The Journal of Open Source Software

FIRECODE: A Computational Chemistry Workflow Driver and Modular Hub

Nicolò Tampellini [1¶]

1 Massachusetts Institute of Technology, United States of America ¶ Corresponding author

DOI: 10.21105/joss.10698

Software

- Review
- Repository
- Archive

Editor: Evan Spotte-Smith

Reviewers:

- @JacksonBurns
- @wongzit

Submitted: 29 April 2026
Published: 18 August 2026

Summary

Computational chemistry, which seeks to model molecular systems for the extraction of properties of interest, is experiencing a significant renaissance thanks to impressive advances in theoretical models enabling ever-faster simulations of complex processes via density functional theory (DFT), semiempirical methods, and machine-learned interatomic potentials (MLIPs). However, the reduced computational cost of these workflows will only translate into faster and more robust modeling with the help of automation frameworks, where trivial human intervention does not become the new bottleneck of computational pipelines. FIRECODE aims at filling this need by providing a modular framework for generating and processing molecular ensembles.

Statement of need

`FIRECODE` is a modular workflow driver for computational chemistry written in Python. The code interfaces with a series of fast modern calculators (xTB (Bannwarth et al., 2019), tblite (Katbashev et al., 2025), AIMNET2 (Anstine et al., 2025), UMA (Wood et al., 2025), usually via ASE (Bahn & Jacobsen, 2002; Larsen et al., 2017)) and exposes a series of workflows and utilities for geometry optimization, frequency analysis, one- and two-ended transition state search, and conformational search. When possible, these routines are implemented in a calculator-independent way, and almost all of these can be arbitrarily chained to one another to define a complex workflow.

The software was born out of necessity over the last five years of research in asymmetric catalysis: `FIRECODE` was designed to streamline the necessary use of multiple software and routines in the computational modeling of small molecules, in a single centralized hub. The program can be called via command line with a minimal plain-text input file defining the workflow, or used as a Python library in Jupyter notebooks. The modular nature of the software enables rapid addition of new features, facilitating external contributions, while also retaining a small installation footprint for the core library. External programs are easily interfaced via API calls or file-based I/O. The concise input syntax also offers a streamlined definition of workflows and software orchestration with no coding experience required.

State of the field

On top of computational chemistry software, workflow utilities orchestrating sequential calculation steps are increasingly recognized as an essential asset in the streamlining of modeling efforts. Recently, version 6.0 of `ORCA` (Neese et al., 2020; Neese, 2022; Tetenberg et al., 2026) (free for academic use) started offering compound job functionality, which enables the chained execution of some core ORCA modules. However, the code is not open-source and functionality is limited to what is included in the ORCA program (*i.e.* no MLIPs, no similarity

pruning, no ensemble processing…). While ORCA workflows are more extensively implemented in `WEASEL` (Faccts), this is commercial, closed-source software.

Related software to the one presented here include AQME, (Alegre-Requena et al., 2023) which has what is perhaps the most similar design philosophy. Key differences with `FIRECODE`, besides code architecture, lie in: 1) the main interaction platform of `AQME` happening via Jupyter notebooks, while the software of this work favors the use of plain-text input files, keeping the inputs as minimal as possible. 2) The present software having a reactivity-centered set of features, chiefly via fast semiempirical and ML calculators, over `AQME`'s focus on DFT and on the generation of ensemble properties like spectra and molecular descriptors.

Other workflow software emerging from the material chemistry community includes `atomate2` (A. Ganose et al., 2025; A. M. Ganose et al., 2025), `AiiDA` (Huber et al., 2020; Uhrin et al., 2021) and `QuAcc` (Rosen, 2026).

Reactivity-centered software, primarily for the automated location of transition states, is also gaining popularity. Examples include autodE (Young et al., 2021) and ML-FSM (Marks & Gomes, 2025). The latter is indeed interfaced as a utility in FIRECODE, given its light weight and flexibility in working with different ASE calculators.

Other examples of related software, specifically targeting the exposure of high-level primitives over structured workflows, include `Cuby` (Řezáč, 2016) (MD and high-level DFT) written in Ruby, and the popular `ASE` (Bahn & Jacobsen, 2002; Larsen et al., 2017), which is extensively used as a lower-level interface for interacting with calculators in `FIRECODE`.

The development of this software from scratch was motivated by several factors. First, a modern and flexible hub for running computational chemistry workflows, particularly tailored to modeling chemical reactivity and MLIPs, was lacking. In this context, we refer to "reactivity workflows" as the sequence of sequential steps from a three-dimensional molecular representation to a conformationally comprehensive characterization of bond breaking and forming events. This requires access to conformational search modules, efficient multistage ensemble optimization routines, similarity pruning, the ability to use arbitrary geometrical constraints throughout, the ability to optimize to local minima and saddle points alike, and the ability to carry out vibrational analysis and calculate related thermochemical corrections to access enthalpy/entropy/free energy values.

While the benefit of developing standalone applications for specific tasks remains, their integration in a single hub is uniquely enabling for automation, and contributes to the popularization of robust atomistic modeling. Second, modern machine-learned interatomic potentials (MLIPs) are often centered around Python libraries like `torch` (Ansel et al., 2024), benefitting greatly from a Python-based workflow manager. Additional features of note of this software also include an implicit delta solvation scheme for gas phase-trained MLIPs [1], and numerical quasi-RRHO thermochemistry (Grimme, 2012).

Software design

The software is developed with the philosophy of maintaining the greatest amount of functionality and module interoperability with the minimal list of dependencies, which should ideally be as modern and lean as possible. All core dependencies are distributed via `pypi`, keeping the installation of the minimal version of the program down to a single command (`uv pip install firecode`). Non-essential dependencies can be installed based on the desired calculator and interfaces to be used, all via conda.

[1] That is, a calculator providing the energy and gradient differences between an atomic structure in an implicit solvent and the same structure in vacuum. These differential contributions are then added to the energy and gradient components of calculations that have no native implementation of solvation models (i.e. gas phase-trained MLIPs).

The software implements various calculators (xTB (Bannwarth et al., 2019), tblite (Katbashev et al., 2025), AIMNET2 (Anstine et al., 2025), UMA (Wood et al., 2025)) via ASE (Bahn & Jacobsen, 2002; Larsen et al., 2017), which can be used interchangeably in various ensemble routines. Other core libraries leveraged throughout the codebase are `numpy` (Harris et al., 2020) for algebraic manipulations, `networkx` (Hagberg et al., 2008) for graph utilities and `prism_pruner` (*GitHub - Ntampellini/Prism_pruner*) for the removal of duplicates. `Sella` (Hermes et al., 2019, 2021, 2022) provides a saddle-point optimizer, and `rdkit` (Greg Landrum, 2026) is used for `ETKDG` (Riniker & Landrum, 2015) conformational searches and SMARTS substructure matching. The `ML-FSM` library (Marks & Gomes, 2025) provides a two-ended TS search utility complementing ASE's implementation of the `NEB` method (Ásgeirsson et al., 2021).

Other standalone conformational search software is instead interfaced via subprocess calls and file-based I/O, like `CREST` (Pracht et al., 2020, 2024), ORCA's `GOAT` (Souza, 2025) and `racerts` (Schmid et al., 2026).

The modular nature of the calculators and interfaces allows the definition of complex workflows in plain-text files (input.txt):

```
firecode input.txt -n test_run
```

A second standalone executable exposes the core optimizer routines acting directly on structure files:

```
firecode_opt conformers.xyz --opt --freq --solvent toluene [...]
```

Contributions by way of adding new routines (or interfaces) to the codebase is made straightforward by design: individual "operators" (routines) are simply functions reading a molecular `.xyz` file and returning the filename of the processed ensemble. Run information can be accessed from the Embedder class or via environment variables.

```python
def center_operator(filename: str, embedder: Embedder) -> str:
    """Example operator centering the molecule."""

    # the Embedder class stores global information
    embedder.avail_gpus # 1
    embedder.options.solvent # "ch2cl2"
    embedder.options.T # 298.15

    # get a copy of the molecule of interest
    mol = embedder.mols[filename]

    # center coordinates
    mol.coords[0] -= np.mean(mol.coords[0], axis=0)

    # save any data you might need later
    embedder.options.centered = True
    embedder.options.last_operator = "center"

    # write to global log
    embedder.log(f"--> Center operator: centered molecule {mol.basename}")

    # write outfile and return its name
    outfile = f"{mol.basename}_centered.xyz"
    mol.to_xyz(outfile)

    return outfile
```

Research impact statement

Since its earliest versions in 2021 (formerly TSCoDe), this software proved essential in orchestrating asymmetric catalysis workflows with ensembles up to thousands of structures (Huth et al., 2024; Marvich et al., 2026; Rozema et al., 2025; Tampellini et al., 2024, 2025; Tampellini, Choi, et al., 2025). These included the generation of large conformational ensembles for flexible catalysts and their non-covalent aggregates (often including specific constraints) and the multilevel optimization and pruning of these ensembles to a smaller number of non-redundant structures. These refined ensembles were then directed to higher-level DFT optimization to achieve quantitative insight into stereoselectivity (*i.e.* a predicted $\Delta\Delta G^{\ddagger}$ value) and obtain other conformationally-comprehensive properties (*i.e.* flexibility, relative populations, number of interaction topologies).

Adoption of `FIRECODE` is starting to grow, with `pypi` reporting ~250 downloads per month at the present time. Part of the core code was exported to the `prism_pruner` library (*GitHub - Ntampellini/Prism_pruner*) to facilitate its adoption by the computational chemistry startup Rowan (*Rowan | ML-Powered Molecular Design and Simulation — Rowansci.com*) (not affiliated with the author).

AI usage disclosure

Generative AI tools (Claude, Gemini) were used in some cases to generate first drafts of Python functions that were then manually curated line-by-line and interfaced with the rest of the codebase by hand. No generative AI tools were used in the writing of this manuscript.

Acknowledgements

We acknowledge years of research support from both Alma Mater Studiorum - University of Bologna (Prof. Giorgio Bencivenni and Prof. Paolo Righi) and Yale University (Prof. Scott J. Miller). These environments provided both the computational resources to run and develop the software as well as the primary problems it was designed to address and was stress-tested against.

References

Alegre-Requena, J. V., Sowndarya S. V, S., Pérez-Soto, R., Alturaifi, T. M., & Paton, R. S. (2023). AQME: Automated quantum mechanical environments for researchers and educators. *WIREs Computational Molecular Science*, *13*(5), e1663. https://doi.org/10.1002/wcms.1663

Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., … Chintala, S. (2024). PyTorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2* (pp. 929–947). ACM. https://doi.org/10.1145/3620665.3640366

Anstine, D. M., Zubatyuk, R., & Isayev, O. (2025). AIMNet2: A neural network potential to meet your neutral, charged, organic, and elemental-organic needs. *Chemical Science*, *16*(23), 10228–10244. https://doi.org/10.1039/d4sc08572h

Ásgeirsson, V., Birgisson, B. O., Bjornsson, R., Becker, U., Neese, F., Riplinger, C., & Jónsson, H. (2021). Nudged elastic band method for molecular reactions using energy-weighted

springs combined with eigenvector following. *Journal of Chemical Theory and Computation*, *17*(8), 4929–4945. https://doi.org/10.1021/acs.jctc.1c00462

Bahn, S. R., & Jacobsen, K. W. (2002). An object-oriented scripting interface to a legacy electronic structure code [Article]. *Comput. Sci. Eng.*, *4*(3), 56–66. https://doi.org/10.1109/5992.998641

Bannwarth, C., Ehlert, S., & Grimme, S. (2019). GFN2-xTB—an accurate and broadly parametrized self-consistent tight-binding quantum chemical method with multipole electrostatics and density-dependent dispersion contributions. *Journal of Chemical Theory and Computation*, *15*(3), 1652–1671. https://doi.org/10.1021/acs.jctc.8b01176

Faccts. *Weasel - FACCTs — faccts.de*. https://www.faccts.de/weasel/.

Ganose, A. M., Sahasrabuddhe, H., Asta, M., Beck, K., Biswas, T., Bonkowski, A., Bustamante, J., Chen, X., Chiang, Y., Chrzan, D. C., Clary, J., Cohen, O. A., Ertural, C., Gallant, M. C., George, J., Gerits, S., Goodall, R. E. A., Guha, R. D., Hautier, G., … Jain, A. (2025). Atomate2: Modular workflows for materials science. *Digital Discovery*, *4*(7), 1944–1973. https://doi.org/10.1039/d5dd00019j

Ganose, A., Riebesell, J., George, J., Shen, J., Rosen, A. S., Naik, A. A., nwinner, Wen, M., rdguha1995, Kuner, M., Kaplan, A., Petretto, G., Ertural, C., Horton, M., Zhu, Z., Sahasrabuddhe, H., Schmidt, J., Kingsbury, R., Purcell, T., … Wang, S. (2025). *Materialsproject/atomate2: v0.0.21* (Version v0.0.21). Zenodo. https://doi.org/10.5281/zenodo.15603088

GitHub - ntampellini/prism_pruner: Similarity pruning engine for conformational ensembles — github.com. https://github.com/ntampellini/prism_pruner.

Greg Landrum, B. K., Paolo Tosco. (2026). *Rdkit/rdkit: 2026_03_3 (Q1 2026)*. https://doi.org/10.5281/zenodo.591637

Grimme, S. (2012). Supramolecular binding thermodynamics by dispersion-corrected density functional theory. *Chemistry - A European Journal*, *18*(32), 9955–9964. https://doi.org/10.1002/chem.201200497

Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring network structure, dynamics, and function using NetworkX. *Proceedings of the 7th Python in Science Conference*, 11–15. https://doi.org/10.25080/tcwv9851

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., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hermes, E. D., Sargsyan, K., Najm, H. N., & Zádor, J. (2019). Accelerated saddle point refinement through full exploitation of partial Hessian diagonalization. *Journal of Chemical Theory and Computation*, *15*(11), 6536–6549. https://doi.org/10.1021/acs.jctc.9b00869

Hermes, E. D., Sargsyan, K., Najm, H. N., & Zádor, J. (2021). Geometry optimization speedup through a geodesic approach to internal coordinates. *The Journal of Chemical Physics*, *155*(9). https://doi.org/10.1063/5.0060146

Hermes, E. D., Sargsyan, K., Najm, H. N., & Zádor, J. (2022). Sella, an open-source automation-friendly molecular saddle point optimizer. *Journal of Chemical Theory and Computation*, *18*(11), 6974–6988. https://doi.org/10.1021/acs.jctc.2c00395

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 [Journal Article]. *Scientific Data*, *7*(1), 300. https://doi.org/10.1038/s41597-020-00638-4

Huth, S. E., Tampellini, N., Guerrero, M. D., & Miller, S. J. (2024). Catalytic enantioselective sulfoxidation of functionalized thioethers mediated by aspartic acid-containing peptides. *Organic Letters*, *26*(32), 6872–6877. https://doi.org/10.1021/acs.orglett.4c02452

Katbashev, A., Stahn, M., Rose, T., Alizadeh, V., Friede, M., Plett, C., Steinbach, P., & Ehlert, S. (2025). Overview on building blocks and applications of efficient and robust extended tight binding. *The Journal of Physical Chemistry A*, *129*(10), 2667–2682. https://doi.org/10.1021/acs.jpca.4c08263

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. https://doi.org/10.1088/1361-648X/aa680e

Marks, J., & Gomes, J. (2025). Incorporation of internal coordinates interpolation into the freezing string method. *Journal of Chemical Theory and Computation*, *21*(23), 12110–12120. https://doi.org/10.1021/acs.jctc.5c01492

Marvich, H. M., Langner, O. C., Tampellini, N., & Miller, S. J. (2026). A bifunctional aminoxyl-bipyridine peptide catalyst for the atroposelective copper-catalyzed aerobic oxidation of biaryl diols. *Journal of the American Chemical Society*, *148*(22), 23032–23041. https://doi.org/10.1021/jacs.6c04560

Neese, F. (2022). Software update: The ORCA program system—version 5.0. *WIREs Computational Molecular Science*, *12*(5), e1606. https://doi.org/10.1002/wcms.1606

Neese, F., Wennmohs, F., Becker, U., & Riplinger, C. (2020). The ORCA quantum chemistry program package. *The Journal of Chemical Physics*, *152*(22), 224108. https://doi.org/10.1063/5.0004608

Pracht, P., Bohle, F., & Grimme, S. (2020). Automated exploration of the low-energy chemical space with fast quantum chemical methods. *Physical Chemistry Chemical Physics*, *22*(14), 7169–7192. https://doi.org/10.1039/C9CP06869D

Pracht, P., Grimme, S., Bannwarth, C., Bohle, F., Ehlert, S., Feldmann, G., Gorges, J., Müller, M., Neudecker, T., Plett, C., Spicher, S., Steinbach, P., Wesołowski, P. A., & Zeller, F. (2024). CREST—a program for the exploration of low-energy molecular chemical space. *The Journal of Chemical Physics*, *160*(11), 114110. https://doi.org/10.1063/5.0197592

Řezáč, J. (2016). Cuby: An integrative framework for computational chemistry. *Journal of Computational Chemistry*, *37*(13), 1230–1237. https://doi.org/10.1002/jcc.24312

Riniker, S., & Landrum, G. A. (2015). Better informed distance geometry: Using what we know to improve conformation generation. *Journal of Chemical Information and Modeling*, *55*(12), 2562–2574. https://doi.org/10.1021/acs.jcim.5b00654

Rosen, A. (2026). *Quacc – the quantum accelerator*. Zenodo. https://doi.org/10.5281/ZENODO.7720998

Rowan | ML-Powered Molecular Design and Simulation — rowansci.com. https://rowansci.com/.

Rozema, S. D., Tampellini, N., Rein, J., Sigman, M. S., Lin, S., & Miller, S. J. (2025). Experimental lineage and computational analysis of a general aminoxyl-based oxidation catalyst: Generality from substrate-specific interactions. *ACS Catalysis*, *15*(20), 17548–17557. https://doi.org/10.1021/acscatal.5c05893

Schmid, S. P., Seng, H., Kläy, T., & Jorner, K. (2026). Rapid generation of transition-state conformer ensembles via constrained distance geometry. *Journal of Chemical Information and Modeling*, *66*(5), 2777–2790. https://doi.org/10.1021/acs.jcim.5c02794

Souza, B. de. (2025). GOAT: A global optimization algorithm for molecules and atomic clusters. *Angewandte Chemie International Edition*, *64*(18), e202500393. https://doi.org/10.1002/anie.202500393

Tampellini, N., Choi, E. S., & Miller, S. J. (2025). Peptide-catalyzed asymmetric amination of sulfenamides enabled by DFT-guided catalyst optimization. *Journal of the American Chemical Society*, *147*(44), 41122–41129. https://doi.org/10.1021/jacs.5c15618

Tampellini, N., Mercado, B. Q., & Miller, S. J. (2024). Scaffold-oriented asymmetric catalysis: Conformational modulation of transition state multivalency during a catalyst-controlled assembly of a pharmaceutically relevant atropisomer. *Chemistry – A European Journal*, *30*(30), e202401109. https://doi.org/10.1002/chem.202401109

Tampellini, N., Mercado, B. Q., & Miller, S. J. (2025). Enantiocontrolled cyclization to form chiral 7- and 8-membered rings unified by the same catalyst operating with different mechanisms. *Journal of the American Chemical Society*, *147*(5), 4624–4630. https://doi.org/10.1021/jacs.4c17080

Tetenberg, T., Neugebauer, H., Plett, C., Santhosh, N., Bursch, M., & Riplinger, C. (2026). ORCA meets python - the ORCA python interface OPI. *Journal of Chemical Theory and Computation*, *22*(10), 4951–4967. https://doi.org/10.1021/acs.jctc.5c02141

Uhrin, M., Huber, S. P., Yu, J., Marzari, N., & Pizzi, G. (2021). Workflows in AiiDA: Engineering a high-throughput, event-based engine for robust and modular computational workflows. *Computational Materials Science*, *187*, 110086. https://doi.org/10.1016/j.commatsci.2020.110086

Wood, B. M., Dzamba, M., Fu, X., Gao, M., Shuaibi, M., Barroso-Luque, L., Abdelmaqsoud, K., Gharakhanyan, V., Kitchin, J. R., & Levine, D. S. (2025). UMA: A family of universal models for atoms. *arXiv Preprint*. https://doi.org/10.48550/arXiv.2506.23971

Young, T. A., Silcock, J. J., Sterling, A. J., & Duarte, F. (2021). autodE: Automated calculation of reaction energy profiles – application to organic and organometallic reactions. *Angewandte Chemie International Edition*, *60*(8), 4266–4274. https://doi.org/10.1002/anie.202011941