

UMA-ASE: A FAIR-Aware Environment for Machine-Learned Interatomic Potential Workflows with UMA and ASE

Carles Bo^{1,2}, Gabriela Dias Da Silva¹, Farzaneh Hosseini^{1,2}, Ajmal Rahman Mullukkandy^{1,2}, Hugo Salazar-Lozas¹, Albert Solé-Daura^{1,2}, Yingying Wang^{1,2}

1. Institute of Chemical Research of Catalonia (ICIQ)

2. Universitat Rovira i Virgili (URV)

Abstract

Machine-learned interatomic potentials (MLIPs) are rapidly changing the operating regime of computational chemistry by narrowing the traditional gap between first-principles accuracy and large-scale atomistic simulation. UMA-ASE was developed to make this transition operational through a reproducible, FAIR-aware environment that couples the Universal Model for Atoms (UMA) with the Atomic Simulation Environment (ASE) and exposes the resulting workflows through both a command-line interface and a browser-based application. The platform combines structure intake and generation from SMILES, job submission, GeoOpt and TS search, molecular dynamics, vibrational analysis, visualization, post-processing, queue management, and integration with ioChem-BD. In this manuscript we position UMA-ASE within the current MLIP landscape and define the design goals of the new software. We emphasize reproducibility artifacts such as structured logs, machine-readable .use recipes, result bundles, and repository-oriented interoperability. We conclude that FAIR-aware scientific tools and user-friendly interfaces are becoming essential complements to modern ML methods, and that LLM-assisted software development is likely to accelerate this trend further.

Keywords

machine-learned interatomic potentials, atomistic simulation, computational chemistry, FAIR data, reproducibility, ioChem-BD, ASE, UMA

UMA-ASE: A FAIR-Aware Environment for Machine-Learned Interatomic Potential Workflows with UMA and ASE

Carles Bo^{1,2,*}, Gabriela Dias da Silva¹, Farzaneh Hosseini^{1,2}, Ajmal Rahman Mullukkandy^{1,2}, Hugo Salazar-Lozas¹, Albert Solé-Daura^{1,2}, Yingying Wang^{1,2}

¹*Institute of Chemical Research of Catalonia (ICIQ), Avinguda dels Països Catalans 16, 43007 Tarragona, Spain*

²*Universitat Rovira i Virgili (URV), Avinguda dels Països Catalans 26, 43007 Tarragona, Spain*

*Corresponding author

Abstract. Machine-learned interatomic potentials (MLIPs) are rapidly changing the operating regime of computational chemistry by narrowing the traditional gap between first-principles accuracy and large-scale atomistic simulation. UMA-ASE was developed to make this transition operational through a reproducible, FAIR-aware environment that couples the Universal Model for Atoms (UMA) with the Atomic Simulation Environment (ASE) and exposes the resulting workflows through both a command-line interface and a browser-based application. The platform combines structure intake and generation from SMILES, job submission, GeoOpt and TS search, molecular dynamics, vibrational analysis, visualization, post-processing, queue management, and integration with ioChem-BD. In this manuscript we position UMA-ASE within the current MLIP landscape and define the design goals of the new software. We emphasize reproducibility artifacts such as structured logs, machine-readable .use recipes, result bundles, and repository-oriented interoperability. We conclude that FAIR-aware scientific tools and user-friendly interfaces are becoming essential complements to modern ML methods, and that LLM-assisted software development is likely to accelerate this trend further.

Keywords: machine-learned interatomic potentials, atomistic simulation, computational chemistry, FAIR data, reproducibility, ioChem-BD, ASE, UMA

1 Introduction

Computational chemistry is undergoing a visible methodological transition, but this transition is best understood historically. In the early 1990s, broadly applicable classical approximations such as QEq and UFF provided fast, periodic-table-spanning estimates useful for prescreening and initial structure

refinement.[1, 2] Through the 1990s and 2000s, domain-optimized empirical force fields such as AMBER, CHARMM, and OPLS became central to large-scale molecular simulation because of their efficiency, while reactive force fields such as ReaxFF expanded the accessible chemistry to bond-making and bond-breaking processes at the cost of substantial system-specific parametrization effort.[3–6]

The MLIP era began with high-dimensional neural-network potentials based on atom-centered symmetry functions, which showed that near first-principles accuracy could be learned from reference data while preserving molecular-dynamics usability.[7] During the late 2010s and early 2020s, broader reviews and catalysis-focused informatics studies consolidated this shift and framed machine learning as a practical layer inside computational chemistry rather than an isolated method.[8–11]

The current phase is characterized by rapid model scaling and architectural specialization. NequIP established the practical impact of E(3)-equivariant graph neural networks; PFP/Matlantis demonstrated production-oriented, broad-coverage deployment; MACE and CHGNet expanded accuracy, scaling behavior, and charge-aware applicability; AIMNet2 broadened transferable coverage for neutral and charged organic chemistry; SO3LR pushed pretrained biomolecular simulations toward larger, physically augmented regimes; and Orb-v3 made the performance–speed–memory trade space explicit for increasingly universal atomistic models.[12–18] In parallel, UMA through FAIR-Chem positioned universal pretrained checkpoints as reusable infrastructure across molecules, materials, and catalysis, while lightweight semiempirical interfaces such as TBLite reinforced practical interoperability with xTB-style workflows.[19–24]

Recent evidence now includes both capability and caution: broad catalyst-screening benchmarks suggest useful ranking performance for UMA, while large cross-domain stress tests continue to map where MLIPs remain fragile, especially in highly fluxional or near-degenerate regions of chemical space.[25–28] At the same time, perspective work in computational catalysis increasingly describes ML potentials, generative AI, reinforcement learning, and LLM-driven workflows as converging components of a single reactivity-discovery stack, with data quality and mechanistic interpretability as persistent bottlenecks.[29]

At the same time, the rapid growth of MLIP capability has exposed a second bottleneck. In real scientific work, useful simulation is not only about predicting one energy or one force. Users need to move structures across file formats, preserve metadata such as charge, spin multiplicity, and periodic cell information, launch and monitor jobs, inspect trajectories and vibrational modes, perform first-pass analyses, package outputs, and retain enough provenance for the calculation to be interpretable and reproducible after the fact. ASE remains central here because it provides a flexible Python framework for atomic structures, optimizers, dynamics engines, and file I/O that can anchor these workflows in a reusable scientific software layer.[30]

The data-management side is equally important. ioChem-BD demonstrated early on that computational chemistry increasingly requires repository-oriented infrastructure for structured results, especially when CML-based records are expected to remain reusable rather than frozen as isolated text outputs.[31] The broader argument that computational results databases can accelerate scientific discovery, for example in catalysis, further reinforces the need to couple simulation tools with exchangeable and queryable scientific records.[32] These concerns align directly with the FAIR data principles introduced by Wilkinson *et al.*, which stress that scientific outputs should be findable, accessible, interoperable, and reusable.[33]

Beyond repository archiving, related ecosystem work also uses structured quantum-chemistry data and workflows to enable AI-assisted simulation setup and execution. Recent examples include LLM pipelines for generating ORCA input decks and autonomous multi-agent systems in the El Agente Quntur fam-

ily (e.g., El Agente Q) that plan, run, and debug quantum-chemistry tasks from natural-language prompts.[34, 35]

UMA-ASE was developed in response to this combined methodological and infrastructural shift. The project couples UMA-based calculations to ASE workflows and exposes them through both a scriptable command-line interface and a browser-based user interface. The goal is not simply to wrap a modern MLIP in a web front end, but to provide a coherent environment in which ease of use, workflow continuity, and reproducibility are treated as first-class design requirements. In the following sections, we describe the design goals of UMA-ASE, discuss the role of each main webapp tab, and examine how the software approaches reproducibility and FAIR-aware practice.

2 Design Goals of UMA-ASE

UMA-ASE was designed around four main objectives.

1. **Lower the barrier to MLIP-based simulation.** A user should be able to import a structure, inspect or edit it, select a task, and launch a calculation without manually stitching together multiple scripts, viewers, and file converters.
2. **Preserve scriptability.** The browser interface should orchestrate the same scientific workflows that remain available from the command line, so that interactive and automated usage remain consistent.
3. **Capture provenance automatically.** Logs, timestamps, run metadata, result bundles, and .use recipe files should be created as part of normal execution rather than being left to manual record keeping.
4. **Connect simulation to FAIR-aware data practice.** Structured repository records should be usable as inputs, and calculation outputs should be easy to package, share, and replay.

These goals shape the software architecture directly. ASE provides the structure handling, optimizers, dynamics engines, and file I/O.[30] UMA, through FAIR-Chem, provides the MLIP-backed calculator and checkpoint ecosystem.[19] The UMA-ASE CLI implements the actual scientific workflows for static calculations, geometry optimization, vibrational analysis, transition-state search, and molecular dynamics, while the web interface adds orchestration, persistent job management, and visualization. This keeps the browser layer from becoming a separate, hard-to-audit execution path.

3 Capabilities of the Web Application

The UMA-ASE web interface is organized around seven main tabs: **Load/Edit**, **Calculate**, **Visualize**, **Analyze**, **My Jobs**, **ioChem-BD**, and **About**. The partition is deliberately workflow-oriented: the interface follows the sequence of actions that users perform in practice rather than mirroring backend modules.

Table 1: Main tabs in the UMA-ASE web application and their primary roles.

Tab	Primary role	Representative capabilities
Load/Edit	Structure intake and preparation	Import from local files, ioChem-BD, PubChem, PDBe, and SMILES/InChI; inspect 2D and 3D representations; edit atoms; generate conformers.
Calculate	Job setup and execution	Single jobs, multi-job submission, molecular dynamics, checkpoint/task selection, streaming logs, and result bundling.
Visualize	Inspection of results	JSmol-based visualization of structures, trajectories, live MD capture, and vibrational modes.
Analyze	Post-processing	RMSD reports, XYZ pair analysis, and simple geometry analysis with downloadable reports.
My Jobs	Queue and history management	Persistent queue, per-job actions, restart/kill/download/delete operations, and automatic log/visualization loading.
ioChem-BD	Repository interoperability	Browse personal and shared trees, select destinations, and integrate with structured external records.
About	Embedded documentation	In-app README rendering, dynamic section navigation, and license access.

3.1 Load/Edit

The **Load/Edit** tab is the intake layer for atomic systems. It accepts local geometries such as .xyz, .pdb, .mol, and .cif; ioChem-BD handles and full URLs; PubChem names or CIDs; PDBe entry identifiers; and SMILES/InChI strings through RDKit-based structure generation. This breadth matters because real computational projects often begin from heterogeneous sources rather than from a single standardized format.

The tab centralizes structural metadata in a way that is immediately useful for downstream execution. Charge, multiplicity, atom count, formula, 2D sketch, IUPAC name, and conformer state are exposed in the *Atomic System Details* card whenever those properties are available or can be inferred. This creates a direct bridge between *chemical interpretation* and *calculation setup*. Users do not need to remember whether multiplicity was encoded in a CML scalar, a comment line, or a separate metadata field: the tab resolves that information into the fields needed by the calculation layer.

The tab also handles format-sensitive logic. Non-periodic CML records are preserved as CML for direct visualization, whereas periodic CML records are converted to CIF before being loaded into JSmol so that the unit cell is retained in the viewer. The same structure is then translated into an ASE-compatible representation for execution: static jobs use XYZ, while periodic molecular dynamics uses CIF. In practice this hides file-format friction without hiding scientific metadata.

Two further capabilities make **Load/Edit** more than a file loader. First, conformer generation and navigation are integrated directly into the input workflow. Users can generate conformers from SMILES or from the currently loaded structure, browse isomer/conformer collections, and inspect UFF energies when they are available. Second, atom-level editing is possible directly in the browser through JSmol selection plus undo/redo support. These operations are valuable for exploratory model-building before

launching a new MLIP calculation.

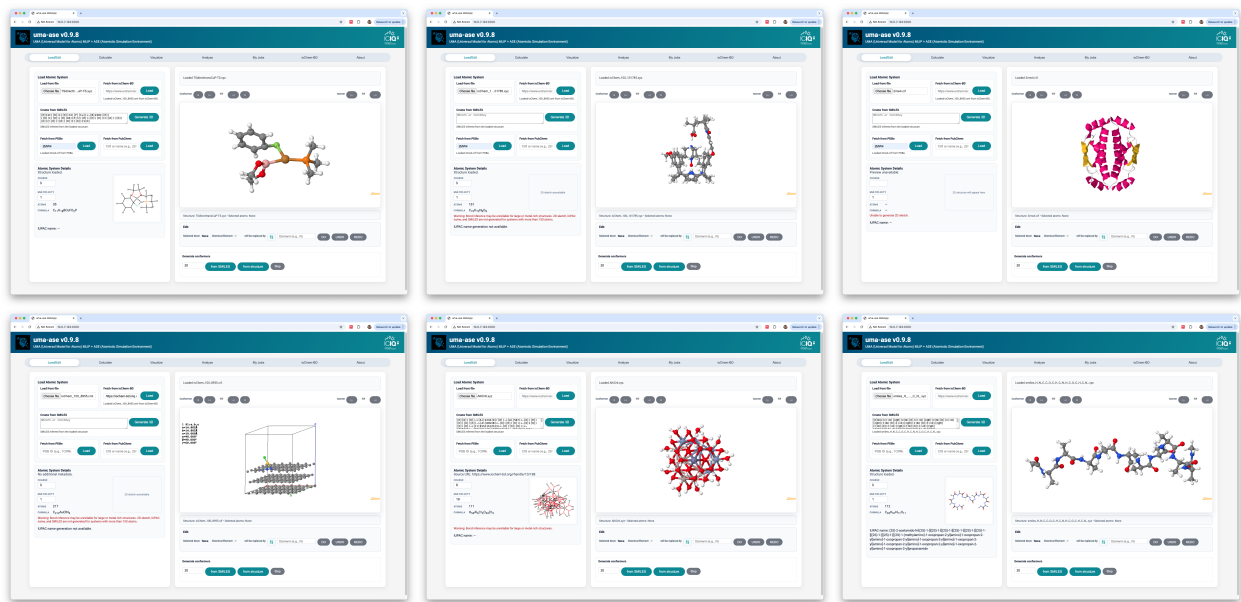


Figure 1: Mosaic view of some **Load/Edit** panel examples, highlighting structure intake, metadata inspection, IUPAC name generation, 2D image, conformer handling, and lightweight atom editing in a single preparatory workspace.

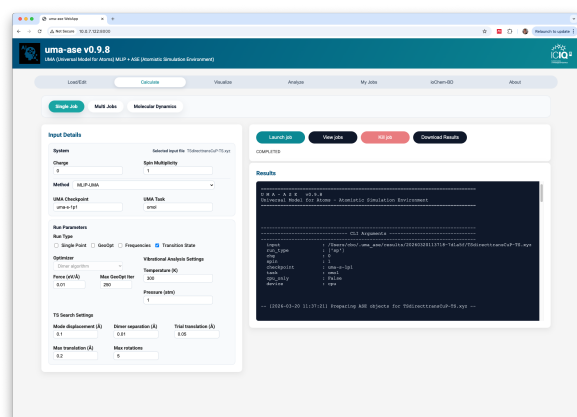
3.2 Calculate

The **Calculate** tab is the main execution hub. It combines three operational modes: *Single Job*, *Multi Jobs*, and *Molecular Dynamics*. Across these modes, the interface keeps a shared representation of the selected system, charge, multiplicity, the active method, the UMA checkpoint, and the UMA task. In the current implementation, the user-facing method is **MLIP-UMA**, while the checkpoint identifies the concrete model release used in the run.

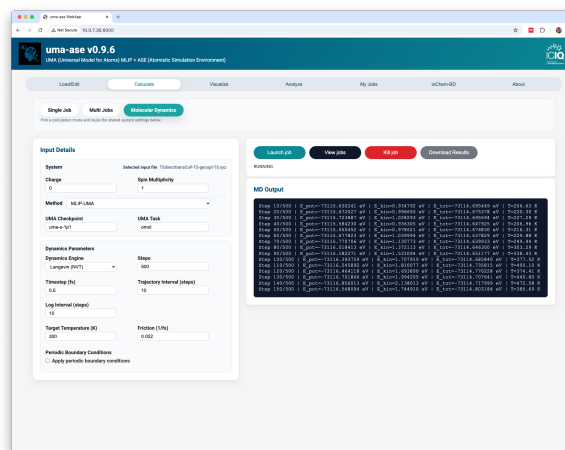
Single-job mode covers the main atomistic workflows expected from a compact MLIP environment: single-point energy evaluation, geometry optimization, vibrational analysis, and transition-state search. These workflows are presented as a coherent task family sharing the same system metadata and calculator settings. This improves usability and reduces the chance of inconsistent job setup.

Multi-job mode reflects how high-throughput work is actually performed in practice. UMA-ASE treats each selected file as an independent queued job rather than as one opaque batch record. This makes failures easier to isolate and preserves provenance on a per-system basis, while still allowing the results to be downloaded as one bundle.

The molecular-dynamics submode extends the same design to ASE-based dynamics. Thermostat and barostat controls, periodic cell parameters, and PBC activation are exposed through the interface while remaining connected to the same structure metadata used elsewhere in the application. A live MD output panel complements the persistent job system so that the calculation can be inspected during execution as well as after completion.



(a) Static and multi-job setup



(b) Molecular-dynamics setup

Figure 2: The **Calculate** tab unifies static calculations (sp, geoopt, freqs, TS), multi-job submission, and molecular dynamics while reusing the same system and method metadata.

3.3 Visualize

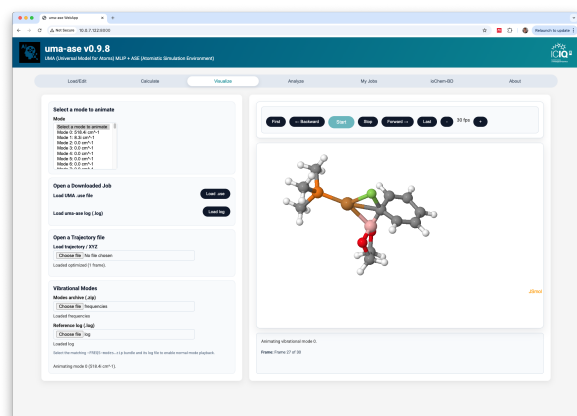
The **Visualize** tab is the interpretation layer of UMA-ASE. It handles static structures, optimized trajectories, live MD capture, completed MD runs, and vibrational-mode animation from packaged *-FREQS-modes.zip archives. Separating this functionality from **Calculate** allows the application to distinguish between execution state and inspection state, which becomes especially valuable when a running or completed job updates visualization assets without clobbering the calculation log.

For vibrational analysis, the tab can load mode bundles together with the matching log and populate the normal-mode selector for completed frequency or TS jobs. For molecular dynamics, the viewer preserves periodic cell information for completed periodic runs and recenters playback to the frame center of mass, which yields a more interpretable visual trajectory than a default origin-biased view. The same tab is also used for tracking running jobs, but in that mode it updates only the visualization-relevant widgets until the calculation finishes.

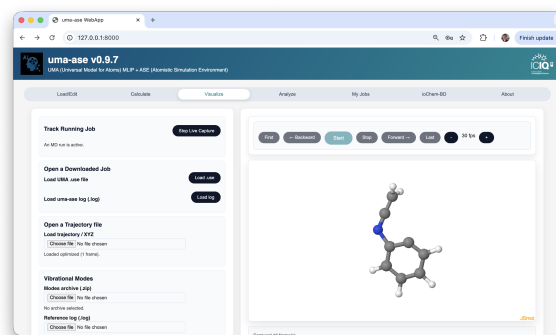
3.4 Analyze

The **Analyze** tab provides browser-triggered post-processing workflows that complement the calculation layer. In the current implementation, three analyses are available: *RMSD Report*, *XYZ Pair Analysis*, and *Simple Geometry Analysis*. These workflows deliberately remain inside the same application context rather than forcing users into a separate scripting environment for routine structural interpretation.

RMSD Report focuses on geometric variation across a selected set of structures. XYZ Pair Analysis extends this to pairwise metrics and derived visual summaries, including 3D scatter and charge-aware views. Simple Geometry Analysis provides compact geometric summaries and histograms for one geometry-centered dataset. A shared action bar and progress bar unify the interaction pattern across these analyses, while downloadable artifacts preserve the results beyond the browser session.

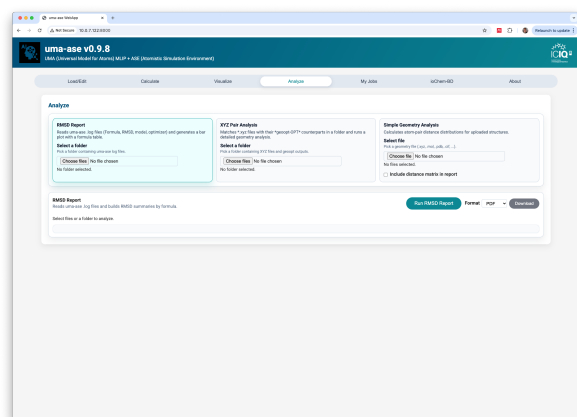


(a) Structure and trajectory view

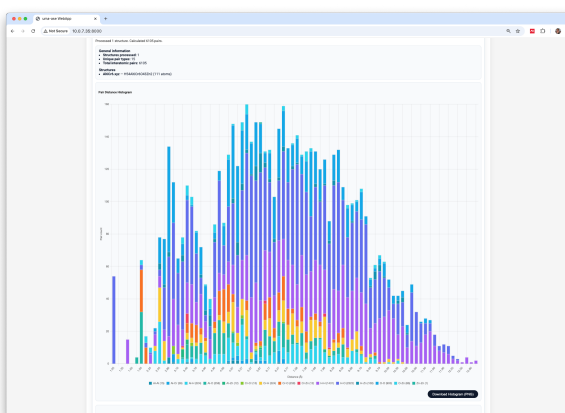


(b) Vibrational and trajectory inspection

Figure 3: The **Visualize** tab provides result-aware inspection for structures, trajectories, live MD capture, and vibrational modes.



(a) RMSD and pairwise workflows



(b) Geometry distributions and reports

Figure 4: The **Analyze** tab embeds routine post-processing workflows directly inside the same environment used for job submission and visualization.

3.5 My Jobs

The **My Jobs** tab is the persistent queue and history interface. It records queued, running, completed, and interrupted jobs together with submission times, elapsed runtime, retention indicators, and storage usage. It also provides per-job actions such as view, rename, restart MD, kill, download, and delete, as well as bulk download and bulk delete.

Scientifically, the most important feature of this tab is its result-aware loading behavior. Single-point jobs restore the log in **Calculate**; frequency jobs restore the log and the normal-mode data; geometry optimizations and TS jobs restore the log and optimized trajectories; and MD jobs restore the log while opening the completed trajectory in **Visualize**. Running jobs are opened differently: they switch to a visualization-oriented tracking mode and update only the relevant panels until they complete. This behavior preserves context rather than forcing every job type into one generic reload pattern.

The tab also reflects the fact that UMA-ASE is not designed as a transient browser front end. Queue state is persisted across server restarts, which means the job list functions as a lightweight scientific workbench history rather than as a disposable cache.

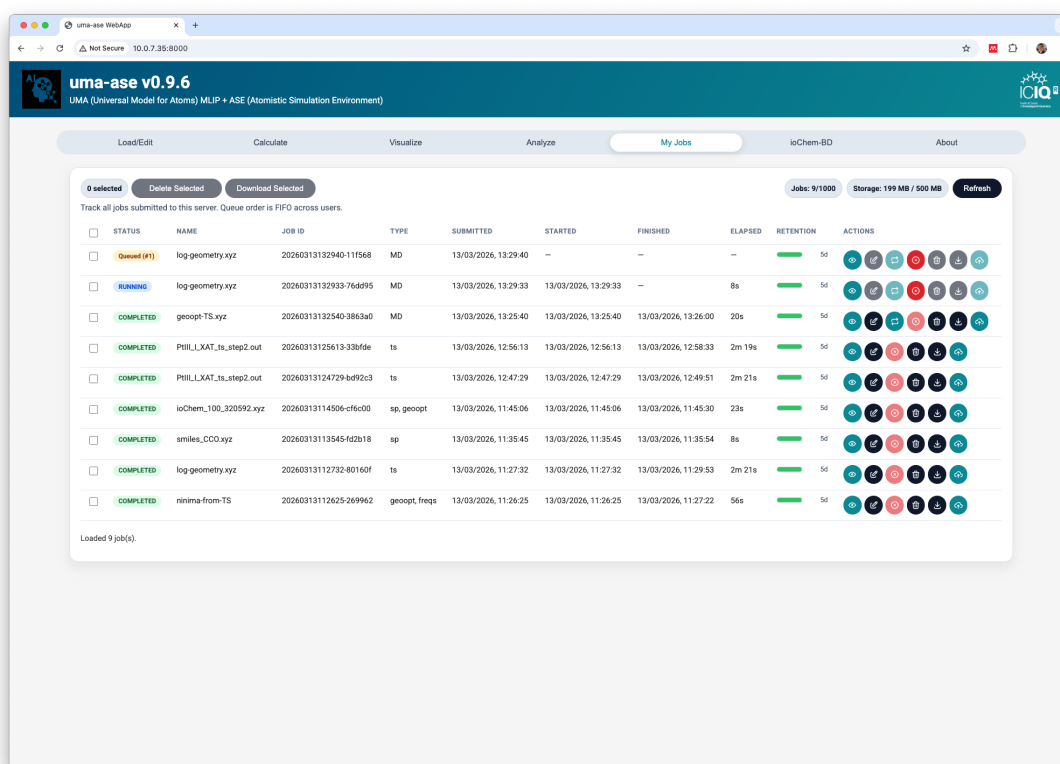


Figure 5: The **My Jobs** tab provides persistent queue management, result-aware reloading, and per-job operational controls.

3.6 ioChem-BD

The **ioChem-BD** tab is the most explicit expression of UMA-ASE’s FAIR-oriented design. ioChem-BD is a platform for managing, storing, and publishing computational chemistry results in structured form, with Chemical Markup Language (CML) playing a central role in data transformation and downstream reporting.[31] The repository perspective was later argued to be important not only for data storage, but also for accelerating discovery through more reusable computational records.[32]

Within UMA-ASE, this tab provides a route between structured external records and executable new calculations. Users can log in, browse personal and shared trees, filter the active tree, select destination paths, and use handle-based records as inputs in new simulations. The scientific value is twofold. First, it reduces the cost of moving from archived data to new MLIP-based exploration: charge, multiplicity, and periodic information can be inherited from the source record rather than manually reconstructed. Second, it embeds UMA-ASE inside a broader open-data workflow in which calculations are not isolated local artifacts but potential contributions to shared, machine-readable knowledge.

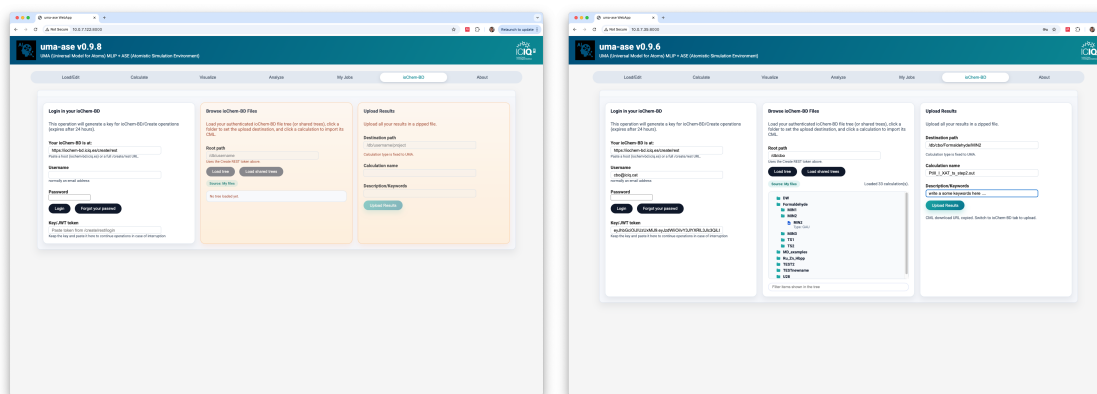


Figure 6: The **ioChem-BD** tab connects UMA-ASE to structured repository workflows for browsing, importing, exporting, and organizing computational records.

3.7 About

The **About** tab displays project documentation within the application itself. Although it is not a simulation tab in the narrow sense, it supports reproducibility in a practical way: users can inspect interface behavior, option meaning, and output structure without leaving the environment in which calculations are launched and interpreted. In the current README-based layout, no dedicated standalone screenshot is needed because the tab simply renders the embedded project documentation and license overlay.

4 REST-API Options and Endpoint Families

Beyond the graphical workflow, UMA-ASE exposes a broad REST API layer used by the web client and by potential external integrations. Most scientific routes are namespaced under `/api/uma-ase/...`, with JSON request/response payloads for control operations and dedicated binary/file routes for downloadable

artifacts. In practice, the API surface follows the same workflow partition as the interface: setup and diagnostics, structure intake, workflow execution, analysis, and job lifecycle management.

Table 2: High-level summary of REST-API option families in UMA-ASE.

Family	Representative methods	Representative endpoints and role
System and diagnostics	GET	<code>/api/uma-ase/readme</code> , <code>/api/uma-ase/config</code> , <code>/api/uma-ase/chem/status</code> , and <code>/api/uma-ase/chem/diagnostics</code> expose runtime meta-data and dependency status used for health and configuration checks.
Input resolution and import	POST/GET	Routes such as <code>/api/uma-ase/chem/resolve</code> , <code>/api/uma-ase/pubchem/fetch</code> , <code>/api/uma-ase/pdbe/fetch</code> , and <code>/api/uma-ase/iochem/*</code> normalize external identifiers and import structures/CML records into executable workflows.
Workflow execution	POST	<code>/api/uma-ase/run</code> and <code>/api/uma-ase/md/run</code> submit asynchronous single-point, optimization, transition-state, frequency, and MD calculations with user-selected checkpoints and task settings.
Analysis workflows	POST/GET	<code>/api/uma-ase/analyze</code> , <code>/api/uma-ase/analyze/driver</code> , and <code>/api/uma-ase/analyze/simple-geometry</code> generate post-processing artifacts, while tokenized download routes expose generated reports.
Job lifecycle and downloads	GET/POST	<code>/api/uma-ase/jobs</code> , <code>/api/uma-ase/job/<job_id></code> , <code>cancel/delete/rename</code> actions, and file endpoints (<code>log</code> , <code>trajectory</code> , <code>optimized structure</code> , <code>frequencies</code> , <code>bundle</code> , <code>.use</code>) provide queue control and reproducibility artifact retrieval.
Chemistry helpers	POST	Helper routes (e.g., conformer generation, preview, and structure-details endpoints) support fast pre-run validation and molecular metadata extraction directly from API clients.

This REST layer is important for reproducibility and interoperability because every browser action can be mapped to explicit, scriptable HTTP calls. As a result, advanced users can automate parts of the pipeline programmatically while preserving consistency with the same backend execution path used by the interactive interface.

5 Reproducibility, Open Data, and FAIR-Aware Practice

UMA-ASE gives reproducibility a more explicit role than is common in lightweight simulation front ends. Each run emits a structured log with timestamps, phase separators, metadata blocks, timing information, and output paths. These logs are intended to remain readable by humans, but they are also sufficiently systematic to function as audit trails of what happened during execution.

In parallel, each job produces a machine-readable `.use` file that records the parameters needed to reproduce the run. This includes the geometry payload, the job kind, the run sequence, the checkpoint, the

task, and the run-specific settings required for static calculations, transition-state search, or molecular dynamics. The `.use` file is especially important because it provides a bridge between the web interface and the CLI: a job prepared graphically can later be replayed in a scriptable context.

Result bundling adds a third layer to this reproducibility stack. Geometry files, trajectories, normal-mode archives, and auxiliary outputs are grouped by workflow outcome and can be downloaded together. This is convenient for ordinary use, but it also matters for scientific record keeping because it discourages the fragmentation of one calculation into many disconnected files with unclear provenance.

Table 3: Primary reproducibility artifacts generated by UMA-ASE.

Artifact	Purpose
<code>.log</code> file	Human-readable run record with timestamps, metadata blocks, setup timings, energies, structural summaries, and output paths.
<code>.use</code> recipe	Machine-readable description of the job inputs and parameters required to replay the calculation.
Result bundle	Collection of geometry, trajectory, vibrational, and auxiliary files grouped by workflow outcome for transfer or archiving.
Queue meta-data	Persistent record of job identity, status, timestamps, storage usage, and retention state.
ioChem-linked input	Handle- or URL-based import of structured records, preserving important metadata from the source record.

These choices align naturally with FAIR data stewardship.[33] A result is more reusable when its inputs and parameters can be reconstructed; it is more interoperable when its metadata are structured and exchangeable; and it is more findable and accessible when it can be linked to a repository-oriented workflow. ioChem-BD integration extends this logic from local reproducibility to open-data practice by connecting newly launched simulations with structured external records and repository-compatible outputs.[31, 32]

From this perspective, UMA-ASE should not be seen only as a convenience layer on top of UMA and ASE. It is also an experiment in how MLIP-based simulation environments can operationalize FAIR-aware behavior without forcing users into a heavyweight laboratory information-management system. The software remains lightweight, but it treats provenance and exchange as part of the scientific workflow.

6 Conclusions and Outlook

UMA-ASE was developed to make modern MLIP workflows usable in a form that fits the practical rhythm of computational chemistry. Its contribution is not merely to expose a pretrained model through a graphical front end, but to organize the surrounding scientific tasks that determine whether an MLIP is truly useful in practice: structure intake, metadata handling, execution, visualization, post-processing, job persistence, and repository-oriented interoperability.

The current MLIP landscape makes this kind of environment increasingly relevant. Models such as MACE, NequIP, CHGNet, Orb-v3, UMA, and PFP/Matlantis illustrate that atomistic simulation is moving toward a regime in which broad-coverage learned potentials can be used productively across many systems and

tasks. As these models continue to improve, the limiting factor for many users will increasingly be workflow quality rather than raw model capability.

For that reason, FAIR-aware tools and user-friendly scientific interfaces should be viewed as enabling infrastructure rather than as secondary embellishments. Reproducibility artifacts such as logs, machine-readable recipes, and repository-compatible bundles are not optional extras in fast ML workflows; they are what make those workflows scientifically inspectable and reusable. The same argument applies to structured interoperability with platforms such as ioChem-BD, which helps connect local simulation work to broader open-data practice.

Over roughly the past four decades, the way computational chemists interact with computing infrastructure has changed profoundly: from punched-card and command-only workflows on non-graphical terminals to touch-first interfaces and immersive 3D environments, increasingly connected in a near-transparent way to HPC centers and cloud backends.[36] At the same time, theoretical chemistry methods continue to co-evolve with advances in computing hardware and software ecosystems.[11, 37–39] Just as density-functional theory once transformed routine atomistic modeling at scale, modern machine-learned methods are now driving a new methodological inflection point.[8, 40] In that context, UMA-ASE illustrates a practical feedback loop in contemporary research: AI-assisted coding agents can accelerate development of new ML-enabled tools, and those tools can in turn accelerate scientific discovery under expert researchers supervision.

Finally, the pace at which such environments can now be built and refined is itself changing. ML methods are expanding what is computationally feasible, while LLM-assisted software development is reducing the friction of implementing, iterating, and documenting complex scientific interfaces. That combination is accelerating the emergence of a new class of chemistry software: tools that are simultaneously model-aware, workflow-aware, and FAIR-aware. UMA-ASE is a concrete example of that direction.

Acknowledgments

During development of UMA-ASE, OpenAI Codex was used to assist with software implementation tasks, including code drafting and refactoring, and code reporting. During manuscript preparation, Prism tools were used to support writing, editing, and figure/manuscript workflow operations. In all cases, the authors reviewed, validated, and edited AI-assisted outputs, and take full responsibility for the final content. The authors acknowledge the Spanish Ministry of Science, Innovation and Universities MCIN/AEI/10.13039/501100011033 (PID2023-153344NB-I00 and CEX2024-001469-S), the ICIQ Foundation, and the CERCA program of the Generalitat de Catalunya for funding. This work was carried out within the framework of the European project Open Science Clusters’ Action for Research and Society (OSCARS), funded under grant agreement N°101129751.

Software availability. The UMA-ASE source code is available from the project repository at <https://gitlab.com/carlesbo/uma-ase>.

Author contributions. A.M., F.H., Y.W., G.D., H.S., and A.S. conducted software testing, identified and reported bugs, and contributed practical ideas that improved functionality and user experience. H.S. supervised software testing and prototype implementation and performed extensive validation tests. A.S.

contributed the XYZ Pair Analysis module. C.B. conceived the original idea, designed and supervised code development, acquired funding, and wrote the original draft. All authors contributed to and approved the final manuscript.

References

- [1] Anthony K. Rappé and William A. Goddard. Charge equilibration for molecular dynamics simulations. *The Journal of Physical Chemistry*, 95(8):3358–3363, 1991. doi: 10.1021/j100161a070. URL <https://doi.org/10.1021/j100161a070>.
- [2] Anthony K. Rappé, Colwell J. Casewit, Kenneth S. Colwell, William A. Goddard, and William M. Skiff. Uff, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *Journal of the American Chemical Society*, 114(25):10024–10035, 1992. doi: 10.1021/ja00051a040. URL <https://doi.org/10.1021/ja00051a040>.
- [3] William D. Cornell, Piotr Cieplak, Christopher I. Bayly, Ian R. Gould, Kenneth M. Merz, David M. Ferguson, David C. Spellmeyer, Thomas Fox, James W. Caldwell, and Peter A. Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *Journal of the American Chemical Society*, 117(19):5179–5197, 1995. doi: 10.1021/ja00124a002. URL <https://doi.org/10.1021/ja00124a002>.
- [4] Alexander D. MacKerell, Donald Bashford, Michele Bellott, Roland L. Dunbrack, Jeffrey D. Evanseck, Martin J. Field, Scott Fischer, Jiali Gao, Haibo Guo, Daniel Joseph-McCarthy, et al. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *The Journal of Physical Chemistry B*, 102(18):3586–3616, 1998. doi: 10.1021/jp973084f. URL <https://doi.org/10.1021/jp973084f>.
- [5] William L. Jorgensen, David S. Maxwell, and Julian Tirado-Rives. Development and testing of the opls all-atom force field on conformational energetics and properties of organic liquids. *Journal of the American Chemical Society*, 118(45):11225–11236, 1996. doi: 10.1021/ja9621760. URL <https://doi.org/10.1021/ja9621760>.
- [6] Adri C. T. van Duin, Siddharth Dasgupta, Francois Lorant, and William A. Goddard. Reaxff: A reactive force field for hydrocarbons. *The Journal of Physical Chemistry A*, 105(41):9396–9409, 2001. doi: 10.1021/jp004368u. URL <https://doi.org/10.1021/jp004368u>.
- [7] Jörg Behler. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *The Journal of Chemical Physics*, 134(7):074106, 2011. ISSN 0021-9606. doi: 10.1063/1.3553717. URL <https://doi.org/10.1063/1.3553717>.
- [8] Gabriel R Schleder, Antonio C M Padilha, Carlos Mera Acosta, Marcio Costa, and Adalberto Fazzio. From DFT to machine learning: recent approaches to materials science—a review. *Journal of Physics: Materials*, 2(3):032001, 2019. doi: 10.1088/2515-7639/ab084b.
- [9] Takashi Toyao, Zen Maeno, Satoru Takakusagi, Takashi Kamachi, Ichigaku Takigawa, and Ken-ichi Shimizu. Machine Learning for Catalysis Informatics: Recent Applications and Prospects. *ACS Catalysis*, 10(3):2260–2297, 2020. ISSN 2155-5435. doi: 10.1021/acscatal.9b04186.

- [10] Oliver T. Unke, Stefan Chmiela, Huziel E. Saucedo, Michael Gastegger, Igor Poltavsky, Kristof T. Schütt, Alexandre Tkatchenko, and Klaus-Robert Müller. Machine Learning Force Fields. *Chemical Reviews*, 121(16):10142–10186, 2021. ISSN 0009-2665. doi: 10.1021/acs.chemrev.0c01111.
- [11] John A. Keith, Valentin Vassilev-Galindo, Bingqing Cheng, Stefan Chmiela, Michael Gastegger, Klaus-Robert Müller, and Alexandre Tkatchenko. Combining Machine Learning and Computational Chemistry for Predictive Insights Into Chemical Systems. *Chemical Reviews*, 121(16):9816–9872, 2021. ISSN 0009-2665. doi: 10.1021/acs.chemrev.1c00107.
- [12] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13:2453, 2022. doi: 10.1038/s41467-022-29939-5.
- [13] Toshiyuki Takamoto, Tatsuya Shinagawa, Yuji Motoyama, Kenta Nakago, Yuma Kashima, Yoshihiro Koyama, Issei Suzuki, Jens Kärtnner, Klaus-Robert Müller, Yuki Ozaki, et al. Towards universal neural network potential for material discovery applicable to arbitrary combination of 45 elements. *Nature Communications*, 13:2991, 2022. doi: 10.1038/s41467-022-30687-9.
- [14] Ilyes Batatia, Dávid Péter Kovács, Gregor N. C. Simm, Christoph Ortner, and Gábor Csányi. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. *Advances in Neural Information Processing Systems*, 36:11423–11436, 2023. Originally released as arXiv:2206.07697.
- [15] Bowen Deng, Peichen Zhong, Jialu Koh, Liang He, Yuqi Wo, Anubhav Jain, and Shyue Ping Ong. CHGNet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nature Machine Intelligence*, 5:1031–1041, 2023. doi: 10.1038/s42256-023-00716-3.
- [16] Dylan M. Anstine, Roman Zubatyuk, and Olexandr Isayev. AIMNet2: a neural network potential to meet your neutral, charged, organic, and elemental-organic needs. *Chemical Science*, 16(23): 10228–10244, 2025. doi: 10.1039/D4SC08572H. URL <https://doi.org/10.1039/D4SC08572H>.
- [17] Adil Kabylda, J. Thorben Frank, Sergio Suárez-Dou, Almaz Khabibrakhmanov, Leonardo Medrano Sandonas, Oliver T. Unke, Stefan Chmiela, Klaus-Robert Müller, and Alexandre Tkatchenko. Molecular simulations with a pretrained neural network and universal pairwise force fields. *Journal of the American Chemical Society*, 147(37):33723–33738, 2025. doi: 10.1021/jacs.5c09558. URL <https://doi.org/10.1021/jacs.5c09558>.
- [18] Benjamin Rhodes, Sander Vandenhaute, Vaidotas Šimkus, James Gin, Jonathan Godwin, Tim Duignan, and Mark Neumann. Orb-v3: atomistic simulation at scale. *arXiv*, 2025. URL <https://arxiv.org/abs/2504.06231>.
- [19] FAIR-Chem Consortium. FAIR-Chem: machine learning for atoms, molecules, and materials, 2026. URL <https://fair-chem.github.io/>. Accessed 2026-03-15.
- [20] Yoav Levine, Niv Levi, Béla Pásztor, Andreas Veit, Bastien Lesieur, Simon Duval, Chaitanya Joshi, Patricia Suriana, Florian Strub, Rodolphe Vuilleumier, Tony Lelièvre, Rishabh Agarwal, Soha Hassoun, Somesh Jha, Santiago Miret, Youssef Mroueh, Boris Kozinsky, Gabriel Srouji, and Albert

- Musaelian. Open molecules 2025 (OMol25): Advancing ai for molecular discovery, 2025. URL <https://arxiv.org/abs/2505.08762>.
- [21] Frank Neese, Frank Wennmohs, Ute Becker, and Christoph Riplinger. The ORCA quantum chemistry program package. *The Journal of Chemical Physics*, 152(22):224108, 2020. doi: 10.1063/5.0004608. URL <https://doi.org/10.1063/5.0004608>.
- [22] Frank Neese. Software update: The ORCA program system—version 5.0. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 12(5):e1606, 2022. doi: 10.1002/wcms.1606. URL <https://doi.org/10.1002/wcms.1606>.
- [23] Frank Neese. Software update: The ORCA program system—version 6.0. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 15(2):e70019, 2025. doi: 10.1002/wcms.70019. URL <https://doi.org/10.1002/wcms.70019>.
- [24] Sebastian Ehlert. tblite: Light-weight tight-binding framework, 2023. URL <https://tblite.readthedocs.io/>. Accessed 2026-03-18.
- [25] Adarsh V. Kalikadien and Evgeny A. Pidko. Performance of meta’s universal model for atoms across the conformational and configurational space of diverse transition-metal catalysts. *The Journal of Physical Chemistry A*, 2026. doi: 10.1021/acs.jpca.5c07061. URL <https://doi.org/10.1021/acs.jpca.5c07061>. Received 2025-10-14; revised 2026-01-06; accepted 2026-01-07.
- [26] Shuhao Zhang, Małgorzata Z. Makoś, Ryan B. Jadrich, Elfi Kraka, Kipton Barros, Benjamin T. Nebgen, Sergei Tretiak, Olexandr Isayev, Nicholas Lubbers, Richard A. Messerly, and Justin S. Smith. Exploring the frontiers of condensed-phase chemistry with a general reactive machine learning potential. *Nature Chemistry*, 16(5):727–734, 2024. doi: 10.1038/s41557-023-01427-3. URL <https://doi.org/10.1038/s41557-023-01427-3>.
- [27] Maksim Kulichenko, Benjamin Nebgen, Nicholas Lubbers, Justin S. Smith, Kipton Barros, Alice E. A. Allen, Adela Habib, Emily Shinkle, Nikita Fedik, Ying Wai Li, and Richard A. Messerly. Data generation for machine learning interatomic potentials and beyond. *Chemical Reviews*, 124(24):13681–13714, 2024. doi: 10.1021/acs.chemrev.4c00572. URL <https://doi.org/10.1021/acs.chemrev.4c00572>.
- [28] Igor Poltavsky, Mirela Puleva, Anton Charkin-Gorbunin, Grégory Fonseca, Ilyes Batatia, Nicholas J. Browning, Stefan Chmiela, Mengnan Cui, J. Thorben Frank, Stefan Heinen, Bingqing Huang, et al. Crash testing machine learning force fields for molecules, materials, and interfaces: molecular dynamics in the TEA challenge 2023. *Chemical Science*, 16(8):3738–3754, 2025. doi: 10.1039/d4sc06530a. URL <https://doi.org/10.1039/d4sc06530a>.
- [29] Konstantinos D. Vogiatzis, Clémence Corminboeuf, Ainara Nova, Kjell Jorner, Johannes Kästner, Markus Meuwly, Philippe Schwaller, Victor Böttcher, Maria Drosou, Edwin Fako, Hannes Hoppe, Zarko Ivkovic, Nestor Iwanjko, Dimitrios A. Pantazis, Stefan P. Schmid, Kalman Szenes, Auguste Tetenoire, and Markus Reiher. Boosting computational catalysis and chemical reactivity with artificial intelligence. *Journal of the American Chemical Society*, 2026. doi: 10.1021/jacs.5c17786. ASAP article.

- [30] Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E. Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N. Groves, Bjørk Hammer, Cory Hargus, Eric D. Hermes, Paul C. Jennings, Peter Bjerre Jensen, James Kermode, Esben Leonhard Kolsbjerg, Joseph Kubal, Kristen Kaasbjerg, Steen Lysgaard, Jón Bergmann Maronsson, Tristan Maxson, Thomas Olsen, Lars Pastewka, Andrew Peterson, Carsten Rostgaard, Jakob Schiøtz, Ole Schütt, Mikkel Strange, Kristian S. Thygesen, Tejs Vegge, Lasse Vilhelmsen, Michael Walter, Zhenhua Zeng, and Karsten W. Jacobsen. The atomic simulation environment—a Python library for working with atoms. *Journal of Physics: Condensed Matter*, 29(27):273002, 2017. doi: 10.1088/1361-648X/aa680e.
- [31] M. Álvarez-Moreno, C. de Graaf, N. López, F. Maseras, J. M. Poblet, and C. Bo. Managing the computational chemistry big data problem: The iochem-bd platform. *Journal of Chemical Information and Modeling*, 55(1):95–103, 2015. doi: 10.1021/ci500593j.
- [32] Carles Bo, Feliu Maseras, and Núria López. The role of computational results databases in accelerating the discovery of catalysts. *Nature Catalysis*, 1:809–810, 2018. doi: 10.1038/s41929-018-0176-4.
- [33] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J. G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A. C. 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joost Kok, Scott J. Lusher, Maryann E. Martone, Barend Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, 3:160018, 2016. doi: 10.1038/sdata.2016.18.
- [34] Pieter Floris Jacobs and Robert Pollice. Developing large language models for quantum chemistry simulation input generation. *ChemRxiv*, 2024. doi: 10.26434/chemrxiv-2024-9g2w2. URL <https://doi.org/10.26434/chemrxiv-2024-9g2w2>.
- [35] Yunheng Zou, Austin H. Cheng, Abdulrahman Aldossary, Jiaru Bai, Shi Xuan Leong, Jorge Arturo Campos-Gonzalez-Angulo, Changhyeok Choi, Cher Tian Ser, Gary Tom, Andrew Wang, Zijian Zhang, Ilya Yakavets, Han Hao, Chris Crebolder, Varinia Bernales, and Alan Aspuru-Guzik. El agente: An autonomous agent for quantum chemistry. *SSRN*, 2025. doi: 10.2139/ssrn.5245665. URL <https://doi.org/10.2139/ssrn.5245665>.
- [36] Callum Devereux, George Booth, Christopher J. Woods, Adrian J. Mulholland, and David R. Glowacki. Interactive molecular dynamics in virtual reality for modelling materials and catalysts. *Journal of Molecular Graphics and Modelling*, 125:108606, 2023. doi: 10.1016/j.jmgm.2023.108606. URL <https://doi.org/10.1016/j.jmgm.2023.108606>.
- [37] Jack Dongarra and David Keyes. The co-evolution of computational physics and high-performance computing. *Nature Reviews Physics*, 6:621–627, 2024. doi: 10.1038/s42254-024-00750-z. URL <https://doi.org/10.1038/s42254-024-00750-z>.

- [38] Edward F. Valeev. Many-body quantum chemistry on massively parallel computers. *Chemical Reviews*, 121(3):1203–1231, 2021. doi: 10.1021/acs.chemrev.0c00006. URL <https://doi.org/10.1021/acs.chemrev.0c00006>.
- [39] Bingqing Cheng, John A. Keith, Valentin Vassilev-Galindo, Stefan Chmiela, Michael Gastegger, Klaus-Robert Müller, and Alexandre Tkatchenko. The next revolution in computational simulations: Harnessing ai and quantum computing in molecular dynamics. *Current Opinion in Structural Biology*, 86:102919, 2024. doi: 10.1016/j.sbi.2024.102919. URL <https://doi.org/10.1016/j.sbi.2024.102919>.
- [40] Walter Kohn and Lu Jeu Sham. Self-consistent equations including exchange and correlation effects. *Physical Review*, 140(4A):A1133–A1138, 1965. doi: 10.1103/PhysRev.140.A1133. URL <https://doi.org/10.1103/PhysRev.140.A1133>.