results Reproduced V1.1, Artifacts Available V1.1, Artifacts Evaluated Reusable V1.1

Incentives

- ICSE "Artifacts Evaluated Reusable"

14:00 - 14:20 ☆ Big Bangs and Small Pops: On Critical Cyclomatic Complexity and Developer Integration Behavior
Daniel Ståhl Ericsson AB, Antonio Martini University of Oslo, Norway, Torvald Mårtensson Saab AB
INDUSTRY PROGRAM SEIP

14:20 - 14:40 ☆ Predictive Test Selection
INDUSTRY PROGRAM SEIP
Mateusz Machalica Facebook, Inc., Alex Samylkin Facebook, Inc., Meredith Porth Facebook, Inc., Satish Chandra Facebook

14:40 - 15:00 ☆ Assessing Transition-based Test Selection Algorithms at Google
INDUSTRY PROGRAM SEIP
Claire Leong Google / UNSW, Abhayendra Singh Google, Inc, Mike Papadakis University of Luxembourg, Yves Le Traon University of Luxembourg, John Micco Netflix

15:00 - 15:20 ☆ Automated Reporting of Anti-Patterns and Decay in Continuous Integration
Carmine Vassallo University of Zurich, Sebastian Proksch University of Zurich, Harald Gall University of Zurich, Massimiliano Di Penta University of Sannio
TECHNICAL TRACK
Pre-print

Incentives

- Reproducibility Section of Information Systems Journal
  - https://www.elsevier.com/journals/information-systems/0306-4379/guide-for-authors

THE DATA IS AS IMPORTANT AS THE CODE!

For certain types of important digital objects, there are well-curated, deeply-integrated, special-purpose repositories such as Genbank3, Worldwide Protein Data Bank (wwPDB4), and UniProt5 in the life sciences; Space Physics Data Facility (SPDF; http://spdf.gsfc.nasa.gov/) and Set of Identifications, Measurements and Bibliography for Astronomical Data (SIMBAD6) in the space sciences. These foundational and critical core resources are continuously curating and capturing highvalue reference datasets and fine-tuning them to enhance scholarly output, provide support for both human and mechanical users, and provide extensive tooling to access their content in rich, dynamic ways. However, not all datasets or even data types can be captured by, or submitted to, these repositories. Many important datasets emerging from traditional, low-throughput bench science don't fit in the data models of these special-purpose repositories, yet these datasets are no less important with respect to integrative research, reproducibility, and reuse in general.

Apparently in response to this, we see the emergence of numerous general-purpose data repositories, at scales ranging from institutional (for example, a single university), to open globally-scoped repositories such as Dataverse7, FigShare (http://figshare.com), Dryad8, Mendeley Data (https://data.mendeley.com/), Zenodo (http://zenodo.org/), DataHub (http://datahub.io), DANS (http://www.dans.knaw.nl/), and EUDat9. Such repositories accept a wide range of data types in a wide variety of formats, generally do not attempt to integrate or harmonize the deposited data, and place few restrictions (or requirements) on the descriptors of the data deposition. The resulting data ecosystem, therefore, appears to be moving away from centralization, is becoming more diverse, and less integrated, thereby exacerbating the discovery and re-usability problem for both human and computational stakeholders.

A specific example of these obstacles could be imagined in the domain of gene regulation and expression analysis. Suppose a researcher has generated a dataset of differentially-selected polyadenylation sites in a non-model pathogenic organism grown under a variety of environmental conditions that stimulate its pathogenic state. The researcher is interested in comparing the alternatively-polyadenylated genes in this local dataset, to other examples of alternative-polyadenylation, and the expression levels of these genes—both in this organism and related model organisms—during the infection process. Given that there is no special-purpose archive for differential polyadenylation data, and no model organism database for this pathogen, where does the researcher begin?

We will consider the current approach to this problem from a variety of data discovery and integration perspectives. If the desired datasets existed, where might they have been published, and how would one begin to search for them, using what search tools? The desired search would need to filter based on specific species, specific tissues, specific types of data (Poly-A, microarray, NGS), specific conditions (infection), and specific genes—is that information ('metadata') captured by the repositories, and if so, what formats is it in, is it searchable, and how? Once the data is discovered, can it be downloaded? In what format(s)? Can that format be easily integrated with private in-house data (the local dataset of alternative polyadenylation sites) as well as other data publications from third-parties and with the community's core gene/protein data repositories? Can this integration be done automatically to save time and avoid copy/paste errors? Does the researcher have permission to use the data from these third-party researchers, under what license conditions, and who should be cited if a data-point is re-used?

SCIENTIFIC DATA

Amended: Addendum

OPEN

SUBJECT CATEGORIES
» Research data
» Publication characteristics

Comment: The FAIR Guiding Principles for scientific data management and stewardship

Mark D. Wilkinson *et al.*#

Received: 10 December 2015
Accepted: 12 February 2016
Published: 15 March 2016

There is an urgent need to improve the infrastructure supporting the reuse of scholarly data. A diverse set of stakeholders—representing academia, industry, funding agencies, and scholarly publishers—have come together to design and jointly endorse a concise and measureable set of principles that we refer to as the FAIR Data Principles. The intent is that these may act as a guideline for those wishing to enhance the reusability of their data holdings. Distinct from peer initiatives that focus on the human scholar, the FAIR Principles put specific emphasis on enhancing the ability of machines to automatically find and use the data, in addition to supporting its reuse by individuals. This Comment is the first formal publication of the FAIR Principles, and includes the rationale behind them, and some exemplar implementations in the community.

https://www.nature.com/articles/sdata201618

FAIR

Findable

Accessible

Interoperable

Reusable

Box 2 | The FAIR Guiding Principles

To be Findable:
F1. (meta)data are assigned a globally unique and persistent identifier
F2. data are described with rich metadata (defined by R1 below)
F3. metadata clearly and explicitly include the identifier of the data it describes
F4. (meta)data are registered or indexed in a searchable resource

To be Accessible:
A1. (meta)data are retrievable by their identifier using a standardized communications protocol
A1.1 the protocol is open, free, and universally implementable
A1.2 the protocol allows for an authentication and authorization procedure, where necessary
A2. metadata are accessible, even when the data are no longer available

To be Interoperable:
I1. (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
I2. (meta)data use vocabularies that follow FAIR principles
I3. (meta)data include qualified references to other (meta)data

To be Reusable:
R1. meta(data) are richly described with a plurality of accurate and relevant attributes
R1.1. (meta)data are released with a clear and accessible data usage license
R1.2. (meta)data are associated with detailed provenance
R1.3. (meta)data meet domain-relevant community standards

Provenance of these slides

- BENUREAU, F., ROUGIER, N. Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. Frontiers in Neuroinformatics. V.11, article 69, 2018.
- FREIRE, J.; CHIRIGATI, F. Provenance and the Different Flavors of Computational Reproducibility. IEEE Data Engineering Bulletin. V. 41:15-26, 2018.
- GOBLE, C. What is reproducibility? The Rbrouhaha, In:First International Workshop on Reproducible Open Science (Hannover), 2016.
- MUNAFÒ, M. et al. A manifesto for reproducible science. Nature Human Behaviour. V. 1: article 21, 2017.
- SIMONSOHN, U., NELSON, L., SIMMONS, J. P-Curve: A Key to the File-Drawer. Journal of Experimental Psychology: General. V. 143(2):534-547, 2014.
- WILKINSON, M. et al. The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data. V. 3, Article 160018, 2016.