The Journal of Open Source Software

SMACT: Semiconducting Materials by Analogy and Chemical Theory

Daniel W. Davies[1], **Keith T. Butler**[2], **Adam J. Jackson**[3], **Jonathan M. Skelton**[4], **Kazuki Morita**[1], **and Aron Walsh**[1, 5]

1 Department of Materials, Imperial College London, London, UK **2** SciML, STFC Scientific Computing Division, Rutherford Appleton Laboratories, UK **3** Department of Chemistry, University College London, London, UK **4** School of Chemistry, University of Manchester, Manchester, UK **5** Department of Materials Science and Engineering, Yonsei University, Seoul, Korea

DOI: 10.21105/joss.01361

Software

- Review
- Repository
- Archive

Submitted: 28 March 2019
Published: 10 June 2019

The paradigm of data-driven science is revolutionising the materials discovery process. There are now many databases containing experimental and calculated materials properties and extensive codes available for applying data mining, machine learning, and other statistical approaches (a well-maintained list is available here). While we use these tools to push forward in the quest to learn as much as we can from existing materials, it is becoming clear that the search space for new materials remains relatively uncharted.

The discovery of new chemical compounds (combinations of elements arranged in a particular way in space) underpins materials discovery. The `smact` Python library is designed to facilitate a top-down approach where sets of element combinations are generated then screened using chemical filters. It is possible to screen for candidates that make "chemical sense" according to the well-established principles of electron valence and charge neutrality. The methodology is inspired by the seminal work of Goodman and Pamplin who carried out similar procedures by hand, predicting the existence of new semiconductors by analogy with existing compounds (Goodman, 1958; Pamplin, 1964).

Once a set of compositions is generated, further functions built into `smact` can be used to filter for candidates with target properties using data-driven models. These functions can predict key electronic structure properties such as the optical band gap using the solid-state energy scale (Pelatt, Ravichandran, Wager, & Keszler, 2011), evaluate sustainability metrics using the Herfidahl-Hirschman Index of resource availability (Gaultois et al., 2013), and predict stability using a statistical oxidation states model (D. W. Davies, Butler, Isayev, & Walsh, 2018).

Core components: The element and species classes are at the heart of `smact`. Elements are elements of the periodic table. Species are elements in a particular oxidation state and (optionally) coordination environment. These classes provide access to tabulated data and the properties of these objects are leveraged by the screening functions. For example, atomic radii can be used in the application of radius-ratio rules (Goldschmidt, 1929) and electronegativities can be used to estimate electronic properties (Nethercot, 1974). In a typical workflow, screening functions are applied to lists of elements or species sets. While other chemistry toolkits such as `OpenBabel` (O'Boyle et al., 2011), the Atomic Simulation Environment (`ASE`) (Larsen et al., 2017) and `Pymatgen` (Ong et al., 2013) can also be used to access tabulated element data, `smact` is distinctive in that it primarily deals with chemical composition and associated properties, as opposed to molecular or crystal structure.

High-throughput workflows: The number of possible element combinations is enormous, exceeding 4×10^{12} for four-component compounds (D. W. Davies et al., 2016). For this reason, functions from `smact` can be applied at low computational cost to facilitate the screening of vast areas of chemical space rapidly on a desktop computer. This is made possible by (i) a data_loader module which implements a data-caching system to avoid a large amount of I/O and (ii) using Python's built-in `multiprocessing` library, as shown in the example workflows.

Interfacing to machine learning: Materials design is begining to benefit from the development of powerful machine learning techniques, with many supervised learning models being built to predict important properties (K. T. Butler, Davies, Cartwright, Isayev, & Walsh, 2018). The `smact` library can provide a large, unseen chemical space to which trained models can be applied. The compositions generated by `smact` can be featurised using the `matminer` Python library (Ward et al., 2018) or converted to objects used in `Pymatgen`.

Author contributions

DWD, AJJ and KTB contributed equally to the primary code base of the `smact` package and, along with AW, made the majority of decisions about which features should be available and how they should be implemented. JMS improved the code performance by implementing the data_loader module. KM implemented atomic polarizability and associated tests. The first draft of this manuscript was written by DWD with input from all co-authours.

Acknowledgements

The development of this code has benefited through discussions with and contributions from many members of the Walsh research group including Andrew Morris, Timothy Gaunlett, Jarvist M. Frost, Suzanne K. Wallace.

References

Butler, K. T., Davies, D. W., Cartwright, H., Isayev, O., & Walsh, A. (2018). Machine learning for molecular and materials science. *Nature*, *559*(7715), 547–555. doi:10.1038/s41586-018-0337-2

Davies, D. W., Butler, K. T., Isayev, O., & Walsh, A. (2018). Materials discovery by chemical analogy: Role of oxidation states in structure prediction. *Faraday Discuss.*, *211*(0), 553–568. doi:10.1039/C8FD00032H

Davies, D. W., Butler, K. T., Jackson, A. J., Morris, A., Frost, J. M., Skelton, J. M., & Walsh, A. (2016). Computational Screening of All Stoichiometric Inorganic Materials. *Chem*, *1*(4), 617–627. doi:10.1016/j.chempr.2016.09.010

Gaultois, M. W., Sparks, T. D., Borg, C. K. H., Seshadri, R., Bonificio, W. D., & Clarke, D. R. (2013). Data-driven review of thermoelectric materials: Performance and resource considerations. *Chemistry of Materials*, *25*(15), 2911–2920. doi:10.1021/cm400893e

Goldschmidt, V. M. (1929). Crystal structure and chemical constitution. *Trans. Faraday Soc.*, *25*(0), 253–283. doi:10.1039/TF9292500253

Goodman, C. (1958). The prediction of semiconducting properties in inorganic compounds. *Journal of Physics and Chemistry of Solids*, *6*(4), 305–314. doi:https://doi.org/10.1016/0022-3697(58)90050-7

Larsen, A. H., Mortensen, J. J., Blomqvist, J., Castelli, I. E., Christensen, R., Dułak, M., Friis, J., et al. (2017). The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter*, *29*(27), 273002. Retrieved from http://stacks.iop.org/0953-8984/29/i=27/a=273002

Nethercot, A. H. (1974). Prediction of fermi energies and photoelectric thresholds based on electronegativity concepts. *Phys. Rev. Lett.*, *33*(18), 1088–1091. doi:10.1103/PhysRevLett.33.1088

Ong, S. P., Richards, W. D., Jain, A., Hautier, G., Kocher, M., Cholia, S., Gunter, D., et al. (2013). Python materials genomics (pymatgen): A robust, open-source python library for materials analysis. *Computational Materials Science*, *68*, 314–319. doi:https://doi.org/10.1016/j.commatsci.2012.10.028

O'Boyle, N. M., Banck, M., James, C. A., Morley, C., Vandermeersch, T., & Hutchison, G. R. (2011). Open babel: An open chemical toolbox. *Journal of Cheminformatics*, *3*(1), 33. doi:10.1186/1758-2946-3-33

Pamplin, B. (1964). A systematic method of deriving new semiconducting compounds by structural analogy. *Journal of Physics and Chemistry of Solids*, *25*(7), 675–684. doi:https://doi.org/10.1016/0022-3697(64)90176-3

Pelatt, B. D., Ravichandran, R., Wager, J. F., & Keszler, D. A. (2011). Atomic solid state energy scale. *Journal of the American Chemical Society*, *133*(42), 16852–16860. doi:10.1021/ja204670s

Ward, L., Dunn, A., Faghaninia, A., Zimmermann, N. E., Bajaj, S., Wang, Q., Montoya, J., et al. (2018). Matminer: An open source toolkit for materials data mining. *Computational Materials Science*, *152*, 60–69. doi:https://doi.org/10.1016/j.commatsci.2018.05.018