Developing and maintaining reproducible workflows for bioinformatics data which are platform independent

This manuscript (permalink) was automatically generated from utsw-bicf/manu-repro-workflows@2db4382 on November 13, 2019.

Authors

- **Venkat S Malldi**
  0000-0002-0144-0564 · vsmalladi · katatonikkat
  Department of Bioinformatics, University of Texas Southwestern Medical Center, Dallas, Texas, United States of America; Bioinformatics Core Facility, University of Texas Southwestern Medical Center, Dallas, Texas, United States of America · Funded by CPRIT RP150596

- **Holly Ruess**
  0000-0001-9148-6672 · HollyRuess
  Department of Bioinformatics, University of Texas Southwestern Medical Center, Dallas, Texas, United States of America; Bioinformatics Core Facility, University of Texas Southwestern Medical Center, Dallas, Texas, United States of America · Funded by CPRIT RP150596

Abstract

High-throughput sequencing technologies generate large amounts of data, which should be analyzed and processed using standard operating procedures (SOPs), with an emphasis on ensuring reproducibility. The bioinformatics analysis workflows for any type of data have varied computational requirements at each step, especially in regards to parallelization and memory, which must be accounted for in their design. In addition, the outputs often result in further massive quantities of data, including statistical analyses and potential biologically-significant pathway information. Therefore, a workflow system requires: (1) defined computational resource allocation; (2) parallelization across samples, and if possible within each step; (3) serial steps executed only if input criteria are met; (4) steps restarted in the case of failure; (5) workflow reproducibility, and (6) visualization of workflow output to aid researchers in understanding complex data. Git projects allow us to maintain version control and test new development with continuous integration, verifying that the pipeline functionality has remained constant between updates. Nextflow provides the features necessary to run workflows on a high-performance compute cluster using a scheduler (e.g. SLURM or SGE) allowing for parallel and serial jobs simultaneously, as well as handling fail states and resuming failed jobs. Singularity containers allow for the implementation of the workflow across various computing environment (e.g. local server, cluster, or cloud-computing platforms), while maintaining versions of programs within a given pipeline or step, and minimizing the effect of inter-platform differences. We will be discussing the challenges and successes of integrating the above components into making our BICF ChiP-seq Analysis Workflow (doi:10.5281/zenodo.2648844).

References