

AQME: Automated Quantum Mechanical Environments for Researchers and Educators

Juan V. Alegre-Requena,^{1,‡,*} Shree Sowndarya S. V.,^{2,‡} Turki M. Alturaifi,² Raúl Pérez-Soto,² Robert S. Paton^{2,*}

¹ Departamento de Química Inorgánica, Instituto de Síntesis Química y Catálisis Homogénea (ISQCH) CSIC-Universidad de Zaragoza, Zaragoza 50009, Spain.

² Department of Chemistry, Colorado State University, Fort Collins, Colorado 80523, USA

‡ These authors contributed equally.

jv.alegre@csic.es; robert.paton@colostate.edu

Abstract

AQME, Automated Quantum Mechanical Environments, is a free and open-source Python package for the rapid deployment of automated workflows using cheminformatics and quantum chemistry. AQME workflows integrate tasks performed across multiple computational chemistry packages and data formats, preserving all computational protocols, data, and metadata for machine and human users to access and reuse. AQME has a modular structure of independent modules that can be implemented in any sequence, allowing the users to use only the desired parts of the program. The code is intended for researchers with basic familiarity with the Python programming language. The CSEARCH module interfaces to molecular mechanics and semi-empirical QM (SQM) conformer generation tools (e.g., RDKit and Conformer–Rotamer Ensemble Sampling Tool, CREST) starting from various initial structure formats. The CMIN module enables geometry refinement with SQM and neural network potentials, such as ANI-1. The QPREP module interfaces with multiple QM programs, such as Gaussian, ORCA, and PySCF. The QCORR module processes QM results, storing structural, energetic, and property data while also enabling automated error handling (i.e., convergence errors, wrong number of imaginary frequencies, isomerization, etc.) and job resubmission. The QDESCP module provides easy access to QM ensemble-averaged molecular descriptors and computed properties, such as NMR spectra. Overall, AQME provides automated, transparent, and reproducible workflows to produce, analyze and archive computational chemistry results. SMILES inputs can be used, and many aspects of tedious human manipulation can be avoided. Installation and execution on Windows, macOS, and Linux

platforms has been tested, and the code has been developed to support access through Jupyter Notebooks, the command line, and job submission (e.g., Slurm) scripts. Examples of pre-configured workflows are available in various formats, and hands-on video tutorials illustrate their use.

1. INTRODUCTION

Continued improvements to computer hardware and algorithms have meant that quantum chemical studies are increasingly applied to study ever-larger and conformationally more flexible molecules and molecular datasets of increasing size. Computational high-throughput screening, chemical space exploration and molecular optimization, and the construction of high-quality datasets to train emerging machine learning models rely on the ability to execute, analyze and store the results of large quantum chemistry campaigns.¹⁻⁸ These tasks typically require more than one calculation per molecule. Instead, a sequence of molecule building, conformational analysis and refinement, optimization and thermochemical analysis, property prediction, and ensemble averaging is often performed. Each step may involve a different model chemistry (e.g., MM, SQM, QM, NN) executed by a distinct package.

Further, these efforts are multiplied by the number of molecules. Automating these multi-step workflows minimizes manual effort and human error and enables the complex protocols and their associated data and metadata to be fully captured and reused.⁹⁻¹³ Automated workflows for QM calculations, including those that address the important challenge of transition state (TS) location and conformational analysis (e.g., Wheeler's QChASM¹⁴ and Duarte's autodE¹⁵), have emerged as powerful software tools, such as AARON,¹⁶ ACE,¹⁷ Aiiida,¹⁸ Auto-QChem,¹⁹ CatVS,²⁰ Chemistream,²¹ ChemShell,²² FireWorks,²³ molSimplify,²⁴ PyADF,²⁵ and QMflows,²⁶ among others. In this work, we focus on the development of automated end-to-end workflow software to perform multi-step computational tasks spanning multiple programs and theoretical methods.

The modular design of AQME provides opportunities for workflow customization, use in Jupyter notebooks, and integration into Python projects.²⁷ Ready-to-use examples with different degrees of complexity are provided via GitHub,²⁸ supplemented by hands-on video tutorials.²⁹ Additionally, users can create workflows for new applications as desired, even in fields that are not considered in the existing examples. For example, a researcher can create a workflow to calculate a reaction energy profile and, afterwards, tune the module combination to generate QM molecular descriptors to use in machine learning (ML) models.

Currently, the program contains five modules designed for different tasks (Figure 1) that can be executed in any order; individual modules can be skipped if required. The input format can be a SMILES representation or many types of structure formats (SDF, PDB, XYZ, among others). The first module, CSEARCH, automates conformational analysis. This is interfaced with molecular mechanics potentials through RDKit³⁰ and semi-empirical potentials through xTB. Searches can be performed using RDKit or CREST,³¹ or using internally-coded systematic or Monte Carlo torsion sampling protocols. Then, CMIN is designed to refine the geometries and relative energies obtained in the initial conformer generation by using semi-empirical methods (xTB)³² and ML potentials (ANI-1).³³ However, this module can also be used to process multiple 3D input formats independently. The next module, QPREP, converts a wide variety of 3D formats into input files for QM calculations with several packages, such as Gaussian,³⁴ ORCA,³⁵ and PySCF.³⁶ Tedious tasks such as creating Gen(ECP) sections or including final lines (i.e., NBO extra keywords) in the input files are handled automatically. QCORR is a cclib-based³⁷ module that detects issues and errors in QM output files, structures all output data, and creates ready-to-submit input files to correct those issues. User-specified criteria (i.e., spin contamination, isomerization, etc.) can be defined to filter output data. The last module, QDESCP, is designed to generate Boltzmann ensemble-averaged molecular QM properties or SQM descriptors, which can be readily used in ML models. Commonly used descriptors such as atomic charges, bond orders, dipole moment, and solvation energy are included.

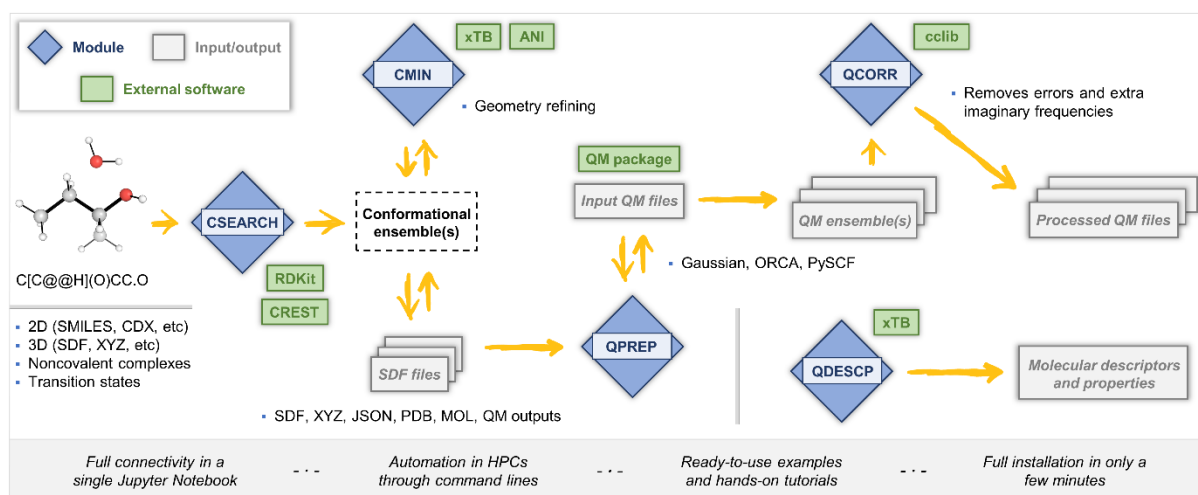


Figure 1. General workflow including the modules available in AQME and their connectivity with external programs.

INSTALLATION AND TECHNICAL DETAILS

The program presented is a free, open-source Python-based software that can be installed via conda-forge (`conda install -c conda-forge aqme`) or Python Package Index (`pip install aqme`). All the dependencies required to run AQME are installed automatically, except for RDKit and Openbabel when using pip install. Along with the software, we set up tests through Pytest, Circle CI, and Codecov, which allows us to ensure a correct functioning of a significant proportion of the code. Also, multiple code analyzers (CodeFactor, Codacy, and LGTM) were employed to improve the quality standards and readability of the deployed code.

MODULES OF AQME

AQME is divided into modules that can be called as part of a workflow or separately. Four main applications enclose these modules: i) conformer generation and geometry refinement (CSEARCH and CMIN), ii) generation of QM input files (QPREP), iii) post-processing of output files (QCORR), and iv) generation of Boltzmann weighted descriptors (QDESCP). In this section, the technical details of the modules are disclosed in more detail.

Conformer Generation and Geometry Refinement: The CSEARCH and CMIN Modules

The availability of numerous conformer generators for small molecules (i.e., ConfGen,³⁸ OMEGA,³⁹ Frog2,⁴⁰ etc.) highlights the central importance of this task. In the CSEARCH module, AQME gathers multiple types of conformational search tools. It can be used simply as an interface to the external conformer generation protocols available in RDKit or CREST, or to perform torsion-based sampling internally while making use of the MM or SQM potentials in those packages (Figure 2A). When starting from SMILES strings, CSEARCH attempts to derive molecular charge and multiplicity, although this can be manually overwritten (*charge* and *mult* options). The number of simultaneous processes is controlled by the *max_workers* option; the number of processors used by each process with the *nprocs* option.

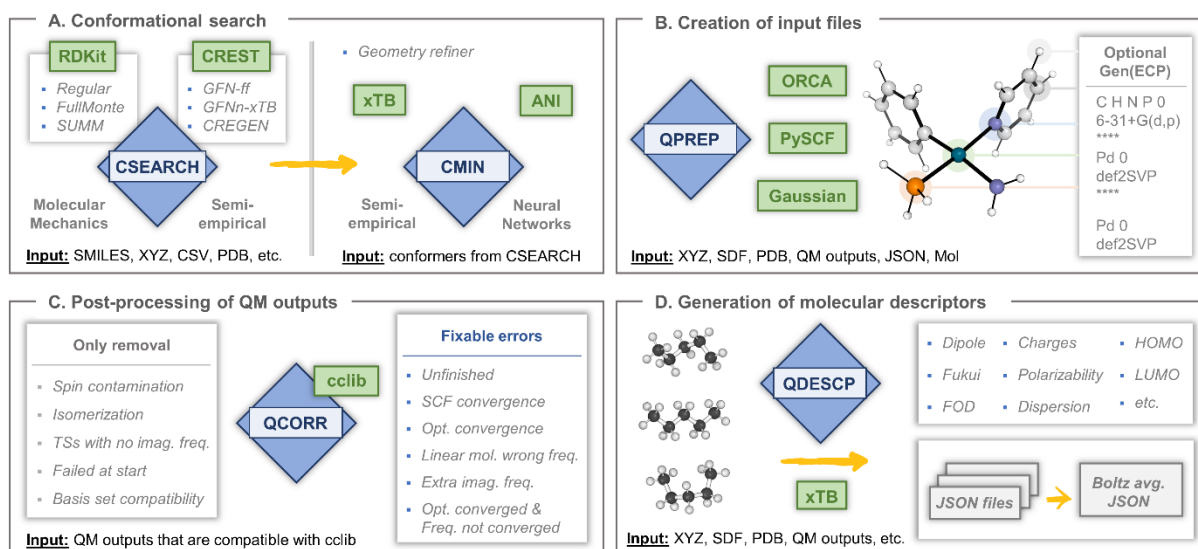


Figure 2. Modules in AQME. (A) Methods and parameter options in CSEARCH and CMIN modules. (B) Recognition of atoms to include in the GenECP section of an example organometallic complex when using `gen_atoms=["Pd"]`. (C) Outline of error fixed in the QCORR module. Additional creation of input files for calculations for properties such as NMR, NBO, or higher-level single point energy evaluation. (D) Generation of Boltzmann averaged molecular properties using the QDESCP module for conformer ensembles.

The Grimme group's CREST generates conformers by extensive metadynamics sampling using semi-empirical methods (GFNn-xTB) or force fields (GFN-FF), with an additional genetic Z-matrix crossing step at the end. User-defined constraints for atom positions, bond distances, angles and dihedral angles enable approximate TS structures to be sampled. When starting from 1D and 2D structures, RDKit is used to generate the necessary 3D input for CREST. Before the sampling, two initial xTB optimizations are performed to avoid errors. During the first optimization, the calculations are performed with all the bonds frozen in addition to any user-defined constraints. This preparatory calculation avoids problems related to the superimposition of molecules when the input molecules are generated from 1D or 2D inputs. In the second optimization, only the user-defined constraints are included. Afterward, the CREST search is carried out, including any additional keywords specified in the `crest_keywords` option (i.e., `crest_keywords="--nci --cbonds 0.5"`).

MM-based searches are faster and may be necessary for large systems or large numbers of molecules. The first method (specified through `program="RDKit"`) uses the ETKDG protocol, with various

parameters we have tuned based on experience to obtain the most significant conformers while reducing the number of duplicates. These protocols have been extensively benchmarked in studying flexible druglike compounds and natural products (see the ESI). When the initial conformational sampling fails, the program automatically tries to address the problems through a series of changes to its initial protocol (i.e., changing MMFF for UFF, using random coordinates for molecular embedding, etc.), making the protocol more robust. CSEARCH also tries to overcome other severe limitations in the conformational sampling of molecules that contain transition metals or atoms with uncommon hybridization (i.e., pentacoordinate P atoms). Additionally, common templates for organometallic compounds, such as linear, trigonal planar, and square-planar geometries, can be used. These geometries are not usually obtained with standard RDKit protocols (i.e., square-planar metal complexes lead to tetrahedral structures), which may then neglect an essential aspect of conformational behavior. Internal torsional sampling approaches are also implemented, such as the systematic unbounded multiple minimum (SUMM)⁴¹ and Monte Carlo Multiple Minimum (MCMM) algorithms.^{42,43} The SUMM approach surveys dihedral angles that are progressively varied by a user-specified increment, while MCMM applies random values to a random subset of the rotatable torsions. These two methods require more time than the standard RDKit sampling, but they might render more accurate results in cases with complex conformational spaces. Finally, the CMIN module refines the energies and geometries of the structures obtained with RDKit or other low-level methods before optimizing with more demanding levels of theory, such as density functional theory (DFT). xTB or ANI-1 methods typically result in a reordering of relative conformational stabilities closer to QM results and the removal of duplicate conformations.

QM Input File Preparation: The QPREP Module

The QPREP module is designed to convert multiple formats (SDF, XYZ, PDB, JSON, LOG/OUT) into input files for QM programs ready to be submitted without further modification. QPREP currently generates inputs for Gaussian, ORCA, and PySCF. One of the most convenient features of this module is that the Gen and GenECP specifications from Gaussian input files are automatically written. When the user specifies atom types to use in `gen_atoms`, QPREP detects the atom types of the molecule and separates them into two groups for the input GenECP section. For example, the users can set the 6-31+G(d,p) basis set for C, H, N, and P atoms while using def2-SVP for Pd atoms (Figure 2B). This automated protocol avoids the tedious manual setup of all the types of atoms in the GenECP part, which

is especially helpful when working with different families of compounds or with big molecular datasets. The input keywords for the generated input files are specified through the *qm_input* option, and other parameters can be edited as preferred, such as charge, multiplicity, generation of CHK files, number of processors, and memory. Also, the user can include final lines after the molecular coordinates (i.e., NBO keywords).

Post-processing of Output Files: The QCORR Module

Typically, a tedious manual search and correction for error terminations, convergence issues, and extra imaginary frequencies is necessary after running QM calculations. Based on our experience with structure optimizations and frequency calculations for large databases (i.e., many thousands) of organic compounds, such occurrences are relatively common.⁴⁴ QCORR structures output data and automatically detects issues or errors, creating new input files that try to correct those issues, a cycle that can be repeated several times (Figure 2C). Usually, only one QCORR analysis is necessary since the number of calculations with issues is reduced considerably.⁴⁴ Structural isomerizations can be automatically detected and filtered. Also, calculations with spin contamination will be removed if $\langle S^2 \rangle$ differs from $s(s+1)$ by more than 10%, as previously suggested.⁴⁵ The filter can be disabled or adjusted to other thresholds with the *s2_threshold* option. QCORR uses the *cclib* Python library and stores all parsed data as JSON files since this format allows other Python tools to retrieve the information easily. By default, QCORR uses this information to detect all calculations for consistency in terms of calculation type and software version.

Generation of Boltzmann Weighted Properties and Descriptors: The QDESCP Module

Boltzmann-weighted values are generally advisable for molecules with multiple conformers when comparing computed results with experimental observables. AQME couples with xTB to generate molecular properties for large chemical databases. Starting from 3D geometries, atomic properties such as charges, fractional occupation densities, Fukui indices, D3-dispersion coefficients, and molecular properties including dipole moment, HOMO-LUMO gap, polarizability, energy are determined for every conformation of a molecule (Figure 2D). Then, Boltzmann averaged properties are calculated and stored separately. This protocol enables the generation of molecular descriptors as starting points for ML models. Additionally, QDESCP can be used to compute Boltzmann averaged nuclear magnetic

resonance (NMR) chemical shifts from DFT calculations. The user can specify slope and intercept to scale the results to the tetramethylsilane (TMS) scale using tools such as The Tantillo group's CHESHIRE repository,⁴⁶ rendering simulated spectra that can be compared directly with experimental spectra.

END-TO-END WORKFLOWS

In this section, we detail four illustrative workflows that have been adapted for different applications regularly carried out in our group, such as calculating energy profiles and generating molecular descriptors for ML models. These workflows are available on Figshare⁴⁴ along with associated data and metadata, in three formats: Jupyter Notebook, SLURM script, and command-line script. Hands-on tutorials have been uploaded to YouTube.²⁹ For large systems or datasets, we typically execute end-to-end AQME workflows on a cluster using SLURM commands: the overall time taken is dominated by the QM calculation steps.

The Conformational Distribution and ¹H chemical Shifts of Strychnine from SMILES Input

Strychnine is a natural alkaloid produced by different plants of the genus *Strychnos*, whose complexity and pharmacological properties have attracted many organic synthetic groups over time.⁴⁷ Recently, John, Reinscheid, and coworkers reported two different conformers in a 97:3 ratio observed in NMR studies.⁴⁸ The following AQME workflow aims to identify these two structures and simulate an averaged NMR spectra starting from a SMILES string, using a combination of i) RDKit conformer sampling, ii) Gaussian geometry optimization with B3LYP/6-31+G(d,p), iii) fixing errors and imaginary frequencies of the output files, iv) GoodVibes⁴⁹ calculation of Boltzmann distributions using Gibbs free energies at 298.15 K, and (v) Boltzmann averaged shielding tensor calculations (empirically-scaled to obtain chemical shifts) with B3LYP/6-311+G(2d,p), SMD = CHCl₃ (Figure 3). Using the CSEARCH module with RDkit yields two conformers which are utilized further for DFT optimization. The calculated Boltzmann distribution for the two conformers is 99:1, which correlates well with the experimental observation of 97:3. Furthermore, the predicted ¹H chemical shifts present a low mean average error (MAE, 0.14 ppm for nine known ¹H signals) compared to the experimental values.⁵⁰ This workflow did not require manual intervention and suggests that further applicability of AQME to automate NMR prediction and organic structure elucidation merits investigation.

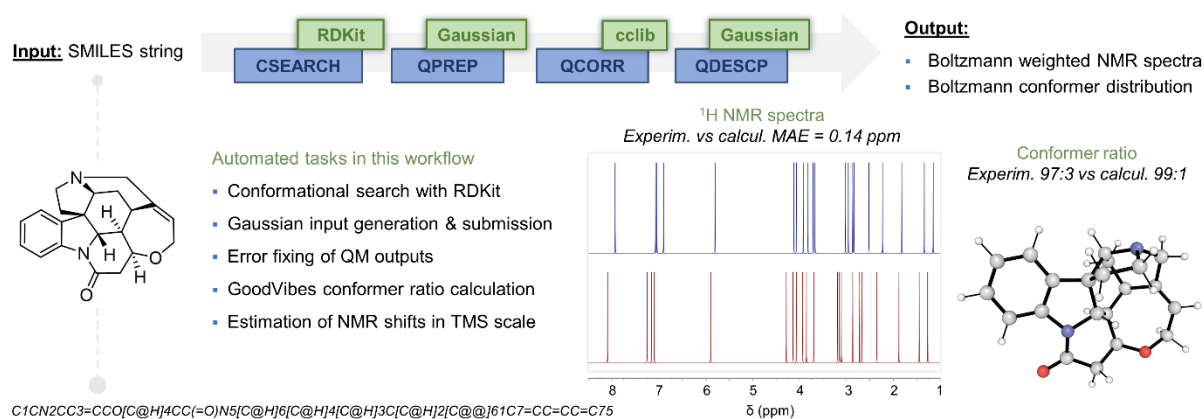


Figure 3. End-to-end workflow to calculate the conformer distribution and scaled (TMS) ¹H NMR chemical shifts of strychnine in CHCl₃.

Comparing Diels-Alder Activation Barriers from Multiple SMILES Inputs

AQME can also be employed to generate energy profiles. Figure 4 summarizes a workflow where a CSV input containing a list of SMILES is used to generate the reaction energy profiles for the Diels-Alder cycloadditions of multiple cyclic dienophiles, each reacting with cyclopentadiene.⁵¹ This workflow might be especially useful to calculate reaction energies between reagents and products from two lists of SMILES. TSs might also be included by specifying the corresponding atomic and structural constraints, with no limit in the number of TSs provided in the same reaction pathway. The approach represents a constraint-based complementary method to other more specialized tools that calculate energy profiles, such as autodE.¹⁵

The workflow presented illustrates a somewhat typical multi-step combination of SQM conformer sampling, geometry optimizations and single point energy corrections with different levels of theory, and the generation of a potential energy surface diagram. AQME links together the following tasks: i) CREST conformer sampling, ii) Gaussian geometry optimization (B3LYP/def2-TZVP), iii) fixing errors and imaginary frequencies of the output files, iv) ORCA single point energy corrections using DLPNO-CCSD(T)/cc-pV(DT)Z, and v) Boltzmann weighted thermochemistry calculation and PES generation with GoodVibes at 298.15 K. Manual intervention and the use of Spreadsheets is avoided.

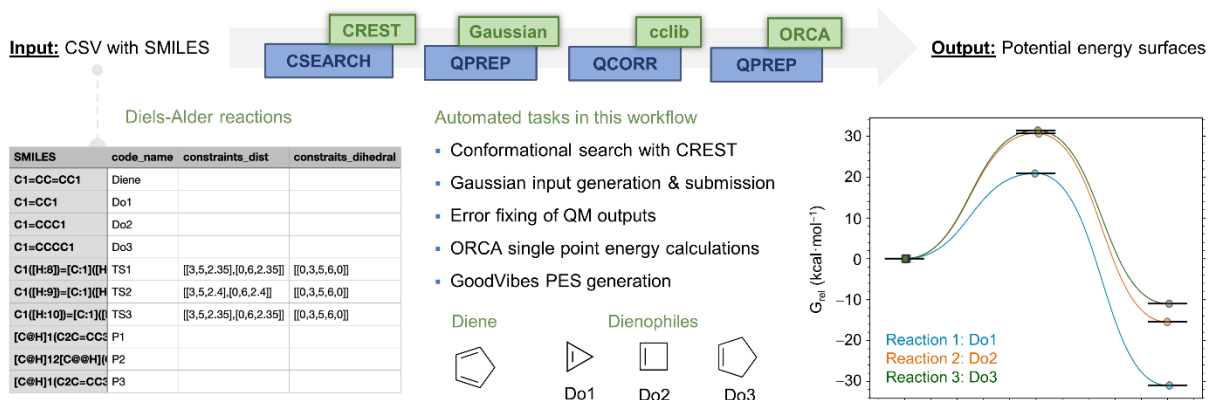


Figure 4. End-to-end workflow to generate the energy profile of multiple Diels-Alder reactions using a CSV as the input file.

Generating QM or SQM Molecular Descriptors for a Large Dataset

Statistical and ML applications in chemistry are often enhanced by using feature vectors or parameters derived from QM calculations.⁵² For example, QM-derived atomic charges or populations are often used in multivariate linear regression or neural network models.⁵³ Figure 5 shows a workflow that employs the SMILES-containing ESOL database⁵⁴ to predict aqueous solubility with a message passing graph neural network model. There are 1128 datapoints available, which are split into training (1028), validation (50) and test (50) sets. The protocol includes i) RDKit conformer sampling, ii) xTB descriptor generation (Boltzmann weighted), and iii) neural fingerprint (nfp)⁵⁵ based graph neural network (GNN) model creation. In the first step, the CSEARCH module creates 3D conformations using the SMILES strings of the database with RDKit. Then, QDESCP uses xTB to generate molecular and atomic properties such as dipole moments, charges, Fukui indexes, dispersion parameters, polarizability, and HOMO-LUMO gaps, among others. These properties are provided individually for each conformer and as Boltzmann averaged values.

The Boltzmann weighted xTB parameters are then used as descriptors in a GNN model. The GFN2-xTB atomic properties are encoded as node features and the molecular properties are passed as global features in the input graph structure. This graphical representation of the molecule with embedded atomic and molecular properties is used to build a message passing GNN. The GNN model utilized the AdamW optimizer, and model performance was assessed by measuring the mean absolute error during training for 500 epochs. The model showed an R^2 and MAE of 0.8 in the held-out test set of 50 molecules. Hyperparameter tuning can be performed along with feature selection to further improve this

accuracy. Other ML models, such as random forests can also be employed in this workflow instead of the GNN shown.

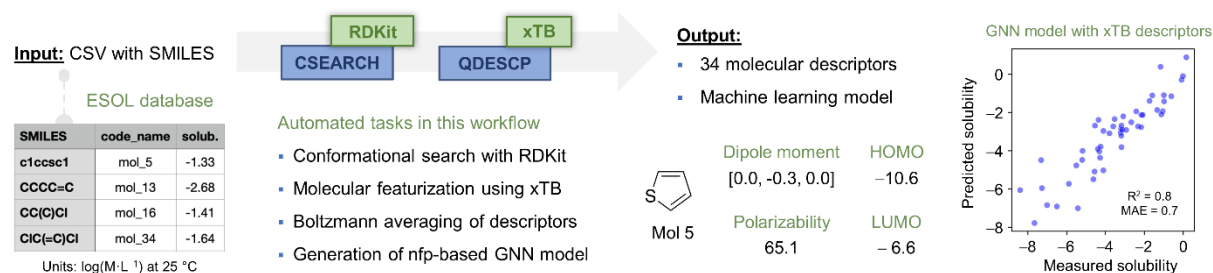


Figure 5. End-to-end workflow to create and use molecular descriptors in a GNN model starting from a database of SMILES strings.

Conclusion

AQME is an open-source Python package for building computational workflows to perform multi-step protocols efficiently that combine different packages and model chemistries. The approach is well-suited to general tasks incorporating conformational analysis, geometry refinement, QM optimizations, and ensemble-averaged property predictions. Representative examples of chemical shift prediction, reaction energy profile calculation, and dataset featurization are shown here, in each case operating as a fully automated ('end-to-end') workflow from the supplied inputs to desired Boltzmann-averaged outputs. Inputs can be supplied in SMILES or several 3D structure formats. This approach captures and preserves protocols used at every stage of the research process: there are no extraneous Spreadsheets involved and all steps can be reproduced by other researchers.

The software consists of independent modules that can be combined in any order. The CSEARCH module performs conformational sampling or interfaces to external conformational analysis tools using MM (RDKit) or SQM (xTB, CREST) levels of theory. CMIN allows the refinement of conformer ensembles with xTB or ANI-1 methods. The QPREP module converts files with different 3D input formats into input files for external QM programs, such as Gaussian, ORCA, and PySCF. The QCORR module analyzes QM output data by systematically processing output files. Filters for e.g., convergence errors, imaginary frequencies, and undesired structural isomerization, can be implemented to create new inputs automatically. The QDESCP module produces Boltzmann-weighted descriptors and properties. The modular structure means AQME can be used in Jupyter notebook environments or reused and imported by other Python projects.

Author Contributions

Juan V. Alegre-Requena: Conceptualization (lead); writing–original draft (lead); writing–review and editing (lead). **Shree Sowndarya S. V.:** Conceptualization (lead); writing–original draft (lead); writing–review and editing (lead). **Turki M. Alturaifi:** Conceptualization (supporting); writing–original draft (supporting); writing–review and editing (supporting). **Raúl Pérez-Soto:** Conceptualization (supporting); writing–original draft (supporting); writing–review and editing (supporting). **Robert S. Paton:** Conceptualization (lead); writing–original draft (supporting); writing–review and editing (supporting).

Conflict of Interest

There is no conflict of interest.

Funding Information

J.V.A.-R. acknowledges financial support through a Juan de la Cierva Incorporación contract from the Ministry of Science and Innovation (MCIN) and the State Research Agency (AEI) of Spain, and the European Union (NextGenerationEU/PRTR) under grant reference IJC2020-044217-I. R.S.P. acknowledges support from the NSF (CHE-1955876) and computational resources from the RMACC Summit supercomputer supported by the National Science Foundation (ACI-1532235 and ACI-1532236), the University of Colorado Boulder and Colorado State University, and the Extreme Science and Engineering Discovery Environment (XSEDE) through allocation TG-CHE180056. The results and analysis presented in this paper were partially possible thanks to the access granted to computing resources at the Galicia Supercomputing Center, CESGA, including access to the FinisTerae supercomputer.

Data Availability Statement

All the data presented in this work has been made publicly available in FigShare (<https://doi.org/10.6084/m9.figshare.20043665>), including i) Electronic Supporting Information, ii) Jupyter notebooks and bash scripts, along with the corresponding results of the end-to-end workflows,

and iii) QCORR benchmarking of termination types in Gaussian geometry optimizations using the “opt freq” standard keywords.

ORCID

Juan V. Alegre-Requena 0000-0002-0769-7168

Shree Sowndarya S. V. 0000-0002-4568-5854

Turki M. Alturaifi 0000-0002-6379-1669

Raúl Pérez-Soto 0000-0002-6237-2155

Robert S. Paton 0000-0002-0104-4166

References

1. Patrascu, M. B.; Pottel, J.; Pinus, S.; Bezanson, M.; Norrby, P.-O.; Moitessier, N., From desktop to benchtop with automated computational workflows for computer-aided design in asymmetric catalysis. *Nat. Catal.* **2020**, *3*, 574-584.
2. Sanchez-Lengeling, B.; Aspuru-Guzik, A., Inverse molecular design using machine learning: Generative models for matter engineering. *Science* **2018**, *361*, 360-365.
3. Ahn, S.; Hong, M.; Sundararajan, M.; Ess, D. H.; Baik, M. H., Design and Optimization of Catalysts Based on Mechanistic Insights Derived from Quantum Chemical Reaction Modeling. *Chem. Rev.* **2019**, *119*, 6509-6560.
4. Durand, D. J.; Fey, N., Computational Ligand Descriptors for Catalyst Design. *Chem. Rev.* **2019**, *119*, 6561-6594.
5. Freeze, J. G.; Kelly, H. R.; Batista, V. S., Search for Catalysts by Inverse Design: Artificial Intelligence, Mountain Climbers, and Alchemists. *Chem. Rev.* **2019**, *119*, 6595-6612.
6. Welborn, V. V.; Head-Gordon, T., Computational Design of Synthetic Enzymes. *Chem. Rev.* **2019**, *119*, 6613-6630.
7. Wagner, J. R.; Lee, C. T.; Durrant, J. D.; Malmstrom, R. D.; Feher, V. A.; Amaro, R. E., Emerging Computational Methods for the Rational Discovery of Allosteric Drugs. *Chem. Rev.* **2016**, *116*, 6370-90.
8. Colón, Y. J.; Snurr, R. Q., High-throughput computational screening of metal–organic frameworks. *Chem. Soc. Rev.* **2014**, *43*, 5735-5749.

9. Gómez-Bombarelli, R.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Duvenaud, D.; Maclaurin, D.; Blood-Forsythe, M. A.; Chae, H. S.; Einzinger, M.; Ha, D. G.; Wu, T.; Markopoulos, G.; Jeon, S.; Kang, H.; Miyazaki, H.; Numata, M.; Kim, S.; Huang, W.; Hong, S. I.; Baldo, M.; Adams, R. P.; Aspuru-Guzik, A., Design of efficient molecular organic light-emitting diodes by a high-throughput virtual screening and experimental approach. *Nat. Mater.* **2016**, *15*, 1120-7.
10. Dunstan, M. T.; Jain, A.; Liu, W.; Ong, S. P.; Liu, T.; Lee, J.; Persson, K. A.; Scott, S. A.; Dennis, J. S.; Grey, C. P., Large scale computational screening and experimental discovery of novel materials for high temperature CO₂ capture. *Energy Environ. Sci.* **2016**, *9*, 1346-1360.
11. Chakraborty, S.; Xie, W.; Mathews, N.; Sherburne, M.; Ahuja, R.; Asta, M.; Mhaisalkar, S. G., Rational Design: A High-Throughput Computational Screening and Experimental Validation Methodology for Lead-Free and Emergent Hybrid Perovskites. *ACS Energy Lett.* **2017**, *2*, 837-845.
12. Robert, J. H.; Bentzien, J.; Marie, L. A.; Marian, Y. H.; Jonathan, M. J.; Vielmetter, J.; Kundu, A.; Bassil, I. D., Combining computational and experimental screening for rapid optimization of protein properties. *Proc. Natl. Acad. Sci.* **2002**, *99*, 15926-15931.
13. Sokolov, A. N.; Atahan-Evrenk, S.; Mondal, R.; Akkerman, H. B.; Sánchez-Carrera, R. S.; Granados-Focil, S.; Schrier, J.; Mannsfeld, S. C.; Zoombelt, A. P.; Bao, Z.; Aspuru-Guzik, A., From computational discovery to experimental characterization of a high hole mobility organic crystal. *Nat. Commun.* **2011**, *2*, 437.
14. Ingman, V. M.; Schaefer, A. J.; Andreola, L. R.; Wheeler, S. E., QChASM: Quantum chemistry automation and structure manipulation. *WIREs Comput. Mol. Sci.* **2021**, *11*, e1510.
15. Young, T. A.; Silcock, J. J.; Sterling, A. J.; Duarte, F., autodE: Automated Calculation of Reaction Energy Profiles— Application to Organic and Organometallic Reactions. *Angew. Chem. Int. Ed.* **2021**, *60*, 4266-4274.
16. Guan, Y.; Ingman, V. M.; Rooks, B. J.; Wheeler, S. E., AARON: An Automated Reaction Optimizer for New Catalysts. *J Chem Theory Comput* **2018**, *14*, 5249-5261.
17. Corbeil, C. R.; Thielges, S.; Schwartzentruber, J. A.; Moitessier, N., Toward a Computational Tool Predicting the Stereochemical Outcome of Asymmetric Reactions: Development and Application of a Rapid and Accurate Program Based on Organic Principles. *Angew. Chem. Int. Ed.* **2008**, *47*, 2635-2638.

18. Pizzi, G.; Cepellotti, A.; Sabatini, R.; Marzari, N.; Kozinsky, B., AiiDA: automated interactive infrastructure and database for computational science. *Comput. Mat. Sci.* **2016**, *111*, 218-230.
19. Żurański, A. M.; Wang, J. Y.; Shields, B. J.; Doyle, A. G., Auto-QChem: an automated workflow for the generation and storage of DFT calculations for organic molecules. *React. Chem. Eng.* **2022**, *7*, 1276-1284.
20. Rosales, A. R.; Wahlers, J.; Limé, E.; Meadows, R. E.; Leslie, K. W.; Savin, R.; Bell, F.; Hansen, E.; Helquist, P.; Munday, R. H.; Wiest, O.; Norrby, P.-O., Rapid virtual screening of enantioselective catalysts using CatVS. *Nat. Catal.* **2019**, *2*, 41-45.
21. Tech-X Corporation, Chemistream.
<https://txcorp.com/images/docs/chemistream/latest/index.html>.
22. Metz, S.; Kästner, J.; Sokol, A. A.; Keal, T. W.; Sherwood, P., ChemShell—a modular software package for QM/MM simulations. *WIREs Comput. Mol. Sci.* **2014**, *4*, 101-110.
23. Jain, A.; Ong, S. P.; Chen, W.; Medasani, B.; Qu, X.; Kocher, M.; Brafman, M.; Petretto, G.; Rignanese, G.-M.; Hautier, G.; Gunter, D.; Persson, K. A., FireWorks: a dynamic workflow system designed for high-throughput applications. *Concur. Comput. Pract. Exper.* **2015**, *27*, 5037-5059.
24. Ioannidis, E. I.; Gani, T. Z.; Kulik, H. J., molSimplify: A toolkit for automating discovery in inorganic chemistry. *J. Comput. Chem.* **2016**, *37*, 2106-17.
25. Jacob, C. R.; Beyhan, S. M.; Buló, R. E.; Gomes, A. S. P.; Götz, A. W.; Kiewisch, K.; Sikkema, J.; Visscher, L., PyADF — A scripting framework for multiscale quantum chemistry. *J. Comput. Chem.* **2011**, *32*, 2328-2338.
26. Zapata, F.; Ridder, L.; Hidding, J.; Jacob, C. R.; Infante, I.; Visscher, L., QMflows: A Tool Kit for Interoperable Parallel Workflows in Quantum Chemistry. *J. Chem. Inf. Model* **2019**, *59*, 3191-3197.
27. Van Rossum, G.; Drake, F. L., *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.
28. AQME v1.3, Alegre-Requena, J. V.; Sowndarya, S.; Pérez-Soto, R.; Alturaifi, T. M.; Paton, R. S., **2022**. <https://github.com/jvalegre/aqme>.
29. The Alegre Group YouTube channel.
<https://www.youtube.com/channel/UCHRqI8N61bYxWV9BjbUI4Xw>.
30. RDKit: Open-source cheminformatics. <http://www.rdkit.org>.

31. Grimme, S., Exploration of Chemical Compound, Conformer, and Reaction Space with Meta-Dynamics Simulations Based on Tight-Binding Quantum Chemical Calculations. *J. Chem. Theory Comput.* **2019**, *15*, 2847-2862.
32. Grimme, S.; Bannwarth, C.; Shushkov, P., A Robust and Accurate Tight-Binding Quantum Chemical Method for Structures, Vibrational Frequencies, and Noncovalent Interactions of Large Molecular Systems Parametrized for All spd-Block Elements ($Z = 1-86$). *J. Chem. Theory Comput.* **2017**, *13*, 1989-2009.
33. Smith, J. S.; Isayev, O.; Roitberg, A. E., ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chem. Sci.* **2017**, *8*, 3192-3203.
34. Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Petersson, G. A.; Nakatsuji, H.; Li, X.; Caricato, M.; Marenich, A. V.; Bloino, J.; Janesko, B. G.; Gomperts, R.; Mennucci, B.; Hratchian, H. P.; Ortiz, J. V.; Izmaylov, A. F.; Sonnenberg, J. L.; Williams; Ding, F.; Lipparini, F.; Egidi, F.; Goings, J.; Peng, B.; Petrone, A.; Henderson, T.; Ranasinghe, D.; Zakrzewski, V. G.; Gao, J.; Rega, N.; Zheng, G.; Liang, W.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Vreven, T.; Throssell, K.; Montgomery Jr., J. A.; Peralta, J. E.; Ogliaro, F.; Bearpark, M. J.; Heyd, J. J.; Brothers, E. N.; Kudin, K. N.; Staroverov, V. N.; Keith, T. A.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, A. P.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.; Cossi, M.; Millam, J. M.; Klene, M.; Adamo, C.; Cammi, R.; Ochterski, J. W.; Martin, R. L.; Morokuma, K.; Farkas, O.; Foresman, J. B.; Fox, D. J. *Gaussian 16 Rev. C.01*, Wallingford, CT, 2016.
35. Neese, F., The ORCA program system. *WIREs Comput. Mol. Sci.* **2012**, *2*, 73-78.
36. Sun, Q.; Berkelbach, T. C.; Blunt, N. S.; Booth, G. H.; Guo, S.; Li, Z.; Liu, J.; McClain, J. D.; Sayfutyarova, E. R.; Sharma, S.; Wouters, S.; Chan, G. K.-L., PySCF: the Python-based simulations of chemistry framework. *WIREs Comput. Mol. Sci.* **2018**, *8*, e1340.
37. O'Boyle, N. M.; Tenderholt, A. L.; Langner, K. M., cclib: A library for package-independent computational chemistry algorithms. *J. Comput. Chem.* **2008**, *29*, 839-845.
38. Watts, K. S.; Dalal, P.; Murphy, R. B.; Sherman, W.; Friesner, R. A.; Shelley, J. C., ConfGen: a conformational search method for efficient generation of bioactive conformers. *J. Chem. Inf. Model.* **2010**, *50*, 534-546.

39. Hawkins, P. C. D.; Skillman, A. G.; Warren, G. L.; Ellingson, B. A.; Stahl, M. T., Conformer Generation with OMEGA: Algorithm and Validation Using High Quality Structures from the Protein Databank and Cambridge Structural Database. *J. Chem. Inf. Model.* **2010**, *50*, 572-584.
40. Miteva, M. A.; Guyon, F.; Tufféry, P., Frog2: Efficient 3D conformation ensemble generator for small compounds. *Nucleic Acids Res.* **2010**, *38*, W622-627.
41. Goodman, J. M.; Still, W. C., An unbounded systematic search of conformational space. *J. Comput. Chem.* **1991**, *12*, 1110-1117.
42. Chang, G.; Guida, W. C.; Still, W. C., An internal-coordinate Monte Carlo method for searching conformational space. *J. Am. Chem. Soc.* **1989**, *111*, 4379-4386.
43. FullMonte, Paton, R. S. <https://github.com/bobbypaton/FullMonte>.
44. Alegre-Requena, J. V.; Sowndarya, S.; Paton, R. S., AQME paper examples. *Figshare Dataset* **2022**. <https://doi.org/10.6084/m9.figshare.20043665>.
45. Young, D., Computational Chemistry: A Practical Guide for Applying Techniques to Real World Problems. *Wiley* **2001**, 228.
46. Lodewyk, M. W.; Siebert, M. R.; Tantillo, D. J., Computational Prediction of ¹H and ¹³C Chemical Shifts: A Useful Tool for Natural Product, Mechanistic, and Synthetic Organic Chemistry. *Chem. Rev.* **2012**, *112*, 1839-1862.
47. Bonjoch, J.; Solé, D., Synthesis of Strychnine. *Chem. Rev.* **2000**, *100*, 3455-3482.
48. Schmidt, M.; Reinscheid, F.; Sun, H.; Abromeit, H.; Scriba, G. K. E.; Sönnichsen, F. D.; John, M.; Reinscheid, U. M., Hidden Flexibility of Strychnine. *Eur. J. Org. Chem.* **2014**, *2014*, 1147-1150.
49. Luchini, G.; Alegre-Requena, J. V.; Funes-Ardoiz, I.; Paton R.S., GoodVibes: automated thermochemistry for heterogeneous computational chemistry data. *F1000Research* **2020**, *9*, 291.
50. SDBSWeb (National Institute of Advanced Industrial Science and Technology, date of access). Compound name: Strychnine, solvent: CDCl₃ SDBS No.: 7596. <https://sdb.db.aist.go.jp/sdb/cgi-bin/landingpage?sdbno=7596>
51. Liu, F.; Paton, R. S.; Kim, S.; Liang, Y.; Houk, K. N., Diels–Alder Reactivities of Strained and Unstrained Cycloalkenes with Normal and Inverse-Electron-Demand Dienes: Activation Barriers and Distortion/Interaction Analysis. *J. Am. Chem. Soc.* **2013**, *135*, 15642-15649.

52. Gallegos, L. C.; Luchini, G.; St. John, P. C.; Kim, S.; Paton, R. S., Importance of Engineered and Learned Molecular Representations in Predicting Organic Reactivity, Selectivity, and Chemical Properties. *Acc. Chem. Res.* **2021**, *54*, 827-836.
53. Stuyver, T.; Coley, C. W., Quantum chemistry-augmented neural networks for reactivity prediction: Performance, generalizability, and explainability. *J. Chem. Phys.* **2022**, *156*, 084104.
54. Delaney, J. S., ESOL: Estimating Aqueous Solubility Directly from Molecular Structure. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1000-1005.
55. St. John, P. C.; Ward, L.; Sowndarya, S., **2022**). NREL/nfp: soft rdkit import (0.3.10). Zenodo. <https://doi.org/10.5281/zenodo.6475665>.