Omnibenchmark for continuous and open benchmarking (in bioinformatics)

www.omnibenchmark.org

Daniel Incicau [1], Reto Gerber [1], Izaskun Mallona[1,2], Almut Lütge[1,2,3], Ben Carillo[1], Anthony Sonrel[1,2], Charlotte Soneson[2,4], and Mark D. Robinson[1,2]

1. Department of Molecular Life Sciences, University of Zurich | 2. SIB Swiss Institute of Bioinformatics
3. Swiss Data Science Center | 4. Friedrich Miescher Institute for Biomedical Research

Introduction

Omnibenchmark provides **community-driven**, **extensible** and **continuously-updating** benchmarks.

Omnibenchmark defines, executes and versions evaluation pipelines by leveraging a formal benchmark specification and a set of (reusable) benchmarking modules.

Omnibenchmark Design and Usage

Figure 1a shows how to start a new benchmark using Omnibenchmark. After Conceptualization, the Configuration of the benchmark is specified in a single yaml file. Software and Storage need to be set up, followed by implementing methods and selecting metrics. A benchmark can be run by a single CLI command that generates the performance metrics. Finally, the results can be interactively visualized.

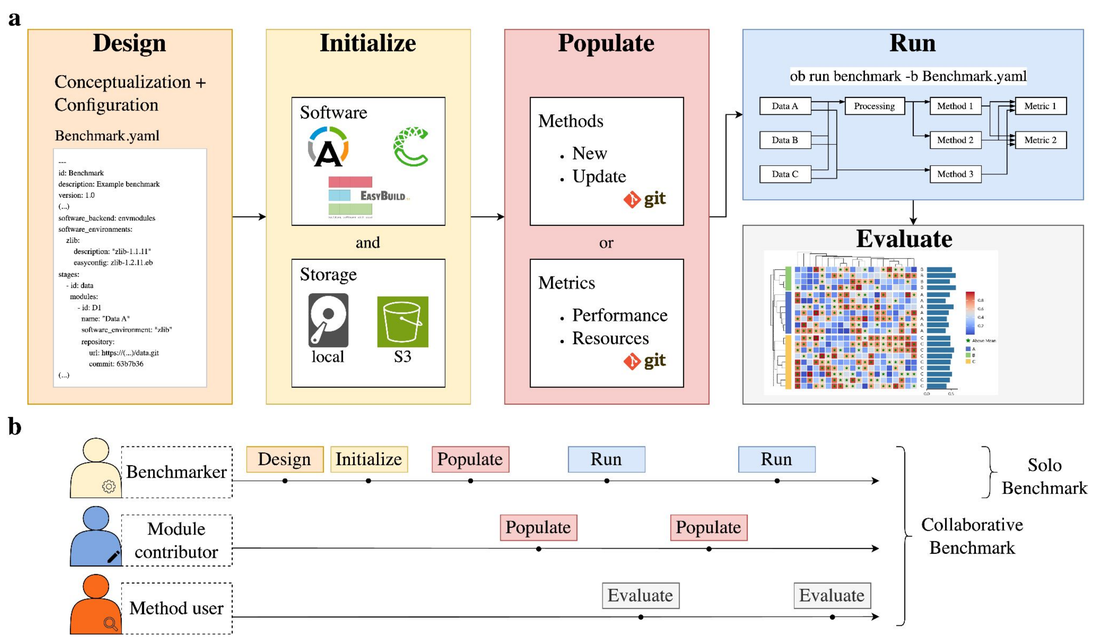

Figure 1: *Omnibenchmark Workflow and Roles a) Benchmarking setup, from configuration to evaluation. b) Roles include Benchmarker (setup), Module Contributors (add/update methods and metrics), and Method Users (analyze results).*

Different roles can be defined (Figure 1b), starting with the **'Benchmarker'**, who is responsible for starting and running a benchmark by initializing the configuration file, specifying the inputs and outputs of each stage, defining (initial) software stacks and providing storage. '**Module contributors'** can add or update datasets / methods / metrics to the benchmark and **'Method users'** evaluate the results to pick the best suited method for their use case.

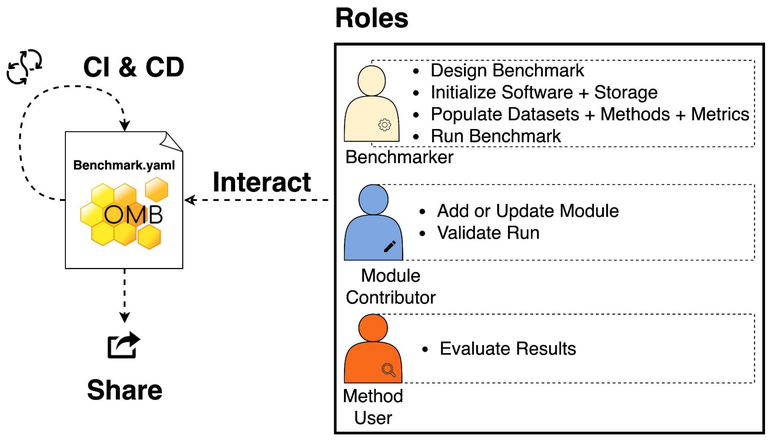

Figure 2*: Summary of Omnibenchmark Workflow and Roles*

Because the structure of a benchmark is defined by a single configuration file, it is the central entry point for updates and contributions (Figure 2). Changes to the configuration file can, for example, directly trigger execution via continuous integration / continuous development (CI/CD).

Case study: clustering

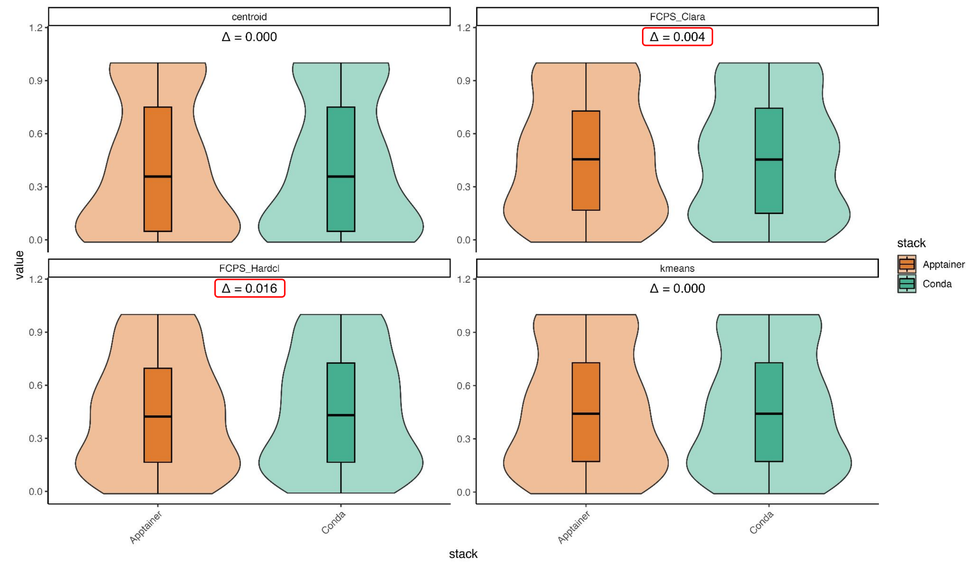

Figure 3: *Benchmark comparison of clustering methods executed in Conda and Apptainer environments. Violin plots display the distribution of Adjusted Rand Index (ARI) scores for each method, with pairwise median differences (Δ) shown above the plots. The comparison is performed across **62 datasets**, using the same code.*

The supported software backends (**Apptainer**, **Conda**, **easybuild**/**lmod**) allow flexibility, but can result in slightly different results as shown for an example clustering benchmark (Figure 3).

In this figure, the choice of clustering methods highlights how software backends can impact clustering performance. While Centroid and K-means clustering yield consistent results across backends, the FCPS (Fundamental Clustering Problems Suite) shows variations depending on the backend used.

Demo CI/CD

Easily trigger your own benchmark run by scanning the QR code below! This will initiate a predefined CI/CD workflow that uses the **Omnibenchmark CLI** (1) on an example benchmark design.

How it works:

1. Scan the QR code
   - *Takes you to GitHub Actions*
2. Ask the presenters to trigger the benchmarking pipeline
   - *The workflows starts running the benchmark automatically*
3. Monitor the results in real time
   - *Track the execution and outcomes directly on Github*

Try it out !

Acknowledgements

- We thank the FOSS community for providing the building blocks of Omnibenchmark (Easybuild, git, Snakemake, conda, apptainer, etc)
- We thank the Renku team at the Swiss Data Science Center (SDSC)
- We thank members of Robinson Lab for their constructive feedback.
- This work has been supported with funding from the Swiss National Science Foundation (grants 200021_212940 and 310030_204869) as well as support from swissuniversities P5 Phase B funding (project 23-36_14).

1. Mallona, I., Luetge, A., Carrillo, B., Incicau, D., Gerber, R., Sonrel, A., Soneson, C., & Robinson, M. D. (2024). Omnibenchmark (alpha) for continuous and open benchmarking in bioinformatics. arXiv preprint arXiv:2409.17038 https://arxiv.org/abs/2409.17038.

Swiss Institute of Bioinformatics

Universität Zürich[UZH]