

Molass Library: A Python Package for SEC-SAXS Data Analysis

Masatsuyo Takahashi ¹ and Nobutaka Shimizu ^{1,2,3}✉

¹ Structural Biology Research Center, Institute of Materials Structure Science, High Energy Accelerator Research Organization (KEK), Japan ² Photon Factory, Institute of Materials Structure Science, High Energy Accelerator Research Organization (KEK), Japan ³ Life Science Research Infrastructure Group, R&D of Technology and Systems for Synchrotron Radiation Applications Division, RIKEN SPring-8 Center, RIKEN, Japan   Corresponding author

DOI: [10.21105/joss.10403](https://doi.org/10.21105/joss.10403)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Bonan Zhu](#) 

Reviewers:

- [@marshallmcdonnell](#)
- [@KedoKudo](#)

Submitted: 21 October 2025

Published: 30 August 2026

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

Molass Library is a modern, open-source Python package designed for the analysis of SEC-SAXS (Small-Angle X-ray Scattering coupled with Size Exclusion Chromatography) experimental data. It refactors the original MOLASS architecture ([Yonezawa et al., 2023](#)), currently hosted at the Japanese synchrotron radiation facilities, [Photon Factory](#) and [SPring-8](#). By leveraging the Python ecosystem and supporting interactive scripting in Jupyter notebooks, Molass Library provides enhanced flexibility, reproducibility, and extensibility compared to its predecessor.

A typical SEC-SAXS experiment involves two interconnected processes:

- **SEC** – Size Exclusion Chromatography
- **SAXS** – Small-Angle X-ray Scattering

Effective analysis requires seamless integration of both domains, which Molass Library facilitates through a unified, scriptable workflow.



Figure 1: Logo of Molass Library designed by K. Yatabe

Statement of Need

Analysis of SEC-SAXS data is inherently multi-step and complex. A typical workflow includes:

1. Circular (Azimuthal) averaging
2. Background subtraction
3. Trimming of data
4. Baseline correction
5. Low-rank factorization
6. Radius of gyration (R_g) estimation – Guinier plot ([Guinier, 1939](#))
7. Folding state estimation – Kratky plot ([Kratky, 1963](#))
8. Electron density calculation

Molass Library currently implements steps 3–7. For steps 1 and 2, users may employ SAngler

(Shimizu et al., 2016) or device-specific software, while for step 8, DENSS (Grant, 2018) is bundled.

State of the field

Several software tools address SEC-SAXS data processing. The proprietary ATSAS suite (Manalastas-Cantos et al., 2021) and the open-source BioXTAS RAW (Hopkins, 2024) provide comprehensive GUI-based workflows widely used in the field. For the specific challenge of overlapping peak decomposition—required when chromatographic peaks overlap or significant interparticle interference occurs—CHROMIXS defers such analysis to “other methods” (Panjkovich & Svergun, 2018), whereas EFAMIX defines quantitative conditions for reliable separation (signal-to-noise ratio $\geq 10^3$ for reliable separation of multiple components; peak asymmetry parameter $\tau \leq 2$; peak center-to-center distance $\geq (2 \times \text{peak width})$ for reliable decomposition) (Konarev et al., 2021). REGALS employs a two-stage framework combining EFA with L1 regularization (Meisburger et al., 2021) and is therefore guided by assumptions underlying EFA in strongly overlapping cases. Molass Library takes a complementary approach: rather than extending these existing GUI-based tools, it provides a Python-first, Jupyter-centric platform in which overlapping peak decomposition is handled through explicit parametric elution curve modeling combined with global optimization, constrained by physicochemical consistency derived from SEC theory (size-dependent elution behavior) and SAXS analysis (radius of gyration consistency across elution fractions).

Software Design

Molass Library is designed to support sustainable software development through an open framework that enables broad community participation and contribution. This design philosophy prioritizes explicit, readable code over graphical interfaces, making the analysis methodology transparent for learning, review, and long-term maintenance. We are actively enhancing AI-readiness through systematic usability testing with AI agents and iterative improvements to API discoverability and inline documentation. The architecture emphasizes modularity and clear separation of concerns: elution curve models (EGH, SDM, EDM, etc.) are isolated as parametric functions; low-rank factorization uses standard linear algebra (Moore-Penrose pseudoinverse); and visualization is decoupled from computation. By integrating established packages (NumPy, SciPy, pybaselines, ruptures) rather than reimplementing core functionality, we reduce custom code volume and enhance long-term viability. This approach allows domain researchers to maintain and extend the code using AI-assisted development tools.

Molass Library maintains a runtime dependency on `molass_legacy` for rigorous optimization and some computational functions during incremental refactoring. The dependency is invisible to end users—Molass Library provides a clean, Pythonic API while the internal implementation is gradually migrated. This approach allows sustainable development by a small facility-based team: refactoring occurs incrementally when fixing bugs or adding features, rather than through disruptive rewrites.

Research Impact Statement

The MOLASS legacy code has demonstrated scientific value through research conducted using its GUI tool (Furukawa et al., 2025; Jiang et al., 2023), as well as through its contribution to the development of SEC-SAXS data analysis systems at synchrotron facilities (Yonezawa et al., 2023). MOLASS is regarded within the community as providing a rigorous peak decomposition framework for SEC-SAXS analysis (Matsui et al., 2024).

Molass Library extends this demonstrated impact by converting the legacy GUI-centered workflow into a scriptable, notebook-based Python library. This architecture makes the

decomposition procedure easier to inspect, reproduce, and extend, allowing facility scientists and domain researchers to prototype new elution-curve models and analysis strategies in Jupyter notebooks, validate them against SEC-SAXS data, and expose them through a stable API for routine use. By separating methodological development from user-facing workflows, Molass Library provides a maintainable route for translating advances in SEC-SAXS analysis into reproducible community software.

Notable package dependencies

Molass Library is built on robust scientific Python libraries, including NumPy ([Harris et al., 2020](#)), SciPy ([Virtanen et al., 2020](#)), and Matplotlib ([Hunter, 2007](#)). It further integrates:

- **pybaselines** ([Erb, 2025](#)) for advanced baseline correction
- **ruptures** ([Truong et al., 2020](#)) for change point detection
- **scipy.signal.find_peaks** for peak recognition
- **scikit-learn** ([Pedregosa et al., 2011](#)) for grouping peaks by the k-means algorithm

By adopting these well-maintained packages, Molass Library reduces custom code and enhances reliability. The transition from a GUI-based workflow (previously using Tkinter) to Jupyter-based scripting further streamlines reproducibility and collaboration.

Theoretical Foundation

A central feature of Molass Library is its implementation of **low-rank factorization** using elution curve models, enabling decomposition of overlapping chromatographic peaks, a common challenge in SEC-SAXS analysis. The decomposition is formulated as:

$$M = P \cdot C \quad (1)$$

where:

- M : measured data matrix
- P : matrix of component scattering curves
- C : matrix of component elution curves

In practice, methods find P and C that minimize $\|M - PC\|^2$ (plus regularization terms) rather than achieving exact equality.

The optimal solution in the least-squares sense is given by:

$$P = M \cdot C^+ \quad (2)$$

where C^+ denotes the Moore-Penrose pseudoinverse ([Penrose, 1955, 1956](#)). This approach enables robust separation of components, even in the presence of overlapping peaks, provided suitable constraints are applied. Moreover, when interparticle interference is significant, the second-order virial approximation of the structure factor yields $I(q, c) \approx cA(q) + c^2B(q)$, which maps naturally onto Eq. (1) by augmenting C with an additional c^2 row vector per affected component.

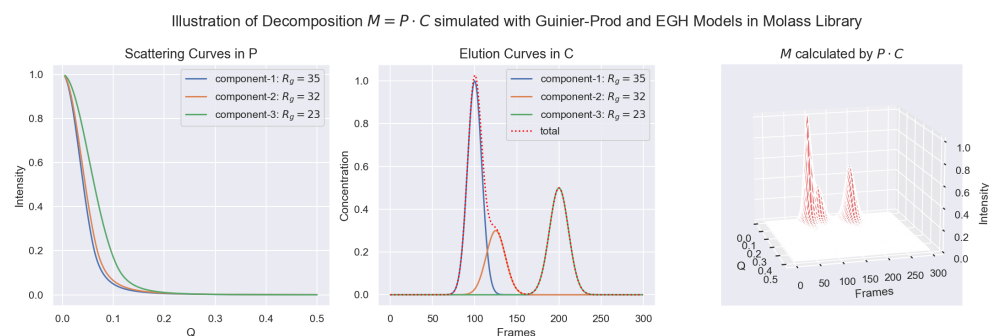


Figure 2: Illustration of decomposition using simulated data

Elution Curve Modeling

To address underdeterminedness and enhance interpretability, Molass Library incorporates several established elution curve models:

- **EGH:** Exponential Gaussian Hybrid (Lan & Jorgenson, 2001)
- **SDM:** Stochastic Dispersive Model (Felingner et al., 1999)
- **EDM:** Equilibrium Dispersive Model (Ur Rehman et al., 2021)
- **LKM:** Lumped Kinetic Model (Felingner et al., 2004)
- **GRM:** General Rate Model (Qamar et al., 2014)

EGH, with fewer constraints, is used as a first fitter which can be “upgraded” to other physical models. The theoretical model hierarchy $\text{EDM} \subset \text{LKM} \subset \text{GRM}$ reflects increasing physical detail (equilibrium dispersive $\rightarrow$ lumped kinetic $\rightarrow$ resolved film/pore transport), but the current implementation prioritizes analytical tractability: EDM supports nonlinear isotherms, while LKM and GRM are restricted to linear isotherms where characteristic-function methods apply. Combined, these models allow users to impose domain-specific constraints, thereby enhancing both the accuracy and the physical relevance of the chromatographic peak decomposition.

Availability and Documentation

Molass Library is freely available under an open-source license at <https://github.com/biosaxs-dev/molass-library>. Comprehensive documentation, including tutorials and theoretical background, is provided at [Molass Essence](#).

AI usage disclosure

Generative AI tools (GitHub Copilot with Claude Sonnet 4, Claude Opus 4, and earlier models including Claude 3.5 Sonnet and ChatGPT-4) were used for the following aspects of this submission:

- **Paper text:** AI assistance was used to improve English grammar, sentence structure, and clarity. All technical content, design decisions, and scientific claims were authored by the human authors.
- **Code generation:** AI contributed to implementing routine code (e.g., standard NumPy operations, Matplotlib boilerplate) as well as targeted bug fixes and API improvements identified through AI-driven usability testing. All core algorithms, elution curve models, and scientific logic were designed and directed by the human authors.
- **Model development:** AI collaboration was integral to the rapid development of LKM (Lumped Kinetic Model) and GRM (General Rate Model) support (2026). The library’s

scriptable architecture enabled iterative AI-assisted coding sessions that would have been impractical with GUI-only code. The human authors directed all model design decisions, numerical implementations, and validation procedures.

- **Documentation:** Significant portions of the tutorial notebooks and API documentation were initially drafted with AI assistance, then reviewed, corrected, and refined by the human authors.
- **AI-friendliness improvement cycle:** As described in the Software Design section, AI agents were used as systematic testers of the library's API. Usability issues discovered during AI-driven analysis sessions were logged as GitHub Issues and resolved iteratively, improving API discoverability and error reporting.

All AI-generated content was reviewed, validated, and edited by the human authors, who made all core design decisions and remain fully responsible for the accuracy and integrity of this submission. The development history, including all AI-assisted changes, is recorded in the repository's Git log and GitHub Issues.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP25K07250, and partially by the Research Support Project for Life Science and Drug Discovery (Basis for Supporting Innovative Drug Discovery and Life Science Research (BINDS)) from AMED under Grant Number JP25ama121001.

References

- Erb, D. (2025). *pybaselines: A Python library of algorithms for the baseline correction of experimental data* (Version v1.2.1). Zenodo. <https://doi.org/10.5281/zenodo.16790579>
- Felinger, A., Cavazzini, A., & Dondi, F. (2004). Equivalence of the microscopic and macroscopic models of chromatography: Stochastic–dispersive versus lumped kinetic model. *Journal of Chromatography A*, 1043, 149–157. <https://doi.org/10.1016/j.chroma.2004.05.081>
- Felinger, A., Cavazzini, A., Remelli, M., & Dondi, F. (1999). Stochastic–dispersive theory of chromatography. *Analytical Chemistry*, 71(20), 4472–4479. <https://doi.org/10.1021/ac990412u>
- Furukawa, A., Yonezawa, K., Negami, T., Yoshimura, Y., Hayashi, A., Nakayama, J., Adachi, N., Senda, T., Shimizu, K., Terada, T., & Shimizu, N. (2025). A dynamic structural unit of phase-separated heterochromatin protein 1 α as revealed by integrative structural analyses. *Nucleic Acids Research*, 53(6), gkaf154. <https://doi.org/10.1093/nar/gkaf154>
- Grant, T. D. (2018). Ab initio electron density determination directly from solution scattering data. *Nature Methods*, 15(3), 191–193. <https://doi.org/10.1038/nmeth.4581>
- Guinier, A. (1939). La diffraction des rayons x aux très petits angles : Application à l'étude de phénomènes ultramicroscopiques. *Ann. Phys.*, 11(12), 161–237. <https://doi.org/10.1051/anphys/193911120161>
- Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- Hopkins, J. B. (2024). *BioXTAS RAW 2: new developments for a free open-source program for small-angle scattering data reduction and analysis*. *Journal of Applied Crystallography*, 57(1), 194–208. <https://doi.org/10.1107/S1600576723011019>

- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>
- Jiang, X., Ogawa, T., Yonezawa, K., Danev, R., & Kikkawa, M. (2023). The two-step cargo recognition mechanism of heterotrimeric kinesin. *EMBO Reports*, 24(11), e56864. <https://doi.org/10.15252/embr.202356864>
- Konarev, P. V., Graewert, M. A., Jeffries, C. M., Fukuda, M., Cheremnykh, T. A., Volkov, V. V., & Svergun, D. I. (2021). EFAMIX, a tool to decompose inline chromatography SAXS data from partially overlapping components. *Protein Science*, 31, 269–282. <https://doi.org/10.1002/pro.4237>
- Kratky, O. (1963). X-ray small angle scattering with substances of biological interest in diluted solutions. *Progress in Biophysics and Molecular Biology*, 13, 105–IN1. [https://doi.org/10.1016/S0079-6107\(63\)80015-2](https://doi.org/10.1016/S0079-6107(63)80015-2)
- Lan, K., & Jorgenson, J. W. (2001). A hybrid of exponential and gaussian functions as a simple model of asymmetric chromatographic peaks. *Journal of Chromatography A*, 915(1), 1–13. [https://doi.org/10.1016/S0021-9673\(01\)00594-5](https://doi.org/10.1016/S0021-9673(01)00594-5)
- Manalastas-Cantos, K., Konarev, P. V., Hajizadeh, N. R., Kikhney, A. G., Petoukhov, M. V., Molodenskiy, D. S., Panjkovich, A., Mertens, H. D. T., Gruzinov, A., Borges, C., Jeffries, C. M., Svergun, D. I., & Franke, D. (2021). ATSAS 3.0: expanded functionality and new tools for small-angle scattering data analysis. *Journal of Applied Crystallography*, 54(1), 343–355. <https://doi.org/10.1107/S1600576720013412>
- Matsui, T., Rajkovic, I., Mooers, B. H. M., Liu, P., & Weiss, T. M. (2024). Adaptable SEC-SAXS data collection for higher quality structure analysis in solution. *Protein Science*, 33(4), e4946. <https://doi.org/10.1002/pro.4946>
- Meisburger, S. P., Xu, D., & Ando, N. (2021). REGALS: a general method to deconvolve X-ray scattering data from evolving mixtures. *IUCrJ*, 8(2), 225–237. <https://doi.org/10.1107/S2052252521000555>
- Panjkovich, A., & Svergun, D. I. (2018). CHROMIXS: automatic and interactive analysis of chromatography-coupled small-angle X-ray scattering data. *Bioinformatics*, 34(11), 1944–1946. <https://doi.org/10.1093/bioinformatics/btx846>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Penrose, R. (1955). A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3), 406–413. <https://doi.org/10.1017/S0305004100030401>
- Penrose, R. (1956). On best approximate solutions of linear matrix equations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 52(1), 17–19. <https://doi.org/10.1017/S0305004100030929>
- Qamar, S., Abbasi, J. N., Javeed, S., & Seidel-Morgenstern, A. (2014). Analytical solutions and moment analysis of general rate model for linear liquid chromatography. *Chemical Engineering Science*, 107, 192–205. <https://doi.org/10.1016/j.ces.2013.12.019>
- Shimizu, N., Yatabe, K., Nagatani, Y., Saijyo, S., Kosuge, T., & Igarashi, N. (2016). Software development for analysis of small-angle x-ray scattering data. *AIP Conference Proceedings*, 1741(1), 050017. <https://doi.org/10.1063/1.4952937>
- Truong, C., Oudre, L., & Vayatis, N. (2020). Selective review of offline change point detection methods. *Signal Processing*, 167, 107299. <https://doi.org/10.1016/j.sigpro.2019.107299>

- Ur Rehman, J., Muneer, A., & Qamar, S. (2021). Analysis of equilibrium dispersive model of liquid chromatography considering a quadratic-type adsorption isotherm. *Thermal Science*, 26(3), 2069–2080. <https://doi.org/10.2298/TSCI201229179U>
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., Walt, S. J. van der, Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... Contributors, S. 1.0. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17(3), 261–272. <https://doi.org/10.1038/s41592-019-0686-2>
- Yonezawa, K., Takahashi, M., Yatabe, K., Nagatani, Y., & Shimizu, N. (2023). MOLASS: Software for automatic processing of matrix data obtained from small-angle x-ray scattering and UV-visible spectroscopy combined with size-exclusion chromatography. *Biophysics and Physicobiology*, 20(1), e200001. <https://doi.org/10.2142/biophysico.bppb-v20.0001>