

The drake R package: a pipeline toolkit for reproducibility and high-performance computing

William Michael Landau¹

¹ Eli Lilly and Company

DOI: [10.21105/joss.00546](https://doi.org/10.21105/joss.00546)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Submitted: 26 January 2018

Published: 26 January 2018

Licence

Authors of JOSS papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Summary

The [drake](#) R package (Landau 2018) is a workflow manager and computational engine for data science projects. Its primary objective is to keep results up to date with the underlying code and data. When it runs a project, [drake](#) detects any pre-existing output and refreshes the pieces that are outdated or missing. Not every runthrough starts from scratch, and the final answers are reproducible. With a user-friendly R-focused interface, [comprehensive documentation](#), and [extensive implicit parallel computing support](#), [drake](#) surpasses the analogous functionality in similar tools such as [Make](#) (Stallman 1998), [remake](#) (FitzJohn 2017), [memoise](#) (Wickham et al. 2017), and [knitr](#) (Xie 2017).

In reproducible research, [drake](#)'s role is to provide tangible evidence that a project's results are re-creatable. [Drake](#) quickly detects when the code, data, and output are synchronized. In other words, [drake](#) helps determine if the starting materials would produce the expected output if the project were to start over and run from scratch. This approach decreases the time and effort it takes to evaluate research projects for reproducibility.

Regarding high-performance computing, [drake](#) interfaces with a [wide variety of technologies](#) to deploy the steps of a data analysis project. Options range from local multicore computing to serious distributed computing on a cluster. In addition, the parallel computing is implicit. In other words, [drake](#) constructs the directed acyclic network of the workflow and determines which steps can run simultaneously and which need to wait for dependencies. This automation eases the cognitive and computational burdens on the user, enhancing the readability of code and thus reproducibility.

References

- FitzJohn, Rich. 2017. "remake: Make-like build management." <https://github.com/richfitz/remake>.
- Landau, William Michael. 2018. "drake: an R-focused pipeline toolkit for reproducibility and high-performance computing." <https://github.com/ropensci/drake>. <https://doi.org/10.5281/zenodo.1160697>.
- Stallman, Richard. 1998. *GNU Make, Version 3.77*. Free Software Foundation.
- Wickham, Hadley, Jim Hester, Kirill Müller, and Daniel Cook. 2017. "memoise: Memoisation of Functions." <https://CRAN.R-project.org/package=memoise>.
- Xie, Yihui. 2017. "knitr: A General-Purpose Package for Dynamic Report Generation in R." <https://CRAN.R-project.org/package=knitr>.