gamma_flow: **G**uided **A**nalysis of **M**ulti-label spectra by **Ma**trix **F**actorization for **L**ightweight **O**perational **W**orkflows

Viola Rädle[a,*], Tilman Hartwig[a], Benjamin Oesen[a], Emily Alice Kröger[b], Julius Vogt[b], Eike Gericke[b], Martin Baron[b]

[a] *Application Lab for AI and Big Data, German Environmental Agency, Leipzig, Germany*
[b] *Federal Office for Radiation Protection, Berlin, Germany*

Abstract

gamma_flow is an open-source Python package for real-time analysis of spectral data. It supports classification, denoising, decomposition, and outlier detection of both single- and multi-component spectra. Instead of relying on large, computationally intensive models, it employs a novel supervised approach to non-negative matrix factorization (NMF) for dimensionality reduction. This ensures a fast, efficient, and adaptable analysis while reducing computational costs. **gamma_flow** achieves classification accuracies above 90% and enables reliable automated spectral interpretation. Originally developed for gamma-ray spectra, it is applicable to any type of one-dimensional spectral data. As an open and flexible alternative to proprietary software, it supports various applications in research and industry.

Keywords: Python, Gamma spectroscopy, Non-negative Matrix Factorization, Classification, Denoising, Spectral Deconvolution

Metadata

The ancillary data table 1 is required for the sub-version of the codebase. Please replace the italicized text in the right column with the correct information about your current code and leave the left column untouched.
Optionally, you can provide information about the current executable software version filling in the left column of Table 2. Please leave the first column as it is. FRAGE: Welche Tabelle sollen wir ausfüllen?

[*] Corresponding author: raedle.htwk@web.de

Nr.	Code metadata description	Metadata
TO DO C1	Current code version	For example v42
C2	Permanent link to code/repository used for this code version	`https://gitlab.opencode.de/uba-ki-lab/gamma_flow`
C3	Permanent link to Reproducible Capsule	TO DO For example: `https://codeocean.com/capsule/0270963/tree/v1`
C4	Legal Code License	BSD 3-Clause "New" or "Revised" License
C5	Code versioning system used	git
C6	Software code languages, tools, and services used	Python
C7	Compilation requirements, operating environments & dependencies	TO DO. Jupyter?
C8	If available Link to developer documentation/manual	`https://gitlab.opencode.de/uba-ki-lab/gamma_flow/-/blob/main/README.md?ref_type=heads`
C9	Support email for questions	raedle.htwk@web.de

Table 1: Code metadata (mandatory)

1. Motivation and significance

In this section, we want you to introduce the scientific background and the motivation for developing the software.

- *Explain why the software is important and describe the exact (scientific) problem(s) it solves.*
- *Indicate in what way the software has contributed (or will contribute in the future) to the process of scientific discovery; if available, please cite a research paper using the software.*
- *Provide a description of the experimental setting. (How does the user use the software?)*
- *Introduce related work in literature (cite or list algorithms used, other software etc.).*

2. Software description

Describe the software. Provide enough detail to help the reader understand its impact.

Nr.	(Executable) software metadata description	Please fill in this column
S1	Current software version	For example 1.1, 2.4 etc.
S2	Permanent link to executables of this version	For example: `https://github.com/combogenomics/DuctApe/releases/tag/DuctApe-0.16.4`
S3	Permanent link to Reproducible Capsule	
S4	Legal Software License	List one of the approved licenses
S5	Computing platforms/Operating Systems	For example Android, BSD, iOS, Linux, OS X, Microsoft Windows, Unix-like , IBM z/OS, distributed/web based etc.
S6	Installation requirements & dependencies	
S7	If available, link to user manual - if formally published include a reference to the publication in the reference list	For example: `http://mozart.github.io/documentation/`
S8	Support email for questions	

Table 2: Software metadata (optional)

2.1. Software architecture

Give a short overview of the overall software architecture; provide a pictorial overview where possible; for example, an image showing the components. If necessary, provide implementation details.

2.2. Software functionalities

Present the major functionalities of the software.

2.3. Sample code snippets analysis (optional)

3. Illustrative examples

Provide at least one illustrative example to demonstrate the major functions of your software/code.
Optional*: you may include one explanatory video or screencast that will appear next to your article, in the right hand side panel. Please upload any video as a single supplementary file with your article. Only one MP4 formatted, with 150MB maximum size, video is possible per article. Recommended video dimensions are 640 x 480 at a maximum of 30 frames / second. Prior to*

submission please test and validate your .mp4 file at `http://elsevier-apps.sciverse.com/GadgetVideoPodcastPlayerWeb/verification` *. This tool will display your video exactly in the same way as it will appear on ScienceDirect.*

Plots:

- Loadings
- Scores (different detector, single-label class.)
- Confusion matrix (different detector, single-label class.)
- Denoised spectrum
- Outlier: Feature importance (decision tree)

Problem: Diese Figures wollten wir eigentlich erst im wiss. Paper bringen! Alternative: Nur graphical abstract? Wirkt auf mich zu wenig für diese Section...

4. Impact

This is the main section of the article and reviewers will weight it appropriately. Please indicate:

- *Any new research questions that can be pursued as a result of your software.*
- *In what way, and to what extent, your software improves the pursuit of existing research questions.*
- *Any ways in which your software has changed the daily practice of its users.*
- *How widespread the use of the software is within and outside the intended user group (downloads, number of users if your software is a service, citable publications, etc.).*
- *How the software is being used in commercial settings and/or how it has led to the creation of spin-off companies.*

Please note that points 1 and 2 are best demonstrated by references to citable publications.

5. Conclusions

[1]

Acknowledgements

Optional. You can use this section to acknowledge colleagues who don't qualify as a co-author but helped you in some way.

References

[1] J. A. Kulesza, T. R. Adams, J. C. Armstrong, S. R. Bolding, F. B. Brown, J. S. Bull, T. P. Burke, A. R. Clark, R. A. A. Forster III, J. F. Giron, T. S. Grieve, C. J. Josey, R. L. Martz, G. W. McKinney, E. J. Pearson, M. E. Rising, C. J. C. Solomon Jr., S. Swaminarayan, T. J. Trahan, S. C. Wilson, A. J. Zukaitis, MCNP® Code Version 6.3.0 Theory & User Manual, Tech. Rep. LA-UR-22-30006, Los Alamos National Laboratory (LANL), Los Alamos, NM (United States) (Sep. 2022). `doi:10.2172/1889957`.

References

[1] Use this style of ordering. References in-text should also use a similar style.

If the software repository you used supplied a DOI or another Persistent IDentifier (PID), please add a reference for your software here. For more guidance on software citation, please see our guide for authors or this article on the essentials of software citation by FORCE 11, of which Elsevier is a member.

Reminder: Before you submit, please delete all the instructions in this document, including this paragraph. Thank you!