Reproducible Computational Workflows with Continuous Analysis

Brett K Beaulieu-Jones[1], Casey S Greene[2]

[1] Graduate Group in Genomics and Computational Biology, Computational Genetics Lab, Institute for Biomedical Informatics, University of Pennsylvania. Email: brettbe@med.upenn.edu

[2] Department of Systems Pharmacology and Translational Therapeutics, Institute for Biomedical Informatics, Institute for Translational Medicine and Therapeutics, University of Pennsylvania

Source Code: https://github.com/greenelab/continuous_analysis
Example Project: https://github.com/greenelab/DAPS/ (Example Project)
License: BSD3

Reproducing experiments is vital to science. Being able to replicate, validate and extend previous work also speeds new research projects. Reproducing computational biology experiments, which are scripted, should be straightforward. But reproducing such work remains challenging and time consuming. In the ideal world we would be able to quickly and easily rewind to the precise computing environment where results were generated. We would then be able to reproduce the original analysis or perform new analyses. We introduce a process termed "continuous analysis" which provides inherent reproducibility to computational research at a minimal cost to the researcher. Continuous analysis combines Docker, a container service similar to virtual machines, with continuous integration, a popular software development technique, to automatically re-run computational analysis whenever relevant changes are made to the source code. This allows results to be reproduced quickly, accurately and without needing to contact the original authors. Continuous analysis also provides an audit trail for analyses that use data with sharing restrictions. This allows reviewers, editors, and readers to verify reproducibility without manually downloading and rerunning any code.