Applications Note

**Bioconductor** toolchain for reproducible bioinformatics pipelines using **Rcwl** and **RcwlPipelines**

Qiang Hu\(^1,*,\) Alan Hutson\(^1,\) Song Liu\(^1,\) Martin Morgan\(^1\) and Qian Liu\(^1,\)*

\(^1\)Department of Biostatistics & Bioinformatics, Roswell Park Comprehensive Cancer Center, Buffalo, NY.

*To whom correspondence should be addressed.

Associate Editor: Peter Robinson

Received on 12/15/2020; revised on 02/28/2021; accepted on 03/22/2021

**Abstract**

**Summary:** The Common Workflow Language (CWL) is used to provide portable and reproducible data analysis workflows across different tools and computing environments. We have developed **Rcwl**, an \(R\) interface to CWL, to provide easier development, use, and maintenance of CWL pipelines from within \(R\). We have also collected more than 100 pre-built tools and pipelines in **RcwlPipelines**, ready to be queried and used by researchers in their own analysis. A single-cell RNA sequencing preprocessing pipeline demonstrates use of the software.

**Availability:** Project website: https://rcwl.org (Rcwl: https://bioconductor.org/packages/Rcwl; RcwlPipelines: https://bioconductor.org/packages/RcwlPipelines)

**Contact:** Qian.Liu@RoswellPark.org; Qiang.Hu@RoswellPark.org.

**Supplementary Information:** Supplementary data are available at Bioinformatics online.

1 Introduction

The bioinformatics community increasingly relies on ‘workflow’ frameworks to manage large and complex biomedical data (Köster and Rahmann, 2012; Di Tommaso et al., 2017). One solution facilitating portable, reproducible, and scalable workflows across a variety of software and hardware environments is the Common Workflow Language (CWL) (Amstutz et al., 2016). The CWL has been widely adopted by large biomedical projects such as The Cancer Genome Atlas (TCGA) (Weinstein et al., 2013) and Galaxy (Afgan et al., 2018). However, as a domain-specific language, the implementation of CWL requires a level of expertise that is often beyond the capabilities of wet-lab researchers and even skilled data scientists. In addition, the impact of CWL pipelines is weakened by poor integration with downstream statistical analysis tools such as \(R\) and Bioconductor (Huber et al., 2015; Amezquita et al., 2020).

Here we introduce a Bioconductor toolchain for use and development of reproducible bioinformatics pipelines in CWL using **Rcwl** and **RcwlPipelines**. **Rcwl** provides a familiar \(R\) interface to, and expands the scope of, CWL. **Rcwl** enables best practices and standardized data flow between different tools, and promotes modularization for easy sharing of established pipelines or critical steps. **RcwlPipelines** manages a collection of commonly used bioinformatics tools and pipeline recipes based on **Rcwl**. **RcwlPipelines** develops a community-driven platform for open source, open development, and open review of best-practice CWL bioinformatics pipelines. **Rcwl** and **RcwlPipelines** reduces the learning curve required to apply findable, accessible, interoperable, and reusable (FAIR) principles to the analysis of multi-omics biological experiments, and to promote community-wide sharing of cloud-ready bioinformatics workflows.

2 Implementation

**Rcwl** is implemented using the \(R\) programming language and Bioconductor S4 data infrastructure, aiming to facilitate data analysts’ ability to build, read, write and execute diverse computational tools and pipelines using the CWL specification. **Rcwl** defines three major categories of functions: composition, visualization and execution (Fig. 1) to construct CWL pipelines, execute them in various computing environments (e.g., on HPC, cloud or through Shiny application), and visualize the pipeline for reporting purposes. Configuration parameters,
Fig. 1. Rcwl & RcwlPipelines functionalities and implementation.

docker requirements (Grüning et al., 2018; da Veiga Leprevost et al., 2017), and job schedulers (e.g., SGE, SLURM) are all managed in R for enhanced reproducibility and scalability. The tools and pipelines developed by Rcwl are readily submitted and executed through R as CWL scripts. Beyond just an interface to CWL, for wrapping command-line tools, Rcwl expands the scope of CWL by enabling direct integration of user-defined R functions to use, e.g., established R/Bioconductor package functions, as components of the pipeline. RcwlPipelines manages a collection of CWL tools and pipelines developed using Rcwl (there are 113 tools and 26 pipelines available at time of writing). RcwlPipelines covers a broad range of bioinformatics areas and provides highly customizable pipelines. For example, the neoantigen prediction pipeline integrates tools for DNA sequence alignment, RNA read quantification, variant calling, annotation, phasing, and HLA typing. A set of RcwlPipelines core functions (Fig. 1) helps users to search and load the best-fit tools for their own data analysis. Expanded examples of code for use of the core functions are provided as Supplementary Material and as Bioconductor package vignettes.

3 Usage

Development and use involve three steps: 1) (developers only) build the CWL tools by specifying the base command, input/output parameters, and docker requirements. 2) Connect Rcwl tools into pipelines by specifying data flow between steps (using the ‘+’ function). 3) Execute the pipeline by assigning values to input parameters and specifying the file path to collect the final results. Jobs can be run on single (runCWL) or multiple (runCWLBatch) computing nodes through the internal integration of Bioconductor (Morgan et al., 2020). An extended example is given in the Supplementary Material, where we present a modified version of the pre-processing pipeline of Single-Cell RNA Sequencing Data (Tekman et al.) using STARsolo for alignment and quantification, and DropletUtils (Lun et al., 2019) for filtering empty droplets from the raw gene-barcode matrix.

4 Conclusion

Here we introduce a Bioconductor toolchain for simplified development, usage, and maintenance of CWL bioinformatics pipelines using Rcwl and RcwlPipelines. The packages provide a route to turn any command-line tools into R tools as well as integrating existing R package functionalities into CWL pipelines, and thus greatly lowered the bar for data analysts to conduct complex bioinformatics workflows directly in R. We are dedicated to implementing more user-oriented core functions and infrastructures, and expanding the current collection of tools and pipelines available to meet wider biomedical research needs. In the future, we will be implementing more specific developing guidelines and benchmark datasets to enhance the reusability and high-quality of pipelines contributed by the community.

Acknowledgements

The research reported in this publication was supported by the National Center for Advancing Translational Sciences of the National Institutes of Health (UL1TR001052-05), National Cancer Institute (U24CA232979, U24180996), and National Human Genome Research Institute (U41HG004059). The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

Conflict of Interest: none declared.

References

Afgan,E. et al. (2018) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Res., 46, W537–W544.
Amezquita,R.A. et al. (2020) Orchestrating single-cell analysis with Bioconductor. Nat. Methods, 17, 137–145.
Amstutz,P. et al. (2016) Common Workflow Language, v1.0.
Dí Tommaso,P. et al. (2017) Nextflow enables reproducible computational workflows. Nat. Biotechnol., 35, 316–319.
Grüning,B. et al. (2018) Bioconda: sustainable and comprehensive software distribution for the life sciences. Nat. Methods, 15, 475–476.
Huber,W. et al. (2015) Orchestrating high-throughput genomic analysis with Bioconductor. Nat. Methods, 12, 115–121.
Köster,J. and Rahmann,S. (2012) Snakemake—a scalable bioinformatics workflow engine. Bioinformatics, 28, 2520–2522.
Lun,A.T.L. et al. (2019) EmptyDrops: distinguishing cells from empty droplets in droplet-based single-cell RNA sequencing data. Genome Biol., 20, 63.
Morgan,M. et al. (2020) BiocParallel: Bioconductor facilities for parallel evaluation. Bioconductor version: Release (3.11).
Tekman,M. et al. (2017) Pre-processing of Single-Cell RNA Data (Galaxy Training Materials).
da Veiga Leprevost,F. et al. (2017) BioContainers: an open-source and community-driven framework for software standardization. Bioinformatics, 33, 2580–2582.
Weinstein,J.N. et al. (2013) The Cancer Genome Atlas Pan-Cancer Analysis Project. Nat. Genet., 45, 1113–1120.