The Journal of Open Source Software

Battflow: an automated workflow for predicting key properties of battery electrolytes

Neubi F. Xavier Jr [1¶], **Matthias J. Golomb** [1], **and Qiong Cai** [1¶]

1 School of Chemistry and Chemical Engineering, University of Surrey, Guildford, GU2 7XH, United Kingdom ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 06 January 2026
Published: unpublished

Summary

Battflow provides an automated workflow that integrates a suite of Python packages, GROMACS (Abraham et al., 2025), and ORCA (Neese, 2012) to predict properties of battery electrolytes. It aims to simplify user interactions when generating large MD/DFT datasets, thereby aiding the high-throughput screening and discovery of new electrolytes. The workflow is linked to a collection within a MongoDB cluster, in which missing calculated properties are identified and flagged for further calculations. The only information needed to start the simulations is the SMILES string and molar concentration of each electrolyte component. A document for each electrolyte composition is then created, and diffusion coefficients and solvation structure statistics are computed from molecular dynamics simulations, while binding energies for solvation clusters and HOMO–LUMO energies of each individual component are updated.

Statement of need

Lithium metal batteries (LMBs) are regarded as a promising solution for meeting market demand for energy storage systems with high specific capacity. Recently, several battery technologies incorporating lithium metal anodes have attracted increasing attention, including lithium–sulfur (Li–S) batteries, lithium–oxygen (Li–O), and lithium–carbon dioxide (Li–CO) batteries. However, the implementation of lithium metal anodes is currently hindered by poor cycle life and uncontrollable side reactions between Li metal and liquid electrolytes. Liquid electrolyte engineering, which involves mixing different molecules to create electrolytes with specific properties, is ultimately the most cost-effective approach for making LMBs viable. However, there are myriad possible electrolyte formulations due to the large number of commercially available molecules, recently synthesised electrolyte-specific compounds, and various strategies for fine-tuning electrolyte components. Consequently, there is an increased need for theory-guided rational design of new electrolyte formulations for LMBs, aiming to reduce research costs and avoid "trial-and-error" approaches.

There is an increasing need for standardised computational data to guide experimental studies on battery electrolytes, particularly in light of the rapid growth of the literature in this field, with approximately 200 papers containing the keywords "battery electrolyte" published per week in 2023 alone. A combination of molecular dynamics (MD) and Density Functional Theory (DFT) simulations for the estimation of transport and electronic properties of the bulk electrolyte and its individual components offers a good balance between accuracy and cost efficiency for properties prediction. However, setting up force fields and simulation settings for molecular dynamics, followed by DFT calculations of relevant Li solvation clusters for several electrolyte compositions, can be a daunting task and error-prone task, even for experienced researchers.

Battflow is intended for both theoreticians and experimentalists and can provide out-of-the-box default settings to run the automated workflow with basic configurations. At present, Battflow extends beyond Li-metal and Li-ion batteries and can be adapted to other alkali metal battery systems, including Na-ion, K-ion, and Zn-based batteries.

Usage and availability

Battflow inputs consist of documents stored within a MongoDB collection. The information required to connect to MongoDB, either through localhost or a remote instance, is provided in `config.yaml`. A `.json` example input file, which should be uploaded to MongoDB as a document, is provided. Battflow reads the `smiles` and `concentrations` fields within each document to build the molecular structure of each component and to create the electrolyte box, respectively. The output is reported in the `simulation_data` field; if any calculated property is flagged as absent, the workflow is triggered to start.

The molecular dynamics workflow consists of: (1) setup of GAFF2 force fields for the molecular components, generated using ACPYPE (Sousa da Silva & Vranken, 2012); (2) creation of the electrolyte box; (3) execution of minimisation, equilibration, and production MD runs using GROMACS (Abraham et al., 2025); and (4) uploading diffusion properties and solvation structure statistics to the MongoDB document. The DFT simulations in ORCA (Neese, 2012) follow the molecular dynamics runs, with calculations performed for the three most prevalent solvation clusters. Binding energies are computed for every solvation cluster and HOMO-LUMO energies are calculated for the entire cluster and separated components, as well. After the analysis, results are uploaded back to MongoDB.

Battflow is available for Linux operating systems and can be downloaded from GitHub (https://github.com/neubifx/Battflow/tree/main) under the GPL-3.0 licence. Additional documentation is available on the repository page and is continuously updated.

Acknowledgements

This project has received funding from the AI for Chemistry: AIchemy Hub (EPSRC grant EP/Y028775/1 and EP/Y028759/1). The authors acknowledge funding from Horizon Europe through the OPERA consortium (Grant Number 101103834) and under the UKRI Horizon Europe Guarantee Extension (Ref Number 10078555), from the Faraday Institution through the LiSTAR programme (Grants FIRG014, FIRG058), and from the Royal Society (IEC\NSFC\211200).

References

Abraham, M., Alekseenko, A., Andrews, B., Basov, V., Bauer, P., Bird, H., Briand, E., Brown, A., Doijade, M., Fiorin, G., Fleischmann, S., Gorelov, S., Gouaillardet, G., Gray, A., Irrgang, M. E., Jalalypour, F., Johansson, P., Kutzner, C., Łazarski, G., … Lindahl, E. (2025). *GROMACS 2025.4 manual* (Version 2025.4). Zenodo. https://doi.org/10.5281/zenodo.17671776

Neese, F. (2012). The ORCA program system [Journal Article]. *WIRES Comput. Molec. Sci.*, *2*(1), 73–78. https://doi.org/10.1002/wcms.81

Sousa da Silva, A. W., & Vranken, W. F. (2012). ACPYPE - AnteChamber PYthon Parser interfacE. *BMC Research Notes*, *5*(1), 367. https://doi.org/10.1186/1756-0500-5-367