Mem. S.A.It. Vol. 94, 102
© SAIt 2023

Memorie della

Performant feature extraction for photometric time series

A. Lavrukhina[1] and K. Malanchev[2,3]

[1] Lomonosov Moscow State University, Faculty of Space Research, Russia
[2] Lomonosov Moscow State University, Sternberg Astronomical Institute, Russia
[3] University of Illinois at Urbana-Champaign, Department of Astronomy, USA
e-mail: lavrukhina.ad@gmail.com

Received: 26-10-2022; Accepted: 24-02-2023

Abstract. Astronomy is entering the era of large surveys of the variable sky such as Zwicky Transient Facility (ZTF) and the forthcoming Legacy Survey of Space and Time (LSST) which are intended to produce up to a million alerts per night. Such an amount of photometric data requires efficient light-curve pre-processing algorithms for the purposes of subsequent data quality cuts, classification, and characterization analysis. In this work, we present the new library "light-curve" for Python and Rust, which is intended for feature extraction from light curves of variable astronomical sources. The library is suitable for machine learning classification problems: it provides a fast implementation of feature extractors, which outperforms other public available codes, and consists of dozens features describing shape, magnitude distribution, and periodic properties of light curves. It includes not only features which had been shown to provide a high performance in classification tasks, but also new features we developed to improve classification quality of selected types of objects. The "light-curve" library is currently used by the ANTARES, AMPEL, and Fink broker systems for analyzing the ZTF alert stream, and has been selected for use with the LSST.

Key words. Methods: data analysis – Methods: statistical – Stars: statistics

1. Introduction

Modern astronomical surveys contain a large amount of data about billions of astronomical sources. For example, the Zwicky Transient Facility (ZTF) (Bellm et al. 2019) obtains about 1 million alerts per night, which total volume is over 70 GB (Patterson et al. 2019). For this reason, there is an urgent need for productive and convenient in use methods of data processing. In this work, we present a new Rust/Python package for feature extraction from light curves of astronomical sources named "light-curve" (Malanchev 2021) and compare its performance with another tool – "feets" library (Cabral et al. 2018), which is written entirely in Python and based on "Numpy" (van der Walt et al. 2011), "SciPy" (Virtanen et al. 2020) and "statsmodels" (Seabold & Perktold 2010).

2. light-curve Python package

The "light-curve" project aims to bring high-performance processing of irregularly sampled

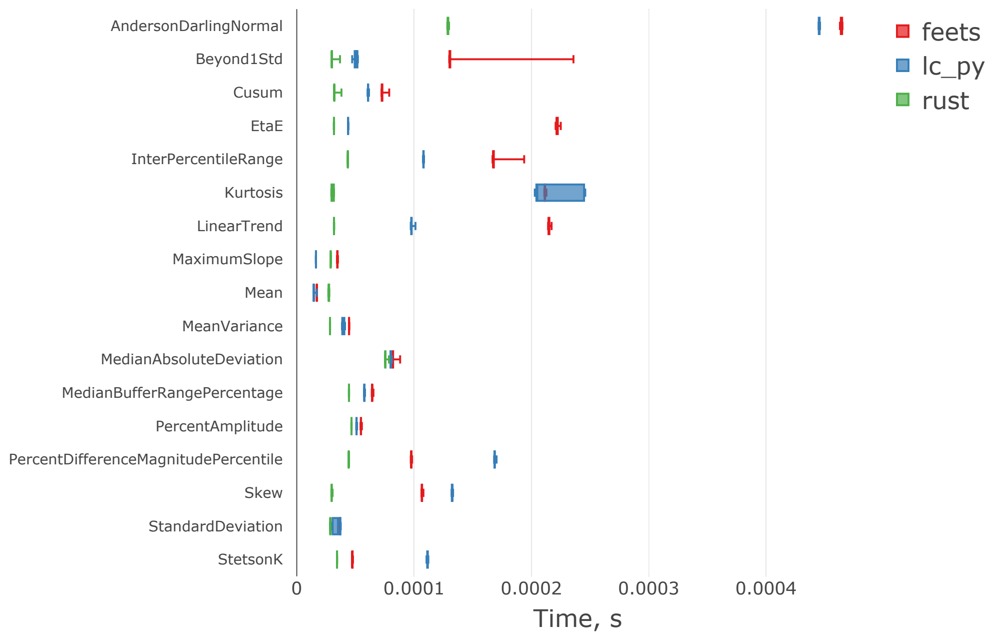

Fig. 1. Benchmark boxplots for the features which are implemented in all discussed packages: ”feets”, ”Rust” implementation of ”light-curve” package, and its Python sub-package. The x-axis is the time in seconds of feature extraction from the light curve consisting of 1,000 data points. ”feets” library performance is labeled as ”feets”, Python version of ”light-curve” as ”lc_py” and the Rust one as ”rust”. The median value, the maximum, the minimum, the first and the third quartiles were counted.

time-series data in Rust and Python. Currently, it consists of several Rust packages including ”light-curve-feature” crate for feature extraction, which was developed as a part of SNAD anomaly-detection pipelines (Malanchev 2021; Aleo et al. 2022; Pruzhinskaya et al. 2022). While this package has proven itself as a fast and thread-parallelizable solution, we wrapped it into ”light-curve” Python package. However, this package consists of two parts: the wrapper sub-package and a pure Python sub-package implemented with ”Numpy” and ”SciPy”. The wrapper sub-package offers high-performance Rust implementation of features with memory-safe Python interoperability, while the second sub-package is utilized for the development of new experimental features. The availability of two separate implementations of the same feature extractors enables us to validate the correctness of both approaches and measure the performance boost provided by the Rust sub-package.

3. Benchmark

For the first benchmark, we used the subset of features, which are implemented in both ”light-curve” and ”feets” packages. Each feature was extracted from a randomly generated light curve consisting of 1,000 observations. The benchmark results are shown in Fig. 1. See the feature descriptions in Malanchev (2021) or `https://docs.rs/light-curve-feature/0.5.2/light_curve_feature/features/index.html`.

The second benchmark, shown in Fig. 2, examines the performance of several feature extractors, including both pure Python and Rust implementations, as the number of observations in a light curve increases. A linear relationship between the number of observations

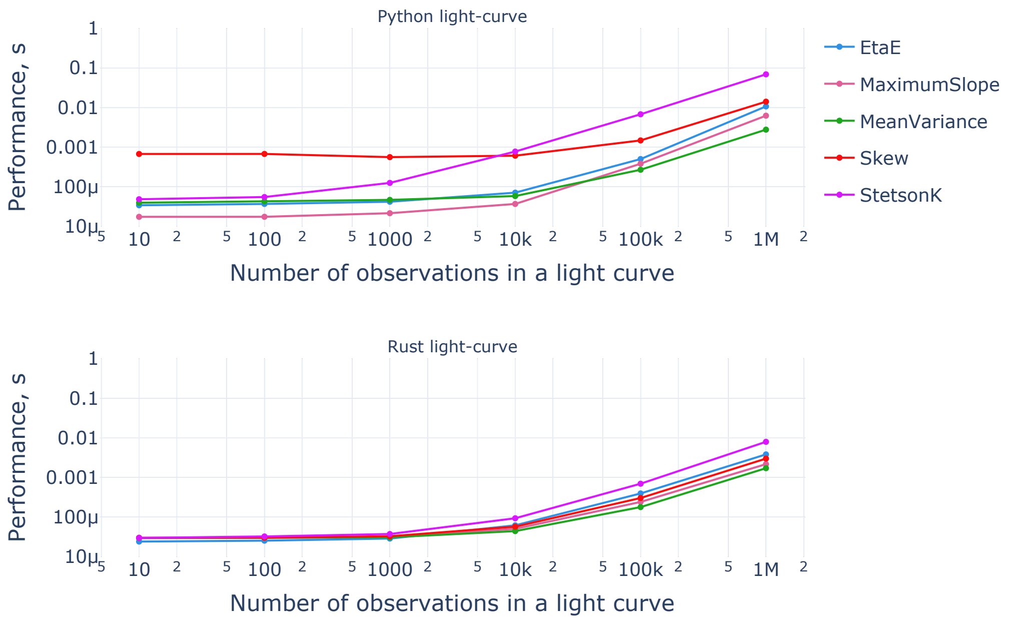

Fig. 2. Benchmark results of several features for both the pure-Python and Rust implementations of the ”light-curve” package, as a function of the number of observations in a light curve. Both the x-axis and y-axis are on a logarithmic scale.

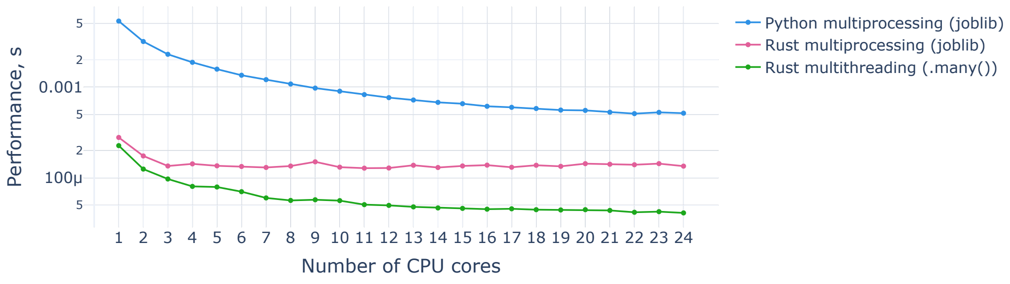

Fig. 3. Processing time per a single light curve for extraction of features subset presented in Fig. 1 versus the number of CPU cores used. The blue curve represents pure-Python implementation and ”Joblib”’s multiprocessing strategy, the red curve represents the same strategy for the Rust implementation, and the green curve represents the built-in multithreading implementation with the Rust sub-package. The dataset consists of 10,000 light curves with 1,000 observations in each.

and the performance of the feature extractors is observed, with a noticeable impact on performance when the number of observations exceeds 10,000. It is likely due to overhead such as checking if the time array is sorted and ensuring the magnitude array does not contain inappropriate values.

The last benchmark (see Fig. 3) is intended to present the dependency of consumed time from the used number of CPU cores for the feature subset, presented in Fig. 1. The features are extracted from a dataset of 10,000 light curves, each consisting of 1,000 observations. The multiprocessing strategies include "Joblib"(Joblib Development Team 2020) for the pure Python sub-package and Rust bindings, as well as the built-in multithreading facility provided by the `many()` method in the Rust bindings. In situations where many features need to be extracted from a large dataset of light curves, the Rust multithreading approach shows the best performance, although performance is still impacted by overhead from the multithreading, such as data serialization and deserialization, and message passing between parent and child threads.

The hardware setup for the benchmarking was a dual-CPU Intel Xeon Gold 5118 server.

4. Conclusions

The results of the test show that the Rust version of "light-curve" package outperforms other implementations by a factor 1.5-50. This superiority is due to Python's dynamic typing and interpreted language nature. Such performance enables the extraction of a large set of "cheap" features in just a few milliseconds per CPU core for 1,000 observations. The library also features fast implementations of the periodogram (Lomb 1976; Scargle 1982) and parametric fits (Bazin et al. 2009; Villar et al. 2019). The time to extract the feature set, including the periodogram, Bazin and Villar fits, is $\approx$ 25 ms $\times$ CPU for six ugrizy LSST 3-year light curves. It allows the processing of all LSST alerts in real-time using a single computational node.

Acknowledgements. The reported study was funded by RFBR, Russia and CNRS, France according to the research project No. 21-52-15024.

References

Aleo, P., Malanchev, K., Pruzhinskaya, M., et al. 2022, New Astronomy, 96, 101846
Bazin, G., Palanque-Delabrouille, N., Rich, J., et al. 2009, A&A, 499, 653
Bellm, E. C., Kulkarni, S. R., Graham, M. J., et al. 2019, PASP, 131, 018002
Cabral, J., Sánchez, B., Ramos, F., et al. 2018, Astronomy and Computing
Joblib Development Team. 2020, Joblib: running Python functions as pipeline jobs
Lomb, N. R. 1976, Ap&SS, 39, 447
Malanchev, K. 2021, light-curve: Light curve analysis toolbox, Astrophysics Source Code Library, record ascl:2107.001
Patterson, M. T., Bellm, E. C., Rusholme, B., et al. 2019, PASP, 131, 018001
Pruzhinskaya, M. V., Ishida, E. E. O., Novinskaya, A. K., et al. 2022, Supernova search with active learning in ZTF DR3
Scargle, J. D. 1982, ApJ, 263, 835
Seabold, S. & Perktold, J. 2010, Proceedings of the 9th Python in Science Conference, 2010
van der Walt, S., Colbert, S. C., & Varoquaux, G. 2011, Computing in Science & Engineering, 13, 22
Villar, V. A., Berger, E., Miller, G., et al. 2019, ApJ, 884, 83
Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, Nature Methods, 17, 261