The Journal of Open Source Software

peaks: a Python package for analysis of angle-resolved photoemission and related spectroscopies

Phil D. C. King [1], **Brendan Edwards** [1], **Shu Mo** [1], **Tommaso Antonelli** [1], **Edgar Abarca Morales** [1], **Lewis Hart** [1], **and Liam Trzaska** [1]

1 SUPA, School of Physics and Astronomy, University of St Andrews, St Andrews, KY16 9SS, UK

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 06 August 2025
Published: unpublished

Summary

The electronic band structure, describing the motion and interactions of electrons in materials, dictates the electrical, optical, and thermodynamic properties of solids. Angle-resolved photoemission spectroscopy (ARPES) provides a direct experimental probe of such electronic band structures, and so is widely employed in the study of functional, quantum, and 2D materials (Damascelli et al., 2003; King et al., 2021; Sobota et al., 2021). peaks (**P**ython **E**lectron spectroscopy **A**nalysis by **K**ing group @ **S**t Andrews) provides a Python package for advanced data analysis of ARPES and related spectroscopic data. It facilitates the fast visualisation and analysis of multi-dimensional datasets, allows for the complex data hierarchy typical to ARPES experiments, and supports lazy data loading and parallel processing, reflecting the ever-increasing data volumes used in ARPES. It is designed to be run in an interactive notebook environment, with extensive inline and pop-out GUI support for data visualisation.

Statement of need

Over recent years, significant technological improvements have developed ARPES into a truly multidimensional spectroscopy. Besides the traditional resolution of energy and up to three momentum directions, temperature, spin, spatial, and temporal-dependent ARPES measurements are becoming increasingly common (King et al., 2021; Sobota et al., 2021), typically requiring efficient handling and advanced analysis of 3-, 4-, and higher-dimensional datasets. Extensive use of international light sources for performing ARPES measurements, during intensive experiment campaigns running over several days, further motivates a collaborative approach to performing data analysis. There is also an ever-increasing push to incorporate machine learning (ML) methods into the analysis pipeline (Ágústsson et al., 2025; Iwasawa et al., 2022; Kim et al., 2021; Melton et al., 2020), while greater transparency and reproducibility in ARPES data analysis can be ensured by the development and utilisation of open-source packages, with clear and transparent metadata handling (Scheffler et al., 2022).

The above requirements all motivate the use of Python as a modern approach to ARPES data analysis. To this end, several packages have been developed. `PyARPES` (Stansbury & Lanzara, 2020) represents a pioneering development in this direction. It appears to no longer be actively maintained by the original author, although a maintained fork does exist (Arafune, 2025). Despite many excellent features, it makes several fundamental convention choices (regarding angular and energy scales and units, alignments, and sign conventions) which, in our view, complicates its use when employed with multiple experimental setups as is typical in the ARPES community, while approximations are used in the critical momentum-space conversions. `pesto` (Polley, 2025) is an excellent easy-to-use alternative, but is heavily oriented towards use with data collected from the Bloch beamline of the Max-IV synchrotron. We have recently discovered `ERLabPy` (Han, 2025) which provides similar functionality to peaks,

although with some differences in the approach to handling the data (e.g. co-ordinate systems). The need to accommodate not only different data formats but also manage distinct angle and sign convention choices for data acquired at multiple facilities can add significant complexity for the user, in particular for on-the-fly processing: this is something that peaks attempts to simplify for the end user, aiding quick and efficient on-the-fly analysis e.g. for sample alignment during intense experimental runs. Other packages that we are aware of tend to focus on a subset of the functions required, e.g. for ARPES data analysis (Das, 2025) or visualistaion (Kramer & Chang, 2021).

peaks

peaks attempts to provide a relatively comprehensive suite of tools for ARPES and related spectroscopic data via a modular approach, supporting the experimentor from initial data acquisition, visualisation, and sample alignment through data processing and more advanced analysis. peaks builds heavily on the `xarray` package (Hoyer & Hamman, 2017), providing a powerful data structure for the N-D labelled data arrays common to ARPES data. This also supports the use of `dask` arrays (Dask Development Team, 2016) for lazy data loading and processing, e.g. for datasets that are beyond the available memory, or to facilitate parallel processing. peaks is intended to be run using interactive notebooks. Data is loaded into `xarray:DataArray`'s using location-specific data loaders to support multiple starting data formats and conventions, reflecting the heterogeneity in existing ARPES setups. Extensive metadata is included in the `DataArray` attributes, making use of `pydantic` (Colvin et al., 2025) models to ensure a consistent metadata framework, while `pint` (Grecco, 2025) is used for ensuring reliable handling of units in both the ARPES dataset and associated metadata. Data can also be loaded into `xarray:DataTree` structures, allowing the user flexibility over grouping data in configurations which reflect the data hierarchy of the underlying experiment, and permitting batch processing or metadata configuration. General data loaders exist for several of the core ARPES spectrometer manufacturers, as well as for the ARPES setups of several central facilities commonly used in the community. A class-based approach for the data loaders provides an efficient route to extending this to new setups in future. For data saved using the standard data formats of one of the common ARPES spectrometer manufactuers, implementing a new loader can be as simple as subclassing the relevant parent class and defining a few sign and unit conventions, while complete loaders can also be developed starting from bespoke data formats. peaks aims to maintain a record of processing steps that have been applied to the data, building up a detailed analysis history which can be easily inspected which – together with use in interactive notebooks – facilitates enhanced data provenance and effective collaborative working on ARPES data analysis.

The use of `xarray:DataArray` accessors allow easy chaining of analysis methods together for most functions. Extensive capabilities are included for data visualisation, including static plots and interactive tools for 2-, 3-, and 4-D datasets (see, e.g. Figure 1). Tools are included for aiding the experimenter in aligning samples for subsequent measurements, with care taken to handle the different conventions used at different experimental facilities in a way that facilitates both standardised data analysis and also the use of peaks for 'on-the-fly' analysis during experiments. Additional core functionality includes tools for ARPES-specific data selection (e.g. momentum (MDC) and energy (EDC) distribution curve extraction), merging, summation, and symmetrisation, data processing (e.g. momentum conversion and Fermi level corrections, data normalisation), and derivative-type methods to aid data visualisation. Capabilities for data fitting (including parallel processing and fitting of lazily-loaded data - see e.g. Figure 2) are included, building on the extensive `lmfit` (Newville et al., 2025) package. Core capabilities for processing time-resolved ARPES data are included, while specific data selection and helper methods are included for spatially-resolved ARPES. Initial functionality (principal component analysis, clustering, and denoising) is included for related unsupervised machine learning analysis of such spatially-resolved ARPES data, built around standard Scikit-learn framework (Pedregosa et al., 2011).

The Journal of Open Source Software

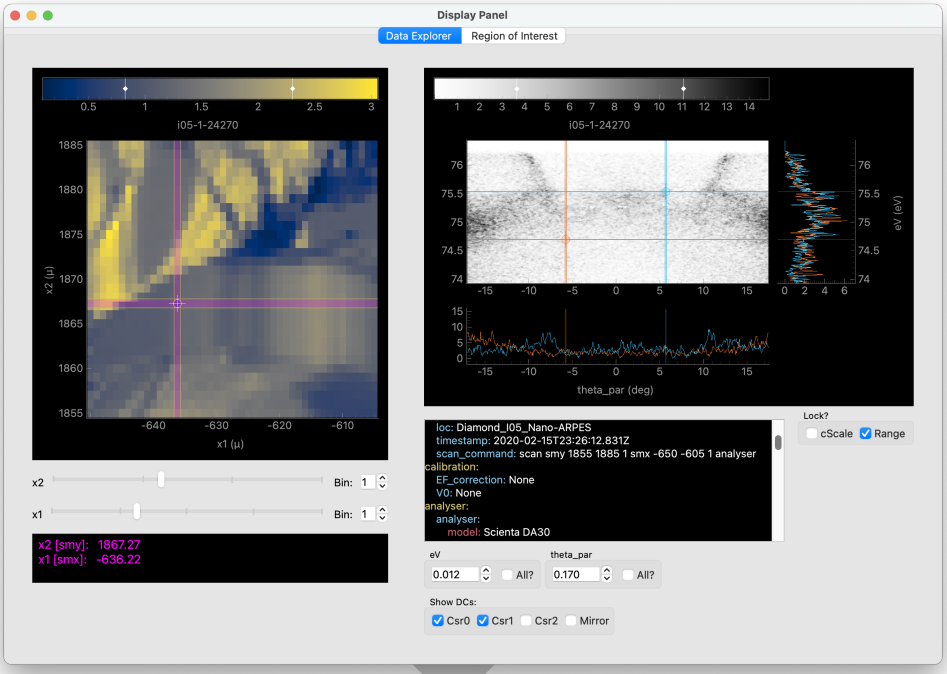

Figure 1: Example data visualisation tool for a 4D dataset (spatially-resolved ARPES). The tab shown allows interactive data exploration, while the secondary tab facilitates region of interest analysis.

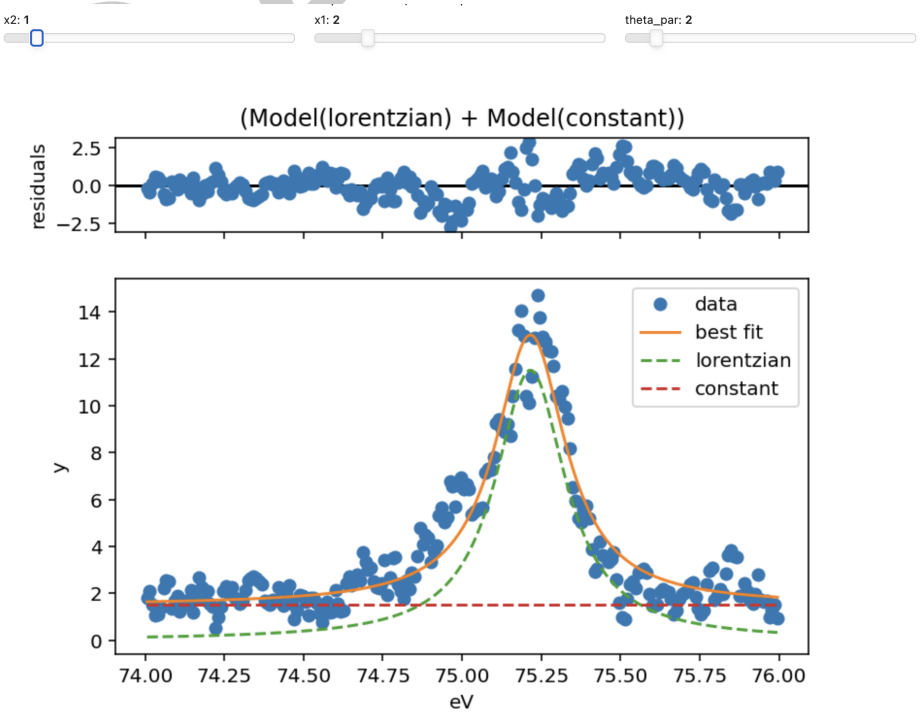

Figure 2: Example fitting of lazily-loaded data. The relevant individual EDC is loaded into memory and the fit performed only when selected by the sliders, facilitating fitting even of very large datasets. Similar approaches allow the parallel fitting of individual EDCs or MDCs from across a large, e.g. spatially-resolved, dataset.

In the future, peaks could be augmented with additional ML approaches tailored to ARPES data analysis, facilitated by the standard `xarray`-based data structures used. The incorporation of additional data structures and functionality for processing spin-resolved ARPES data is also planned.

Acknowledgements

We acknowledge valuable discussions, suggestions, and bug reports/fixes from Marieke Visscher, Gesa Siemann, Naina Kushwaha, Bruno Saika, Phil Murgatroyd, Anđela Živanović, and Igor Marković. We are grateful to past and present members of the King group at the University of St Andrews and collaborators for measurements of the experimental ARPES data that were utilised in the development of the code. We thank the UK Engineering and Physical Sciences Research Council (Grant Nos. EP/X015556/1, EP/T02108X/1, and EP/R025169/1), the Leverhulme Trust (Grant Nos. RL-2016-006 and RPG-2023-256), and the European Research Council (through the QUESTDO project, 714193) for financial support.

References

Ágústsson, S. Ý., Haque, M. A., Truong, T. T., Bianchi, M., Klyuchnikov, N., Mottin, D., Karras, P., & Hofmann, P. (2025). An autoencoder for compressing angle-resolved photoemission spectroscopy data. *Mach. Learn.: Sci. Technol.*, *6*(1), 015019. https://doi.org/10.1088/2632-2153/ada8f2

Arafune, R. (2025). PyARPES corrected (V4). In *GitHub repository*. GitHub. https://github.com/arafune/arpes

Colvin, S., Jolibois, E., Ramezani, H., Garcia Badaracco, A., Dorsey, T., Montague, D., Matveenko, S., Trylesinski, M., Runkle, S., Hewitt, D., Hall, A., & Plot, V. (2025). *Pydantic* (Version v2.11.7). https://github.com/pydantic/pydantic

Damascelli, A., Hussain, Z., & Shen, Z.-X. (2003). Angle-resolved photoemission studies of the cuprate superconductors. *Rev. Mod. Phys.*, *75*(2), 473–541. https://doi.org/10.1103/RevModPhys.75.473

Das, P. (2025). ARPES python tools. In *GitHub repository*. GitHub. https://github.com/pranabdas/arpespythontools

Dask Development Team. (2016). *Dask: Library for dynamic task scheduling*. http://dask.pydata.org

Grecco, H. E. (2025). Pint. In *GitHub repository*. GitHub. https://github.com/hgrecco/pint

Han, K. (2025). ERLabPy. In *GitHub repository*. GitHub. https://github.com/kmnhan/erlabpy

Hoyer, S., & Hamman, J. (2017). Xarray: N-D labeled arrays and datasets in Python. *Journal of Open Research Software*, *5*(1). https://doi.org/10.5334/jors.148

Iwasawa, H., Ueno, T., Masui, T., & Tajima, S. (2022). Unsupervised clustering for identifying spatial inhomogeneity on local electronic structures. *Npj Quantum Mater.*, *7*(1), 24. https://doi.org/10.1038/s41535-021-00407-5

Kim, Y., Oh, D., Huh, S., Song, D., Jeong, S., Kwon, J., Kim, M., Kim, D., Ryu, H., Jung, J., Kyung, W., Sohn, B., Lee, S., Hyun, J., Lee, Y., Kim, Y., & Kim, C. (2021). Deep learning-based statistical noise reduction for multidimensional spectral data. *Review of Scientific Instruments*, *92*(7), 073901. https://doi.org/10.1063/5.0054920

King, P. D. C., Picozzi, S., Egdell, R. G., & Panaccione, G. (2021). Angle, Spin, and Depth Resolved Photoelectron Spectroscopy on Quantum Materials. *Chem. Rev.*, *121*(5),

2816–2856. https://doi.org/10.1021/acs.chemrev.0c00616

Kramer, K., & Chang, J. (2021). Visualization of multi-dimensional data – the data-slicer package. *Journal of Open Source Software*, *6*(60), 2969. https://doi.org/10.21105/joss.02969

Melton, C. N., Noack, M. M., Ohta, T., Beechem, T. E., Robinson, J., Zhang, X., Bostwick, A., Jozwiak, C., Koch, R. J., Zwart, P. H., Hexemer, A., & Rotenberg, E. (2020). K-means-driven Gaussian Process data collection for angle-resolved photoemission spectroscopy. *Mach. Learn.: Sci. Technol.*, *1*(4), 045015. https://doi.org/10.1088/2632-2153/abab61

Newville, M., Otten, R., Nelson, A., Stensitzki, T., Ingargiola, A., Allan, D., Fox, A., Carter, F., & Rawlik, M. (2025). *LMFIT: Non-Linear Least-Squares Minimization and Curve-Fitting for Python* (Version 1.3.3).

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, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Polley, C. (2025). Pesto: Photoemission spectroscopy tools. In *GitLab repository*. GitLab. https://gitlab.com/flashingLEDs/pesto

Scheffler, M., Aeschlimann, M., Albrecht, M., Bereau, T., Bungartz, H.-J., Felser, C., Greiner, M., Groß, A., Koch, C. T., Kremer, K., Nagel, W. E., Scheidgen, M., Wöll, C., & Draxl, C. (2022). FAIR data enabling new horizons for materials research. *Nature*, *604*(7907), 635–642. https://doi.org/10.1038/s41586-022-04501-x

Sobota, J. A., He, Y., & Shen, Z.-X. (2021). Angle-resolved photoemission studies of quantum materials. *Rev. Mod. Phys.*, *93*(2), 025006. https://doi.org/10.1103/RevModPhys.93.025006

Stansbury, C., & Lanzara, A. (2020). PyARPES: An analysis framework for multimodal angle-resolved photoemission spectroscopies. *SoftwareX*, *11*, 100472. https://doi.org/10.1016/j.softx.2020.100472