Simmate: a framework for materials science

Jack D. Sundberg[1¶], **Siona S. Benjamin**[1], **Lauren M. McRae**[1], and **Scott C. Warren**[1,2]

1 Department of Chemistry, The University of North Carolina at Chapel Hill, Chapel Hill, North Carolina 27599, United States **2** Department of Applied Physical Sciences, The University of North Carolina at Chapel Hill, Chapel Hill, North Carolina 27599, United States ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 08 April 2022
Published: unpublished

Summary

Over the past decade, the automation of electronic structure codes has led to many large-scale initiatives for materials discovery, such as the Materials Project (Jain et al., 2013), AFLOW (Curtarolo et al., 2012), OQMD (Saal et al., 2013), JARVIS (Choudhary et al., 2020), and others (Draxl & Scheffler, 2019; Pizzi et al., 2016; Talirz et al., 2020). Each of these projects facilitate the creation and distribution of materials science data to the broader researcher community through databases, workflow libraries, and web interfaces. However, each software ecosystem (i.e. the collection of software used by a specific project) still possesses several pain-points for users attempting to implement new standards for computational research. Proper setup involves learning how (i) workflows are defined/orchestrated, (ii) how databases are built/accessed, and (iii) how website interfaces/APIs make results accessible to the community, where each component requires learning a new package and, more importantly, learning how that package integrates with others. As a result, it can be difficult to integrate several smaller packages when building a production-ready server and database for materials science research.

To address this, we developed the Simulated Materials Ecosystem (Simmate). Simmate strives to simplify the process for researchers who are setting up a full-featured server. For the purposes of beginners, we desired a framework that could run locally without requiring any additional setup. For the purposes of experts, we sought to facilitate scaling of calculations across any number of resources. Simmate accomplishes these by building on top of popular, well-established packages for workflow orchestration, database management, and materials science analysis. This includes integration with high-level, beginner-friendly packages such as Django (Django Collaboration, 2022), Dask (Dask Collaboration, 2022), Prefect (Prefect Collaboration, 2022), and PyMatGen (Ong et al., 2013). The use of popular packages also lets Simmate users take advantage of these packages' large user communities, abundant guides, and robust coding standards, while Simmate handles the integration of these packages in the context of materials science. Because Simmate removes many obstacles to advanced computation, we anticipate that this code will be utilized by beginners and experts alike. Thus, our tutorials are written for researchers that have never used the command-line or python However, as users become comfortable, they can begin exploring underlying API and integrated packages for advanced features.

Simmate is designed specifically for materials science research and the calculation of materials properties. While our current implementation is focused on periodic crystals and *ab-initio* calculations, the framework is built around abstract data types and functionality; this allows easy integration of third-party software and databases. For example, we currently distribute data from other providers (COD (Gražulis et al., 2009), Materials Project (Jain et al., 2013), OQMD (Saal et al., 2013), and JARVIS (Choudhary et al., 2020)) as well as orchestrate calculations from popular DFT codes (e.g. VASP (Kresse & Furthmüller, 1996)). Each of these integrations benefit by inheriting from our core data types, which implement features such as

error handling and job recovery for workflow integrations as well as the automatic generation of Python APIs, REST APIs, and website interfaces for results and databases. Moreover, data can be converted to other useful Python objects such as those from PyMatGen (Ong et al., 2013) or ASE (Larsen et al., 2017), allowing further analysis of the materials. Together, these features help Simmate bridge the gap between the existing ecosystems of materials science software, while making production-ready implementations as easy as possible.

Acknowledgements

S.C.W. acknowledges support of this research by NSF grant DMR-1905294. L.M.M. acknowledges support by the NSF Graduate Research Fellowship (GRF) under grant DGE-1650116. J.D.S acknowledges support by the NSF GRF under grant DGE-1650114. While developing this software, computational resources were provided, in part, by the Research Computing Center at the University of North Carolina at Chapel Hill.

References

Choudhary, K., Garrity, K. F., Reid, A. C. E., DeCost, B., Biacchi, A. J., Hight Walker, A. R., Trautt, Z., Hattrick-Simpers, J., Kusne, A. G., Centrone, A., Davydov, A., Jiang, J., Pachter, R., Cheon, G., Reed, E., Agrawal, A., Qian, X., Sharma, V., Zhuang, H., … Tavazza, F. (2020). The joint automated repository for various integrated simulations (JARVIS) for data-driven materials design. *Npj Computational Materials*, *6*(173). https://doi.org/10.1038/s41524-020-00440-1

Curtarolo, S., Setyawan, W., Hart, G. L. W., Jahnatek, M., Chepulskii, R. V., Taylor, R. H., Wang, S., Xue, J., Yang, K., Levy, O., Mehl, M. J., Stokes, H. T., Demchenko, D. O., & Morgan, D. (2012). AFLOW: An automatic framework for high-throughput materials discovery. *Computational Materials Science*, *58*, 218–226. https://doi.org/10.1016/j.commatsci.2012.02.005

Dask Collaboration. (2022). Dask: Parallel computing with task scheduling. In *GitHub repository*. GitHub. https://github.com/dask/dask

Django Collaboration. (2022). Django: The web framework for perfectionists with deadlines. In *GitHub repository*. GitHub. https://github.com/django/django

Draxl, C., & Scheffler, M. (2019). The NOMAD laboratory: From data sharing to artificial intelligence. *Journal of Physics: Materials*, *2*(3), 036001. https://doi.org/10.1088/2515-7639/ab13bb

Gražulis, S., Chateigner, D., Downs, R. T., Yokochi, A. F. T., Quirós, M., Lutterotti, L., Manakova, E., Butkus, J., Moeck, P., & Le Bail, A. (2009). Crystallography Open Database – an open-access collection of crystal structures. *Journal of Applied Crystallography*, *42*(4), 726–729. https://doi.org/10.1107/S0021889809016690

Jain, A., Ong, S. P., Hautier, G., Chen, W., Richards, W. D., Dacek, S., Cholia, S., Gunter, D., Skinner, D., Ceder, G., & Persson, K. A. (2013). Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL Materials*, *1*(1), 011002. https://doi.org/10.1063/1.4812323

Kresse, G., & Furthmüller, J. (1996). 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., Marcin 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. https://doi.org/10.1088/1361-648x/aa680e

Ong, S. P., William Davidson Richards, W. M., 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*, 314–319. https://doi.org/10.1016/j.commatsci.2012.10.028

Pizzi, G., Cepellotti, A., Sabatini, R., Marzari, N., & Kozinsky, B. (2016). AiiDA: Automated interactive infrastructure and database for computational science. *Computational Materials Science*, *111*, 218–230. https://doi.org/10.1016/j.commatsci.2015.09.013

Prefect Collaboration. (2022). Prefect: The easiest way to automate your data. In *GitHub repository*. GitHub. https://github.com/PrefectHQ/prefect

Saal, J. E., Kirklin, S., Aykol, M., Meredig, B., & Wolverton, C. (2013). Materials design and discovery with high-throughput density functional theory: The open quantum materials database (OQMD). *JOM*, *65*, 1501–1509. https://doi.org/10.1007/s11837-013-0755-4

Talirz, L., Kumbhar, S., Passaro, E., Yakutovich, A. V., Granata, V., Gargiulo, F., Borelli, M., Uhrin, M., Huber, S. P., Zoupanos, S., Adorf, C. S., Andersen, C. W., Schütt, O., Pignedoli, C. A., Passerone, D., VandeVondele, J., Schulthess, T. C., Smit, B., Pizzi, G., & Marzari, N. (2020). Materials cloud, a platform for open computational science. *Scientific Data*, *7*(299, 1). https://doi.org/10.1038/s41597-020-00637-5