The Journal of Open Source Software

MAFw: A Modular Analysis Framework for Streamlining and Optimizing Data Analysis Workflows

Antonio Bulgheroni [1¶] **and Michael Krachler** [1]

1 European Commission - Joint Research Centre (JRC) - Karlsruhe - Germany ¶ Corresponding author

DOI: 10.21105/joss.08449

Software

- Review
- Repository
- Archive

Editor: Chris Vernon

Reviewers:

- @kaustavbhattacharjee
- @sandeep-ps
- @lizliz

Submitted: 12 May 2025
Published: 02 October 2025

Summary

The Modular Analysis Framework (MAFw) is a versatile platform engineered to streamline the development of sophisticated data analysis pipelines. Featuring a flexible and extensible architecture, MAFw enables users to easily define, customize, and chain multiple processing steps, fostering efficient and reproducible analyses.

The idea behind MAFw is well described in the use case study presented in the documentation. The researcher needs to break down their analysis strategy in simple, self-contained steps (data import, conversion, filtering and aggregation, just to name a few examples) to be coded in a subclass of the MAFw basic unit, the Processor. Processors can be dasy-chained to make a pipeline via steering files and then executed by the built-in execution engine.

Statement of need

In recent years, the volume and complexity of data generated across diverse scientific disciplines have grown exponentially. Consequently, robust data analysis has become a vital element of modern scientific research, demanding efficient and adaptable tools capable of managing and processing large-scale datasets.

MAFw addresses this challenge by offering a flexible, modular framework designed to empower data scientists to execute complex analytical tasks within a well-structured environment. Currently, researchers often juggle multiple steps — such as data ingestion, processing, and visualization — using disparate tools, which can be time-consuming, error-prone, and hinder code reusability across different projects.

By providing a customizable, component-based platform, MAFw streamlines the entire analysis workflow. It enables each analytical step to be executed reproducibly within a unified environment, supported by a centralized database that stores intermediate results and guides the progression of analyses.

This design not only enhances efficiency and reproducibility, but also facilitates collaboration and knowledge sharing among researchers, ultimately advancing scientific discovery through more reliable and adaptable data analysis practices.

MAFw conceptual design

Historical roots and other similar libraries

MAFw draws inspiration from **MARLIN** (Aplin et al., 2012; Gaede, 2006), an object-oriented C++ application framework for particle physicists. The authors transitioned from C++ to

Python to streamline onboarding for data scientists and leverage the extensive array of available analytical tools.

Apart from MarlinMT (Ete, Remi et al., 2020), the evolution of Marlin including multi-threading and parallel processing, another commonly-used package performing similar tasks is **GAUDI** (Barrand et al., 2001), developed and widely adopted by the high-energy physics (HEP) community. Compared to GAUDI, MAFw is surely lighter weight and not bound to other software libraries typical of the HEP sector, making MAFw potentially more accessible to a wider user base.

MAFw was initially developed and successfully used for analyzing autoradiography images (Krachler et al., 2024, 2025); later was expanded with enhanced database interfaces and plugin systems for broader applicability.

The three pillars behind MAFw

The three primary driving forces behind MAFw are:

1. **Modularity**: Complex analytical tasks are decomposed into self-contained operations within the Processor container, which scientists can subclass to implement sophisticated operations tailored to specific needs.
2. **Enhanced Ease of Handling Large Datasets**: MAFw leverages database-assisted I/O for better scalability and efficiency, enabling quick access to parameters and efficient selection of data subsets for analysis.
3. **Pipelines and Scalability**: The framework offers a flexible execution engine that enables the definition of comprehensive analysis pipelines through text files, facilitating seamless deployment within container orchestration environments.

The `Processor`: MAFw core element

The Processor is a self-contained execution unit performing tasks such as data filtering and transformation. Designed for reusability, Processors can be combined to form complex pipelines. Each Processor includes methods for initialization, processing, and finalizing tasks, with flexible looping modes enhancing its adaptability for specific applications.

The Processor execution logic schematic flowchart is shown in Figure 1, where the elements with colored background are available to the user for customization via subclassing. A more detailed description on how to create a processor subclass is available in the general documentation of MAFw.

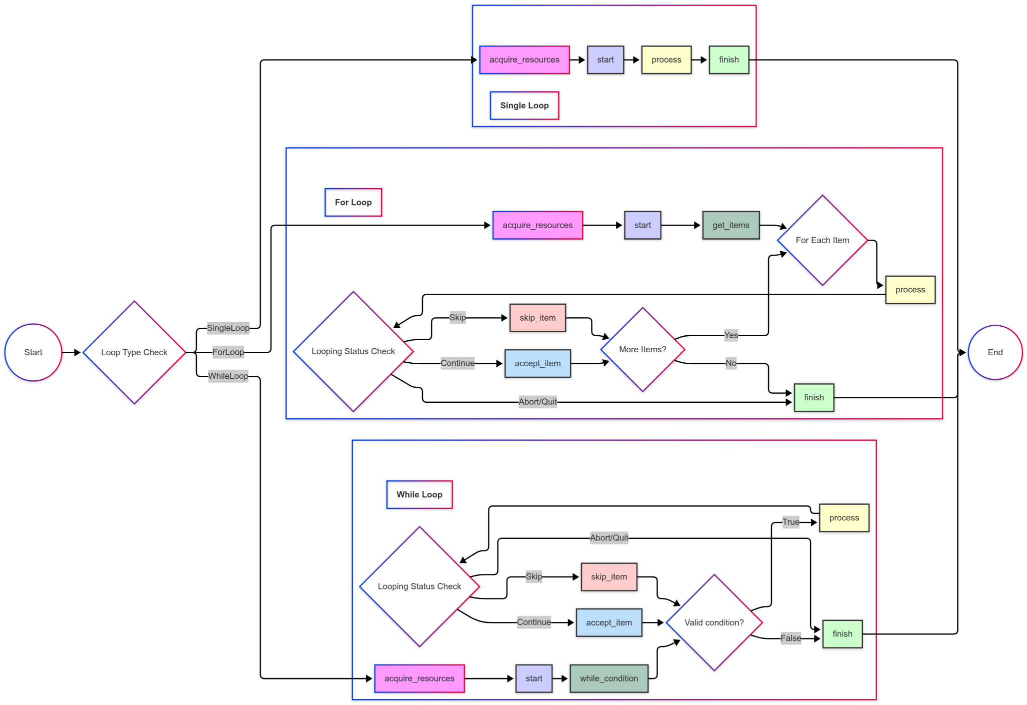

Figure 1: The flowchart of the processor execution logic. Elements with colored background are available to the user for customization.

Extensibility via a simple plugin mechanism

MAFw supports user-developed custom processors through a plugin system based on pluggy, enabling seamless integration of specialized functionality and fostering a collaborative community where plugin libraries can be shared and compared.

Modularity and pipelines

MAFw achieves modularity through Processors combined into complex data processing pipelines configured via TOML steering files. This approach eliminates the need for dedicated scripts, as the built-in execution engine automatically interprets the steering file to run the pipeline. Moreover, the pipeline execution can be seamlessly performed on a high-throughput computer system.

A database as I/O support

MAFw provides a database interface built on the Peewee Object-Relational Mapping (ORM) system, supporting multiple database backends (MySQL, PostgreSQL, SQLite). This ORM abstraction simplifies database interactions, allowing users to define data models and execute queries directly in Python.

The database component plays a crucial role in storing intermediate results and guiding analyses, with support for filters and triggers to automate pipeline execution.

Conclusion

MAFw offers a versatile and highly customizable platform that empowers data scientists to execute complex analytical tasks with ease. Its modular architecture, robust plugin system, and seamless database integration position it as an ideal solution for researchers with diverse analytical requirements.

By delivering a comprehensive framework for data analysis, MAFw simplifies workflow management, allowing researchers to concentrate on their core research questions rather than the technical complexities of data processing.

Acknowledgments

The authors would like to thank the JRC.T.4 and JRC.T.6 units for providing the essential tools necessary for code development and deployment.

References

Aplin, S., Engels, J., Gaede, F., Graf, N. A., Johnson, T., & McCormick, J. (2012). LCIO: A persistency framework and event data model for HEP. *2012 IEEE Nuclear Science Symposium and Medical Imaging Conference Record (NSS/MIC)*, 2075–2079. https://doi.org/10.1109/NSSMIC.2012.6551478

Barrand, G., Belyaev, I., Binko, P., Cattaneo, M., Chytracek, R., Corti, G., Frank, M., Gracia, G., Harvey, J., Herwijnen, E. van, Maley, P., Mato, P., Probst, S., & Ranjard, F. (2001). GAUDI — a software architecture and framework for building HEP data processing applications. *Computer Physics Communications*, *140*(1), 45–55. https://doi.org/10.1016/S0010-4655(01)00254-5

Ete, Remi, Gaede, Frank, Benda, Julian, & Grasland, Hadrian. (2020). MarlinMT - parallelising the marlin framework. *EPJ Web Conf.*, *245*, 05022. https://doi.org/10.1051/epjconf/202024505022

Gaede, F. (2006). Marlin and LCCD—software tools for the ILC. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, *559*(1), 177–180. https://doi.org/10.1016/j.nima.2005.11.138

Krachler, M., Martinez Ferri, A. I., & Bulgheroni, A. (2024). Potential of digital autoradiography for characterization of uranium materials. *Microchemical Journal*, *206*, 111448. https://doi.org/10.1016/j.microc.2024.111448

Krachler, M., Martinez Ferri, A. I., & Bulgheroni, A. (2025). Assessing uranium enrichment levels using digital autoradiography. *Analytica Chimica Acta*, 344073. https://doi.org/10.1016/j.aca.2025.344073