香港中文大學
The Chinese University of Hong Kong

A Review of Automated Workflow Pipelines for Computational Chemists

Huiwen Tan
10 Jul 2026

small methods

Review | Full Access

A Review of Automated Workflow Pipelines for Computational Chemists

Tong Wu, Mingzi Sun, Bolong Huang

First published: 24 June 2025 | https://doi.org/10.1002/smtd.202500308 | VIEW METRICS

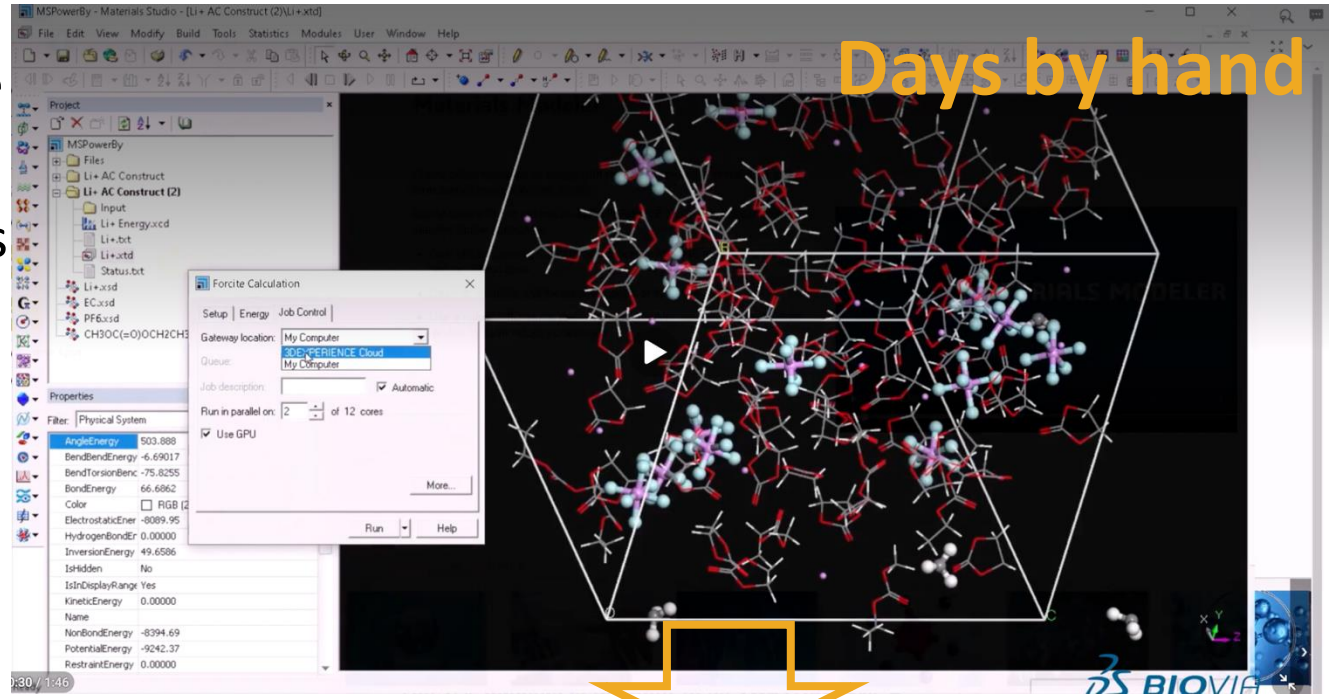
01

Background

Motivation and Key Challenges in HT-DFT Automation

Motivation: Why Automation Matters

- AI/ML-driven chemistry requires large, reliable *ab initio* datasets
- High-throughput DFT often involves thousands of calculations
- GUI-based workflows are not scalable for model building and data processing
- **Case study:** Huang et al. generated **10,164** CO_2RR adsorption models on GDY-ACs within seconds using scripts
- **Automation turns repetitive manual operations into reproducible, scalable workflows**



Seconds by tool

Three Key Challenges in HT-DFT

1. **Large-Scale Input Generation**

   Manual GUI operations become impractical when thousands of structures must be generated and saved.

2. **Parameter Optimization**

   DFT settings such as k-point density, cutoff energy, XC functional, and initial structures require careful optimization; however, MP, OQMD, and AFLOW often use fixed parameters for rapid screening.

3. **Data Silo Effect**

   Modern quantum chemistry software uses 20+ proprietary formats; one cross-software workflow may require 15–20 custom parsing scripts.

Library name	Number of supported formats	Key features
`Open Babel`	110+	Conversion of molecular topologies, support to `SMILES` (Simplified Molecular Input Line Entry System)[52] format
`cclib`	15+	Bridge computational chemistry software and cheminformatic software libraries
`c2x`	8	Conversion of input and output formats to allow usage of multiple data processing tools
`ASE`	40+	Structure data manipulations, ML interfaces
`IOData`	20+	Basis set conversion and molecular wavefunction information parsing

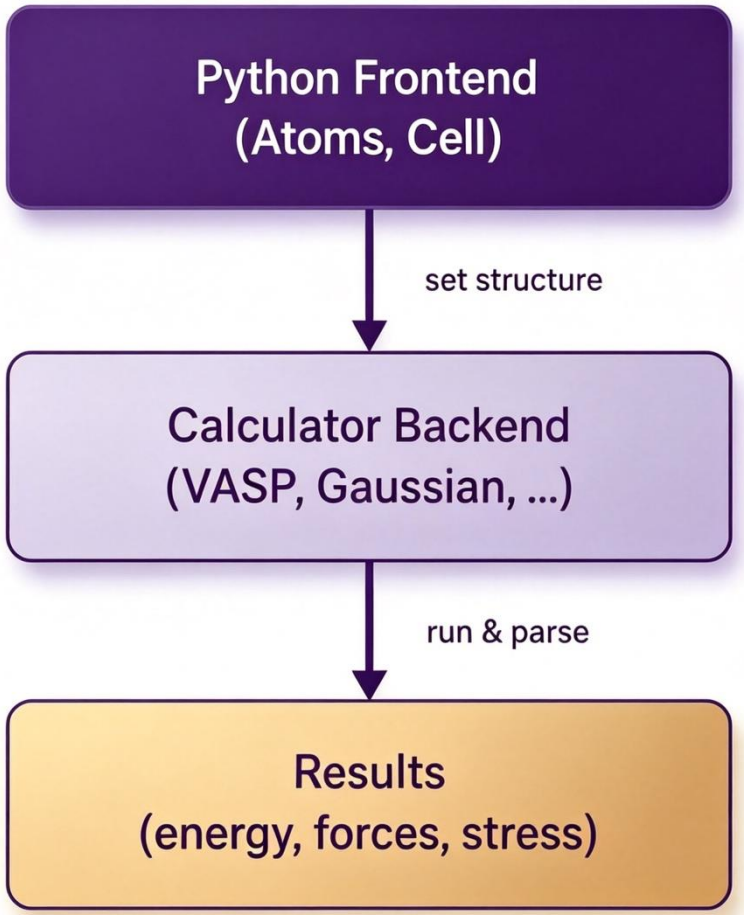
02

Software Tools — Code Libraries

ASE, Open Babel, and cclib

ASE: Atomic Simulation Environment

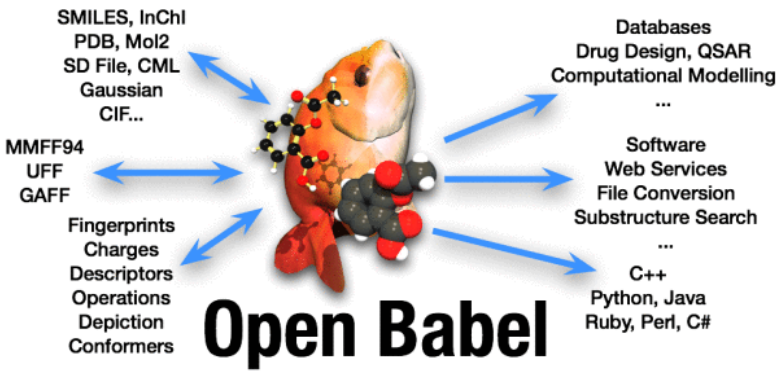
- Represents structures as Python `Atoms` objects
- Converts common formats such as POSCAR, CIF, and XYZ into programmable structures
- Uses calculator interfaces to connect with 30+ simulation codes (VASP, Gaussian, GROMACS, …)
- Enables scripted structure manipulation, e.g., automatic slab cleaving for surface screening
- Open-source and extensible, making it easy to build customized workflows
- **Limitation:** ASE is a low-level coding library; users still need programming skills and often pair it with workflow managers.



Open Babel & cclib

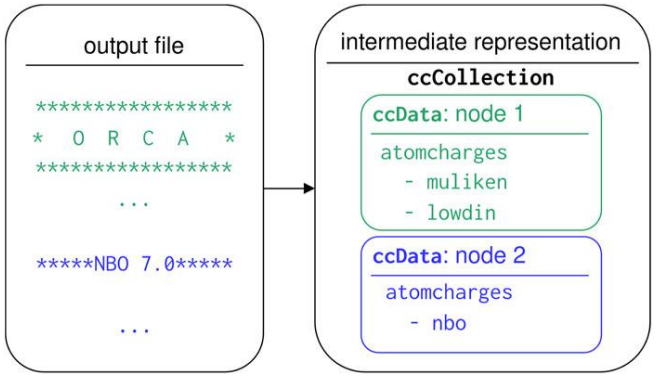
Open Babel — Structure Conversion & Cheminformatics

- Converts 110+ chemical file formats
- Handles SMILES, bond perception, atom typing, and 3D structure generation
- Supports fingerprints for substructure and similarity searches
- Widely adopted in downstream chemistry tools



cclib — Quantum Chemistry Output Parsing

- Parses outputs from 15+ quantum chemistry packages
- Extracts 70+ attributes (energies, molecular orbitals, vibrations, charges, …) into a unified `ccData` object



- Bridges QM outputs to downstream tools (Avogadro, orbkit, Goodvibes)

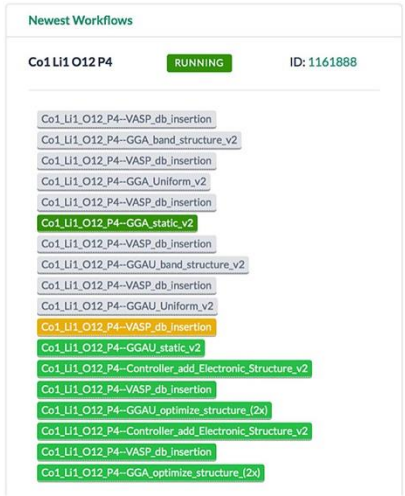
03

Software Tools — Workflow Frameworks

AiiDA, FireWorks, ChemShell, molSimplify, autodE, and AARON

AiiDA & FireWorks — General-Purpose Workflow Frameworks

Framework	Key features	Target fields	Citations	ML integration	Database integration
`AiiDA`	• DAG-based provenance • plugin system	• Verification and validation (e.g., test pseudopotentials) • New functional development • General purpose high-throughput materials screening	379[16]	Medium (via plugins)	True
`FireWorks`	• Dynamic workflows • GPU scaling support • Automatic job provenance and duplicate tracking	• Database construction (`Materials Project`)[46] • General purpose high-throughput materials screening	357[9]	Low	True



ChemShell & molSimplify — Domain-Specific Frameworks

Framework	Key features	Target fields	Citations	ML integration	Database integration
`ChemShell`	• QM/MM hybrid simulations	• Enzymes • Organometallic • Large systems • Biochemistry	300[84]	Low	False
`molSimplify`	• Ligand screening • force-field pre-optimization	• Catalysis • Coordination chemistry	117[22]	Medium	True

autodE & AARON— Reaction Mechanism Tools

Framework	Key features	Target fields	Citations	ML integration	Database integration
`autodE`	• Automated TS search • pathway mapping	• Reaction mechanisms • Organic • Organometallic	42[85]	Low	False
`AARON`	• Automatic TS structure building • Automatic conformational search • Stereoselectivity prediction	• Asymmetric catalysis • Organocatalysis • Transition metal catalysis	98[86]	Medium	True

04

Future Directions and Conclusions

Future Directions of Automation Workflows

1. **Database Integration & Provenance**
   - Build FAIR, reusable, and cross-compatible computational databases
   - Track calculation parameters and post-processing history for reproducibility
2. **ML/AI-Integrated Workflows**
   - Use active learning and ML potentials to reduce expensive DFT labeling
   - Generate descriptors and train predictive models directly from workflow outputs
3. **Chemical Space Exploration**
   - Automate reaction network discovery and molecular/structure generation
   - Combine rule-based sampling with ML-driven candidate generation and filtering
4. **AI-Assisted Parameter Optimization**
   - Predict suitable k-points, cutoff energies, and other settings before DFT runs
   - Use LLM/chatbot interfaces cautiously to lower the barrier for non-expert users

Conclusions

- Automation is essential for high-throughput computational chemistry, where GUI-based workflows cannot scale.
- Code libraries (ASE, Open Babel, cclib) serve as modular **building blocks**;
- Workflow frameworks (AiiDA, FireWorks) act as **orchestrators**, connecting input generation, job execution, error handling, and data management.
- Domain-specific tools (ChemShell, molSimplify, autodE, AARON) address specialized tasks, including QM/MM simulations, inorganic complex generation, and automated reaction mechanism exploration.
- Future workflows will increasingly combine AI-driven optimization, database integration, and FAIR-compliant reproducibility.

Thank You

ASE

- Reference: *J. Phys.: Condens. Matter* **2017**, 29, 273002.
- Website: https://docs.ase-lib.org/index.html
- Code: https://gitlab.com/ase/ase

Open Babel

- Reference: *J. Cheminform.* **2011**, 3, 33.
- Website: https://openbabel.org/
- Code: https://github.com/openbabel/openbabel

cclib

- Reference: *J. Comp. Chem.* **2008**, 29, 839-845.; *J. Chem. Phys.* **2024**, 161, 042501.
- Website: https://cclib.github.io/index.html
- Code: https://github.com/cclib/cclib

AiiDA

- Reference: *Comput. Mater. Sci.* **2016**, 111, 218-230.; *Sci. Data* **2020**, 7, 300.; *Comp. Mat. Sci.* **2021**, 187, 110086.
- Website: https://aiida.net/
- Code: https://github.com/aiidateam/aiida-core

Fireworks

- Reference: *Concurrency Computat.: Pract. Exper.* **2015**, 27, 5037–5059.
- Website: https://materialsproject.github.io/fireworks/
- Code: https://github.com/materialsproject/fireworks

ChemShell

- Reference: *J. Mol. Struct. THEOCHEM* **2003**, 632, 1–28.; *WIREs Comput. Mol. Sci.* **2014**, 4 101-110.; *J. Chem. Theory Comput.* **2019**, 15, 1317-1328.; *Phys. Chem. Chem. Phys.* **2023**, 25, 21816-21835.
- Website: https://www.chemshell.org/
- Code: Py-ChemShell can be downloaded free of charge from the ChemShell website under the open-source GNU LGPL v3 licence.

molSimplify

- Reference: *J. Comput. Chem.* **2016**, 37, 2106–2117.; *Ind. Eng. Chem. Res.* **2018**, 57, 42, 13973–13986.; *J. Chem. Inf. Model.* **2026**, 66, 5, 2753–2767.
- Website: https://molsimplify.mit.edu/; https://molsimplify.readthedocs.io/
- Code: https://github.com/hjkgrp/molsimplify

autodE

- Reference: *Angew. Chem. Int. Ed.* **2021**, 60, 4266.
- Website: https://duartegroup.github.io/autodE/index.html
- Code: https://github.com/duartegroup/autodE

AARON

- Reference*: J. Chem. Theory Comput.* **2018**, 14, 10, 5249–5261.; *WIREs Comput. Mol. Sci.* **2021**, 11, e1510.
- Website: https://github.com/QChASM/Aaron/wiki
- Code: https://github.com/QChASM/Aaron