The Journal of Open Source Software

easyunfold: A Python package for unfolding electronic band structures

Bonan Zhu[1,2,3], **Seán R. Kavanagh**[1,2,4], **and David Scanlon**[1,2,3,5]

1 Department of Chemistry, University College London, London, United Kingdom **2** Thomas Young Centre, University College London, London, United Kingdom **3** The Faraday Institution, Didcot, United Kingdom **4** Department of Materials, Imperial College London, London, United Kingdom **5** School of Chemistry, University of Birmingham, Birmingham, United Kingdom

DOI: 10.21105/joss.05974

Software

- Review
- Repository
- Archive

Editor: Mojtaba Barzegari

Reviewers:

- @srmnitc
- @awvwgk

Submitted: 12 September 2023
Published: 03 January 2024

Summary

The electronic band structure is an important property for understanding and designing solid crystalline materials in many fields such as photovoltaic, catalytic, thermoelectric and transparent-conducting materials. Obtaining the band structure for an ideal crystal through first-principles density functional theory (DFT) calculations is a well-established routine operation(Ganose et al., 2018). However, the materials of interest are often complex and the simulation cells may contain multiple primitive cells of the archetypal structure when, for example, modelling disordered or defective materials(Kim et al., 2020). Repeating the unit cell in real space results in folded band structures, as illustrated in Figure 1, complicating its interpretation and analysis. Band structure unfolding maps the electronic structure from supercell calculations back to the reciprocal lattice of the primitive cell, thereby enabling researchers to understand structure-property relationships and compare the effect of various crystal imperfections on an equal footing. With `easyunfold`, we provide a simple, easy-to-use, yet powerful and flexible tool which implements the band structure unfolding workflow using plane-wave DFT codes, from input file generation to publication-quality plotting.

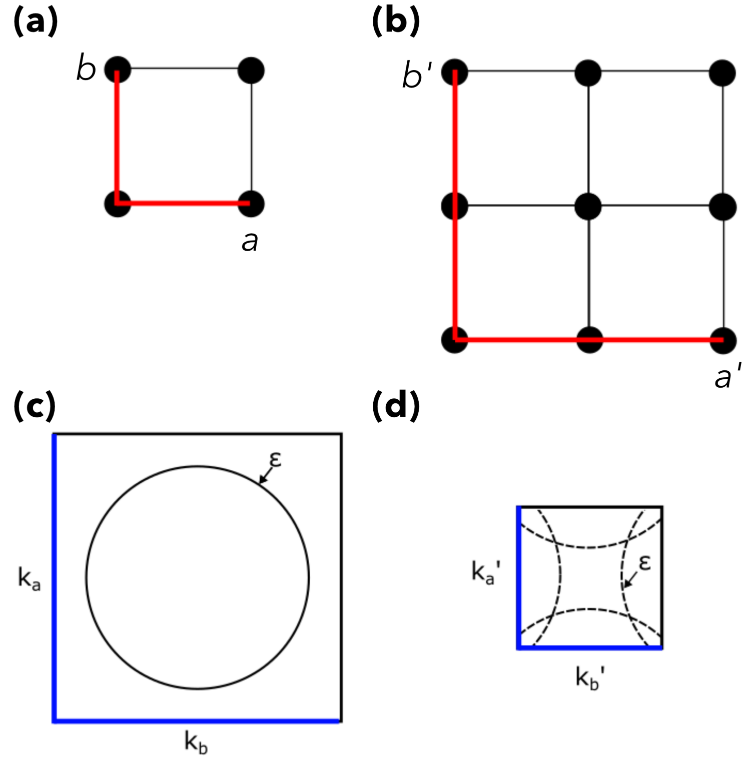

Figure 1: Schematics of band folding in a 2D lattice: (a) The primitive cell in real space; (b) The 2×2 supercell in real space. (c) A slice (ε) of the band structure in the Brillouin zone of the primitive cell; (d) The same band structure slice (ε) in the Brillouin zone of the supercell. Band structure unfolding is the process of transforming the band structure slice in (d) back to the Brillouin zone of the primitive cell (i.e. the form shown in (c)).

Statement of need

There are existing packages that provide band structure unfolding capabilities, such as `BandUp` (Tsirkin, 2017), and `VaspBandUnfolding` (QijingZheng, 2020). `easyunfold` is written in Python with a focus on user-friendliness, data provenance, reproducibility, and publication-quality figure generation. An example output of the effective band structure produced is shown in Figure 2 for a $2 \times 2 \times 2$ MgO supercell containing a neutral oxygen vacancy. Figure 3 shows the orbital-projected effective band structure of a $\mathrm{Cs_2(Sn_{0.5}, Ti_{0.5})Br_6}$ vacancy-ordered perovskite alloy (Kavanagh et al., 2022; Liga et al., 2023), in the Brillouin zone of the primitive $Fm\bar{3}m$ unit cell. A key feature of `easyunfold` is to provide data serialization compliant with the FAIR principles (Wilkinson et al., 2016). Both the input settings and calculated outputs are stored in a single JSON file. This enables the unfolded band structure to be re-plotted and further analysed without reprocessing the wave function data, which can be time-consuming and require large storage space.

The package is designed with flexibility in mind. `easyunfold` can split a supercell calculation into multiple runs. This functionality is essential for compute-heavy and memory-intensive hybrid functional calculations, particularly if spin-orbit coupling is used, where the required computational resources would quickly become intractable otherwise. Upon completion of all calculations, their wave functions can be collected to generate a single effective band structure. The atomic and orbital projections can be used to colour the effective band structure (Figure 2), and the electronic density of states (DOS) can optionally be plotted alongside (Figure 3), which can provide valuable information for analysing the underlying structure-property relationships.

We chose Python as the programming language due to its low barrier-to-entry, flexibility and popularity in the materials modelling field. An object-oriented approach is used when designing the package to allow abstractions when reading and processing wave function data. The code currently supports two of the most popular DFT codes, VASP (Kresse & Furthmüller, 1996a, 1996b) and CASTEP (Clark et al., 2005), and others can be added with a small amount of coding effort. `easyunfold` depends on common scientific computing packages such as `numpy` (Harris et al., 2020) and `matplotlib` (Hunter, 2007). The Atomic Simulation Environment (ase) (Larsen et al., 2017) is used for reading input crystal structures from a wide range of formats.

`easyunfold` is designed for researchers with or without prior knowledge of Python. A command-line interface is provided as the primary way of using the package. Thanks to the Python API, `easyunfold` can be easily integrated with workflow engines such as `AiiDA` (Huber et al., 2020) and `atomate` (Mathew et al., 2017) as well as disorder/defect modelling codes such as `icet` (Ångqvist et al., 2019) and `doped` (Kavanagh, 2022). `easyunfold` has been used in several scientific publications (Huang et al., 2022; Nicolson et al., 2023; Wang et al., 2022) and graduate student research projects.

The combination of user-friendliness, flexibility, and efficiency will improve the accessibility of band structure unfolding and help train new researchers.

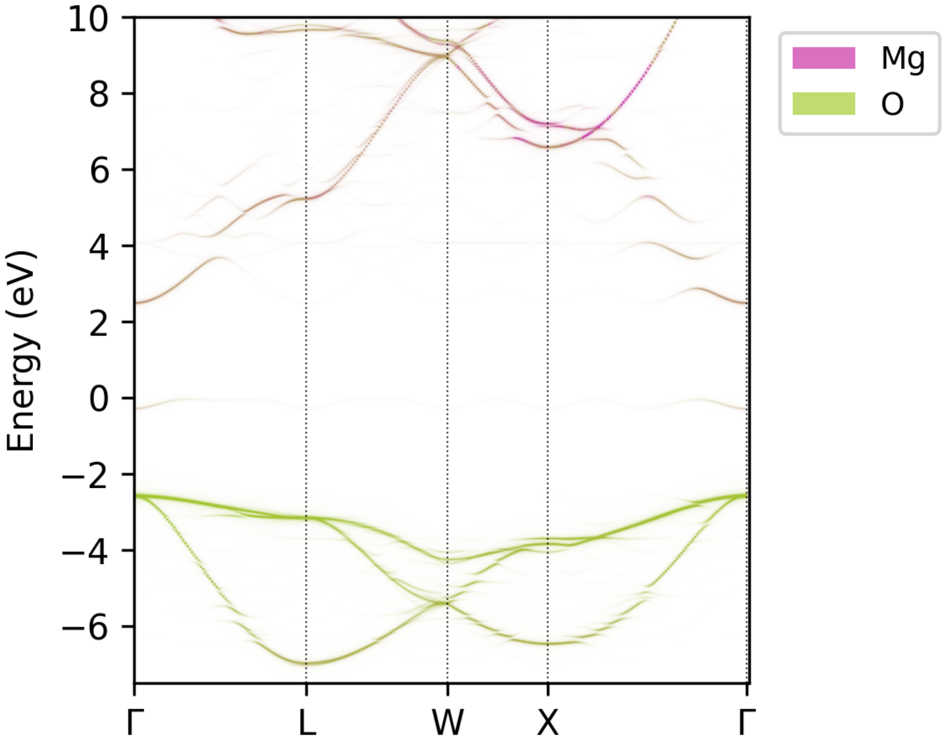

Figure 2: Atom-projected effective band structure of a $2 \times 2 \times 2$ MgO supercell showing a localised mid-gap state resulting from a neutral oxygen vacancy (using a relatively small supercell containing 63 atoms).

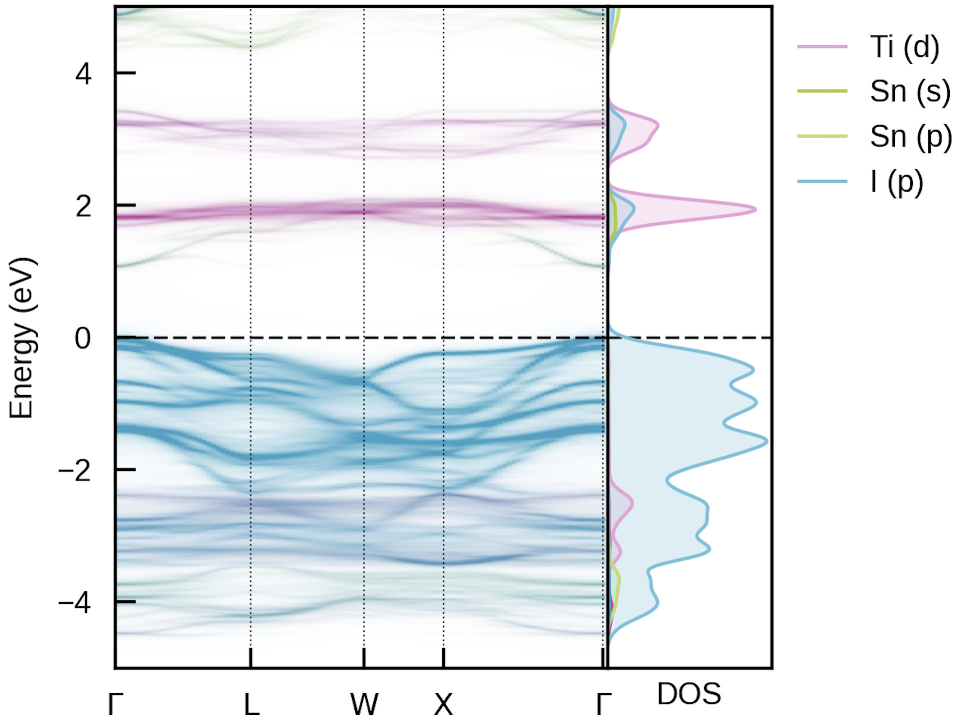

Figure 3: Orbital-projected effective band structure of a disordered $\mathrm{Cs_2(Sn,Ti)Br_6}$ vacancy-ordered perovskite alloy (Kavanagh et al., 2022; Liga et al., 2023).

Theory

The mathematics of band structure unfolding has been discussed in detail in the work of (Popescu & Zunger, 2012). Here, we only give a brief summary of the key ingredients.

Our goal is to reconstruct the energies of the Kohn-Sham states in the reciprocal lattice of the primitive cell ($E(\vec{k})$), given a more complex (folded) $E(\vec{K})$ of the supercell.

Each $\vec{k}$-point in the primitive cell's first Brillouin zone can be mapped to a $\vec{K}$ in that of the supercell:

$$\vec{K} = \vec{k} - \vec{G}_0.$$

with $\vec{G}_0$ being a reciprocal lattice vector of the supercell. Conversely, each $\vec{K}$ in the supercell can be unfolded into a set of $\vec{k}_i$:

$$\vec{k}_i = \vec{K} + \vec{G}_i, \quad i = 1, ..., N$$

where $\vec{k}_i$ is a set of N $\vec{k}$-points in the Brillouin zone of the primitive cell, with N equalling the number of primitive cells that the supercell includes in real space. For a given $\vec{k}$, there is a unique $\vec{K}$ that it folds to, as the first equation can be seen as to *wrap* the $\vec{k}$ into the smaller reciprocal unit cell of the supercell. On the other hand, a single $\vec{K}$ may map to multiple $\vec{k}_i$s, as different $\vec{k}$-points in the Brillouin zone of the primitive cell (Figure 1c) can fold onto the same point in that of the supercell (Figure 1d).

To establish the relationship between $E(\vec{k})$ and $E(\vec{K})$, one can project $\langle \vec{K} m|$, where m labels the band, on a set of primitive cell Bloch states $\langle \vec{k}_i|$ and compute the spectral weight:

$$P_{\vec{K}m}(\vec{k}_i) = \sum_n |\langle \vec{K}m|\vec{k}_i n\rangle|^2.$$

where P represents the probability of finding a set of primitive cell states $\langle\vec{k}_i n$ contributing to the supercell state $\langle\vec{K}m|$, or the amount of Bloch character $\vec{k}_i$ preserved in $\langle\vec{K}m|$ at the same energy.

Presenting the spectral weights directly in two-dimensional plots can be problematic due to the existence of states (m) with degenerate or closely-spaced energies. A more interpretable representation of the effective band structure is the spectral function, defined as:

$$A(\vec{k}_i, E) = \sum_m P_{\vec{K}m}(\vec{k}_i)\delta(E_m - E).$$

where A is the total intensity summed from all contributing $P_{\vec{K}m}$ at $\vec{k}_i$ and energy E. In practice, the δ function is replaced with a Gaussian or Lorentzian function which smears the contribution across a discretised energy grid.

For a plane wave basis, the $P_{\vec{K}m}(\vec{k}_i)$ can be calculated as (equation 15 (Popescu & Zunger, 2012)):

$$P_{\vec{K}m}(\vec{k}_i) = \sum_{\vec{g}} |C_{\vec{K}m}(\vec{g} + \vec{G}_i)|^2,$$

where $\vec{g}$ are the reciprocal lattice vectors of the primitive cell, and $C_{\vec{K}m}$ is the plane wave coefficient. Since the supercell lattice is commensurate with that of the primitive cell, the plane wave coefficients to be summed, $C_{\vec{K}m}(\vec{g} + \vec{G}_i)$, are included in the basis set of the supercell. This means that a single supercell calculation producing the wave function (plane wave coefficients) at each $\vec{K}$ required is sufficient for reconstructing the unfolded band structure.

The path of $\vec{k}$ in the primitive cell is often constructed with its symmetry in mind, and each $\vec{k}$ along the path can be mapped to multiple equivalent $\vec{k}_s$ points given the symmetry of the reference structure. However, these $\vec{k}_s$ points may no longer be equivalent if the supercell has broken symmetry – which is almost always the case when performing band structure unfolding. This can be addressed by first expanding each $\vec{k}$ based on the symmetry operations of the primitive cell, followed by a reduction using the symmetry of the supercell. The spectral weight at each $\vec{k}$ is then a weighted combination of that from a set of $\vec{k}_s'$ points that are inequivalent under the symmetry of the supercell.

Acknowledgements

We acknowledge bug fixes from Adair Nicolson and help from Joe Willis and Sabrine Hachmioune in testing the code and providing feedback.

References

Ångqvist, M., Muñoz, W. A., Rahm, J. M., Fransson, E., Durniak, C., Rozyczko, P., Rod, T. H., & Erhart, P. (2019). ICET – a python library for constructing and sampling alloy cluster expansions. *Advanced Theory and Simulations*, *2*(7), 1900015. https://doi.org/10.1002/adts.201900015

Clark, S. J., Segall, M. D., Pickard, C. J., Hasnip, P. J., Probert, M. I. J., Refson, K., & Payne, M. C. (2005). *Zeitschrift Für Kristallographie - Crystalline Materials*, *220*(5-6), 567–570. https://doi.org/10.1524/zkri.220.5.567.65075

Ganose, A. M., Jackson, A. J., & Scanlon, D. O. (2018). Sumo: Command-line tools for plotting and analysis of periodic *ab initio* calculations. *Journal of Open Source Software*, *3*(28), 717. https://doi.org/10.21105/joss.00717

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., Fernández del Río, J., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*, 357–362. https://doi.org/10.1038/s41586-020-2649-2

Huang, Y.-T., Kavanagh, S. R., Righetto, M., Rusu, M., Levine, I., Unold, T., Zelewski, S. J., Sneyd, A. J., Zhang, K., Dai, L., Britton, A. J., Ye, J., Julin, J., Napari, M., Zhang, Z., Xiao, J., Laitinen, M., Torrente-Murciano, L., Stranks, S. D., … Hoye, R. L. Z. (2022). Strong absorption and ultrafast localisation in $NaBiS_2$ nanocrystals with slow charge-carrier recombination. *Nature Communications*, *13*(1), 4960. https://doi.org/10.1038/s41467-022-32669-3

Huber, S. P., Zoupanos, S., Uhrin, M., Talirz, L., Kahle, L., Häuselmann, R., Gresch, D., Müller, T., Yakutovich, A. V., Andersen, C. W., Ramirez, F. F., Adorf, C. S., Gargiulo, F., Kumbhar, S., Passaro, E., Johnston, C., Merkys, A., Cepellotti, A., Mounet, N., … Pizzi, G. (2020). AiiDA 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance. *arXiv:2003.12476 [Cond-Mat]*. http://arxiv.org/abs/2003.12476

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

Kavanagh, S. R. (2022). Defect oriented python environment distribution (doped). In *GitHub repository*. GitHub. https://github.com/SMTG-Bham/doped

Kavanagh, S. R., Savory, C. N., Liga, S. M., Konstantatos, G., Walsh, A., & Scanlon, D. O. (2022). Frenkel Excitons in Vacancy-Ordered Titanium Halide Perovskites (Cs_2TiX_6). *The Journal of Physical Chemistry Letters*, *13*(47), 10965–10975. https://doi.org/10.1021/acs.jpclett.2c02436

Kim, S., Hood, S. N., Park, J.-S., Whalley, L. D., & Walsh, A. (2020). Quick-start guide for first-principles modelling of point defects in crystalline materials. *Journal of Physics: Energy*, *2*(3), 036001. https://doi.org/10.1088/2515-7655/aba081

Kresse, G., & Furthmüller, J. (1996a). Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Computational Materials Science*, *6*(1), 15–50. https://doi.org/10.1016/0927-0256(96)00008-0

Kresse, G., & Furthmüller, J. (1996b). Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys. Rev. B*, *54*, 11169–11186. https://doi.org/10.1103/PhysRevB.54.11169

Larsen, A. H., Mortensen, J. J., Blomqvist, J., Castelli, I. E., Christensen, R., Dułak, M., Friis, J., Groves, M. N., Hammer, B., Hargus, C., Hermes, E. D., Jennings, P. C., Jensen, P. B., Kermode, J., Kitchin, J. R., Kolsbjerg, E. L., Kubal, J., Kaasbjerg, K., Lysgaard, S., … Jacobsen, K. W. (2017). The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter*, *29*(27), 273002. http://stacks.iop.org/0953-8984/29/i=27/a=273002

Liga, S. M., Kavanagh, S. R., Walsh, A., Scanlon, D. O., & Konstantatos, G. (2023). Mixed-Cation Vacancy-Ordered Perovskites ($Cs_2Ti_{1-x}Sn_xX_6$; X = I or Br): Low-Temperature Miscibility, Additivity, and Tunable Stability. *The Journal of Physical Chemistry C*. https://doi.org/10.1021/acs.jpcc.3c05204

Mathew, K., Montoya, J. H., Faghaninia, A., Dwarakanath, S., Aykol, M., Tang, H., Chu, I., Smidt, T., Bocklund, B., Horton, M., Dagdelen, J., Wood, B., Liu, Z.-K., Neaton, J., Ong, S. P., Persson, K., & Jain, A. (2017). Atomate: A high-level interface to generate, execute, and analyze computational materials science workflows. *Computational Materials Science*, *139*, 140–152. https://doi.org/10.1016/j.commatsci.2017.07.030

Nicolson, A., Breternitz, J., Kavanagh, S. R., Tomm, Y., Morita, K., Squires, A. G., Tovar, M., Walsh, A., Schorr, S., & Scanlon, D. O. (2023). Interplay of static and dynamic disorder in the mixed-metal chalcohalide $\mathrm{Sn_2SbS_2I_3}$. *Journal of the American Chemical Society*. https://doi.org/10.1021/jacs.2c13336

Popescu, V., & Zunger, A. (2012). Extracting E versus $\vec{k}$ effective band structure from supercell calculations on alloys and impurities. *Physical Review B*, *85*(8), 085201. https://doi.org/10.1103/PhysRevB.85.085201

QijingZheng. (2020). VaspBandUnfolding. In *GitHub repository*. GitHub. https://github.com/QijingZheng/VaspBandUnfolding

Tsirkin, S. (2017). BandUP: Band unfolding code for plane-wave based calculations. In *GitHub repository*. GitHub. https://github.com/band-unfolding/bandup

Wang, Y., Kavanagh, S. R., Burgués-Ceballos, I., Walsh, A., Scanlon, D. O., & Konstantatos, G. (2022). Cation disorder engineering yields $\mathrm{AgBiS_2}$ nanocrystals with enhanced optical absorption for efficient ultrathin solar cells. *Nature Photonics*, *16*(3), 235–241. https://doi.org/10.1038/s41566-021-00950-4

Wilkinson, M. D., Dumontier, M., Aalbersberg, Ij. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., Silva Santos, L. B. da, Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., … Mons, B. (2016). The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, *3*(1), 160018. https://doi.org/10.1038/sdata.2016.18