The Journal of Open Source Software

ThermoParser: Streamlined Analysis of Thermoelectric Properties

Kieran B. Spooner[1,2,3], **Maud Einhorn**[2,3], **Daniel W. Davies**[2,3], **and David O. Scanlon**[1,2,3]

1 School of Chemistry, University of Birmingham, Birmingham B15 2TT, United Kingdom **2** Department of Chemistry, University College London, 20 Gordon Street, London WC1H 0AJ, United Kingdom **3** Thomas Young Centre, University College London, Gower Street, London WC1E 6BT, United Kingdom

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Mojtaba Barzegari

Reviewers:

- @enricgrau
- @fnattino
- @espottesmith

Submitted: 08 January 2024
Published: unpublished

Summary

Thermoelectric materials, which convert heat into electricity, could be an important renewable energy source to help slow the encroaching climate crisis, not only by displacing fossil fuels, but by recycling waste heat, which makes up around 50 % of generated energy (Firth et al., 2019). With the growing computational capacity and development of several codes to calculate the key properties of thermoelectrics, they have become an increasingly popular area of computational materials research in recent years. Thermal transport packages include Phonopy (Togo & Tanaka, 2015), Phono3py (Togo et al., 2015), ShengBTE/ almaBTE (Li et al., 2014), ALAMODE (Tadano et al., 2014), TDEP (Hellman et al., 2011) and HiPhive (Eriksson et al., 2019); and electronic transport packages include BoltzTraP (G. K. H. Madsen & Singh, 2006), BoltzTraP2 (G. K. Madsen et al., 2018), EPW (Noffsinger et al., 2010), EPA (Samsonidze & Kozinsky, 2018), EPIC STAR (Deng et al., 2020), AMSET (Alex M. Ganose et al., 2021), Perturbo (Zhou et al., 2021), TOSSPB (Pöhls & Mozharivskyj, 2022) and ElecTra (Graziosi et al., 2023). While separate packages are required for such different calculations, this makes data analysis complex, needing to load in different file formats, account for different data arrangements (e.g. array shapes), and convert to consistent units, even before one begins analysing anything. ThermoParser deals with these time-consuming and error-prone problems by loading data from multiple codes into a consistent data format with informative metadata, and facilitates the post-processing of thermoelectric properties by using this to accurately calculate and visualise them through an easy-to-use command-line interface (CLI) and a fully customisable Python package. Some of its utility can be seen by its use in the literature (sometimes under its former name, ThermoPlotter) (Brlec et al., 2022; Han et al., 2024; Herring Rodriguez et al., 2023; Kavanagh et al., 2021; Spooner et al., 2021; Willis et al., 2023).

Statement of Need

To the best of our knowledge no package exists for processing data from the vastly different sets of calculations needed to study thermoelectrics computationally. While there are several codes to aid with individual aspects of thermoelectric calculations, such as the inbuilt analysis sections of Phonopy and AMSET or packages such as Phono3py-Power-Tools (Skelton, 2020) and sumo (Alex M. Ganose et al., 2018), they are specialised to either the phononic or electronic side. ThermoParser brings three key novelties to the existing software landscape:

- The automatic parsing of outputs from multiple codes for both electronic and phononic calculations;
- A data system which is transparent in the origin, arrangement and units of the data,

customisable, and accessible regardless of Python aptitude;
- Plotting tools for the creation of publication-ready figures through an intuitive Python API, accessible to Python novices while fully customisable for making complex, information-rich graphics, with the most common plots also availble via a CLI.

To complement these capabilities, there is also a range of ancillary functions which streamline all parts of the process.

ThermoParser

ThermoParser is a Python package for analysing and plotting thermoelectric properties. The main dependencies are matplotlib (Hunter, 2007) for plotting, pymatgen (Ong et al., 2013) for symmetry analysis, numpy for calculations and click for the CLI. The package interfaces with Phonopy, Phono3py, AMSET and BoltzTraP.

The package is modular, with a separate function for loading from each code, plotting each graph-type and preparing each axis arrangement, as well as numerous helper functions for calculating properties, manipulating data, and fomatting outputs. Therefore, in order to add support for a new code, calculated property or plot-type, one needs only create a single function able to read/write the common format, and perhaps some metadata, and the rest should just work. Everything loaded into the common format has a `meta` directory, containing all the metadata needed to understand the data: the code it was loaded from, the shape of the array (e.g. if the first index is temperature) and the units, as well as any conditions imposed, such as if the data has been reduced to a particular temperature. All dependent variables are also loaded by default. A `tprc.yaml` config file is supplied, which enables the user to automatically convert units, and update the units as they appear in the metadata and on axis labels. This data is used by CLI functions provided to retrieve data from files, which verbosely describes the conditions under which it was measured for maximum transparency.

Plotting simple plots can be done easily via the CLI, but the Python interface is designed to be accessible to those unfamiliar with Python, with the simplest plots requiring just four lines of code. More complex plots can be achieved by making full use of the Python API, but oftentimes still only require adding an extra `plot` line. In general, raw data is converted to high-quality plots in four stages:

1. Axes: Choose pre-sized axes suitable for publication or presentation from `tp.axes`.
2. Load: Use the `data.load` module to parse outputs of other codes, standardising variable names, formatting and ensuring consistency across code versions. There is also a module which efficiently runs BoltzTraP and saves the output.
3. Add: Use the `plot` module to add plots to the axes. Options exist for scaling data, multiple plots per set of axes, and other customisation.
4. Save: Use `fig.savefig` or similar.

The `tprc.yaml` file allows users to set a range of defaults, including axis labels, tick locators, style sheets and more. While a gallery is curated to demonstrate all plotting functionality, we highlight some key examples in Figures 1 and 2.

Figure 1a shows a phonon dispersion with an element-decomposed DoS sharing the y-axis. Different environments for the same atom type can be specified, if desired, and it is possible to overlay multiple dispersions to assess calculation convergence. Figure 1b shows the same dispersion with lifetime projected on the colour axis. Data from both Phono3py and Phonopy is parsed, and ThermoParser internally calculates mean free path and phonon lifetime. Several other ways of projecting a range of properties onto a phonon dispersion are implemented, including the broadened bands plot (`tp.plot.frequency.add_wideband`), which is more commonly seen in the literature (Togo et al., 2015). This also demonstrates the utility of ThermoParser's consistent data format: as well as Phono3py data, Gruneisen parameter data from Phonopy can be projected onto phonon dispersions in the same way, by changing only which data is loaded and setting `quantity='gruneisen'` rather than `'lifetime'`.

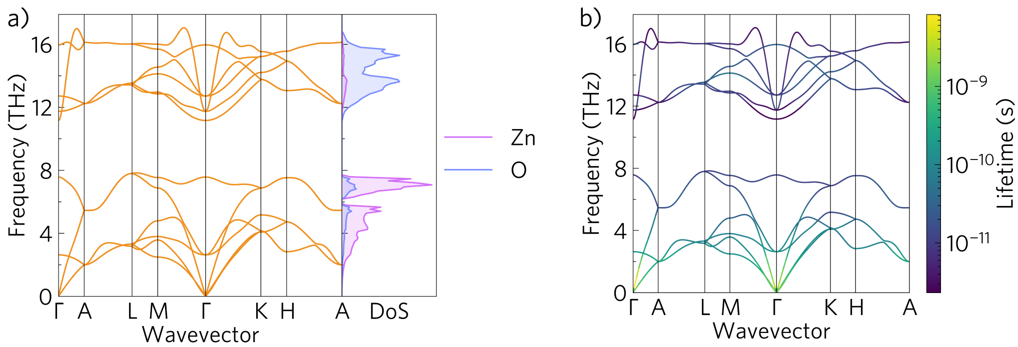

Figure 1: Phonon dispersions for ZnO with a) DoS and b) phonon lifetime projected on the colour axis.

Figure 2 is a waterfall plot of mean free path against frequency overlaying a DoS plot, clearly showing the relationship between elemental composition and scattering. Scaling the linear-scaled DoS data to the log-scaled waterfall axes would be time-consuming on a case-by-case basis, whereas the `tp.plot.frequency.add_dos` function wil autodetect the data range and rescale appropriately if the `scale` argument is set to `True`.

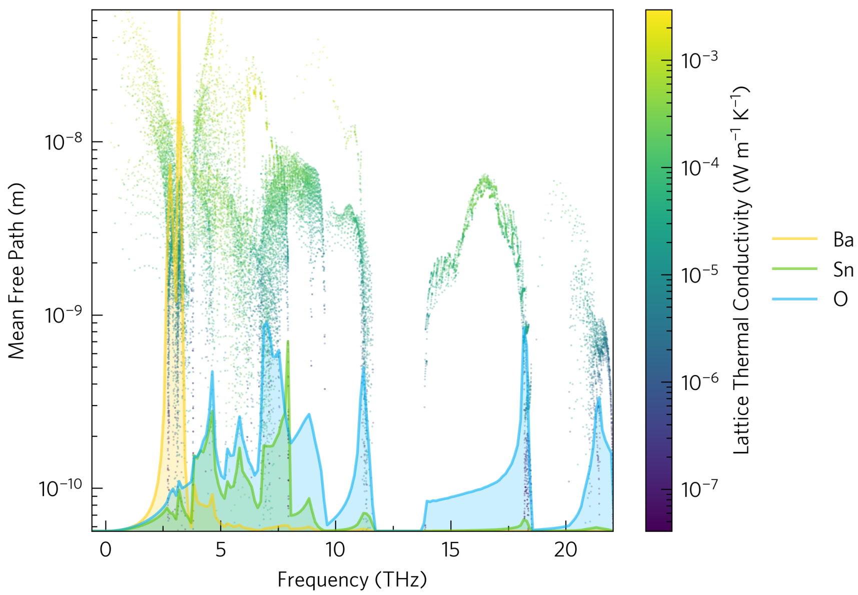

Figure 2: Waterfall plot overlaid on a DoS for $BaSnO_3$.

In the future, ThermoParser could be expanded to include an increased number of analysis types and supported codes. On top of this, support for uploading experimental data into the ThermoParser format, including the appropriate metadata, could allow easier comparison of theoretical and experimental results.

Author Contributions

K.B.S.: Conceptualization, data curation, formal analysis, investigation, methodology, software, validation, visualization, writing - original draft, writing - review and editing. M.E.: Formal analysis, software, visualization, writing - review and editing. D.W.D.: Software, writing - original draft, writing - review and editing. D.O.S.: Funding acquisition, project administration, resources, supervision. The code is currently maintained by KBS.

Conflicts of Interest

There are no conflicts to declare.

Acknowledgements

KBS, DWD and DOS acknowledge support from the European Research Council (grant 758345). This work made use of the ARCHER2 UK National Supercomputing Service via the Materials Chemistry Consortium, which is funded by EPSRC (EP/L000202), and resources made available via the UK Materials and Molecular Modelling Hub, which is partially funded by EPSRC (EP/P020194/1). K.B.S and D.O.S acknowledge the University of Birmingham's BlueBEAR HPC service (http://www.birmingham.ac.uk/bear); the Baskerville Tier 2 HPC service (https://www.baskerville.ac.uk/), which was funded by the EPSRC and UKRI through the World Class Labs scheme (EP/T022221/1) and the Digital Research Infrastructure programme (EP/W032244/1) and is operated by Advanced Research Computing at the University of Birmingham; and the Sulis Tier 2 HPC platform hosted by the Scientific Computing Research Technology Platform at the University of Warwick, which is funded by EPSRC Grant EP/T022108/1 and the HPC Midlands+ consortium. We would like to acknowledge contributions of ideas and time from Katarina Brlec, Bonan Zhu, Seán R. Kavanagh, Warda Rahim, Joe Willis, Luisa Herring–Rodriguez and Sabrine Hachmioune.

References

Brlec, K., Spooner, K. B., Skelton, J. M., & Scanlon, D. O. (2022). $Y_2Ti_2O_5S_2$ — a promising n-type oxysulphide for thermoelectric applications. *Journal of Materials Chemistry A*, *10*(32), 16813. https://doi.org/10.1039/D2TA04160J

Deng, T., Wu, G., Sullivan, M. B., Wong, Z. M., Hippalgaonkar, K., Wang, J.-S., & Yang, S.-W. (2020). EPIC STAR: A reliable and efficient approach for phonon-and impurity-limited charge transport calculations. *Npj Computational Materials*, *6*(1), 46. https://doi.org/10.1038/s41524-020-0316-7

Eriksson, F., Fransson, E., & Erhart, P. (2019). The hiphive package for the extraction of high-order force constants by machine learning. *Advanced Theory and Simululations*, *2*(5), 1800184. https://doi.org/10.1002/adts.201800184

Firth, A., Zhang, B., & Yang, A. (2019). Quantification of global waste heat and its environmental effects. *Appl. Energy*, *235*, 1314. https://doi.org/10.1016/j.apenergy.2018.10.102

Ganose, Alex 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

Ganose, Alex M., Park, J., Faghaninia, A., Woods-Robinson, R., Persson, K. A., & Jain, A. (2021). Efficient calculation of carrier scattering rates from first principles. *Nature Communications*, *12*(1), 2222. https://doi.org/10.1038/s41467-021-22440-5

Graziosi, P., Li, Z., & Neophytou, N. (2023). ElecTra Code: Full-band electronic transport properties of materials. *Computer Physics Communications*, *287*, 108670. https://doi.org/10.1016/j.cpc.2023.108670

Han, D., Zhu, B., Cai, Z., Spooner, K. B., Rudel, S. S., Schnick, W., Bein, T., Scanlon, D. O., & Ebert, H. (2024). Discovery of multi-anion antiperovskites X_6NFSn_2 (X= Ca, Sr) as promising thermoelectric materials by computational screening. *Matter*, *7*(1), 158. https://doi.org/10.1016/j.matt.2023.10.022

Hellman, O., Abrikosov, I. A., & Simak, S. (2011). Lattice dynamics of anharmonic solids from first principles. *Physical Review B*, *84*(18), 180301. https://doi.org/10.1103/PhysRevB.84.180301

Herring Rodriguez, L., Spooner, K. B., Einhorn, M., & Scanlon, D. O. (2023). $Sr_2Sb_2O_7$: A novel earth abundant oxide thermoelectric. *Journal Materials Chemistry C*, *11*(27), 9124. https://doi.org/10.1039/D3TC01003A

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science and Engineering*, *9*(3), 90. https://doi.org/10.1109/mcse.2007.55

Kavanagh, S. R., Savory, C. N., Scanlon, D. O., & Walsh, A. (2021). Hidden spontaneous polarisation in the chalcohalide photovoltaic absorber $Sn_2SbS_2I_3$. *Materials Horizons*, *8*(10), 2709. https://doi.org/10.1039/D1MH00764E

Li, W., Carrete, J., Katcho, N. A., & Mingo, N. (2014). ShengBTE: A solver of the Boltzmann transport equation for phonons. *Computer Physics Communications*, *185*(6), 1747. https://doi.org/10.1016/j.cpc.2014.02.015

Madsen, G. K. H., & Singh, D. J. (2006). BoltzTraP. A code for calculating band-structure dependent quantities. *Computer Physics Communications*, *175*(1), 67. https://doi.org/10.1016/j.cpc.2006.03.007

Madsen, G. K., Carrete, J., & Verstraete, M. J. (2018). BoltzTraP2, a program for interpolating band structures and calculating semi-classical transport coefficients. *Computer Physics Communications*, *231*, 140. https://doi.org/10.1016/j.cpc.2018.05.010

Noffsinger, J., Giustino, F., Malone, B. D., Park, C.-H., Louie, S. G., & Cohen, M. L. (2010). EPW: A program for calculating the electron–phonon coupling using maximally localized wannier functions. *Computer Physics Communications*, *181*(12), 2140. https://doi.org/10.1016/j.cpc.2010.08.027

Ong, S. P., Richards, W. D., Jain, A., Hautier, G., Kocher, M., Cholia, S., Gunter, D., Chevrier, V. L., Persson, K. A., & Ceder, G. (2013). Python materials genomics (pymatgen): A robust, open-source python library for materials analysis. *Computational Materials Science*, *68*, 314. https://doi.org/10.1016/j.commatsci.2012.10.028

Pöhls, J.-H., & Mozharivskyj, Y. (2022). TOSSPB: Thermoelectric optimization based on scattering-dependent single-parabolic band model. *Computational Materials Science*, *206*, 111152. https://doi.org/10.1016/j.commatsci.2021.111152

Samsonidze, G., & Kozinsky, B. (2018). Accelerated screening of thermoelectric materials by first-principles computations of electron–phonon scattering. *Advanced Energy Materials*, *8*(20), 1800246. https://doi.org/10.1002/aenm.201800246

Skelton, J. M. (2020). Phono3py-power-tools. In *GitHub repository*. GitHub. https://github.com/skelton-group/Phono3py-Power-Tools

Spooner, K. B., Ganose, A. M., Leung, W. W., Buckeridge, J., Williamson, B. A., Palgrave, R. G., & Scanlon, D. O. (2021). $BaBi_2O_6$: A promising n-type thermoelectric oxide with the $PbSb_2O_6$ crystal structure. *Chemistry of Materials*, *33*(18), 7441. https://doi.org/10.1021/acs.chemmater.1c02164

Tadano, T., Gohda, Y., & Tsuneyuki, S. (2014). Anharmonic force constants extracted from first-principles molecular dynamics: Applications to heat transfer simulations. *Journal of Physics: Condensed Matter*, *26*(22), 225402. https://doi.org/10.1088/0953-8984/26/22/225402

Togo, A., Chaput, L., & Tanaka, I. (2015). Distributions of phonon lifetimes in brillouin zones. *Physical Review B*, *91*(9, 9), 094306. https://doi.org/10.1103/PhysRevB.91.094306

Togo, A., & Tanaka, I. (2015). First principles phonon calculations in materials science. *Scripta Materialia*, *108*, 1. https://doi.org/10.1016/j.scriptamat.2015.07.021

Willis, J., Spooner, K. B., & Scanlon, D. O. (2023). On the possibility of p-type doping in barium stannate. *Applied Physics Letters*, *123*(16), 162103. https://doi.org/10.1063/5.0170552

Zhou, J.-J., Park, J., Lu, I.-T., Maliyov, I., Tong, X., & Bernardi, M. (2021). Perturbo: A software package for ab initio electron–phonon interactions, charge transport and ultrafast dynamics. *Computer Physics Communications*, *264*, 107970. https://doi.org/10.1016/j.cpc.2021.107970