Pacific Symposium on Biocomputing 27:412-416(2022)

Packaging Biocomputing Software to Maximize Distribution and Reuse

William S. Bush, Nicholas Wheeler
Cleveland Institute for Computational Biology, Department of Population and Quantitative Health Sciences, Case Western Reserve University
Cleveland, OH, 44106, USA
Email: wsb36@case.edu; nrw16@case.edu

Christian Darabos
Biomedical Data Science Department, Information Technology and Consulting, Dartmouth College
Hanover, NH, 03755, USA
Email: Christian.Darabos@dartmouth.edu

Brett Beaulieu-Jones
Department of Biomedical Informatics, Harvard Medical School,
Boston, MA, 02115, USA
Email: Brett_Beaulieu-Jones@hms.harvard.edu

The majority of publications in computational biology and biocomputing develop or apply software approaches to relevant biological problems to some degree. While journals and conferences often prompt authors to make their source code available, these are often only basic requirements. Investigators often wish their software and tools were widely usable to the scientific community, but there are limited resources available to maximize the distribution and provide easy use of developed software. Even when authors adhere to standards of source code availability, the growing problem of system configuration issues, language and library version conflicts, and other implementation issues often impede the broad distribution, availability of software tools, and reproducibility of research. There are a variety of solutions to these implementation issues, but the learning curve for applying these solutions can be steep. This tutorial demonstrates tools and approaches for packaging and distribution of published code, and provides methodological practices for the broad and open sharing of new biocomputing software.

1. Rationale for Tutorial

A cornerstone of biocomputing and computational biology is the release of new algorithms for data analysis, often in the form of an author-developed software implementation. With the ever-increasing need for algorithmic processing of experimental data in scientific studies, the reproducibility of individual studies has declined (Baker and Penny 2016; Monya and Dan 2016). The lack of reproducibility and open sharing of methods has had downstream impacts into more expensive clinical research, leading to an estimated $200 billion of wasted research funds (Chalmers and Glasziou 2009). Despite improvements in certain aspects of reproducibility in recent years (Wallach, Boyack, and Ioannidis 2018), there are still opportunities for improvement. In their Ten simple rules for reproducible computational research, Sandve and colleagues enumerate the need for archiving exact versions of external programs, version controlling all custom scripts, storing

intermediate data and raw output, and providing public access to scripts, runs and results(Sandve et al. 2013).

The traditionally accepted approach for standardization, distribution, and version control of software is the use of package repositories. The Comprehensive R Archive Network (CRAN) is an extensively mirrored repository of distributions, extensions, and documentation for the R statistical package (Hornik 2018). Similarly, Bioconductor serves as an extension of the R environment for computational biology and bioinformatics packages (Gentleman et al. 2004). These are both reminders that there is an "R" in "reproducible" (Ochs 2020), and that R packages may serve as a useful framework for managing and organizing research projects (Vuorre and Crump 2020). Analogs of these repositories in the *conda* framework have also been developed for the Python language (Dale et al. 2018), and custom software and version control is now routinely stored and managed using Git and GitHub (Chacon and Straub 2014).

While package management systems have dramatically improved version control and accessibility of software, duplicating the precise software environment used to process experimental data in a publication has long remained a major challenge, as reviewed in a recent challenge to run ten-year-old code (Perkel 2020). Within the last few years, the dramatic rise of containerization technologies like Docker (Merkel 2014) have for the first time allowed seamless distribution of data, software, and its native processing environment together as a single entity. As a result, Docker is now a commonly used tool for reproducible research in multiple fields (Boettiger 2014; Cito, Jurgen; Gall 2016; Wiebels and Moreau 2021). Containerization technology has been adapted for bioinformatics tasks (Belmann et al. 2015), deployed into custom bioinformatics registries (Moreews et al. 2015), and specifically adapted to high-performance computing environments (Kurtzer, Sochat, and Bauer 2017). Containers have been especially useful in the distribution of complex workflows with dependencies on multiple software tools, such as the processing of next-generation sequencing data (Kim et al. 2017; Schulz et al. 2016). The BioContainers Community has produced a list of recommendations for standardizing bioinformatics packages and containers (Gruening et al. 2019).

Even with software version control and entire software environments available for download, specific analysis steps within a publication may not be documented with sufficient detail to reproduce an analysis. While package management systems have dramatically improved version control and accessibility of software, and containerization allows duplication of the precise software environment, the exact process for analyzing experimental data may still prove difficult to reproduce without detailed documentation. To address these challenges, Jupyter notebooks have emerged as a composite digital document that seamlessly blends code (from a variety of languages), documentation, and data visualization in an easy-to-follow format (Kluyver et al. 2016; Perkel 2021). They have been specifically touted for improving research reproducibility (Beg et al. 2021; Rule et al. 2019), and Jupyter notebooks themselves have been researched to identify practices that can improve reproducibility (Pimentel et al. 2021). Jupyter notebooks have gained popularity in other computation-heavy fields like astronomy (Wofford et al. 2019), however their stability and accessibility is not always persistent after publication. While there are also repositories for storing Jupyter notebooks, specific practices are needed to ensure long-term availability of accessed documents (Bouquin et al. 2018).

In a second iteration of this tutorial, we outline a technology stack that ensures high availability and easy distribution of software, encapsulated data, software environment, and analysis approaches. Docker containers are proposed as a foundational layer, providing a stable, version-controlled operating system along with its associated programming languages and packages, and data files that can be cached within the environment. R and Python packages are the distribution method for custom software implementations, and are accessible within distributable containers. Jupyter notebooks provide detailed documentation of all analysis steps in an interactive fashion. Altogether, the collection of approaches outlined in this tutorial will ensure maximal distribution, ease of use, and reproducibility of biocomputing research projects (Beaulieu-Jones and Greene 2017). In recent years, methods built upon this process have emerged to reduce technical hurdles and fit specific domains (Krafczyk et al. 2021; Kwon, Kim, and Ahn 2018; Nüst et al. 2020; Peikert and Brandmaier 2021; Sheffield 2019; Yenni et al. 2019).

2. Tutorial Speakers

William S. Bush, Ph.D. is an Associate Professor in the Department of Epidemiology and Biostatistics and Assistant Director for Computational Methods in the Cleveland Institute for Computational Biology at Case Western Reserve University. Dr. Bush received his Ph.D. at Vanderbilt University in Human Genetics in 2008 and then continued as a post-doctoral fellow in the Neurogenomics Training Program at Vanderbilt. As a human geneticist and bioinformatician, Dr. Bush's research interests include understanding the functional impact of genetic variation, developing statistical and bioinformatics approaches for integrating functional genomics knowledge into genetic analysis, and the use of electronic medical records for translational research.

Nicholas Wheeler, Ph.D. is a Research Scientist in the Cleveland Institute for Computational Biology at Case Western Reserve University. Dr. Wheeler is a macromolecular scientist and engineer by training with extensive expertise in the use of "big data" technologies for large scale data aggregation and analysis. Dr. Wheeler manages genomic datasets and their associated meta-data within a Spark/Hadoop cluster, with extensions to the open-source HAIL platform for genomic analysis, which ensures standardization and reproducibility of experimental analyses. Over the course of his career, Dr. Wheeler has created, validated, and submitted multiple R and Python packages into public repositories.

Brett Beaulieu-Jones, Ph.D. is an Instructor of Biomedical Informatics in the Kohane lab at Harvard University. He received his PhD from the Perelman School of Medicine at the University of Pennsylvania under the supervision of Dr. Jason Moore and Dr. Casey Greene. Dr. Beaulieu-Jones' doctoral research focused on using machine learning-based methods to more precisely define phenotypes from large-scale biomedical data repositories, e.g. those contained in clinical records. He is currently performing large-scale data integration (genomic, therapeutic, imaging) to both better understand the etiology of complex diseases and conditions.

Christian Darabos, Ph.D. is an Instructor in Quantitative Biomedical Sciences at the Geisel School of Medicine and the Interim Sr. Director for Research Computing at Dartmouth College. He co-leads the Reproducible Research initiatives at Dartmouth College and supports a series of workshops and tutorials which are designed to educate and support the entire research community on best computational and data practices, informatics and analytics tools, and high-performance computing.

3. Acknowledgements

This work is partially supported by the National Institute on Aging of the National Institutes of Health (NIH) Grants U01 AG058654 and U54 AG052427.

4. References

Baker, Monya, and Dan Penny. 2016. "Is There a Reproducibility Crisis?" *Nature* 533(7604): 452–54.

Beaulieu-Jones, Brett, and Casey Greene. 2017. "Reproducibility of Computational Workflows Is Automated Using Continuous Analysis." *Nature biotechnology* 35(4): 342–46. https://pubmed.ncbi.nlm.nih.gov/28288103/ (October 4, 2021).

Beg, Marijan et al. 2021. "Using Jupyter for Reproducible Scientific Workflows." *Computing in Science & Engineering* 23(02): 36–46.

Belmann, Peter et al. 2015. "Bioboxes: Standardised Containers for Interchangeable Bioinformatics Software." *GigaScience* 4(1).

Boettiger, Carl. 2014. "An Introduction to Docker for Reproducible Research, with Examples from the R Environment." http://arxiv.org/abs/1410.0846 (October 4, 2021).

Bouquin, Daina, Sophie Hou, Matthew Benzing, and Lee Wilson. 2018. *Jupyter Notebooks: A Primer for Data Curators Link w/ Release Notes*. http://datacurationnetwork.org.

Chacon, Scott, and Ben Straub. 2014. *Pro Git*. 2nd ed. Berkely, CA, USA: Apress.

Chalmers, Iain, and Paul Glasziou. 2009. "Avoidable Waste in the Production and Reporting of Research Evidence." *Lancet (London, England)* 374(9683): 86–89. http://www.ncbi.nlm.nih.gov/pubmed/19525005 (October 7, 2019).

Cito, Jurgen; Gall, Harald C. 2016. "Using Docker Containers to Improve Reproducibility in Software Engineering Research | IEEE Conference Publication | IEEE Xplore." *IEEE/ACM 28th International Conference on Software Engineering Companion (ICSE-C)*: 906–7. https://ieeexplore.ieee.org/document/7883438 (October 4, 2021).

Dale, Ryan et al. 2018. "Bioconda: Sustainable and Comprehensive Software Distribution for the Life Sciences." *Nature Methods* 15(7): 475–76.

Gentleman, Robert C et al. 2004. "Bioconductor: Open Software Development for Computational Biology and Bioinformatics." *Genome biology* 5(10): R80. http://www.ncbi.nlm.nih.gov/pubmed/15461798 (October 7, 2019).

Gruening, Bjorn et al. 2019. "Recommendations for the Packaging and Containerizing of Bioinformatics Software." *F1000Research* 7: 742.

Hornik, Kurt. 2018. "R FAQ." https://cran.r-project.org/doc/FAQ/R-FAQ.html.

Kim, Baekdoo et al. 2017. "Bio-Docklets: Virtualization Containers for Single-Step Execution of NGS Pipelines." *GigaScience* 6(8).

Kluyver, Thomas et al. 2016. "Jupyter Notebooks-a Publishing Format for Reproducible Computational Workflows." https://nbviewer.jupyter.org/,.

Krafczyk, M. S. et al. 2021. "Learning from Reproducing Computational Results: Introducing Three Principles and the Reproduction Package." *Philosophical Transactions of the Royal Society A* 379(2197). https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2020.0069 (October 4, 2021).

Kurtzer, Gregory M., Vanessa Sochat, and Michael W. Bauer. 2017. "Singularity: Scientific Containers for Mobility

of Compute." *PLoS ONE* 12(5).
Kwon, ChangHyuk, Jason Kim, and Jaegyoon Ahn. 2018. "DockerBIO: Web Application for Efficient Use of Bioinformatics Docker Images." *PeerJ* 6(11). /pmc/articles/PMC6266945/ (October 4, 2021).
Merkel, Dirk. 2014. "Docker: Lightweight Linux Containers for Consistent Development and Deployment." *Linux J.* 2014(239). http://dl.acm.org/citation.cfm?id=2600239.2600241.
Monya, Baker, and Penny Dan. 2016. "Reproducibility Crisis (Nature)." *Nature* 533: 452–54.
Moreews, François et al. 2015. "BioShaDock: A Community Driven Bioinformatics Shared Docker-Based Tools Registry." *F1000Research* 4.
Nüst, Daniel et al. 2020. "Ten Simple Rules for Writing Dockerfiles for Reproducible Data Science." *PLoS Computational Biology* 16(11). /pmc/articles/PMC7654784/ (October 4, 2021).
Ochs, Andreas. 2020. "There Is an R in Reproducible. Make Your next R Project Reproducible… | by Dr Andreas Ochs | Towards Data Science." https://towardsdatascience.com/there-is-an-r-in-reproducibility-b9120712742f (October 4, 2021).
Peikert, Aaron, and Andreas M. Brandmaier. 2021. "A Reproducible Data Analysis Workflow." *Quantitative and Computational Methods in Behavioral Sciences* 1.
Perkel, Jeffrey M. 2020. "Challenge to Scientists: Does Your Ten-Year-Old Code Still Run?" *Nature* 584(7822): 656–58.
Perkel, Jeffery M. 2021. "Reactive, Reproducible, Collaborative: Computational Notebooks Evolve." *Nature* 593(7857): 156–57.
Pimentel, João Felipe, Leonardo Murta, Vanessa Braganholo, and Juliana Freire. 2021. "Understanding and Improving the Quality and Reproducibility of Jupyter Notebooks." *Empirical Software Engineering 2021 26:4* 26(4): 1–55. https://link.springer.com/article/10.1007/s10664-021-09961-9 (October 4, 2021).
Rule, Adam et al. 2019. "Ten Simple Rules for Writing and Sharing Computational Analyses in Jupyter Notebooks." *PLOS Computational Biology* 15(7): e1007007. https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1007007 (October 4, 2021).
Sandve, Geir Kjetil, Anton Nekrutenko, James Taylor, and Eivind Hovig. 2013. "Ten Simple Rules for Reproducible Computational Research." *PLoS computational biology* 9(10): e1003285. http://www.ncbi.nlm.nih.gov/pubmed/24204232 (October 7, 2019).
Schulz, Wade L., Thomas J.S. Durant, Alexa J. Siddon, and Richard Torres. 2016. "Use of Application Containers and Workflows for Genomic Data Analysis." *Journal of Pathology Informatics* 7(1).
Sheffield, Nathan C. 2019. "Bulker: A Multi-Container Environment Manager." https://osf.io/natsj/ (October 4, 2021).
Vuorre, Matti, and Matthew J. C. Crump. 2020. "Sharing and Organizing Research Products as R Packages." *Behavior Research Methods 2020 53:2* 53(2): 792–802. https://link.springer.com/article/10.3758/s13428-020-01436-x (October 4, 2021).
Wallach, Joshua D., Kevin W. Boyack, and John P. A. Ioannidis. 2018. "Reproducible Research Practices, Transparency, and Open Access Data in the Biomedical Literature, 2015–2017." *PLOS Biology* 16(11): e2006930. https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.2006930 (October 4, 2021).
Wiebels, Kristina, and David Moreau. 2021. "Leveraging Containers for Reproducible Psychological Research:" *https://doi.org/10.1177/25152459211017853* 4(2): 1–18. https://journals.sagepub.com/doi/full/10.1177/25152459211017853 (October 4, 2021).
Wofford, Morgan et al. 2019. "Jupyter Notebooks as Discovery Mechanisms for Open Science: Citation Practices in the Astronomy Community." *Computing in Science & Engineering*: 1–1.
Yenni, Glenda M. et al. 2019. "Developing a Modern Data Workflow for Regularly Updated Data." *PLOS Biology* 17(1): e3000125. https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.3000125 (October 4, 2021).