The Journal of Open Source Software

Rclean: A Tool for Writing Cleaner, More Transparent Code

Matthew K. Lau[1], **Thomas F. J.-M. Pasquier**[2, 3], **Elizabeth Fong**[4], **and Margo Seltzer**[5]

1 Harvard Forest, Harvard University **2** Department of Computer Science, University of Bristol **3** School of Engineering and Applied Science, Harvard University **4** Department of Computer Sciences, Mount Holyoke College **5** Department of Computer Science, University of British Columbia

DOI: 10.21105/joss.01312

Software

- Review
- Repository
- Archive

Submitted: 07 February 2019
Published: 10 March 2019

Summary

The growth of open-source statistical software programming has been explosive in the last decade. In particular, the statistical programming language `R` has grown exponentially to become one of the top ten programming languages in use today. Recently, concerns have arisen over the reproducibility of scientific research (R. D. Peng et al., 2011 Baker (2016) Stodden, Seiler, & Ma (2018)) and the potential issues stemming from the complexity and fragility of analytical software (Pasquier et al., 2017 Chen et al. (2018)). There is now a recognition that simply making the code open is not enough, and that there is a need for improvements to documentation and transparency (Chen et al., 2018). From this perspective, tools that can lower the time and energy required to re-factor and streamline analytical scripts could have a significant impact on scientific reproducibility across all disciplines (Visser et al., 2015). Supporting this objective, we have created `Rclean` which automatically reduces a script to the parts that are specifically relevant to a research product, such as a blog, academic talk or research article.

The `Rclean` package provides a simple, easy to use tool for scientists conducting analyses in the R programming language. Using graph analytic algorithms, `Rclean` isolates the code necessary to produce a specified result (e.g., an object stored in memory or a table or figure written to disk). This process relies on the generation of data provenance (Carata et al., 2014), which is a formal representation of the execution of a computational process (https://www.w3.org/TR/prov-dm/), to rigorously determine the the unique computational pathway from inputs to results. However, as the intended user is a researcher conducting analyses, the process is abstracted and only the minimum information is required and presented to the user to streamline the process of creating "cleaner" code. The output generated by `RClean` is the minimum and sufficient code needed to generate the chosen result.

As statistical programming becomes more common across the sciences, tools that make the production of accessible code will be an important aid for improving scientific reproducibility. `Rclean` has been designed to take advantage of recent advances in data provenance capture techniques to create a minimalistic tool for this purpose. New users can easily install the package from the Comprehensive R Archive Network (CRAN) (Lau, 2018). The package is open-source and welcomes contributions. For example, the existing framework could be extended to support new provenance capture methods, and there is tremendous potential for the use of code cleaning in the creation of more robust capsules (Pasquier et al., 2018). Interested contributors can connect to the project at https://github.com/provtools/Rclean. The project is also tagged and curated at Zenodo (DOI: 10.5281/zenodo.1208640).

Citations

Acknowledgments

This work was improved by discussions with ecologists at Harvard Forest. Much of the work was funded by US National Science Foundation grant SSI-1450277 for applications of End-to-End Data Provenance.

References

Baker, M. (2016). 1,500 scientists lift the lid on reproducibility. *Nature*, *533*, 452–454. doi:10.1038/533452a

Carata, L., Akoush, S., Balakrishnan, N., Bytheway, T., Sohan, R., Seltzer, M., & Hopper, A. (2014). A Primer on Provenance. *Queue*, *12*(3), 10–23. doi:10.1145/2602649.2602651

Chen, X., Dallmeier-Tiessen, S., Dasler, R., Feger, S., Fokianos, P., Gonzalez, J. B., Hirvonsalo, H., et al. (2018). Open is not enough. *Nat. Phys.*, 1. doi:10.1038/s41567-018-0342-2

Lau, M. K. (2018). Rclean: A Tool for Writing Cleaner, more Transparent Code. *Compr. R Arch. Netw.* Retrieved from https://cran.r-project.org/package=Rclean

Pasquier, T., Lau, M. K., Han, X., Fong, E., Lerner, B. S., Boose, E. R., Crosas, M., et al. (2018). Sharing and Preserving Computational Analyses for Posterity with encapsulator. *Comput. Sci. Eng.*, *20*(4), 111–124. doi:10.1109/MCSE.2018.042781334

Pasquier, T., Lau, M. K., Trisovic, A., Boose, E. R., Couturier, B., Crosas, M., Ellison, A. M., et al. (2017). If these data could talk. *Sci. Data*, *4*, 170114. doi:10.1038/sdata.2017.114

Peng, R. D., Hanson, B., Sugden, A., Alberts, B., Peng, R. D., Dominici, F., Zeger, S. L., et al. (2011). Reproducible research in computational science. *Science*, *334*(6060), 1226–7. doi:10.1126/science.1213847

Stodden, V., Seiler, J., & Ma, Z. (2018). An empirical analysis of journal policy effectiveness for computational reproducibility. *Proc. Natl. Acad. Sci. U. S. A.*, *115*(11), 2584–2589. doi:10.1073/pnas.1708290115

Visser, M. D., McMahon, S. M., Merow, C., Dixon, P. M., Record, S., & Jongejans, E. (2015). Speeding Up Ecological and Evolutionary Computations in R; Essentials of High Performance Computing for Biologists. (F. Ouellette, Ed.)*PLOS Comput. Biol.*, *11*(3), e1004140. doi:10.1371/journal.pcbi.1004140