Review: "CoRR — The Cloud of Reproducible Records"

Andrew P. Davison[1]

[1]Computing in Science and Engineering

April 28, 2020

This is a review of manuscript CiSESI-2018-02-0016 submitted to Computing in Science & Engineering: "CoRR — The Cloud of Reproducible Records" (Congo, Traoré, Hill and Wheeler, 2018).

Overview

While a number of tools have been developed over recent years for run-time capture of computational provenance artefacts and metadata (referred to in the manuscript as "Computation Version Control" or CVC), such as Sumatra, CDE, Reprozip, recipy and noWorkflow, these tools generally lack an easy way to review and share provenance records through a web-based graphical interface, such as is provided by Github, Gitlab, Bitbucket, etc. for software version control systems such as Git or Mercurial. In some cases, no graphical interface is available at all. In others, the graphical interface runs only locally, and is not accessible over the web. Where web-based tools do exist (example) they are (i) difficult to deploy and (ii) specific to a given provenance capture tool: there is essentially no interoperability between such tools.

This manuscript reports on CoRR, a collaborative web platform that aims to solve the limitations outlined above by being easy to deploy (using a modern cloud architecture approach) and by providing a common graphical interface for different provenance capture tools. The manuscript outlines the case for CoRR, briefly reviews Sumatra, CDE and Reprozip, presents the CoRR architecture and interface, then presents a case study of using CoRR together with the three provenance capture tools previously reviewed.

CoRR clearly fills an important gap in the landscape of tools for reproducible computational research, and has the potential to expand the usage of run-time provenance capture in scientific computation, just as Github has done for usage of version control systems. The manuscript is a very good fit for the Reproducible Research track of CiSE (Barba *et al.*, 2017).

As such, the manuscript should be **accepted after minor revisions**.

Suggestions for improving the manuscript.

Terminology

While almost everyone agrees on the need to distinguish "repeatable", "reproducible" and "replicable", there is considerable disagreement over what each term refers to (Plesser, 2018). While your use of terms seem to be largely consistent with one of the more widely used conventions, to minimize confusion it would be desirable to cite a source for your definitions (e.g. Fabien *et al.*, 2018, although there are many others).

The term "computational provenance" seems to be well established for what is called "CVC" in the manuscript. Rather than coining a new term, I suggest reusing the existing one. This will better integrate the

article into the wider literature on this topic, and make it easier to find through search engines.

Structure

Overall, I think the manuscript is well structured. However, I think there is a lack of balance between the sections:

- The section "The main elements views in CoRR" seems unecessarily detailed, describing toolbox actions with a level of detail appropriate for a user manual, but not for a CiSE article.
- The manuscript has almost no information on the REST API. While it would not be appropriate to give detailed documentation, some examples of the REST endpoints and the document format(s) used would be helpful.
- The machine learning examples used in the case study are described in considerable detail, but this information is then hardly used in the rest of the manuscript; I think this session could be considerably shortened.
- The "Results" section is very minimal; I would expect at least to see some screenshots of how the different records are represented in CoRR. In addition, the "Reproducibility Effort" is not well explained; the phrase "is the tool representation directly reproducible after a download from CoRR" does not give me a clear idea of what steps are involved.

Inaccuracies and missing references

- *"Sumatra (created in 2009), CDE (created in 2010) and ReproZip (created in 2013) are some of the most used CVC tools."* What is the source for the assertion that these are among the most used tools? Do you have any numbers for this?
- There is a web-accessible API and web UI for Sumatra, "sumatra-server", separate from Sumatra's internal, local web-browser UI. This does support dissemination of records and multiple users. However, it is not widely used and is not interoperable with other provenance capture tools.
- It is not true that *"Sumatra makes copies of the output files only if placed in a folder named Data."* Sumatra does not make copies by default (only if the "archive" option is used), rather it stores the file system path or URL together with a hash of the file contents. It is also not necessary to use a specific folder name, this is fully configurable ("Data" is just the default).

Language/style

There are frequent minor grammatical errors and some typos (e.g. "ReroZip", "CORRR"). I recommend that all co-authors carefully re-read the manuscript to correct these.

Style issues: - Please spell out "OS" in the introduction. From the context, it becomes clear that it means "operating system", but the abbreviation is also often use for "open source". - The sentence *"The uncovering of recent frauds, irreproducibility facts and concerns from major journals publications, have enforced new requirements in scientific results corroborations"* is hard to understand. What are *"irreproducibility facts and concerns"*? What are *"major journals"*? Who is doing the enforcing? - *"We refer to these software and services as CVC tools"*. The antecedent of this sentence is three sentences earlier, which makes it hard to understand what is being referred to. - It is not usual in a scientific context to explicitly use titles, as in "Dr Rosebrock". I think it would be more appropriate to use the name of the book and the author's surname the first time the book is cited, and then use just the surname ("Rosebrock") subsequently.

Discussion of existing literature

In the context of interoperability, which is one of the main benefits of CoRR, the authors should discuss previous efforts to enable interoperability of representations of computational provenance, such as the ProvONE

data model, which builds on the W3C PROV standard. Would it be possible to support PROV-compatible representations (i.e. some form of RDF) through the CoRR API?

Disclaimer

For the sake of transparency, it should be noted that I am the principal developer of Sumatra, one of the software tools reviewed in the manuscript and used in the case study.

References

Lorena A. Barba & George K. Thiruvathukal (2017). Reproducible Research for Computing in Science & Engineering. *Computing in Science & Engineering* **19**:85–87. http://doi.org/10.1109/mcse.2017.3971172

Plesser, H. E. (2018). Reproducibility vs. replicability: A brief history of a confused terminology. *Frontiers in Neuroinformatics* **11**:76. https://doi.org/10.3389/fninf.2017.00076

Fabien C. Y. Benureau, Nicolas P. Rougier (2018). Re-run Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions. *Frontiers in Neuroinformatics* **11**:69. https://doi.org/10.3389/fninf.2017.00069