Replication and Recomputation in Scientific Experiments

Ian Gent
Computer Science
University of St Andrews
ian.gent@st-andrews.ac.uk

Lars Kotthoff
INSIGHT Centre for Data Analytics
University College Cork
lars.kotthoff@insight-centre.org

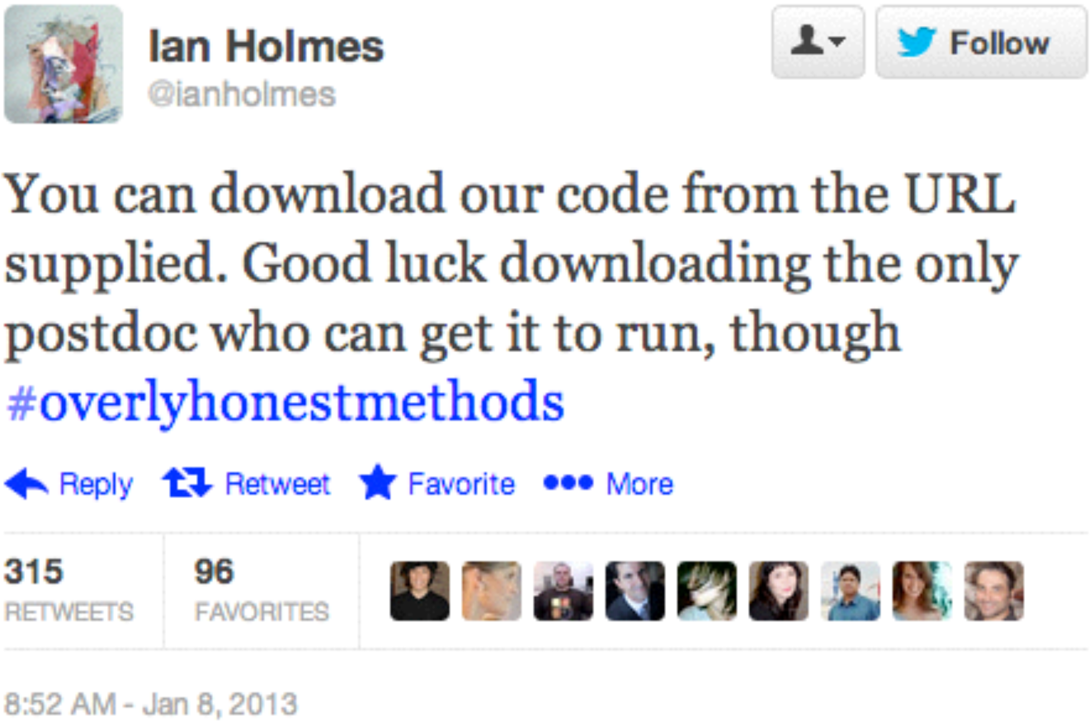
http://recomputation.org/ecai2014

contact@recomputation.org

Replication in Science

- Officially, replication is key to science
- Unofficially?

#overlyhonestmethods



You can download our code from the URL supplied. Good luck downloading the only postdoc who can get it to run, though #overlyhonestmethods

Our definition of “recomputable”

- pragmatic approach
- work with authors of experiments
- ability to rerun experiments and get similar results

Groundbreaking Papers at ECAI 2014

- “Schedule Generation Schemes for Job Shop Problems with Fuzziness”
  - Juan José Palacios, Camino R. Vela, Inés González-Rodríguez and Jorge Puente
- “Bias Reformulation for One-shot Function Induction”
  - Dianhuan Lin, Eyal Dechter, Kevin Ellis, Joshua B. Tenenbaum and Stephen H. Muggleton

Groundbreaking Papers at ECAI 2014

- “Conditioned Belief Propagation Revisited”
  - Thomas Geier, Felix Richter and Susanne Biundo
- “Fast Instantiation of GGP Game Descriptions Using Prolog with Tabling”
  - Jean-Noël Vittaut and Jean Méhat

Groundbreaking Papers at ECAI 2014

- “Automating Gödel’s Ontological Proof of God’s Existence with Higher-order Automated Theorem Provers”
  - Christoph Benzmüller and Bruno Woltzenlogel Paleo
- “DITO: A CSP-based Diagnostic Engine”
  - Yannick Pencolé
- *with more coming - watch this space!*

Why Groundbreaking?

- These papers contain recomputable experiments
- Recomputable?
  - You can download a virtual machine
  - You can run the experiment
    - very simple set of instructions
  - You can see all the details of the experiments
  - You can build on and improve these experiments
  - Or if they are perfect then learn from them
- All available from http://recomputation.org

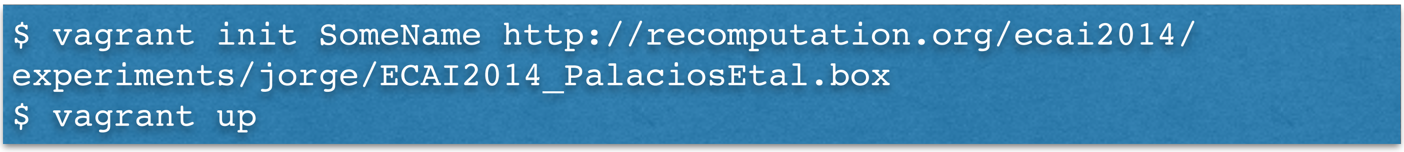
Very Simple Set of Instructions?

- E.g. for one of those groundbreaking papers
- "Schedule Generation Schemes for Job Shop Problems with Fuzziness"
  - Juan José Palacios, Camino R. Vela, Inés González-Rodríguez and Jorge Puente

```
$ vagrant init SomeName http://recomputation.org/ecai2014/
experiments/jorge/ECAI2014_PalaciosEtal.box
$ vagrant up
```

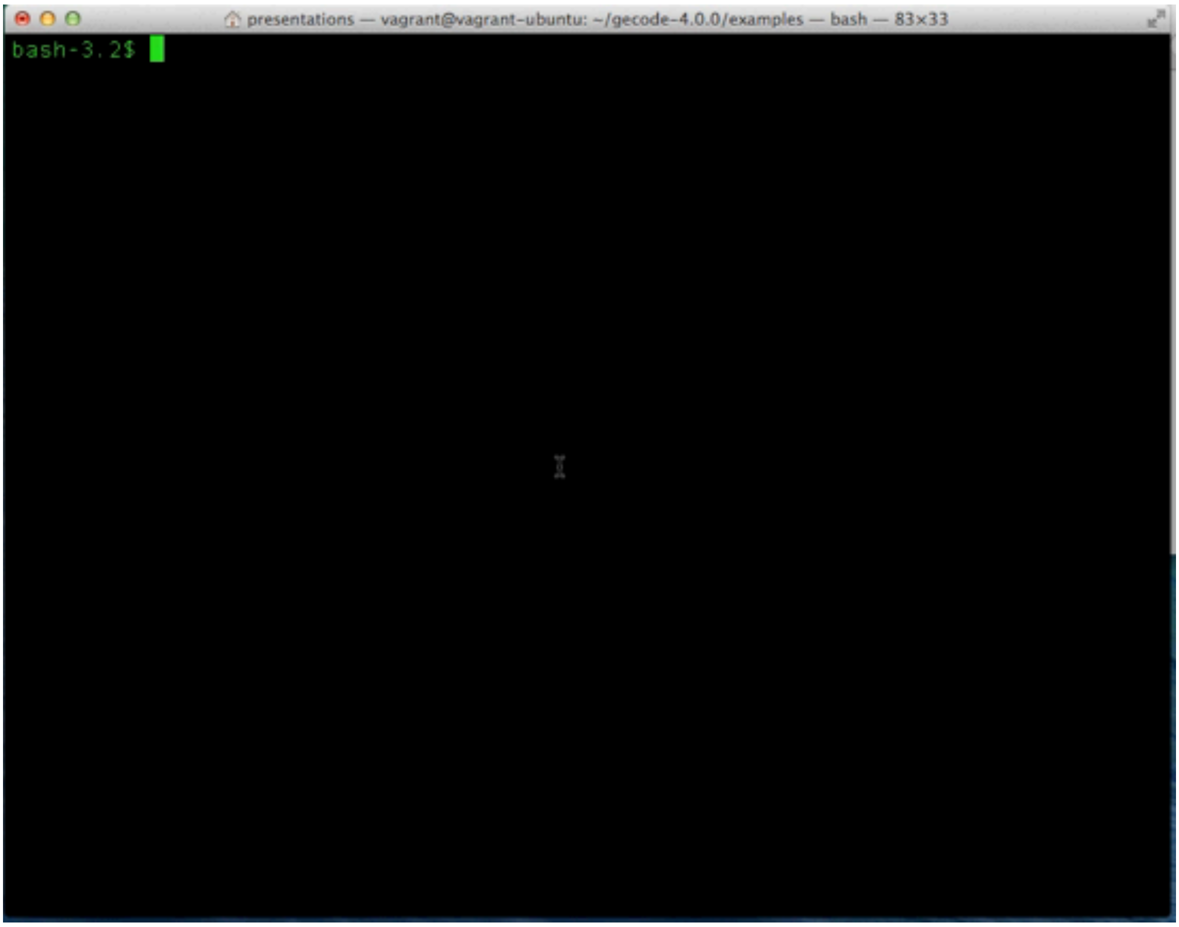
Very Simple Set of Instructions?

- Ok, you also need two free tools
  - VirtualBox
    - https://www.virtualbox.org/
  - Vagrant
    - http://www.vagrantup.com/



```
$ vagrant init SomeName http://recomputation.org/ecai2014/
experiments/jorge/ECAI2014_PalaciosEtal.box
$ vagrant up
```

Very Simple Set of Instructions?



presentations — vagrant@vagrant-ubuntu: ~/gecode-4.0.0/examples — bash — 83×33

```
bash-3.2$ 
```

Acknowledgements

Too many to thank but include...

Torsten Schaub (Programme Chair of ECAI 2014)

Agostino Dovier and Paolo Torroni (Tutorial Chairs)

St Andrews & Cork Universities for resources

EPSRC Impact Acceleration Grant at St Andrews

John McDermott, Lakshitha de Silva for technical help and …

All Authors of Recomputable Experiments from ECAI 2014

About This Tutorial...

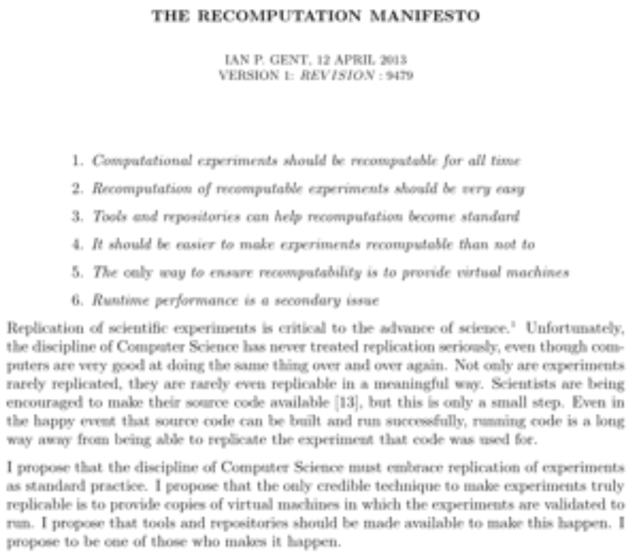
- Part 1: The Recomputation Manifesto
- Part 2: Recomputation @ ECAI 2014
- Postscript: How can you help?

Recomputation: Let's fix this!

- Experiments should be downloadable
- *And runnable easily*

- "Recomputation"
  - word older than the USA
  - adding a meaning
  - exact replication of a computational experiment

Part 1: The Recomputation Manifesto



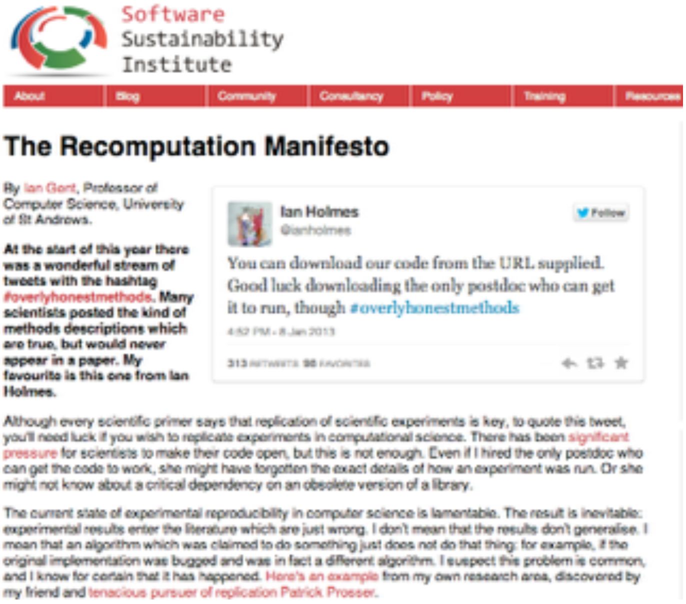
THE RECOMPUTATION MANIFESTO

IAN P. GENT, 12 APRIL 2013
VERSION 1: *REVISION* : 9479

1. *Computational experiments should be recomputable for all time*
2. *Recomputation of recomputable experiments should be very easy*
3. *Tools and repositories can help recomputation become standard*
4. *It should be easier to make experiments recomputable than not to*
5. *The only way to ensure recomputability is to provide virtual machines*
6. *Runtime performance is a secondary issue*

Replication of scientific experiments is critical to the advance of science.[1] Unfortunately, the discipline of Computer Science has never treated replication seriously, even though computers are very good at doing the same thing over and over again. Not only are experiments rarely replicated, they are rarely even replicable in a meaningful way. Scientists are being encouraged to make their source code available [13], but this is only a small step. Even in the happy event that source code can be built and run successfully, running code is a long way away from being able to replicate the experiment that code was used for.

I propose that the discipline of Computer Science must embrace replication of experiments as standard practice. I propose that the only credible technique to make experiments truly replicable is to provide copies of virtual machines in which the experiments are validated to run. I propose that tools and repositories should be made available to make this happen. I propose to be one of those who makes it happen.



The Recomputation Manifesto

By Ian Gent, Professor of Computer Science, University of St Andrews.

At the start of this year there was a wonderful stream of tweets with the hashtag #overlyhonestmethods. Many scientists posted the kind of methods descriptions which are true, but would never appear in a paper. My favourite is this one from Ian Holmes.

Ian Holmes
@ianholmes

You can download our code from the URL supplied. Good luck downloading the only postdoc who can get it to run, though #overlyhonestmethods

4:52 PM - 8 Jan 2013

313 RETWEETS 98 FAVORITES

Although every scientific primer says that replication of scientific experiments is key, to quote this tweet, you'll need luck if you wish to replicate experiments in computational science. There has been significant pressure for scientists to make their code open, but this is not enough. Even if I hired the only postdoc who can get the code to work, she might have forgotten the exact details of how an experiment was run. Or she might not know about a critical dependency on an obsolete version of a library.

The current state of experimental reproducibility in computer science is lamentable. The result is inevitable: experimental results enter the literature which are just wrong. I don't mean that the results don't generalise. I mean that an algorithm which was claimed to do something just does not do that thing: for example, if the original implementation was bugged and was in fact a different algorithm. I suspect this problem is common, and I know for certain that it has happened. Here's an example from my own research area, discovered by my friend and tenacious pursuer of replication Patrick Prosser.

http://tinyurl.com/recomputation1

http://tinyurl.com/recomputation2

Unashamedly...

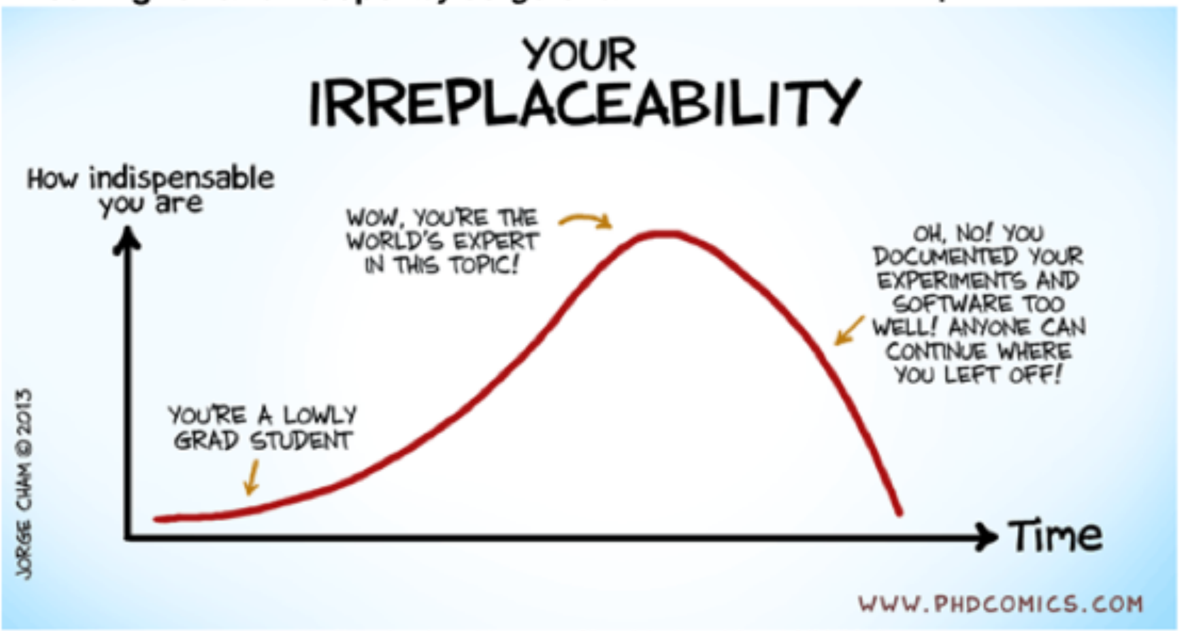
- Recomputation is about exact reproduction
  - not scientifically more important replication
  - sometimes called bit replication
- I want to change the way Computer Science is done
  - this may be a bit overambitious
- I'm going to need help

The Recomputation Manifesto

1. Computational experiments should be recomputable for all time

xkcd/PhD/Dilbert Compliance

Piled Higher and Deeper ***by Jorge Cham*** **www.phdcomics.com**


YOUR IRREPLACEABILITY
How indispensable you are
Time
YOU'RE A LOWLY GRAD STUDENT
WOW, YOU'RE THE WORLD'S EXPERT IN THIS TOPIC!
OH, NO! YOU DOCUMENTED YOUR EXPERIMENTS AND SOFTWARE TOO WELL! ANYONE CAN CONTINUE WHERE YOU LEFT OFF!
JORGE CHAM © 2013
WWW.PHDCOMICS.COM

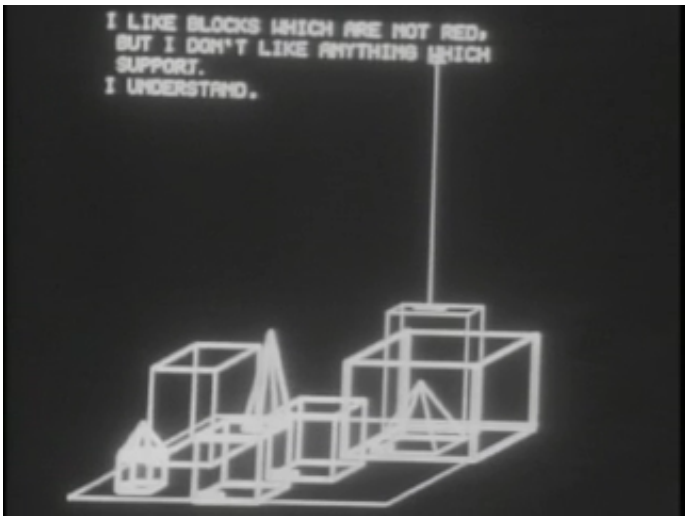
title: "How irreplaceable are you?" originally published 3/20/2013

Galileo’s Telescopes

- Imagine if we could look through Galileo’s telescopes
- And we hadn’t bothered to keep them
  - Or threw away the only postdoc …
- *This has happened in computer science*
  - *Many many times*

Galileo’s Telescopes

- SHRDLU is a famous early AI program
- We have the source code
- But we can’t run it!



```
I LIKE BLOCKS WHICH ARE NOT RED,
 BUT I DON'T LIKE ANYTHING WHICH
 SUPPORT.
I UNDERSTAND.
```

Image AI Lab MIT

The Recomputation Manifesto

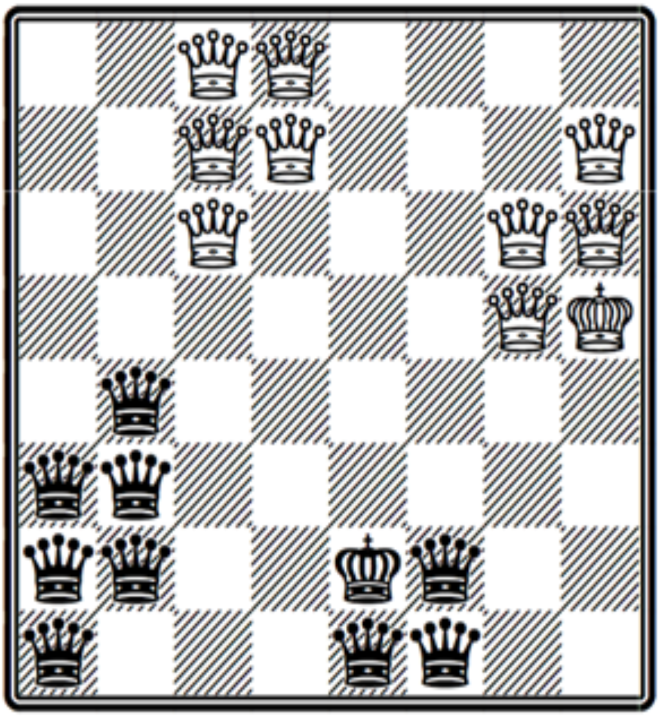
2. Recomputation of recomputable experiments should be very easy

A Chess Puzzle

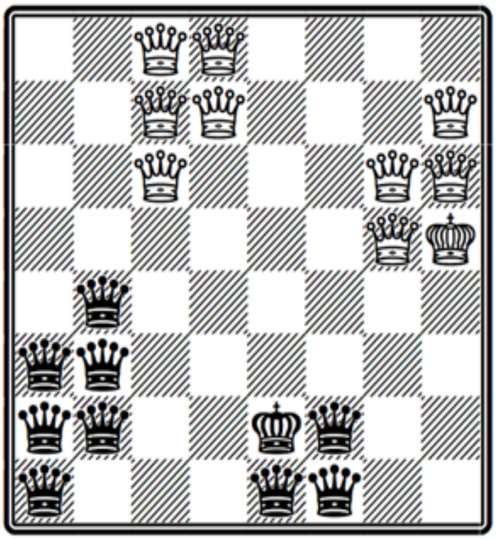
This position contains the king and all nine possible queens of each colour, i.e. the original and eight promoted pawns.

No queen is on the same row, column or diagonal as any piece of the opposite colour.

This is the only chess position for which the description of the previous paragraph is true, excepting rotations and reflections of the chessboard, or swapping black and white.



A Chess Puzzle



How to recompute an experiment for this.

1. Install VirtualBox
2. Install Vagrant
3. Open a terminal and …

```
mkdir anydir

cd anydir

vagrant init experiment1 http://recomputation.org/cp2013/experiment1/
recomputation-QueensPuzzle-b.box

vagrant up
```

The Recomputation Manifesto

3. Tools and repositories can help recomputation become standard

Some good things out there

- Some repositories with various goals, e.g....
  - RunMyCode.org:
  - IPOL Image Processing On Line journal.
  - myExperiment.org
  - SHARE: Sharing Hosted Autonomous Research Environments:
  - ResearchCompendia: Sharing code, data and computational methods
  - CDE
  - ... and many others...
  - ... plus repurposable general tools.
- More are needed

What is missing?

- We want a repository which is ...
- Focussed on computational science *experiments*
  - liberated from caring if entire code base works
- Totally open
  - anyone in the world can get the entire experiment
  - though we'll see some licensing problems
- Totally agnostic as to type of experiment
  - not restricted to e.g. factorial application of algorithms to instances
  - Or to certain languages like Python, R...
- Dedicated to being around for a long time
  - could cause problems when hardware generations change

recomputation.org

If we can compute your experiment now, anyone can recompute it 20 years from now

Main foci are:

- Taking freeform computational experiments
- Providing Virtual Machine versions of experiments
- Keeping experiments around for all time if we can

The Recomputation Manifesto

4. It should be easier to make experiments recomputable than not

It’s easier to do it right than wrong

- Quasi paradoxical
  - probably not very well expressed
- Quote one of the heroes of replication in CS ...
- “It's not really for the benefit of other people. Experience shows the principal beneficiary of reproducible research is you the author yourself.”

Jon Claerbout

It’s easier to do it right than wrong

- I have often had problems rerunning experiments I’ve run in the past
  - often the very recent past
- It should be easy to rerun an old experiment
- And then hack it to be better

The Recomputation Manifesto

5. The only way to ensure recomputability is to provide virtual machines

Controversial point (1)

- Focus on VMs raises a lot of issues
  - bandwidth
  - storage
  - overkill
  - long term persistence...
- I just don’t think there’s an alternative
  - SHRDLU example

Key Response

- It should NOT (always) be necessary for
  - original experimenters to upload VMs
  - recomputers to download them
- (Almost) Dominates other methods
  - If you've got another way of storing experiments...
    - we should be able to run them in a VM anyway
- The more ways of recomputing experiments the better

The Recomputation Manifesto

6. Run time performance is a secondary issue

Controversial point (2)

- What if runtime performance is THE point
  - e.g. finding fastest way to search tree in practice?
- Completely accept this can be true
- Again puts a lot of people off straight away

Key Response

- Trivial response
  - if we can't recompute it all we can't reproduce the times
- Important response
  - if run times are different in different environments
  - this is important and interesting
  - recomputation might bring these things to light

Conceptual Challenges...

- What is an experiment?
- What is a recomputation of an experiment?
- How do we know an experiment is recomputable?
- Can we partially recompute an experiment?

Beyond Recomputation

- Danger you could get the wrong impression
- Replication of experiments is critical
- But there's a lot more than that...
- We should be able to do *much better* science
- Imagine trying to program...
  - without source code control
  - without testing

Beyond Recomputation

- So it’s not just about going back in time
  - – just like git/mercurial isn’t
  - – just like testing isn’t
- Should go throughout experimental lifecycle
  - – just like testing, source code control...
- “Once you have it you can’t imagine how you lived without it”
  - – Lars Kotthoff, September 2013

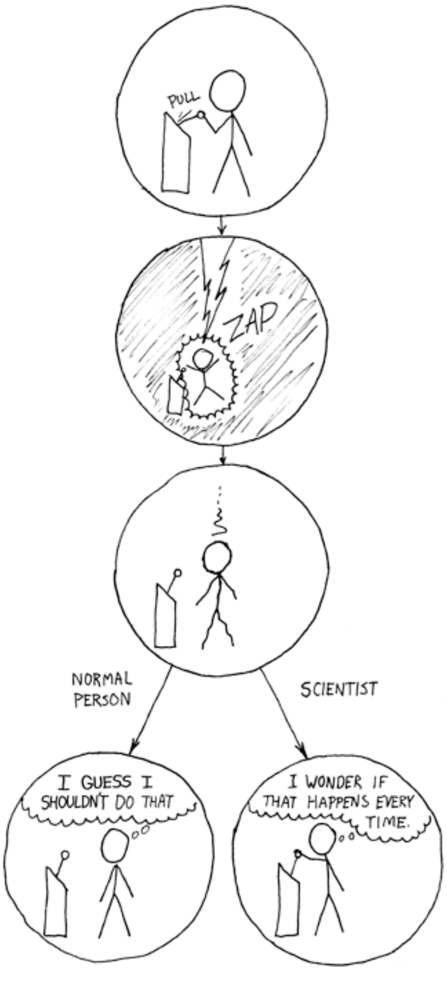
Any questions?

Replication and Recomputation in Scientific Experiments

Ian Gent, **Lars Kotthoff**

19 August 2014

When theory meets practice



PULL
ZAP
...
NORMAL PERSON
SCIENTIST
I GUESS I SHOULDN'T DO THAT
I WONDER IF THAT HAPPENS EVERY TIME.

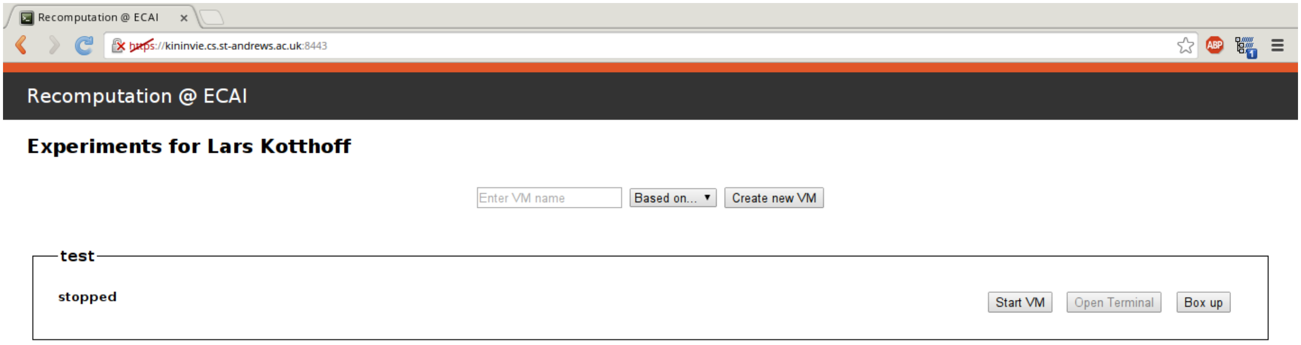
http://xkcd.com/242/

Recomputation @ ECAI

- invited authors of accepted papers to participate and make their experiments recomputable
- 14 expressions of interest
- 6 recomputable experiments available (work in progress)
- `http://recomputation.org/ecai2014/`

Recomputation @ ECAI

- ▷ we asked you to prepare your experiments to be runnable in a virtual machine using our interface



~~https~~://kininvie.cs.st-andrews.ac.uk:8443

Recomputation @ ECAI

Experiments for Lars Kotthoff

test

stopped

Demo

Challenges

- some authors have everything “ready to go”
- others use specific setup that’s hard to replicate (e.g. many manual steps)
- . . . and people in between
- main obstacle: summer time, holiday time!

Challenges: Getting it to work

- need to replicate environment experiments were run in
- experiments may rely on specific software versions
- these may not be available anymore
- excellent motivation for our approach

Challenges: Resources

“So, you will need approximately 50.000.000 seconds [1.6 years] of computation on 2.27 GHz processors, memory limit of 3.9 GB (timeout is 2h).”

Challenges: Resources

"It requires a GPU/CUDA card of a certain version. I don't know if virtual machines work."

Challenges: Resources

“[the script] delegates the theorem proving tasks to Geoff Sutcliffe’s automated theorem proving servers in Miami.”

Challenges: Resources

“My experiments are based on games that humans played with/against a computer-agent in lab conditions. In order to run these experiments, it’s needed to have a research assistant in the lab that will take care on the process of the experiment [. . .]. My experiments were done only by students from specific departments and mean age.”

Challenges: Resources

- experiments may require specific type/number of CPUs, specific amount of memory
- cannot guarantee this
- may be unable to recompute because of this

Challenges: Licenses

- some experiments use proprietary software such as IBM CPLEX
- some authors use other people's tools that they don't own
- cannot include in virtual machine image and distribute
- solution: allow user to provide relevant software on host machine

30 second Vagrant recomputation tutorial

Do try this at home

```
vagrant init myExperiment \
    http://files.vagrantup.com/precise64.box
vagrant up
vagrant ssh
[set up virtual machine and experiments]
vagrant halt
vagrant package --output myExperiment.box
```

This is all you need.

Do try this at home – run automatically

```
vagrant init myExperiment \
    http://files.vagrantup.com/precise64.box
vagrant up
vagrant ssh
[set up, single script recompute.sh to run everything]
vagrant halt
[edit Vagrantfile, add line
config.vm.provision :shell, :inline => "./recompute.sh"
before "end"]
vagrant package --vagrantfile Vagrantfile \
   --output myExperiment.box
```

ENLIST TODAY: CONTACT@RECOMPUTATION.ORG

TOGETHER WE CAN MAKE IT RECOMPUTABLE!

RECOMPUTATION.ORG