Reproducibility

Vanessa Braganholo
REPRODUCIBILITY: WHY DOES IT MATTER?
Ciência vive uma epidemia de estudos inúteis

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

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 reproduzibilidade 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 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.

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
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.

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

**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)*

**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)*

**2 months ago**
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

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

#1 Polyommatus Arasbarani commented 9 months ago

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
Reproducibility of Data-Oriented Experiments in e-Science

http://www.dagstuhl.de/en/program/calendar/semhp/?semnr=16041
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)

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
  \[- S' = S \text{ and } E' = E \text{ and } D' = D \Rightarrow R = R'\]

- **Approximate Reproducibility**: involves producing similar results as the original ones
  \[- S' \neq S \text{ or } E' \neq E \text{ or } D' \neq D \Rightarrow R \sim R'\]

---

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)

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)
  - ...

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.”


Reproduce

Repeat

Re-run

Replicate

Reuse
The R* brouhaha

• For a program to contribute to science, it should be rerunnable (R₁), repeatable (R₂), reproducible (R₃), reusable (R₄), and replicable (R₅)

GOBLE, C. What is reproducibility? The Rbrouhaha, In:First International Workshop on Reproducible Open Science (Hannover), 2016.
R$^1$ - 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 $E' \sim E$ and $D' = D$
  - Note that nothing is said about the result

Example: Random Walk \((R^0)\)

Listing 0: Random walk \((R^0)\)

```python
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
```

Example: Random Walk ($R^0$)

```
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?

Example: Random Walk ($R^0$)

```python
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.

Example: Rerunnable Random Walk (R¹)

```
# 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)
```

Example: Rerunnable Random Walk ($R^1$)

**Listing 1: Re-runnable random walk ($R^1$)**

```python
# 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)
```

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
  - **Previous results** need to be available (so it is possible to compare with current results)
  - $S' = S$ and $E' \sim E$ and $D' = D$ and $R = R'$

Example: Repeatable Random Walk ($R^2$)

```python
# 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

```python
# 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)
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

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.
- An $R^2$ program **will not necessarily** produce the same results all the time over different execution environments.
- $S' = S$ and $E' = E$ and $D' = D$ and $R \sim R'$

Example: Repeatable Random Walk ($R^2$)

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.

```python
# Tested with Python 3
import random

random.seed(1)  # RNG initialization
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))
```
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

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.

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
Example: Reproducible Random Walk ($\mathbb{R}^3$)

LISTING 3: Re-runnable, repeatable, reproducible random walk ($\mathbb{R}^3$)
raw code, archive

```python
# 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
```
Use git to keep track of code versions
# 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()
# Display & save results

```python
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
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.
- The **data and scripts behind the graphs must be published**.

Reusability (R⁴)

- 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**!

---

Reusability (R⁴)

• 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!!)
Reusability (R⁴)

• 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
Example: Reusable Random Walk ($R^4$)

```
# 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
Replicability ($R^5$)

“the replication of important findings by multiple independent investigators is fundamental to the accumulation of scientific evidence”

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 $(E' \neq E \lor D' \neq D) \Rightarrow R \sim R'$
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

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

Summary

Re-run (R₁)
- \( S' = S \) and
- \( E' \sim E \) and
- \( D' = D \)

Repeat (R²)
- \( S' = S \) and
- \( E' \sim E \) and
- \( D' = D \) and
- \( R = R' \)

Reproduce (R³)
- \( S' = S \) and
- \( E' = E \) and
- \( D' = D \) and
- \( R \sim R' \)

Reuse (R⁴)
- Document
- Avoid hardcoded or magic numbers
- Use parameters

Replicate (R⁵)
- \( S' \neq S \) and
- \((E' \neq E \text{ or } D' \neq D) \) and
- \( R \sim R' \)

Code (local) + Environment + Input Data
- Same (Compatible) Output
- Same Environment
- Publicly available + Documentation
Summary

Re-run (R¹)
- \( S' = S \) and
- \( E' \approx E \)
- \( D' = D \)

Repeat (R²)
- \( S' = S \) and
- \( E' = E \)
- \( D' = D \)
- \( R = R' \)

Reproduce (R³)
- \( S' = S \) and
- \( E' = E \)
- \( D' = D \)
- \( R = R' \)

Reuse (R⁴)
- Document
- Avoid hardcoded or magic numbers
- Use parameters

Replicate (R⁵)
- \( S' \neq S \) and
- \( (E' \neq E \text{ or } D' \neq D) \)
- \( R = R' \)

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”

Making Reproducibility Easier

• Scientist should focus on research rather than making their code capture its own provenance
• There are several tools to ease 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\textsuperscript{5}, hypothesizing after the results are known (HARKing)\textsuperscript{7}, poor study design, low statistical power\textsuperscript{2}, analytical flexibility\textsuperscript{51}, \textit{P}-hacking\textsuperscript{4}, publication bias\textsuperscript{3} and lack of data sharing\textsuperscript{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

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

Estimated extent of current adoption: *, <5%; **, 5-30%; ***. 30-60%; ****, >60%. Abbreviations for key stakeholders: J, journals/publishers; F, funders; I, institutions; R, regulators.
<table>
<thead>
<tr>
<th>Theme</th>
<th>Proposal</th>
<th>Examples of initiatives/potential solutions (extent of current adoption)</th>
<th>Stakeholder(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methods</td>
<td>Protecting against cognitive biases</td>
<td>* All of the initiatives listed below (* to ****)</td>
<td>J, F</td>
</tr>
<tr>
<td></td>
<td></td>
<td>** Blinding (**)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Improving methodological training</td>
<td>* Rigorous training in statistics and research methods for future researchers</td>
<td>I, F</td>
</tr>
<tr>
<td></td>
<td></td>
<td>** Rigorous continuing education in statistics and methods for researchers</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Independent methodological support</td>
<td>* Involvement of methodologists in research</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
<td>** Independent oversight</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Collaboration and team science</td>
<td>* Multi-site studies/distributed data collection</td>
<td>I, F</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Team-science consortia</td>
<td></td>
</tr>
<tr>
<td>Reporting and dissemination</td>
<td>Promoting study pre-registration</td>
<td>* Registered Reports</td>
<td>J, F</td>
</tr>
<tr>
<td></td>
<td></td>
<td>** Open Science Framework</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Improving the quality of reporting</td>
<td>Use of reporting checklists</td>
<td>J</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Protocol checklists</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Protecting against conflicts of interest</td>
<td>* Disclosure of conflicts of interest</td>
<td>J</td>
</tr>
<tr>
<td></td>
<td></td>
<td>** Exclusion/containment of financial and non-financial conflicts of interest</td>
<td></td>
</tr>
<tr>
<td>Reproducibility</td>
<td>Encouraging transparency and open science</td>
<td>* Open data, materials, software and so on</td>
<td>J, F, R</td>
</tr>
<tr>
<td>Evaluation</td>
<td>Diversifying peer review</td>
<td>* Pre-registration (**** for clinical trials, * for other studies)</td>
<td>J</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Preprints (* in biomedical/behavioural sciences, **** in physical sciences)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Pre- and post-publication peer review, for example, Publons, PubMed Commons</td>
<td>J</td>
</tr>
<tr>
<td>Incentives</td>
<td>Rewarding open and reproducible practices</td>
<td>Badges (*)</td>
<td>J, I, F</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Registered Reports (*)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transparency and Openness Promotion guidelines (*</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Funding replication studies (*)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Open science practices in hiring and promotion (*)</td>
<td></td>
</tr>
</tbody>
</table>

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

• ACM SIGMOD Most Reproducible Paper Award
• ACM SIGMOD Labels

Taking part in the SIGMOD Reproducibility process enables your paper to take the ACM Results Replicated label. This is embedded in the PDF of your paper in the ACM digital library.

There is an option to also host your data, scripts and code in the ACM digital library as well to make them available to a broad audience, which will award the ACM Artifacts Available label.

ACM Results Replicated label

The experimental results of the paper were replicated by the committee and were found to support the central results reported in the paper.

ACM Artifacts Available label

The experiments (data, code, scripts) are made available to the community.
Incentives

- ICSE “Artifacts Evaluated Reusable”
Incentives

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

Brazilian biomedical science faces reproducibility test

Researchers at more than 60 Brazilian labs will assess the replicability of research by their country’s scientists.
Provenance of these slides

• GOBLE, C. What is reproducibility? The Rbrouhaha, In:First International Workshop on Reproducible Open Science (Hannover), 2016.