



www.omnibenchmark.org

# Omnibenchmark for reproducible and collaborative benchmarking

Daniel Incicau<sup>1</sup>, Izaskun Mallona<sup>1,2</sup>, Almut Luetge<sup>1,2,3</sup>, Ben Carrillo<sup>1</sup>, Reto Gerber<sup>1</sup>, Anthony Sonrel<sup>1,2</sup>, Charlotte Soneson<sup>2,4</sup>, and Mark D. Robinson<sup>1,2</sup>

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



Swiss Institute of Bioinformatics

## Introduction

**Omnibenchmark** is a **command line** tool that facilitates **community-driven, extensible** and **continuous** benchmarks. It formalizes, executes, and versions evaluation pipelines using a simple specification file and reusable modules, thus making benchmarking **reproducible, transparent** and **easy to share**.

## Omnibenchmark Workflow

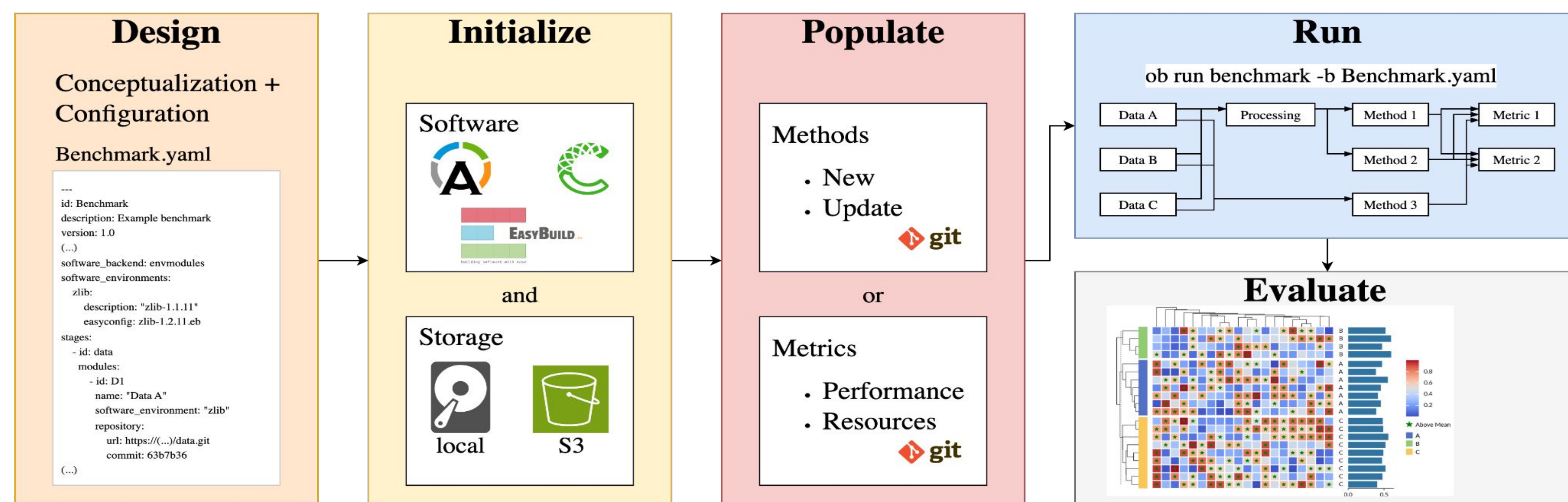


Figure 1: Omnibenchmark setup, from configuration to evaluation.

Figure 1 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.

Different roles can be defined (Figure 2), 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.

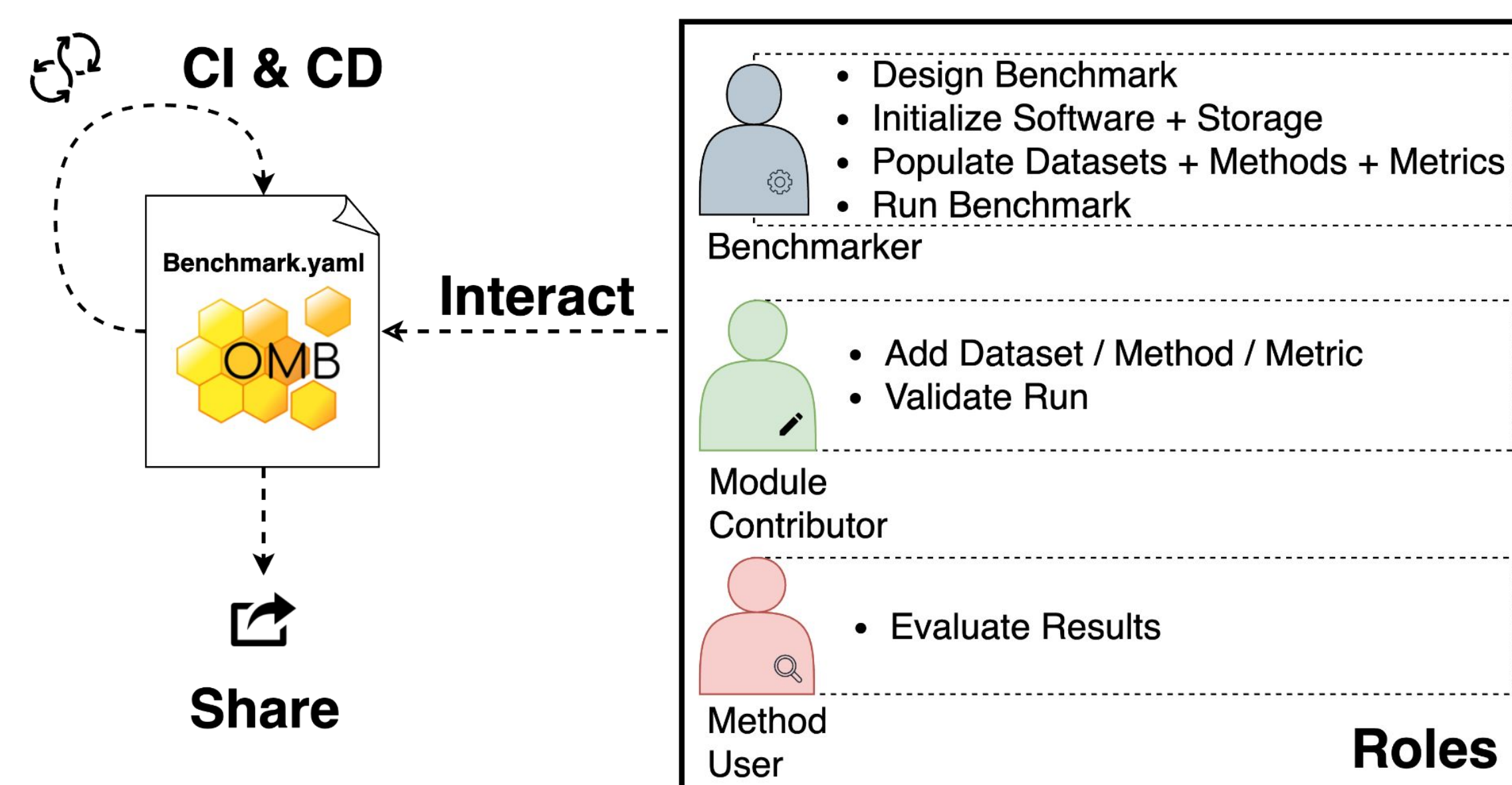


Figure 2: Omnibenchmark workflow and user roles.

## Solo versus Collaborative Benchmarking

**Solo Benchmarking** allows individual researchers to independently evaluate methods using their own infrastructure. A single benchmarker designs the benchmark, adds methods, and runs evaluations locally.

**Collaborative Benchmarking** enables distributed teams to contribute methods simultaneously through a shared benchmark specification. Multiple contributors can add methods via git repositories, with review processes ensuring quality control.

**Omnibenchmark supports both paradigms** through its flexible architecture. Solo users can run benchmarks locally with private storage, while collaborative efforts can utilize shared S3-compatible storage and distributed computing resources. The same *benchmark.yaml* file works for both approaches, allowing to transition from solo development to community collaboration as projects evolve.

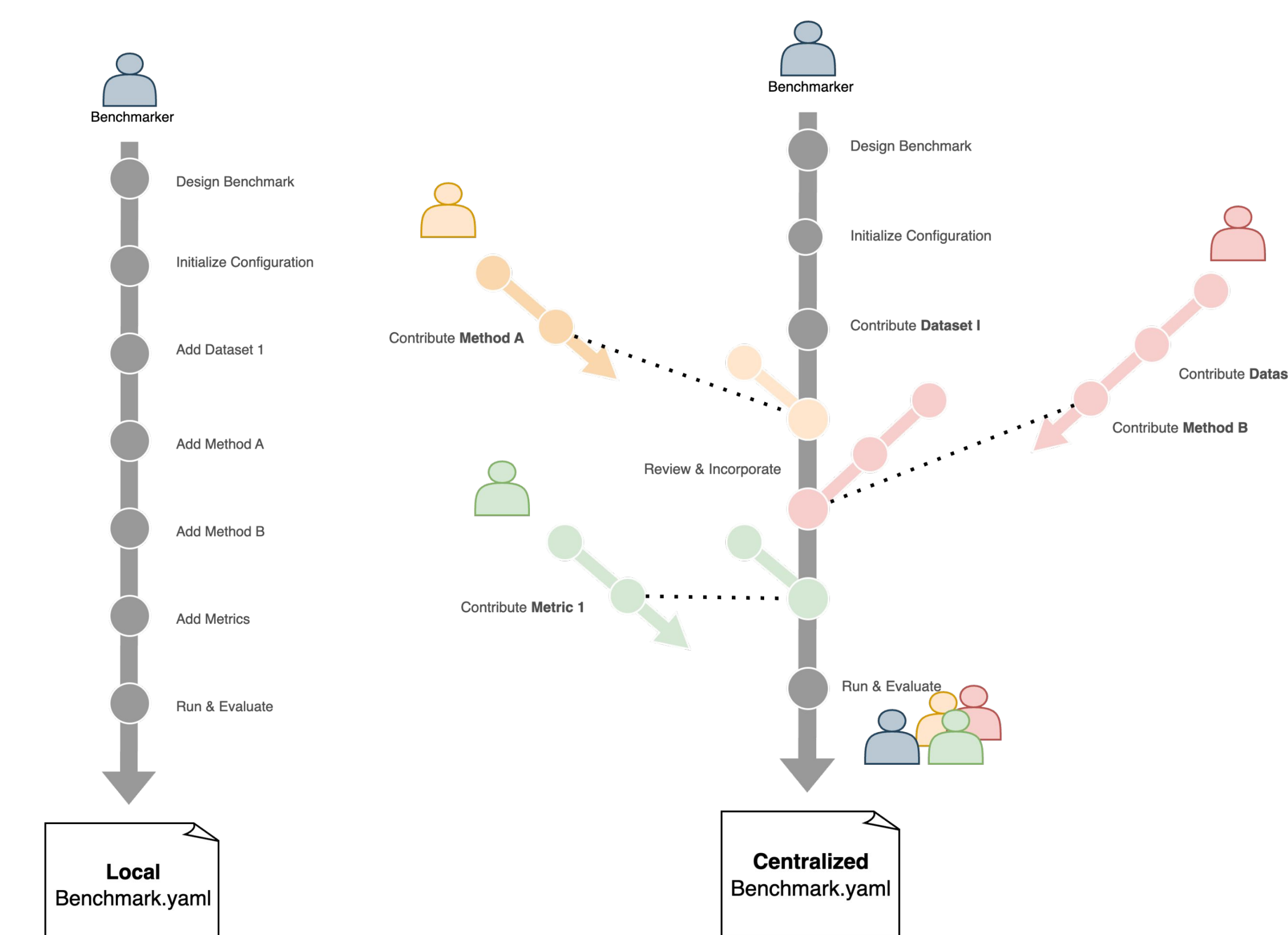


Figure 3: Solo versus collaborative benchmarking workflows.

## Clustering Benchmark Example



Check it out !

A clustering example powered by Omnibenchmark is available on GitHub. It contains the *benchmark.yaml* configuration file with more than 60 datasets, 28 methods and 10 metrics, along with configurations for different software backends.

The continuous benchmark process is highlighted using a Github Actions pipeline that can be triggered automatically by a Pull Request, or manually by the Benchmarker.

## Getting Started

Omnibenchmark is a pip installable python package. With the following commands and some benchmark definition and configuration writing, you can have your own setup.

```
pip install omnibenchmark
ob create benchmark
ob run benchmark [args]
```

## Acknowledgements

- We thank the FOSS community for providing the building blocks of Omnibenchmark (Easybuild, git, Snakemake, conda, aptainer, 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).