Open Computer Programs - More than just a methodology of reproducible research

Journal Club

Holger Schmuhl

Medfloss.org

May 28, 2013
Outline

1. Reproducible Research

2. Beyond Reproducibility

3. Evidence

4. Guide

5. Q & A
The case for open computer programs

Darrel C. Ince¹, Leslie Hatton² & John Graham-Cumming³

Scientific communication relies on evidence that cannot be entirely included in publications, but the rise of computational science has added a new layer of inaccessibility. Although it is now accepted that data should be made available on request, the current regulations regarding the availability of software are inconsistent. We argue that, with some exceptions, anything less than the release of source programs is intolerable for results that depend on computation. The vagaries of hardware, software and natural language will always ensure that exact reproducibility remains uncertain, but withholding code increases the chances that efforts to reproduce results will fail.

Available from: http://dx.doi.org/10.1038/nature10836
At present, debate rages on the need to release computer programs associated with scientific experiments, with policies still ranging from mandatory total release to the release only of natural language descriptions.

”Nature does not require authors to make code available, but we do expect a description detailed enough to allow others to write their own code to do a similar analysis.”

Stated policy on code availability actively hinders reproducibility.
Difficulty of reproducibility

- Idea: Reproduction of a scientific paper’s central finding, rather than exact replication of each specific numerical result.
- Without code, direct reproducibility is impossible . . . and raises needless, and needlessly confusing, roadblocks to reproducibility
- Open code would improve the chances of both direct and indirect reproducibility.
Problem of ambiguity
describing a computer program

- Without code a program can be described
  - Natural language
  - Natural language augmented my mathematics
  - Mathematically

- Can be converted in computer code in multiple ways, each of which may lead to different outcomes.

- Ambiguity of description can occur lexical, syntactic or semantic level.
Perfect description
implementation and execution issues

- Time consuming, require researchers to acquire skills that are only peripheral to their work.
- In case of the “perfect” description, there are still multiple sources of error:
  - Programming errors
  - Order-of-evaluation problem
  - Influence of runtime environment
Challenges
to open computer programs

- Classification of degree of source code accessibility (full, partial, marginal, no code)
- Intellectual property rights (patent, business model)
- Limited access (3rd party libraries)
- Procedure (submission)
- Logistics (hosting)
- Packaging (usage, presentation)
Journal and funding body strictures relating to code implementations of scientific ideas are now largely obsolete.

With some exceptions, anything less than the release of source programs is intolerable for results that depend on computation.

Aside of source code also a description of the hardware and software environment in which the program was executed and developed must be included.
1. Reproducible Research

2. Beyond Reproducibility

3. Evidence

4. Guide

5. Q & A
Cardona A, Tomancak P.
Current challenges in open-source bioimage informatics.
Available from: http://dx.doi.org/10.1038/nmeth.2082
Open vs. Closed Source
why open should be preferred

- It is not good scientific practice to press a button in a piece of software and interpret the results without understanding what the software does.
- Open source software provides the necessary transparency, giving scientists the opportunity to not only fully understand the computational methods but also to adapt and improve them, building on research of others in the best scientific tradition.
- Still time-tested closed source solutions can be applied in routine research. But due to transformative advances in modern biological research, new approaches are required.
Bringing scientists together
collaborative software development

- Computer Scientists
  - See SW engineering as non-scientific activity
  - Interest in underlying maths and algorithms

- Biologists
  - Serious programming expertise, no professional SW engineering practice
  - Just method to achieve their research goals

- Employee/PhD student turnover

Authors believe that an open source software framework is the viable approach to

- Fosters productive collaboration between computer-savvy biologists and bio-application-oriented computer scientists.

- Collect and maintain useful programs that can be reused and expanded by future generations of researchers.
Recognizing the value of contribution to research

- Motivation for developers to invest their time in implementing and distributing new algorithm solutions through open source platforms.
- The measuring stick of success in scientific circles is publication, and therefore software must be published to promote the scientific career of the researchers doing the software engineering work.

Suggestions:
- New type of journal focusing on practical implementations.
- Alternatively new popularity measurements like number of downloads or user ratings.
Support for open source projects

- Lack of sustainable funding.
- Common perception that open source software is developed by enthusiasts in their spare time and as such comes essentially for free.
- Funding agencies are not opposed to paying tens of thousands of dollars to buy commercial licenses for software packages that are rigid, opaque regarding their inner workings.
- Research funds serve scientific research better by paying for development and customization of open source software that benefits not only the funded research group but also all other groups with similar image-processing needs.
Outline

1. Reproducible Research
2. Beyond Reproducibility
3. Evidence
4. Guide
5. Q & A
In computational sciences such as image processing, publishing usually isn’t enough to allow other researchers to verify results. Often, supplementary materials such as source code and measurement data are required. Yet most researchers choose not to make their code available because of the extra time required to prepare it. Are such efforts actually worthwhile, though?

Vandewalle P.
First study
IEEE TIP, 2004-2006

- Analyzed journal: *IEEE Transactions on Image Processing (TIP)*
- Citation counts based on Google Scholar, similar effect but lower citation rates obtained with Web of Science; self-citations not discarded.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>All</th>
<th>2004</th>
<th>2005</th>
<th>2006</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of papers</td>
<td>645</td>
<td>134</td>
<td>182</td>
<td>329</td>
</tr>
<tr>
<td>Code available</td>
<td>66 (10%)</td>
<td>12 (9%)</td>
<td>19 (10%)</td>
<td>35 (11%)</td>
</tr>
<tr>
<td>Average citations (not RR)</td>
<td>41</td>
<td>62</td>
<td>45</td>
<td>31</td>
</tr>
<tr>
<td>Average citations (RR)</td>
<td>198</td>
<td>438</td>
<td>202</td>
<td>114</td>
</tr>
<tr>
<td>Median citations (not RR)</td>
<td>25</td>
<td>37</td>
<td>26</td>
<td>21</td>
</tr>
<tr>
<td>Median citations (RR)</td>
<td>76</td>
<td>88</td>
<td>111</td>
<td>67</td>
</tr>
<tr>
<td>Significance level</td>
<td>6.6 e−11</td>
<td>4.8 e−2</td>
<td>1.4 e−5</td>
<td>9.7 e−7</td>
</tr>
</tbody>
</table>

*TIP = IEEE Transactions on Image Processing. Papers with code available online are denoted as “RR” (all others are denoted as “not RR”).
Second study
IEEE TIP, TPAMI, and TSP, 2004-2008

- Analyzed journals: *IEEE Transactions on Image Processing (TIP)*, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, and *IEEE Transactions on Signal Processing (TSP)*

- Three most highly cited articles per year selected

---

Table 2. Summary of the second study: top-cited papers in high-profile journals (2004–2008).*

<table>
<thead>
<tr>
<th>Criteria</th>
<th>TIP</th>
<th>TPAMI</th>
<th>TSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of papers</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>Code available</td>
<td>13 (87%)</td>
<td>13 (87%)</td>
<td>2 (13%)</td>
</tr>
</tbody>
</table>

Contextualizing the results

First study

- Median number of citations for the papers increases with a factor of 3 when code is available online

Second study

- Best-cited papers 87 percent had code available online (compared to 10 percent as a global average)
- Difference between more theoretical and more applied papers, and different practices within research communities

General

- When was the data analyzed ???
- Provides only a snapshot at a specific moment in time
Outline

1 Reproducible Research
2 Beyond Reproducibility
3 Evidence
4 Guide
5 Q & A
Prlić A, Procter JB.
Ten simple rules for the open development of scientific software

1. Don’t Reinvent the Wheel
2. Code Well
3. Be Your Own User
4. Be Transparent
5. Be Simple
6. Don’t Be a Perfectionist
7. Nurture and Grow Your Community
8. Promote Your Project
9. Find Sponsors
10. Science Counts
1 Reproducible Research
2 Beyond Reproducibility
3 Evidence
4 Guide
5 Q & A
Thank you for your attention!