Reproducible Research: a Perspective

Arnaud Legrand
Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP

UFRGS, October 2019
• RR mainly allows to fight scientific misconduct (e.g., fraud). That’s nice but I’m honnest so just let me do my work!
• RR mainly allows to fight scientific misconduct (e.g., fraud). That’s nice but I’m honnest so just let me do my work!

• RR is all about re-executing the same code, even if the code is stupid and makes wrong computation. It’s pointless!
• RR mainly allows to fight scientific misconduct (e.g., fraud). That’s nice but I’m honnest so just let me do my work!

• RR is all about re-executing the same code, even if the code is stupid and makes wrong computation. It’s pointless!

• My student has done everything with org-jupyter-studio-mode. Now he’s gone and I can’t reuse what he did. See, what’s the point? RR does not help!
• RR mainly allows to fight scientific misconduct (e.g., fraud). That’s nice but I’m honnest so just let me do my work!

• RR is all about re-executing the same code, even if the code is stupid and makes wrong computation. It’s pointless!

• My student has done everything with org-jupyter-studio-mode. Now he’s gone and I can’t reuse what he did. See, what’s the point? RR does not help!

• RR is about controling and checking everything, which slows down the scientific discovery process. Changing the way we work and publish may be harmful!
Past
Electronic Documents Give Reproducible Research a New Meaning

Jon F. Claerbout and Martin Karrenbach, Stanford Univ.

SUMMARY

A revolution in education and technology transfer follows from the marriage of word processing and software command scripts. In this marriage an author attaches to every figure caption a pushbutton or a name tag usable to recalculate the figure from all its data, parameters, and programs. This provides reproducibility in computing.

In 1990, we set this sequence of goals:

- Learn how to merge a publication with its underlying computational analysis.
- Teach researchers how to prepare a document in a form where they themselves can reproduce their own research results a year or more later by “pressing a single button”.
- Learn how to leave finished work in a condition where coworkers can reproduce the calculation including the final illustration by pressing a button in its caption.
- Prepare a complete copy of our local software environment so that graduating students can take their work away with them to other sites, press a button, and reproduce their Stanford work.
- Merge electronic documents written by multiple authors (SEP reports).

- make incremental improvements in electronic-document software
- seek partners for broadening standards (and making incremental improvements).

Our basic goal is reproducible research. The electronic document is our means to this end. In principle, reproducibility in research can be achieved without electronic documents and that is how we started. Our first nonelectronic reproducible document was a textbook in which the paper document contained the name of a program script in every figure caption. The program scripts were organized by book chapter and section so they could be correlated to an accompanying magnetic tape dump of the file system. The magnetic tape also contained all the necessary data to feed the program script.

Now that we have begun using CD-ROM publication, we can go much further. Every figure caption contains a pushbutton that jumps to the appropriate science directory (folder) and initiates a figure rebuild command and then displays the figure, possibly as a movie or interactive program. We normally display seismic images of the earth’s interior, but to reach wider audiences, Figure 1 shows a satellite weather picture which the pushbutton will animate as seen on commercial television. We include all our plot software as well as freely available software from many sources, including compilers and the \LaTeX word processing system. Naturally we cannot include licensed software, but with the exception...

**2006**: Inconsistencies reveal a programming mistake

*a homemade data-analysis program had flipped two columns of data, inverting the electron-density map from which his team had derived the protein structure.*

**5 retraction**s that motivate improved software engineering practices in computational biology
"clinical trials in oncology have the highest failure rate [only 5% are licensed] compared with other therapeutic areas […] difficulty to repeat experiments even in the laboratory of the original investigator"


The scientific process demands the highest standards of quality, ethics and rigor.

8 ACM conferences and 5 journals

EM\textsuperscript{no} = the code cannot be provided

1,500 scientists lift the lid on reproducibility, Nature, May 2016

Social causes

- Fraud, conflict of interest (pharmaceutic, ...)
- No incentive to reproduce/check our own work (afap), nor the work of others (big results!), nor to allow others to check (competition)
- Peer review does not scale: 1+ million articles per year!

Methodological or technical causes

- The many biases (apophenia, confirmation, hindsight, experimenter, ...): bad designs
- Selective reporting, weak analysis (statistics, data manipulation mistakes, computational errors)
- Lack of information, code/raw data unavailable
**Different concerns**

- Social Sciences, Oncology, ...
  - methodology, statistics
- Genomics
  - software engineering, computational reproducibility, provenance, ...
- Computational fluid dynamics
  - numerical issues

Distinguish between:

- experimental science
- observational science
- computational science (simulation)
- (big) data analysis
Present
Reproducible Research: Trying to Bridge the Gap

Inspired by Roger D. Peng’s lecture on reproducible research, May 2014
Reproducible Research: Trying to Bridge the Gap

Inspired by Roger D. Peng’s lecture on reproducible research, May 2014
Try to keep track of the whole chain

Inspired by Roger D. Peng’s lecture on reproducible research, May 2014
Reproducible Research: Trying to Bridge the Gap

"Tricky" and "Easy" refer to parallel computer scientist use cases

Inspired by Roger D. Peng’s lecture on reproducible research, May 2014
Existing Tools, Emerging Standards

Notebooks and workflows

- Jupyter
- R Studio

Software environments

- Python 3-matplotlib
- Python 3-dateutil
- Python 3-six
  
- Python 3
  
- Python 3-numpy

Sharing platforms

- JupyterHub
- arXiv.org
- HAL
- figshare
- Zenodo
- Software Heritage

Real dependencies

- Python dependencies
- Matplotlib library

Fake OS dependencies

- Induced by package granularity
Un document computationnel

Mon ordinateur m'indique que π vaut approximativement

3.141592653589793

Mais calculé avec la méthode des aiguilles de Buffon, on obtiendrait comme approximation :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/sum((x-np.sin(theta)) > 1) / N
```

3.1437198694098765

On peut inclure des formules mathématiques comme \( \frac{1}{\sigma \sqrt{2\pi}} \exp \left( -\frac{(x-\mu)^2}{2\sigma^2} \right) \) et des dessins qui n'ont rien à voir avec π (si ce n'est une constante de normalisation... 😜).
DOCUMENT initial dans son environnement

DOCUMENT final

Un document computationnel

Mon ordinateur m'indique que π vaut approximativement

3.141592653589793

Mais calculé avec la méthode des aiguilles de Buffon, on obtiendrait comme approximation :

```python
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x-np.sin(theta))>1)/N)
```

3.1437198594098765

On peut inclure des formules mathématiques comme \( \frac{1}{\sigma \sqrt{2\pi}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) \) et des dessins qui n'ont rien à voir avec π (si ce n'est une constante de normalisation... ☺).
**Un document computationnel**

Mon ordinateur m’indique que π vaut approximativement

3.141592653589793

Mais calculé avec la méthode des aiguilles de Buffon, on obtiendrait comme approximation :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x-np.sin(theta))>1)/N)
```

3.1437198694098765

On peut inclure des formules mathématiques comme \( \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \) et des dessins qui n’ont rien à voir avec π (si ce n’est une constante de normalisation... 😊).
**TOOL 1: COMPUTATIONAL NOTEBOOKS/LITTERATE PROGRAMMING**

Document initial dans son environnement

Document final

**Un document computationnel**

Mon ordinateur m’indique que π vaut **approximativement**

3.141592653589793

Mais calculé avec la méthode des aiguilles de Buffon, on obtiendrait comme approximation :

```python
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=np.pi/2)
2/(sum((x-np.sin(theta))>1)/N)
```

3.1437198594098765

On peut inclure des formules mathématiques comme \( \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \) et des dessins qui n’ont rien à voir avec π (si ce n’est une constante de normalisation... 😊).
Un document computationnel

Mon ordinateur m’indique que π vaut approximativement

3.141592653589793

Mais calculé avec la méthode des aiguilles de Buffon, on obtiendrait comme approximation :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/sum((x-np.sin(theta)>1)/N)
```

3.1437198694098765

On peut inclure des formules mathématiques comme $\frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ et des dessins qui n’ont rien à voir avec π (si ce n’est une constante de normalisation... 😊).
Un document computationnel

Mon ordinateur m'indique que π vaut approximativement

3.141592653589793

Mais calculé avec la méthode des aiguilles de Buffon, on obtiendrait comme approximation :

```
import numpy as np
N = 1000000
x = np.random.uniform(size=N, low=0, high=1)
theta = np.random.uniform(size=N, low=0, high=pi/2)
2/(sum((x-np.sin(theta))>1)/N)
```

3.1437198594098765

On peut inclure des formules mathématiques comme \( \frac{1}{\sigma \sqrt{2\pi}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) \) et des dessins qui n'ont rien à voir avec π (si ce n'est une constante de normalisation...).
What is hiding behind a simple

```python
import matplotlib
```

Package: python3-matplotlib
Version: 2.1.1-2
Depends: python3-dateutil, python-matplotlib-data (>= 2.1.1-2), python3-pyparsing (>= 1.5.6), python3-six (>= 1.10), python3-tz, libjs-jquery, libjs-jquery-ui, python3-numpy (>= 1:1.13.1), python3-numpy-abi9, python3 (<< 3.7), python3 (>= 3.6~), python3-cycler (>= 0.10.0), python3:any (>= 3.3.2-2~), libc6 (>= 2.14), libfreetype6 (>= 2.2.1), libgcc1 (>= 1:3.0), libpng16-16 (>= 1.6.2-1), libstdc++6 (>= 5.2), zlib1g (>= 1:1.1.4)
What is hiding behind a simple

```
import matplotlib
```

Package: python3-matplotlib
Python and its rapidly evolving environment

```python
1 python2 -c "print(10/3)"
2 python3 -c "print(10/3)"
```

3
3.3333333333333335
**Tool 2: Fighting Software Environments Nightmare**

Python and its rapidly evolving environment

```
1  python2 -c "print(10/3)"
2  python3 -c "print(10/3)"
```
Tool 2: Fighting Software Environments Nightmare

Python and its rapidly evolving environment

```
1. python2 -c "print(10/3)"
2. python3 -c "print(10/3)"
```

Cortical Thickness Measurements (PLOS ONE, June 2012): FreeSurfer: differences were found between the Mac and HP workstations and between Mac OSX 10.5 and OSX 10.6.
Tool 2: Fighting Software Environments Nightmare

Python and its rapidly evolving environment

```python
python2 -c "print(10/3)"
python3 -c "print(10/3)"
```

Cortical Thickness Measurements (PLOS ONE, June 2012): FreeSurfer differences were found between the Mac and HP workstations and between Mac OSX 10.5 and OSX 10.6.

*The half-life of a referenced URL is approximately 4 years from its publication date.*

P. Habibzadeh. Decay of References to Web sites in Articles Published in General Medical Journals: Mainstream vs Small Journals”. Applied Clinical Informatics. 4 (4), 2013

*half life ranged from 2.2 years in EMHJ to 5.3 years in BMJ*
Tool 3: Fighting Information Loss with Archives


*The half-life of a referenced URL is approximately 4 years from its publication date.*

P. Habibzadeh. Decay of References to Web sites in Articles Published in General Medical Journals: Mainstream vs Small Journals”. Applied Clinical Informatics. 4 (4), 2013

*Half life ranged from 2.2 years in EMHJ to 5.3 years in BMJ*

**Article archives**
- arXiv.org
- HAL

**Data archives**
- figshare
- zenodo

**Software Archive**
- Software Heritage

 '>' or ' = awesome collaborations ≠ archive
Manifesto: "I solemnly pledge" (WSSSPE, Lorena Barba, FAIR)

1. I will teach my graduate students about reproducibility
2. All our research code (and writing) is under version control
3. We will always carry out verification and validation
4. We will share data, plotting script & figure under CC-BY
5. We will upload the preprint to arXiv at the time of submission of a paper
6. We will release code at the time of submission of a paper
7. We will add a "Reproducibility" declaration at the end of each paper
8. I will keep an up-to-date web presence

Soft. Engineering, Statistics, and Reproducible Research in the curricula

- Webinars on RR 2016-2017
- MOOC on RR (3rd edition planned for January 2020)
- Book on RR in June 2019
Artifact evaluation and ACM badges

Major conferences

- **Supercomputing**: Artifact Description (AD) mandatory, Artifact Evaluation (AE) still optional, Double blind vs. RR
- **NeurIPS, ICLR**: open reviews, reproducibility challenge

Joelle Pineau @ NeurIPS’18

- ACM SIGMOD 2015-2019, ...

**Mentality is evolving**: people care and make stuff available
Key Concerns for our Community (Room for Improvement)

- Awareness of Experiments and Statistics
  - How are cloud performance currently obtained and reported?, March 2019

- Shared testbeds and Experimental control
  - Grid’5000/SILECS, ...
Future
Publish or Perish (OK, this is past and present)

  - AI: over 1,000 ranked journals ($\times10$ in 15 years)
  - Shorter papers with increasing self references
  - More and more papers without any citation
  - Sharp increase in the number of new authors publishing at a much faster rate given their career age

![Diagram]

- Often rejected
- Rare
- Safe bet
- Rejected
- Often rejected

Quality

Novelty
Publish or Perish (OK, this is past and present)

  - AI: over 1,000 ranked journals (×10 in 15 years)
  - Shorter papers with increasing self references
  - More and more papers without any citation
  - Sharp increase in the number of new authors publishing at a much faster rate given their career age


We redo Science!

We redo
Science!

GitHub.com/Rescience

![Diagram showing the relationship between Quality and Novelty with the categories Often rejected, Safe bet, Rare, Rejected, and Often rejected.](image-url)
To err is human.

Good research requires time and resources

1. **Train yourself and your students on RR, statistics, experiments**
   - Beware of checklists and norms
   - Understand what’s at stake

2. **Make publication practices evolve**

3. **Prepare the Future: Toward literate experimentation?**
   - Reuse, reuse, reuse
   - How to share Experiments
Some advertising

Ten Years Reproducibility Challenge
Rescience Special Issue
Free to Read - Free to Publish

Would you dare to run the code from your past self?
(the one that does not answer mail)

Submission deadline 01/04/2020
http://rescience.github.io/ten-years
In association with Rescience, CNRS, Software Heritage, ReScience, Comité pour la Science Ouverte,
URFIST Bordeaux & Mission de la pédagogie et du numérique pour l'enseignement supérieur

3rd Edition: ≈ Feb. 2020
A new MOOC: ”Advanced RR”

• Software environment control (Docker)
• Scientific workflow (snakemake)
• Managing data (HDF5, archiving)

October 2020?