A few major challenges for Informatics

reproductibility, transparency, explainability

Roberto Di Cosmo
INRIA and IRIF

roberto@dicosmo.org

November 22, 2017

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE

Outline

1. Présentations
2. Plan
3. The Science Crisis
4. The state of Software reproducibility
5. The Software Heritage initiative
6. The rise of Big Data and Deep Learning
7. Conclusion

Qui je suis

professeur d’Informatique, chercheur
20 ans de contribution au Logiciel Libre

1998 *Hold up planétaire* – vulgarisation enjeux sociétaux de l’informatique

1999 *DemoLinux* – première distro live GNU/Linux

2007 *GTLL Systematic* 150 members 40 projects 200Me

2010 *IRILL* `www.irill.org`

2015 *Software Heritage*

Outline

1. Présentations
2. Plan
3. The Science Crisis
4. The state of Software reproducibility
5. The Software Heritage initiative
6. The rise of Big Data and Deep Learning
7. Conclusion

Plan: two distinct parts…

Reproducibility and transparency in Science

- the science crisis
- the role of software
- Software Heritage

Transparency and explanations in the AI era

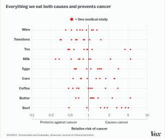
- the rise of big data and machine learning
- when software is only part of the story

Outline

1. Présentations
2. Plan
3. The Science Crisis
4. The state of Software reproducibility
5. The Software Heritage initiative
6. The rise of Big Data and Deep Learning
7. Conclusion

Inconsistencies all around us

What causes cancer?


Everything we eat both causes and prevents cancer

SOURCE: Schoenfeld and Ioannidis, *American Journal of Clinical Nutrition*

Is everything we eat associated with cancer?

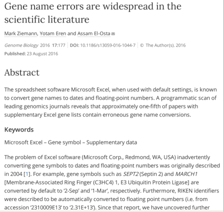
Schoenfeld and Ioannidis, *Amer. Jour. of Clinical Nutrition*, 2013.

Inconsistency *an incompatibility between two propositions that cannot both be true*

SEPT2 is Septin 2 or September 2nd?

Gene name errors are widespread in the scientific literature Ziemann, Eren and El-Osta, *Genome Biology*, 2016.

Corruption *The process by which a computer database or program becomes debased by alteration or the introduction of errors*


Gene name errors are widespread in the scientific literature

Mark Ziemann, Yotam Eren and Assam El-Osta

Genome Biology 2016 **17**:177 | **DOI:** 10.1186/s13059-016-1044-7 | © The Author(s). 2016

Published: 23 August 2016

Abstract

The spreadsheet software Microsoft Excel, when used with default settings, is known to convert gene names to dates and floating-point numbers. A programmatic scan of leading genomics journals reveals that approximately one-fifth of papers with supplementary Excel gene lists contain erroneous gene name conversions.

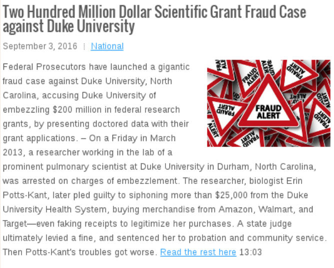
Keywords

Microsoft Excel – Gene symbol – Supplementary data

The problem of Excel software (Microsoft Corp., Redmond, WA, USA) inadvertently converting gene symbols to dates and floating-point numbers was originally described in 2004 [1]. For example, gene symbols such as *SEPT2* (Septin 2) and *MARCH1* [Membrane-Associated Ring Finger (C3HC4) 1, E3 Ubiquitin Protein Ligase] are converted by default to '2-Sep' and '1-Mar', respectively. Furthermore, RIKEN identifiers were described to be automatically converted to floating point numbers (i.e. from accession '2310009E13' to '2.31E+13'). Since that report, we have uncovered further

And it gets worse!

Doctored data?



Two Hundred Million Dollar Scientific Grant Fraud Case against Duke University

September 3, 2016 ⁝ National

Federal Prosecutors have launched a gigantic fraud case against Duke University, North Carolina, accusing Duke University of embezzling $200 million in federal research grants, by presenting doctored data with their grant applications. – On a Friday in March 2013, a researcher working in the lab of a prominent pulmonary scientist at Duke University in Durham, North Carolina, was arrested on charges of embezzlement. The researcher, biologist Erin Potts-Kant, later pled guilty to siphoning more than $25,000 from the Duke University Health System, buying merchandise from Amazon, Walmart, and Target—even faking receipts to legitimize her purchases. A state judge ultimately levied a fine, and sentenced her to probation and community service. Then Potts-Kant's troubles got worse. Read the rest here 13:03

Fraud *wrongful or criminal deception intended to result in financial or personal gain*

What are drugs good for?

Non reproducibile results ...

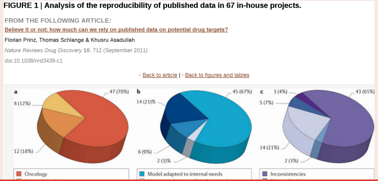


FIGURE 1 | Analysis of the reproducibility of published data in 67 in-house projects.

FROM THE FOLLOWING ARTICLE:

Believe it or not: how much can we rely on published data on potential drug targets?

Florian Prinz, Thomas Schlange & Khusru Asadullah

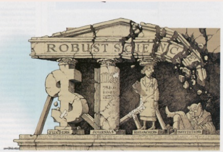
Nature Reviews Drug Discovery **10**, 712 (September 2011)

doi:10.1038/nrd3439-c1

‹ Back to article | ‹ Back to figures and tables

We face a science crisis

"Sub-prime science"? (Nicholas Humprey)



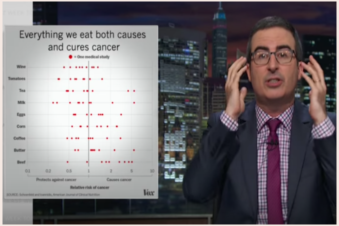
- inconsistencies
- data corruption, fraud
- non reproducible findings…

(picture from Nature, Sep. 2015)

The world starts noticing

The Economist
Washington's lawyer surplus
How to do a nuclear deal with Iran
Investment tips from Nobel economists
Junk bonds are back
The meaning of Sachin Tendulkar
HOW SCIENCE GOES WRONG

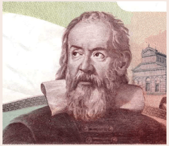
October 2013



John Oliver, *Science* May 2016

How we built our scientific knowledge

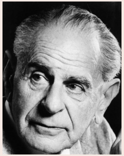
The experimental method



- make an *observation*
- formulate an *hypothesis*
- set up an experiment
- formulate a *theory*

And then we reproduce and verify.

Reproducibility is the key



non-reproducible single occurrences are of no significance to science

Karl Popper, The Logic of Scientific Discovery, 1934

Reproducibility in the digital age

For an experiment involving software, we need

- open access to the scientific article describing it
- open data sets used in the experiment
- source code of all the components
- environment of execution
- stable references between all this

Remark

The first two items are already widely discussed!

… what about *software*?

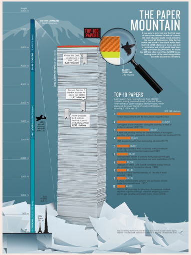
Software is Knowledge

Software is *an essential component* of modern scientific research

Top 100 papers (Nature, October 2014)

> *[...] the vast majority describe experimental methods or software that have become essential in their fields.*

http://www.nature.com/news/the-top-100-papers-1.16224



Pressure to make research code available is now raising

Evaluation of software artefacts (optional)

- tools are usable, in line with expectations
- started as a contest in 2011 (ESEC/FSE) (winner *Vouillon and Di Cosmo*)
- now going mainstream: POPL'17, POPL'16, ECOOP'16, OOPSLA'16, CGO'16, VISSOFT'16, PLDI'16, CGO'15, PPoPP'15, VISSOFT'15, ISSTA'15, OOPSLA'15, PLDI'15, POPL'15, CAV'15, ECOOP'15, FSE'15, ISSTA'14, OOPSLA'14, PLDI'14, ECOOP'14, FSE'14, SAS'13, OOPSLA'13, ECOOP'13, FSE'13, FSE'11

Use the Source, Luke!

Some people claim that having (all) the source of the code used in an experiment is *not worth the effort* (see "Replicability is not Reproducibility: Nor is it Good Science", Chris Drummond, ICML 2009)

Sure, diversity *is* important, but:

- Source code is like the proof used in a theorem: can we really accept *Fermat statements* like "the details are omitted due to lack of space"?
- modern complex systems makes even the simplest experiment depend on a wealth of components and configuration options
- access to *all* the source code is not just necessary to *reproduce*, it is also useful to *evolve and modify*, to *build new experiments* from the old ones

Source code matters!

> "The source code for a work means the preferred form of the work for making modifications to it."
>
> — GPL Licence

Hello World

Program (excerpt of binary)

```
4004e6:  55
4004e7:  48 89 e5
4004ea:  bf 84 05 40 00
4004ef:  b8 00 00 00 00
4004f4:  e8 c7 fe ff ff
4004f9:  90
4004fa:  5d
4004fb:  c3
```

Program (source code)

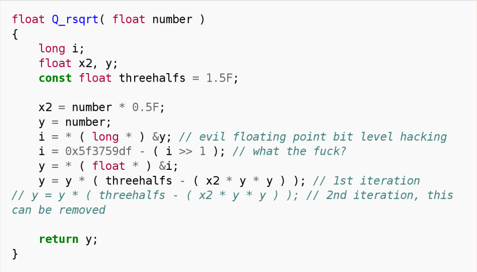
```
/* Hello World program */

#include<stdio.h>

void main()
{
    printf("Hello World");
}
```

Software Source Code is *special*

Harold Abelson, Structure and Interpretation of Computer Programs (1st ed.) 1985

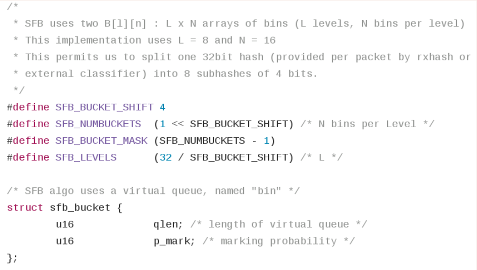
“Programs must be written for people to read, and only incidentally for machines to execute.”

Quake 2 source code (excerpt)



```
float Q_rsqrt( float number )
{
    long i;
    float x2, y;
    const float threehalfs = 1.5F;

    x2 = number * 0.5F;
    y = number;
    i = * ( long * ) &y; // evil floating point bit level hacking
    i = 0x5f3759df - ( i >> 1 ); // what the fuck?
    y = * ( float * ) &i;
    y = y * ( threehalfs - ( x2 * y * y ) ); // 1st iteration
// y = y * ( threehalfs - ( x2 * y * y ) ); // 2nd iteration, this
can be removed

    return y;
}
```

Net. queue in Linux (excerpt)



```
/*
 * SFB uses two B[l][n] : L x N arrays of bins (L levels, N bins per level)
 * This implementation uses L = 8 and N = 16
 * This permits us to split one 32bit hash (provided per packet by rxhash or
 * external classifier) into 8 subhashes of 4 bits.
 */
#define SFB_BUCKET_SHIFT 4
#define SFB_NUMBUCKETS  (1 << SFB_BUCKET_SHIFT) /* N bins per Level */
#define SFB_BUCKET_MASK (SFB_NUMBUCKETS - 1)
#define SFB_LEVELS      (32 / SFB_BUCKET_SHIFT) /* L */

/* SFB algo uses a virtual queue, named "bin" */
struct sfb_bucket {
        u16             qlen; /* length of virtual queue */
        u16             p_mark; /* marking probability */
};
```

Len Shustek, Computer History Museum

“Source code provides a view into the mind of the designer.”

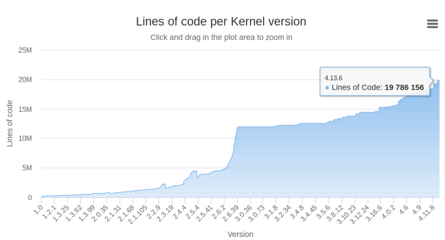
~ 50 years, a lightning fast growth

Apollo 11 Guidance Computer (~60.000 lines), 1969

"When I first got into it, nobody knew what it was that we were doing. It was like the Wild West."

Margaret Hamilton

Linux Kernel



... now in your pockets!

are we taking care of all this?

Outline

1. Présentations
2. Plan
3. The Science Crisis
4. The state of Software reproducibility
5. The Software Heritage initiative
6. The rise of Big Data and Deep Learning
7. Conclusion

Collberg's report from the trenches

Analysis of 613 papers

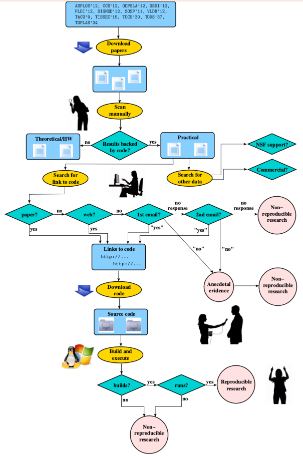
- 8 ACM conferences: ASPLOS'12, CCS'12, OOPSLA'12, OSDI'12, PLDI'12, SIGMOD'12, SOSP'11, VLDB'12
- 5 journals: TACO'9, TISSEC'15, TOCS'30, TODS'37, TOPLAS'34

all very practical oriented

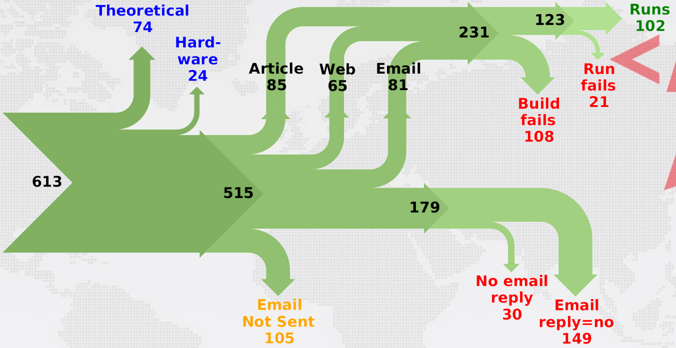
The basic question

can we get the code to build and run?

The workflow



The result



This can be debated (see `http://cs.brown.edu/~sk/Memos/Examining-Reproducibility/`), but…

… that’s a whopping 81% of non reproducible works!

The reasons (or, “the dog ate my program”)

Why so much software fails to pass the test?

Many issues, nice anecdotes, and it finally boils down to

- *Availability*
- *Traceability*
- Environment
- Automation (do *you* use continuous integration?)
- Documentation
- Understanding (including free/open source software)

The first two are important *software preservation issues*

Yes, code is fragile:

it can be destroyed, and we can lose trace of it

Outline

1. Présentations
2. Plan
3. The Science Crisis
4. The state of Software reproducibility
5. The Software Heritage initiative
6. The rise of Big Data and Deep Learning
7. Conclusion

The Software Heritage Project

www.softwareheritage.org

Software Heritage

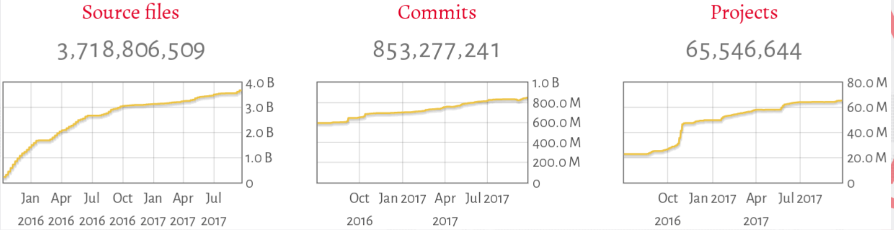
Our mission

<span style="color:red">Collect</span>, <span style="color:red">preserve</span> and <span style="color:red">share</span> the *source code* of *all the software* that is available

Past, present and future

Preserving the past, *enhancing* the present, *preparing* the future

Archive coverage



Source files

3,718,806,509

Commits

853,277,241

Projects

65,546,644

~150 TB blobs, ~5 TB database (as a graph: ~7 B nodes + ~60 B edges)

Our sources

- GitHub — full, up-to-date mirror
- Debian — automation in progress; GNU
- Gitorious, Google Code — processing (Archive Team & Google)
- Bitbucket — WIP

The *richest* source code archive already, … and growing daily!

Going global

April 3rd, 2017: landmark Inria Unesco agreement…

https://www.softwareheritage.org/blog

September 28th, 2017

Mauritius Call on information access

An unique opportunity

Library of Alexandria of code

Take *urgent* action to

- recover the past
- structure the future

A CERN for Software

Photo: ALMA(ESO/NAOJ/NRAO), R. Hills

Build a *common infrastructure*

- software research, better science
- for society as a whole

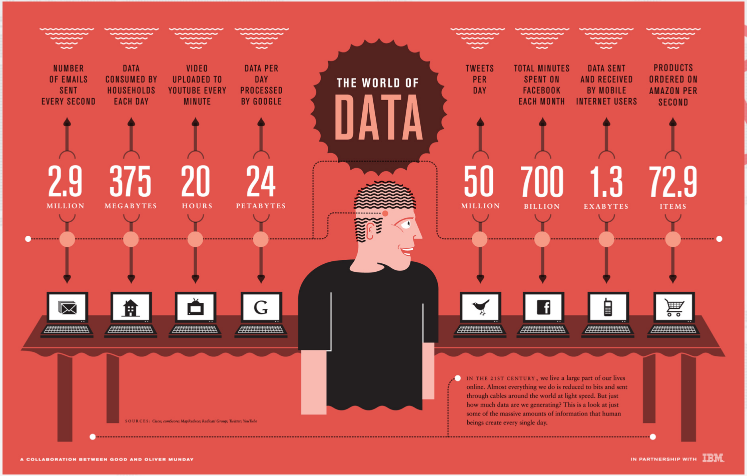
Come in, we're open www.softwareheritage.org

tons of research problems, and our code: `forge.softwareheritage.org`

Outline

1. Présentations
2. Plan
3. The Science Crisis
4. The state of Software reproducibility
5. The Software Heritage initiative
6. The rise of Big Data and Deep Learning
7. Conclusion

(Our) Data collected everywhere into Big Data

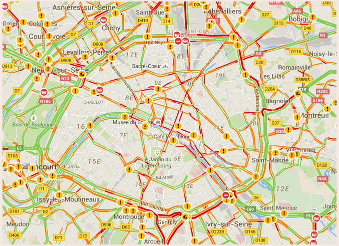


IN THE 21ST CENTURY, we live a large part of our lives online. Almost everything we do is reduced to bits and sent through cables around the world at light speed. But just how much data are we generating? This is a look at just some of the massive amounts of information that human beings create every single day.

SOURCES: *Cisco; comScore; MapReduce; Radicati Group; Twitter; YouTube*

Balancing common interest

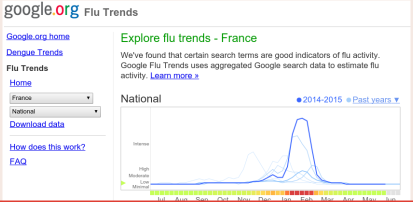
Google traffic + Waze



- données en temps réel
- votre position GSM
- vos signalements

Google Flu trends

- analyse des vos requêtes Google


google.org Flu Trends

Google.org home

Dengue Trends

Flu Trends

Home

France ▾

National ▾

Download data

How does this work?

FAQ

Explore flu trends - France

We've found that certain search terms are good indicators of flu activity. Google Flu Trends uses aggregated Google search data to estimate flu activity. Learn more »

National

● 2014-2015 ● Past years ▼

… and privacy protection

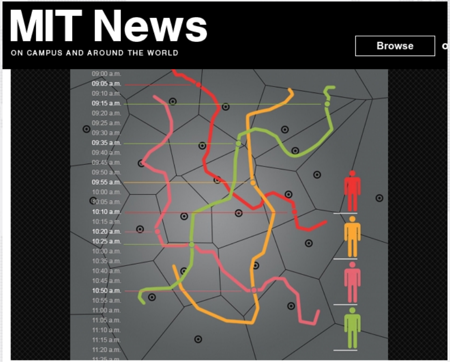
Avec de l'analyse massive des données, Target peut savoir si vous attendez un enfant... avant vous

> *"I had a talk with my daughter. It turns out there's been some activities in my house I haven't been completely aware of"*

Un client de Target, en 2012.

… is not easy!

Hey, just scratch the personal information!



"you are identified by just 4 points, over a year"

Scientific Reports, 2013

The new challenge of algorithmic decisions

Impact on real life

- price of your tickets/goods
- autonomous cars/systems decide on your life
- algorithms decide
  - your credit history
  - the university you get in!

We expect "transparency"

- France: Loi Lemaire " information loyale, claire et transparente"

Deep Learning: impressive progress

AlphaGo

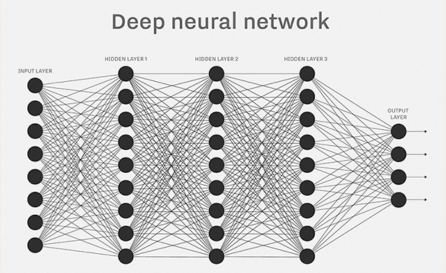
- October 2015, beats Lee Sedol
- October 2017, beaten by AlphaGo Zero

Everyday use

- language recognition, translation, . . .
- most systems moved to deep learning

The challenge

Deep neural networks



- What is an "explanation"?
- open discussion. . .

Open source

- most deep learning frameworks are open source
- it is far from enough!

Outline

1. Présentations
2. Plan
3. The Science Crisis
4. The state of Software reproducibility
5. The Software Heritage initiative
6. The rise of Big Data and Deep Learning
7. Conclusion

Age of responsibility

Technology reshapes society

Code is law (Lawrence Lessig)

- reproducitibility, replicability, science
- safety, security
- privacy, anonimity, trust,
- justice, freedom, transparency ...

Conclusion

The digital revolution is on

- reshaping society
- major ethical and social challenges

We need you!

- learn and teach Computer Science
- contribute to Free Software
- value "responsible" research
- ask questions, join the conversation

Giambattista Vico (1688 - 1744)

Conoscere, è saper fare