Contribution ID: **383** Type: **Poster Presentation**

An approach and tools for research reproducibility across computational biology

Monday, 9 July 2018 19:45 (15 minutes)

Reproducibility of results is a key part of the scientific method; in general, scientific communication aims to describe a result in enough detail that readers and reviewers are able to contextualise the result within their own knowledge, and to reproduce it themselves given the appropriate skills and resources. In the field of computational biology, readers and reviewers face special challenges to contextualise and reproduce results, because of the size of datasets analysed, and the increasing complexity of the computational methods used. To meet these challenges, we present an approach that uses software engineering tools to produce complete 'reference computation environments', containing all software and configuration necessary to reproduce a computational result. Existing tools to reproduce results are limited to a single technology or programming language; our approach is the first to use a single specification to produce the same environment on a desktop computer, in a cloud computing service, or in a managed computing facility. Using these tools, authors can ensure their published results are easily reproducible by readers and reviewers, and are robust to future changes in software and hardware.

The drive for reproducibility in the computational sciences has provoked discussion and effort across a broad range of perspectives: technological, legislative/policy-making, education, and publishing. Nature Biotechnology has recently implemented standards for reproducibility [1] in response to increasing complexity of methods and data requirements in computational biology. Discussion on these topics is not new [2,3,4], but the need to adopt standards for reproducibility of claims made based on computational results is now clear to researchers, publishers and policymakers. Many technologies exist to support and promote reproduction of results: this journal and others have discussed containerisation tools like Docker [5], literate programming approaches such as Sweave [6], knitr [7], iPython [8] or cloud environments like Amazon Web Services [9]. But these technologies are tied to specific programming languages (e.g. Sweave/knitr to R; iPython to Python) or to platforms (e.g. Docker for 64-bit Linux environments only). No single approach to date is able to span the broad range of technologies and platforms represented in computational biology and biotechnology.

To enable reproducibility across computational biology, we demonstrate for the first time an approach and a set of tools that is suitable for all computational work, and not tied to a particular programming language or platform. Our approach extends our previous work in this area [10] to produce 'reference environments' for reproducing results that can be deployed across platforms and can use any and all of the replication tools described above. We achieve this by adopting innovative open-source tools from software engineering [13,14], and we are now involved in the community contributing to the development of these tools, and acting as advocates for scientific computing within that community. We present published examples from a series of papers across research groups in different areas of computational and systems biology, spanning the major languages and technologies in the field (Python/R/MATLAB/Fortran/C/Java). We also include examples reproducing studies in network analysis by Feizi *et al.* [11] and Barzel *et al.* [12], results which have been recently commented upon in this journal [1]. We also present results from another study demonstrating that different output values can be produced from the same data and code given different versions of the MATLAB software used by the Feizi and Barabasi codebases. Since very few reviewers and readers will run the same code using multiple software versions, this could be interpreted as inability to reproduce the published work, underlining the need for a reference environment as produced by our approach. Our approach produces a transparent and flexible process for replication and recomputation of results, but ultimately its most valuable aspect is the decoupling of methods in computational biology from their implementation. Separating the 'how' (method) of a publication from the 'where' (implementation) promotes genuinely open science and benefits the scientific community as a whole.

[1] Editorial. *Nat Biotech.* Nature Publishing Group, a division of Macmillan Publishers Limited. All Rights Reserved.; 2015;33: 319.

[2] Barnes N. *Nature.* 2010;467: 753. doi:10.1038/467753a
[3] Merali Z. *Nature.* Nature Publishing Group; 2010;467: 775–777.
[4] Ball P. *Nature.* 2003;
[5] Docker. [cited 8 May 2015]. Available: https://www.docker.com/
[6] Leisch F. Hardle W, Ronz B, editors. Compstat 2002: Proceedings in Computational Statistics. Heidelberg: Physica-Verlag Gmbh & Co; 2002.
[7] Xie Y. Implementing Reproducible Research. Chapman and Hall/CRC; 2014. doi:doi:10.1201/b16868-3
[8] Pérez F *et al. Comput Sci Eng.* 2007;9: 21–29. doi:10.1109/MCSE.2007.53
[9] "Amazon Web Services." [cited 8 May 2015]. Available: http://aws.amazon.com/
[10] Hurley DG *et al. Brief Bioinform.* Oxford University Press; 2014; bbu043. doi:10.1093/bib/bbu043
[11] Feizi S *et al. Nat Biotechnol.* 2013;31: 726–33. doi:10.1038/nbt.2635
[12] Barzel B *et al. Nat Biotechnol.* NATURE PUBLISHING GROUP, 75 VARICK ST, 9TH FLR, NEW YORK, NY 10013-1917 USA; 2013;31: 720–5. doi:10.1038/nbt.2601
[13] Vagrant. [cited 8 May 2015]. Available: https://www.vagrantup.com/
[14] Packer. [cited 11 Mar 2015]. Available: packer.io/

Primary author: Dr HURLEY, Daniel G (Systems Biology Laboratory, University of Melbourne)

Co-author: Prof. CRAMPIN, Edmund J (Systems Biology Laboratory)

Presenter: Dr HURLEY, Daniel G (Systems Biology Laboratory, University of Melbourne)

Session Classification: Poster Session

Track Classification: Techniques for Mathematical Biology