Received: 13 December 2022 | Revised: 31 January 2023 | Accepted: 2 February 2023

DOI: 10.1002/wcms.1663

SOFTWARE FOCUS

WIREs COMPUTATIONAL MOLECULAR SCIENCE WILEY

AQME: Automated quantum mechanical environments for researchers and educators

Juan V. Alegre-Requena[1] | Shree Sowndarya S. V.[2] | Raúl Pérez-Soto[2] | Turki M. Alturaifi[2] | Robert S. Paton[2]

[1]Dpto. de Química Inorgánica, Instituto de Síntesis Química y Catálisis Homogénea (ISQCH) CSIC-Universidad de Zaragoza, Zaragoza, Spain

[2]Department of Chemistry, Colorado State University, Fort Collins, Colorado, USA

Correspondence
Juan V. Alegre-Requena, Dpto. de Química Inorgánica, Instituto de Síntesis Química y Catálisis Homogénea (ISQCH) CSIC-Universidad de Zaragoza, Zaragoza 50009, Spain.
Email: jv.alegre@csic.es

Robert S. Paton, Department of Chemistry, Colorado State University, Fort Collins, CO 80523, USA.
Email: robert.paton@colostate.edu

Funding information
CESGA; European Union (NextGenerationEU/PRTR), Grant/Award Number: IJC2020-044217-I; Ministerio de Ciencia e Innovación, Grant/Award Number: IJC2020-044217-I; National Science Foundation, Grant/Award Numbers: ACI-1532235, ACI-1532236, CHE-1955876; XSEDE, Grant/Award Number: TG-CHE180056; Gobierno de Aragón-Fondo Social Europeo, Grant/Award Number: E07_20R

Edited by: Peter R. Schreiner, Editor-in-Chief

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 all or only the desired parts of the program. The code has been developed 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. 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 have 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.

Juan V. Alegre-Requena and Shree Sowndarya S. V. contributed equally.

This article is categorized under:
Data Science > Chemoinformatics
Data Science > Computer Algorithms and Programming
Software > Quantum Chemistry

KEYWORDS
automation, cheminformatics, machine learning, quantum mechanics, workflows

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 (ML) 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. For example, 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., molecular mechanics, MM, quantum mechanics, QM, semi-empirical QM, SQM) executed by a distinct package. Further, these efforts are multiplied by the number of molecules. Automating these multistep 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 QChASM[14] and Duarte's autodE[15]), have emerged as powerful software tools, such as AARON,[16] ACE,[17] Aiida,[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 an automated end-to-end workflow software, AQME, to perform multistep 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 other Python projects.[27] Ready-to-use examples with different degrees of complexity are provided via GitHub,[28] supplemented by hands-on video tutorials.[29] These examples can be trivially modified to create workflows that implement different software and levels of theory, or for application to different prediction tasks. For example, a researcher can create a workflow to calculate a reaction energy profile and, afterward, tune the module combination to generate QM molecular descriptors to use in machine learning 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 module is interfaced with molecular mechanics potentials through RDKit[30] and semi-empirical potentials through xTB. Searches can be performed externally using RDKit or CREST,[31] or using internally-coded systematic or Monte Carlo torsion sampling protocols. Then, CMIN refines these geometries and relative energies obtained from the initial conformer generation with semi-empirical methods (xTB)[32] or ML potentials (ANI).[33] However, this module can also be used to independently process 3D input formats. 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 (in Gaussian) creating Gen(ECP) sections or including final lines (i.e., NBO extra keywords) in the input files are handled automatically. QCORR is a cclib-based[37] 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 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.

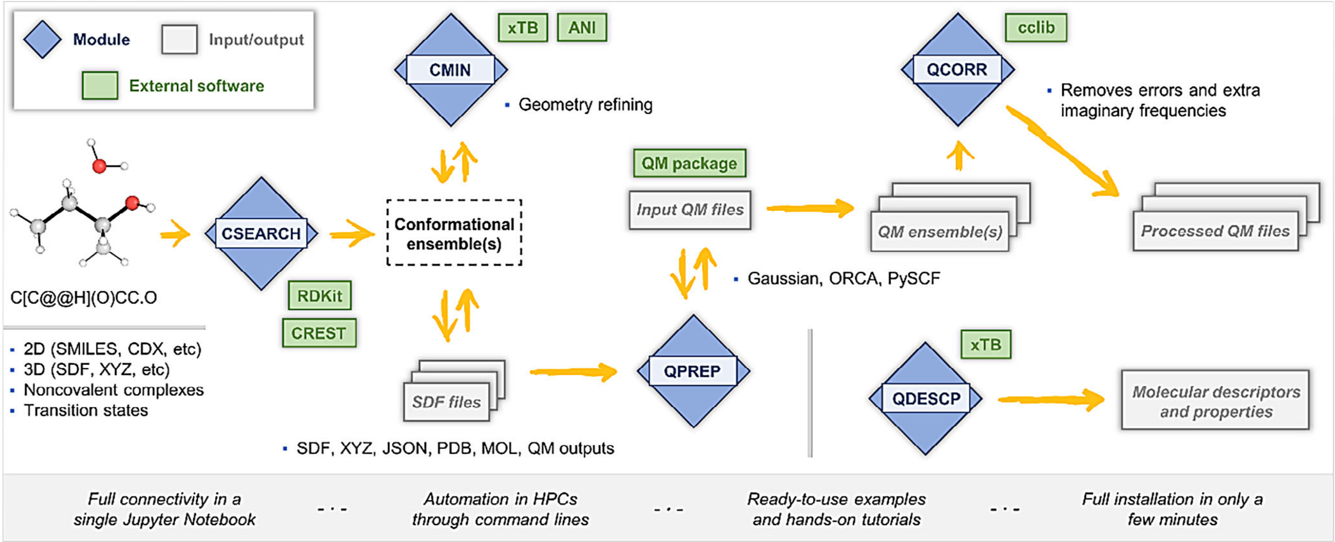

Module
Input/output
External software
C[C@@H](O)CC.O
2D (SMILES, CDX, etc)
3D (SDF, XYZ, etc)
Noncovalent complexes
Transition states
CSEARCH
RDKit
CREST
CMIN
xTB
ANI
Geometry refining
Conformational ensemble(s)
SDF files
SDF, XYZ, JSON, PDB, MOL, QM outputs
QM package
Input QM files
Gaussian, ORCA, PySCF
QPREP
QM ensemble(s)
QCORR
cclib
Removes errors and extra imaginary frequencies
Processed QM files
QDESCP
xTB
Molecular descriptors and properties
Full connectivity in a single Jupyter Notebook
Automation in HPCs through command lines
Ready-to-use examples and hands-on tutorials
Full installation in only a few minutes

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

2 | INSTALLATION AND TECHNICAL DETAILS

The program presented is a free, open-source Python-based software 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. A detailed documentation page is available at Read the Docs.[29]

3 | 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) postprocessing 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.

3.1 | 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.

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

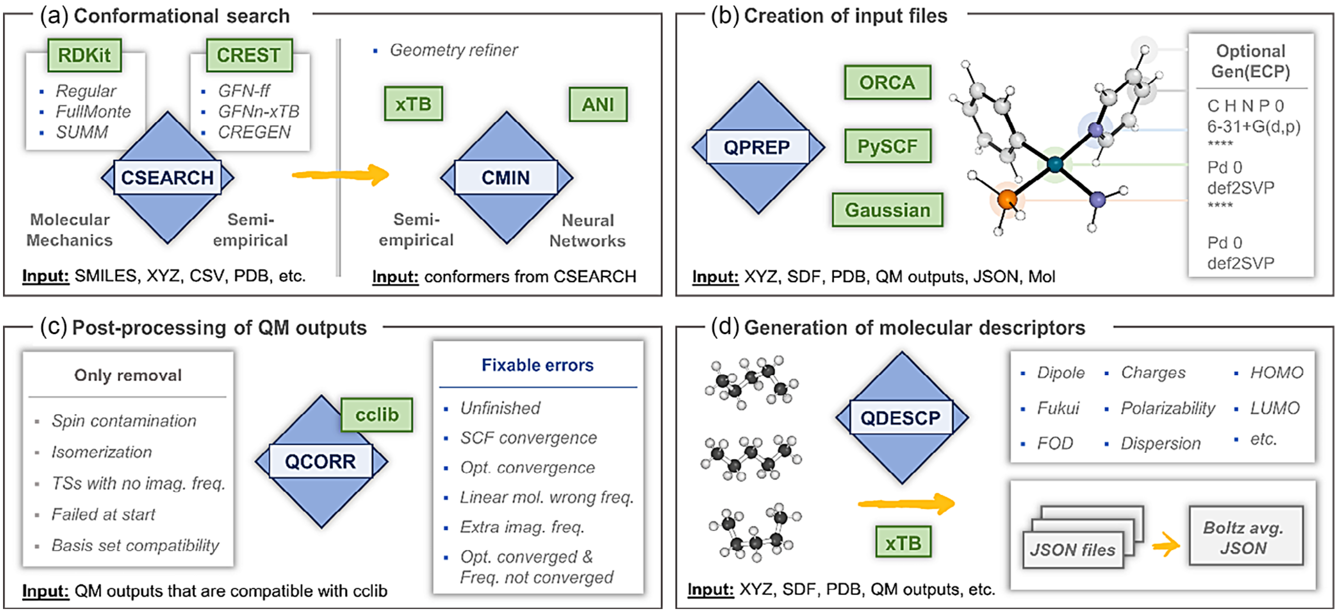


FIGURE 2 Modules in AQME. (a) Methods and parameter options in CSEARCH and CMIN modules. (b) Recognition of atoms to include in the GenECP section of an example organometallic complex when using gen_atoms = ["Pd"]. (c) Outline of error fixed in the QCORR module. Additional creation of input files for 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.

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 (using program="rdkit") performs RDKit-based conformer optimization and filters duplicates based on energy and geometry (with root mean square distance, RMSD). It starts with an energy window to remove high energy conformers (*ewin_csearch* option, default 5 kcal/mol above lowest), then removes conformers with similar energies (*initial_energy_threshold* option, default 0.0001 kcal/mol), and finally removes conformers with similar energy and RMSD (*energy_threshold*, default 0.25 kcal/mol and *rms_threshold*, default 0.25). The default values were chosen based on a benchmark study of flexible druglike compounds and natural products to yield significant conformers while reducing duplicates (see the *Benchmarking of CSEARCH-RDKit and CMIN* section in the AQME_ESI.docx document available in FigShare[41]).

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 (see the *Highlighting the Importance of Specifying the Metal Type: ABEVUZ as an Example* section in the ESI[41]).

Internal torsional sampling approaches, such as the systematic unbounded multiple minimum (SUMM)[42] and Monte Carlo Multiple Minimum (MCMM) algorithms, are also implemented.[43,44] 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 methods typically result in a reordering of relative conformational stabilities closer to QM results and the removal of duplicate conformations.

3.2 | 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. When using SDF and XYZ files, a QM input is generated for each structure in the files. For LOG/OUT calculations, only the final geometry is employed to create inputs for post-optimization single-point calculations (i.e., energy corrections at a higher theory level, TD-DFT calculations, etc.). 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).

3.3 | Postprocessing 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. 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).

We conducted a study of 2709 calculations with organic molecules to find the optimal number of QCORR cycles for ground state optimization and frequency calculations in Gaussian 16 (QCORR_benchmarking.zip file in FigShare[41]). The initial geometries were obtained with a CSEARCH-RDKit standard conformer sampling. In the first round, 24% of the RDKit-generated conformers converged to duplicated QM conformers after optimization. From the unique structures, many outputs had problems: 1.5% showed imaginary frequencies, and 35% failed to converge to a stationary point in frequency calculation (albeit they converged during optimization). QCORR then automatically generated new inputs to fix the issues and these inputs were run. After the second round of QM calculations, 98% of the outputs had no issues, indicating that two QCORR rounds might be sufficient for most organic molecules. In our experience, an additional cycle is normally needed for complex systems with metals or supramolecular aggregates. The *freq_conv* keyword, which checks if a stationary point is found during optimization but not during frequency calculations, may be best avoided for flat or complex energy surfaces.

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.

3.4 | Generation of Boltzmann weighted properties and descriptors: The QDESCP module

When comparing computed results with experimental observables, Boltzmann-weighted values are generally advisable for molecules with multiple conformers. This also applies to molecular descriptors, where the utility of approaches that derive an ensemble average for a particular descriptor, or directly use minimum/maximum values, has been demonstrated.[46,47] AQME is integrated with xTB to compute and curate computed descriptors for large compound 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, the Boltzmann averaged values of each descriptor are calculated and stored separately. This protocol enables the generation of SQM molecular descriptors as starting points for ML models. Additionally, QDESCP can be used to obtain 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,[48] rendering simulated spectra that can be compared directly with experimental spectra.

4 | END-TO-END WORKFLOWS

In this section, we detail three 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 Figshare[41] along with associated data and metadata, and three formats are available in the code and the Read the Docs webpage: 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.

4.1 | The conformational distribution and ^{1}H 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.[49] Recently, John, Reinscheid, and coworkers reported two different conformers in a 97:3 ratio observed in NMR studies.[50] 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) GoodVibes[51] 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 = $CHCl_3$ (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 ^{1}H chemical shifts present a low mean average error (MAE, 0.14 ppm for nine known ^{1}H signals) compared to the experimental values.[52] This workflow did not require manual intervention and suggests that further applicability of AQME to automate NMR prediction and organic structure elucidation merits investigation.

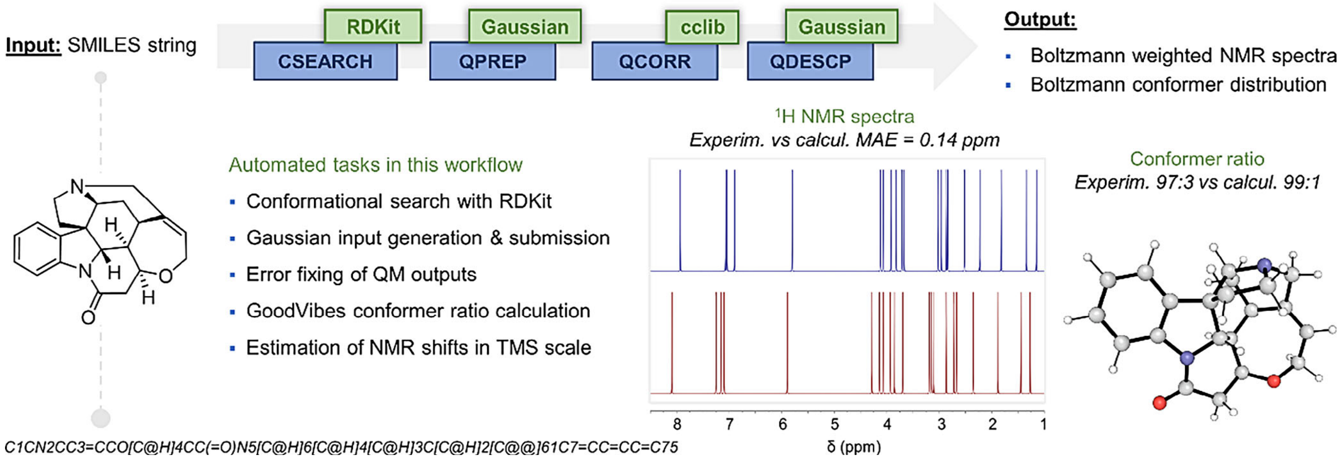

Input: SMILES string
RDKit
CSEARCH
Gaussian
QPREP
cclib
QCORR
Gaussian
QDESCP
Output:
▪ Boltzmann weighted NMR spectra
▪ Boltzmann conformer distribution
Automated tasks in this workflow
▪ Conformational search with RDKit
▪ Gaussian input generation & submission
▪ Error fixing of QM outputs
▪ GoodVibes conformer ratio calculation
▪ Estimation of NMR shifts in TMS scale
^{1}H NMR spectra
Experim. vs calcul. MAE = 0.14 ppm
8
7
6
5
4
3
2
1
δ (ppm)
Conformer ratio
Experim. 97:3 vs calcul. 99:1
N
H
N
H
O
H
O
C1CN2CC3=CCO[C@H]4CC(=O)N5[C@H]6[C@H]4[C@H]3C[C@H]2[C@@]61C7=CC=CC=C75

FIGURE 3 End-to-end workflow to calculate the conformer distribution and scaled (TMS) ^{1}H NMR chemical shifts of strychnine in $CHCl_3$.

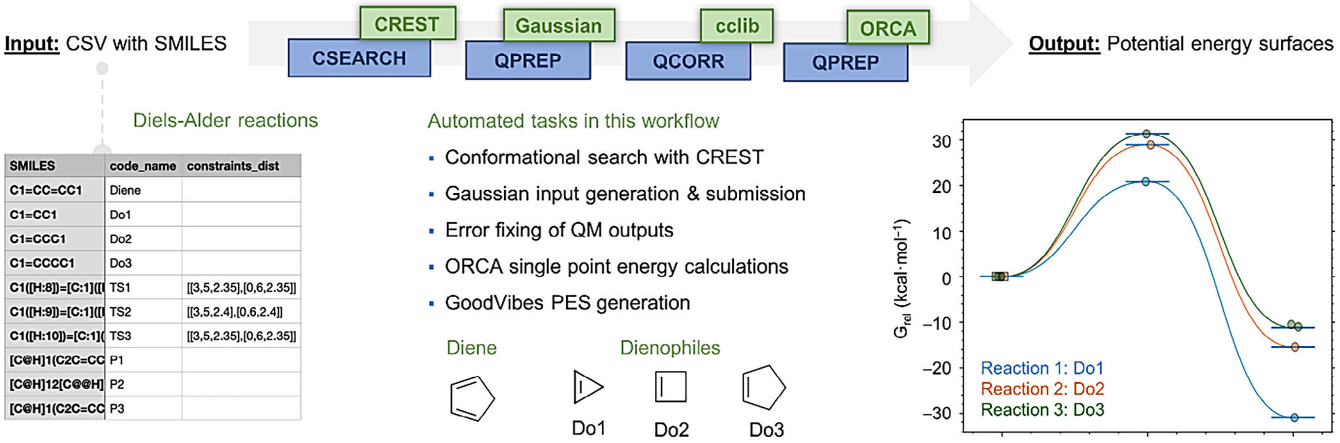
4.2 | Comparing Diels-Alder activation barriers from multiple SMILES inputs

A common computational task involves comparing the reactivity of several different substrates or reagents, for which the same elementary steps are studied separately for each of the related systems. Where transition states are involved, a common approach to conformational analysis involves constraining forming/breaking bonds while flexible regions are explored in much the same way as for a ground state structure, followed by saddle point optimizations. AQME can be employed to generate and compare energy profiles by automating this sequence of steps that is often performed manually. 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.[53] A Jupyter notebook was used to define SMILES strings and to identify the relevant atom numbers to define constraints that are used for the conformational analysis of TSs. Performed manually, these tasks would typically each structure to be built and visualized by the user. This approach is limited to reactions where the TS structures are intuitively known; for automated TS location and energy profile generation without requiring prior knowledge, and mechanistic discovery, tools such as autodE are highly recommended.[15]

The workflow presented illustrates a typical multistep 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)/def2-TZVPP, and (v) Boltzmann weighted thermochemistry calculation and PES generation with GoodVibes at 298.15 K. There is minimal manual intervention and the use of separate spreadsheets to create the PES is avoided.

4.3 | Generating QM or SQM molecular descriptors for a large dataset

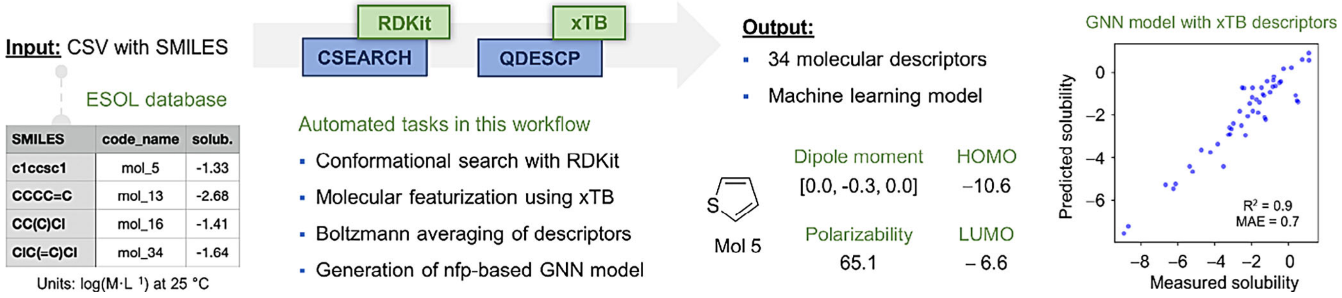
Statistical and ML applications in chemistry are often enhanced by using feature vectors or parameters derived from QM calculations, as opposed to features obtained solely from the 2D-molecular graph.[54] For example, QM-derived atomic charges or populations are often used in multivariate linear regression or neural network models.[55] Figure 5 shows a workflow performed on the SMILES-containing ESOL database,[56] which contains measured aqueous solubilities, that uses a message passing graph neural network (GNN) model. There are 1126 structures with experimental values available, which are split into training (901), validation (175) and test (50) sets. The protocol includes (i) RDKit conformer sampling, (ii) xTB descriptor generation (Boltzmann weighted), and (iii) neural fingerprint (nfp)[57] based 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



SMILES	code_name	constraints_dist
C1=CC=CC1	Diene	
C1=CC1	Do1	
C1=CCC1	Do2	
C1=CCCC1	Do3	
C1([H:8])=[C:1]([[	TS1	[[3,5,2.35],[0,6,2.35]]
C1([H:9])=[C:1]([[	TS2	[[3,5,2.4],[0,6,2.4]]
C1([H:10])=[C:1](	TS3	[[3,5,2.35],[0,6,2.35]]
[C@H]1(C2C=CC	P1	
[C@H]12[C@@H]	P2	
[C@H]1(C2C=CC	P3	

- Conformational search with CREST
- Gaussian input generation & submission
- Error fixing of QM outputs
- ORCA single point energy calculations
- GoodVibes PES generation

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



SMILES	code_name	solub.
c1ccsc1	mol_5	-1.33
CCCC=C	mol_13	-2.68
CC(C)Cl	mol_16	-1.41
ClC(=C)Cl	mol_34	-1.64

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

Boltzmann averaged values. In addition to properties generated from xTB, features in the Lipinski/Descriptors modules of RDKit are also included in the QDESCP analysis.

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 0.9 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.

5 | CONCLUSION

AQME is an open-source Python package for building computational workflows to perform multistep 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 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 example, 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); data curation (lead); investigation (lead); methodology (lead); writing – original draft (lead); writing – review and editing (lead). **Shree Sowndarya S. V.:** Conceptualization (lead); data curation (lead); investigation (lead); methodology (lead); writing – original draft (lead); writing – review and editing (lead). **Raúl Pérez-Soto:** Methodology (supporting); writing – review and editing (supporting). **Turki M. Alturaifi:** Methodology (supporting); writing – review and editing (supporting). **Robert S. Paton:** Conceptualization (supporting); funding acquisition (lead); resources (lead); supervision (lead); writing – review and editing (supporting).

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), the State Research Agency (AEI) of Spain and the European Union (NextGenerationEU/PRTR) under grant reference IJC2020-044217-I, and the Gobierno de Aragón-Fondo Social Europeo (Research Group E07_20R). 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 FinisTerrae supercomputer.

CONFLICT OF INTEREST STATEMENT

The authors have declared no conflicts of interest for this article.

OPEN RESEARCH BADGES

This article has earned an Open Data badge for making publicly available the digitally-shareable data necessary to reproduce the reported results. The data is available at https://figshare.com/articles/dataset/AQME_paper_examples/20043665.

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 https://orcid.org/0000-0002-0769-7168
Shree Sowndarya S. V. https://orcid.org/0000-0002-4568-5854
Raúl Pérez-Soto https://orcid.org/0000-0002-6237-2155
Turki M. Alturaifi https://orcid.org/0000-0002-6379-1669
Robert S. Paton https://orcid.org/0000-0002-0104-4166

RELATED WIREs ARTICLES

ChemML: A machine learning and informatics program package for the analysis, mining, and modeling of chemical and materials data
The MolSSI QCArchive project: An open-source platform to compute, organize, and share quantum chemistry data
QChASM: Quantum chemistry automation and structure manipulation
WebMO: Web-based computational chemistry calculations in education and research
Free and open source software for computational chemistry education

REFERENCES

1. Patrascu MB, Pottel J, Pinus S, Bezanson M, Norrby PO, Moitessier N. From desktop to benchtop with automated computational workflows for computer-aided design in asymmetric catalysis. Nat Catal. 2020;3:574–84.
2. Sanchez-Lengeling B, Aspuru-Guzik A. Inverse molecular design using machine learning: generative models for matter engineering. Science. 2018;361:360–5.
3. Ahn S, Hong M, Sundararajan M, Ess DH, Baik MH. Design and optimization of catalysts based on mechanistic insights derived from quantum chemical reaction modeling. Chem Rev. 2019;119:6509–60.
4. Durand DJ, Fey N. Computational ligand descriptors for catalyst design. Chem Rev. 2019;119:6561–94.
5. Freeze JG, Kelly HR, Batista VS. Search for catalysts by inverse design: artificial intelligence, mountain climbers, and alchemists. Chem Rev. 2019;119:6595–612.
6. Welborn VV, Head-Gordon T. Computational design of synthetic enzymes. Chem Rev. 2019;119:6613–30.

7. Wagner JR, Lee CT, Durrant JD, Malmstrom RD, Feher VA, Amaro RE. Emerging computational methods for the rational discovery of allosteric drugs. Chem Rev. 2016;116:6370–90.
8. Colón YJ, Snurr RQ. High-throughput computational screening of metal–organic frameworks. Chem Soc Rev. 2014;43:5735–49.
9. Gómez-Bombarelli R, Aguilera-Iparraguirre J, Hirzel TD, Duvenaud D, Maclaurin D, Blood-Forsythe MA, et al. 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 MT, Jain A, Liu W, Ong SP, Liu T, Lee J, et al. Large scale computational screening and experimental discovery of novel materials for high temperature CO_2 capture. Energy Environ Sci. 2016;9:1346–60.
11. Chakraborty S, Xie W, Mathews N, Sherburne M, Ahuja R, Asta M, et al. Rational design: a high-throughput computational screening and experimental validation methodology for Lead-free and emergent hybrid perovskites. ACS Energy Lett. 2017;2:837–45.
12. Robert JH, Bentzien J, Marie LA, Marian YH, Jonathan MJ, Vielmetter J, et al. Combining computational and experimental screening for rapid optimization of protein properties. Proc Natl Acad Sci U S A. 2002;99:15926–31.
13. Sokolov AN, Atahan-Evrenk S, Mondal R, Akkerman HB, Sánchez-Carrera RS, Granados-Focil S, et al. From computational discovery to experimental characterization of a high hole mobility organic crystal. Nat Commun. 2011;2:437.
14. Ingman VM, Schaefer AJ, Andreola LR, Wheeler SE. QChASM: quantum chemistry automation and structure manipulation. WIREs Comput Mol Sci. 2021;11:e1510.
15. Young TA, Silcock JJ, Sterling AJ, Duarte F. autodE: automated calculation of reaction energy profiles—application to organic and organometallic reactions. Angew Chem Int Ed. 2021;60:4266–74.
16. Guan Y, Ingman VM, Rooks BJ, Wheeler SE. AARON: an automated reaction optimizer for new catalysts. J Chem Theory Comput. 2018;14:5249–61.
17. Corbeil CR, Thielges S, Schwartzentruber JA, 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;47:2635–8.
18. Pizzi G, Cepellotti A, Sabatini R, Marzari N, Kozinsky B. AiiDA: automated interactive infrastructure and database for computational science. Comput Mater Sci. 2016;111:218–30.
19. Żurański AM, Wang JY, Shields BJ, Doyle AG. Auto-QChem: an automated workflow for the generation and storage of DFT calculations for organic molecules. React Chem Eng. 2022;7:1276–84.
20. Rosales AR, Wahlers J, Limé E, Meadows RE, Leslie KW, Savin R, et al. Rapid virtual screening of enantioselective catalysts using CatVS. Nat Catal. 2019;2:41–5.
21. Tech-X Corporation, Chemistream. Available from: https://txcorp.com/images/docs/chemistream/latest/index.html.
22. Metz S, Kästner J, Sokol AA, Keal TW, Sherwood P. ChemShell—a modular software package for QM/MM simulations. WIREs Comput Mol Sci. 2014;4:101–10.
23. Jain A, Ong SP, Chen W, Medasani B, Qu X, Kocher M, et al. FireWorks: a dynamic workflow system designed for high-throughput applications. Concur Comput Pract Exper. 2015;27:5037–59.
24. Ioannidis EI, Gani TZ, Kulik HJ. molSimplify: a toolkit for automating discovery in inorganic chemistry. J Comput Chem. 2016;37:2106–17.
25. Jacob CR, Beyhan SM, Bulo RE, Gomes ASP, Götz AW, Kiewisch K, et al. PyADF—a scripting framework for multiscale quantum chemistry. J Comput Chem. 2011;32:2328–38.
26. Zapata F, Ridder L, Hidding J, Jacob CR, Infante I, Visscher L. QMflows: a tool kit for interoperable parallel workflows in quantum chemistry. J Chem Inf Model. 2019;59:3191–7.
27. Van Rossum G, Drake FL. Python 3 reference manual. Scotts Valley, CA: CreateSpace; 2009.
28. AQME v1.3 Alegre-Requena JV, Sowndarya S, Pérez-Soto R, Alturaifi TM, Paton RS. 2022. https://github.com/jvalegre/aqme
29. For the the Alegre group YouTube channel: https://www.youtube.com/channel/UCHRqI8N61bYxWV9BjbUI4Xw. For the Read the Docs webpage: https://aqme.readthedocs.io.
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;15:2847–62.
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 JS, Isayev O, Roitberg AE. ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. Chem Sci. 2017;8:3192–203.
34. Frisch MJ, Trucks GW, Schlegel HB, Scuseria GE, Robb MA, Cheeseman JR, et al. Gaussian 16 Rev. C.01. Wallingford, CT: Gaussian, Inc.; 2016.
35. Neese F. The ORCA program system. WIREs Comput Mol Sci. 2012;2:73–8.
36. Sun Q, Berkelbach TC, Blunt NS, Booth GH, Guo S, Li Z, et al. PySCF: the python-based simulations of chemistry framework. WIREs Comput Mol Sci. 2018;8:e1340.
37. O'Boyle NM, Tenderholt AL, Langner KM. Cclib: a library for package-independent computational chemistry algorithms. J Comput Chem. 2008;29:839–45.

38. Watts KS, Dalal P, Murphy RB, Sherman W, Friesner RA, Shelley JC. ConfGen: a conformational search method for efficient generation of bioactive conformers. J Chem Inf Model. 2010;50:534–46.
39. Hawkins PCD, Skillman AG, Warren GL, Ellingson BA, Stahl MT. Conformer generation with OMEGA: algorithm and validation using high quality structures from the protein databank and Cambridge structural database. J Chem Inf Model. 2010;50:572–84.
40. Miteva MA, Guyon F, Tufféry P. Frog2: efficient 3D conformation ensemble generator for small compounds. Nucleic Acids Res. 2010;38: W622–7.
41. Alegre-Requena JV, Sowndarya S, Paton RS. AQME paper examples figshare dataset. 2022. https://doi.org/10.6084/m9.figshare.20043665
42. Goodman JM, Still WC. An unbounded systematic search of conformational space. J Comput Chem. 1991;2:1110–7.
43. Chang G, Guida WC, Still WC. An internal-coordinate Monte Carlo method for searching conformational space. J Am Chem Soc. 1989; 11:4379–86.
44. FullMonte, Paton RS. https://github.com/patonlab/FullMonte.
45. Young D. Computational chemistry: a practical guide for applying techniques to real world problems. Wiley-Interscience, New York; 2001. p. 228.
46. Brethomé AV, Fletcher SP, Paton RS. Conformational effects on physical-organic descriptors: the case of Sterimol steric parameters. ACS Catal. 2019;9:2313–23.
47. Newman-Stonebraker SH, Smith SR, Borowski JE, Peters E, Gensch T, Johnson HC, et al. Univariate classification of phosphine ligation state and reactivity in cross-coupling catalysis. Science. 2021;374:301–8.
48. Lodewyk MW, Siebert MR, Tantillo DJ. Computational prediction of ^{1}H and ^{13}C chemical shifts: a useful tool for natural product, mechanistic, and synthetic organic chemistry. Chem Rev. 2012;112:1839–62.
49. Bonjoch J, Solé D. Synthesis of strychnine. Chem Rev. 2000;100:3455–82.
50. Schmidt M, Reinscheid F, Sun H, Abromeit H, Scriba GKE, Sönnichsen FD, et al. Hidden flexibility of strychnine. Eur J Org Chem. 2014;2014:1147–50.
51. Luchini G, Alegre-Requena JV, Funes-Ardoiz I, Paton RS. GoodVibes: automated thermochemistry for heterogeneous computational chemistry data. F1000Research. 2020;9:291.
52. SDBSWeb (National Institute of Advanced Industrial Science and Technology, date of access). Compound name: strychnine, solvent: $CDCl_3$ SDBS No.: 7596. https://sdbs.db.aist.go.jp/sdbs/cgi-bin/landingpage?sdbsno=7596
53. Liu F, Paton RS, Kim S, Liang Y, Houk KN. 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;135:15642–9.
54. Gallegos LC, Luchini G, St. John PC, Kim S, Paton RS. Importance of engineered and learned molecular representations in predicting organic reactivity, selectivity, and chemical properties. Acc Chem Res. 2021;54:827–36.
55. Stuyver T, Coley CW. Quantum chemistry-augmented neural networks for reactivity prediction: performance, generalizability, and explainability. J Chem Phys. 2022;156:084104.
56. Delaney JS. ESOL: estimating aqueous solubility directly from molecular structure. J Chem Inf Comput Sci. 2004;44:1000–5.
57. St. John PC, Ward L, Sowndarya S. 2022. NREL/nfp: neural fingerprint (0.3.10). Zenodo. https://doi.org/10.5281/zenodo.6475665.

How to cite this article: Alegre-Requena JV, Sowndarya S. V. S, Pérez-Soto R, Alturaifi TM, Paton RS. AQME: Automated quantum mechanical environments for researchers and educators. WIREs Comput Mol Sci. 2023. e1663. https://doi.org/10.1002/wcms.1663