Reaction Chemistry & Engineering

ROYAL SOCIETY OF CHEMISTRY

PERSPECTIVE

View Article Online
View Journal | View Issue

Published on 17 March 2022. Downloaded by Bristol-Myers Squibb Co on 11/8/2023 8:08:09 PM.

Check for updates

Cite this: *React. Chem. Eng.*, 2022, 7, 1276

Auto-QChem: an automated workflow for the generation and storage of DFT calculations for organic molecules†

Andrzej M. Żurański, ‡*[a] Jason Y. Wang, ‡[ab] Benjamin J. Shields [ac] and Abigail G. Doyle *[ab]

Received 27th January 2022,
Accepted 9th March 2022

DOI: 10.1039/d2re00030j

rsc.li/reaction-engineering

This perspective describes Auto-QChem, an automatic, high-throughput and end-to-end DFT calculation workflow that computes chemical descriptors for organic molecules. Tailored toward users without extensive programming experience, Auto-QChem has facilitated more than 38 000 DFT calculations for 17 000 molecules as of January 2022. Starting from string representations of molecules, Auto-QChem automatically (a) generates conformational ensembles, (b) submits and manages DFT calculations on a high-performance computing (HPC) cluster, (c) extracts production-ready features that are suitable for statistical analysis and machine learning model development, and (d) stores resulting calculations in a cloud-hosted and web-accessible database. We describe in detail the design and implementation of Auto-QChem, as well as its current functionalities. We also review three case studies where Auto-QChem was applied to our recent efforts in combining data science approaches in organic chemistry methodology development: (a) the design of a diverse and unbiased aryl bromide substrate scope for a Ni/photoredox catalyzed alkylation reaction, (b) mechanistic studies on the effect of bioxazoline (BiOx) and biimidazoline (BiIm) ligands on enantioselectivity in a Ni/photoredox catalyzed cross-electrophile coupling of epoxides and aryl iodides, (c) the development of a reaction condition optimization framework using Bayesian optimization. In addition, we discuss limitations and future directions of Auto-QChem and similar automated DFT calculation systems.

Introduction

Data-driven synthetic chemistry has witnessed rapid growth in recent years owing to advances in computing power, software, and algorithms, coupled with an increase in data availability from experiment and computation. The recent resurgence of interest in machine learning and other data-driven approaches in organic chemistry has demonstrated their potential as complementary and quantitative approaches for reactivity and selectivity predictions,[1,2] synthesis planning[3] and mechanistic studies.[4] Importantly, the application of machine learning models in organic chemistry requires effective representations of chemical structures.[5] Compared to molecular fingerprints and various learned representations,[6–10] machine learning models trained with chemical descriptors often offer enhanced interpretability. In particular, features derived from density function theory (DFT) calculations are more closely associated with physical and chemical attributes of molecules, thus enabling improved mechanistic understandings. Therefore, these features serve as good candidates for building statistical and machine learning models. However, DFT calculations often require vast computing resources and proficiency in the operation of various software tools, which presents a significant barrier to experimental chemists. These problems are exacerbated by the number of DFT calculations required to featurize datasets that are sufficient for modern machine learning models. An automatic, high-throughput DFT calculation framework has the potential to accelerate the workflow and facilitate the computation of chemical descriptors by non-experts.

Many tools have been developed to automate high-throughput DFT calculations, such as A$_{\mathrm{FLOW}}$,[11] pymatgen,[12] MAST,[13] Atomate,[14] QMflows,[15] Nexus,[16] and AiiDA.[17,18] However, most of these tools are designed to facilitate material science research and are not well-suited for small organic molecules. Downstream applications in machine learning models also require a framework to extract and store

[a] Department of Chemistry, Princeton University, Princeton, NJ 08544, USA. E-mail: zuranski@princeton.edu
[b] Department of Chemistry & Biochemistry, University of California, Los Angeles, Los Angeles, CA 90095, USA. E-mail: agdoyle@chem.ucla.edu
[c] Bristol Myers Squibb, Cambridge, MA 02142, USA

† Electronic supplementary information (ESI) available. See DOI: 10.1039/d2re00030j

‡ Equal contribution.

a large amount of information from DFT calculation results. Databases containing DFT-calculated properties of materials and small molecules[19–22] have also been developed, usually with an underlying high-throughput workflow clearly defined. For example, the open-access VERDE materials database[22] provides numerous calculated photophysical properties of π-conjugated organic molecules. Such databases usually provide exceptional data access through APIs and web interfaces, but end users often do not have direct access to the calculation pipelines. Beyond functionalities, the simplicity and ease of use for non-experts is also an important consideration. The objectives and limitations of current systems prompted us to implement a framework specifically designed for usage requirements of synthetic organic chemists.

A successful and robust high-throughput DFT calculation framework requires several key functionalities: (a) the ability to generate input files with user specifications for selected quantum chemistry software, (b) an interface with high performance computing (HPC) clusters for the submission and retrieval of jobs with error correction mechanisms, and (c) an analysis workflow to automatically extract information from calculation results. More specifically, we are interested in an end-to-end framework that can generate DFT-derived features directly from string representations (such as SMILES[23]) of organic molecules in a high-throughput fashion, as well as provide storage and convenient access to processed data.

With these goals in mind, we developed Auto-QChem, an automated software package that streamlines DFT calculations for organic molecules. Starting from string representations of molecules, Auto-QChem performs initial conformational searches, manages DFT calculations on local HPC cluster, and facilitates cloud data storage and access *via* a web interface.

In this perspective, we first describe the implementation and detailed workflow of Auto-QChem, followed by a technical description of the software architecture. Next, we showcase the applications of Auto-QChem by reviewing three research projects from our group where Auto-QChem has facilitated the calculations of DFT-derived features and downstream model development in organic chemistry. We conclude the paper by discussing some limitations and potential future directions for Auto-QChem and similar automated DFT calculation systems.

Implementation technologies

The Auto-QChem framework is written in Python 3;[24] DFT calculations are performed with Gaussian 16;[25] the database is powered by MongoDB;[26] and the database web interface is written in Python Dash web framework.[27] Both the database and the web interface are hosted on a common Amazon cloud server.[28] The code base is publicly hosted on a GitHub repository (https://github.com/PrincetonUniversity/auto-qchem) together with its functional documentation (https://princetonuniversity.github.io/auto-qchem). The database web interface is publicly available at https://autoqchem.org. The framework is modularized such that all operations can be performed from a single Jupyter notebook.[29] A handful of usage examples are also provided in the GitHub repository.

Computational workflow

The workflow of Auto-QChem (Fig. 1) starts with a set of molecules represented as SMILES strings. Each SMILES string is first converted to a RDKit[30] molecule object. With a user-defined limit on the maximum number of conformers generated, Auto-QChem performs a conformational search for each molecule using one of the following configurable force field methods: (a) a genetic algorithm for stochastic conformer search implemented in OpenBabel,[31] (b) ETKDG distance geometry algorithm[32] implemented in RDKit.

By default, the following calculation workflow is applied: (a) geometry optimization; (b) frequency and thermochemical analysis, including vibrational frequency, molecular volume, natural population analysis (NPA) and nuclear magnetic resonance (NMR) calculations; and (c) a time dependent DFT calculation for vertical excited state transitions. DFT calculation parameters such as functionals, basis sets and solvation models can be specified by the user. For each conformer, an input file with calculation specifications and

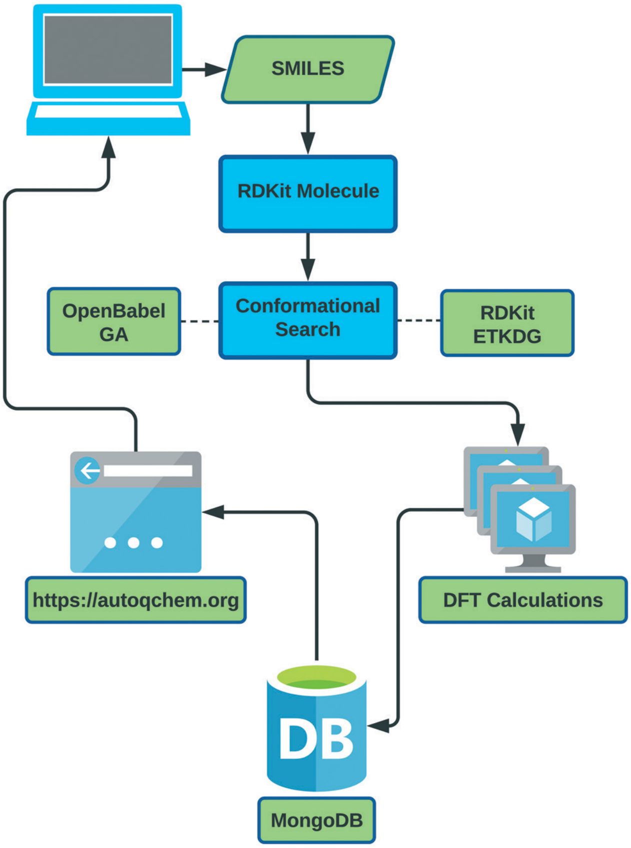


Fig. 1 Computational workflow of Auto-QChem.

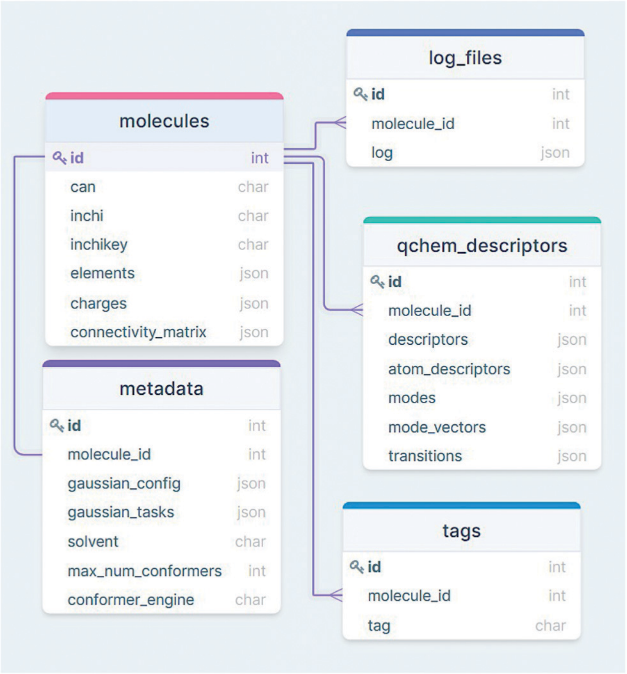
atomic coordinates is generated and submitted to a Slurm scheduler[33] for DFT calculation with Gaussian on a local computer cluster. If a calculation runs out of time or memory, it can be resubmitted with a higher time or resource limit using the last geometry checkpoint. Calculations with unspecified error will be ignored.

Upon successful completion of the DFT calculations, duplicate conformers are removed from the ensemble with a configurable root-mean-square deviation (RMSD) threshold (0.35 Å by default). For each unique conformer, numeric descriptors (Table S1†) are extracted from Gaussian output files. These numeric descriptors and Gaussian output files are then uploaded to the Auto-QChem database.

Database

Data is organized into 5 collections (tables) to support queries and retrieval of the data (Fig. 2):

- **molecules**: master collection that stores information of individual molecules, such as string representations (SMILES, InChI, InChIKey), atomic coordinates, charges, and connectivity matrices.
- **metadata**: one-to-one auxiliary collection that stores the configuration of calculation for each molecule.
- **log_files**: many-to-one collection of raw output files of the calculations (one per conformer).
- **qchem_descriptors**: many-to-one collection of extracted numeric descriptors (one per conformer).
- **tags**: many-to-one collection that stores individual project name tags for easier retrieval and better organization of data.



molecules	
id	int
can	char
inchi	char
inchikey	char
elements	json
charges	json
connectivity_matrix	json

metadata	
id	int
molecule_id	int
gaussian_config	json
gaussian_tasks	json
solvent	char
max_num_conformers	int
conformer_engine	char

log_files	
id	int
molecule_id	int
log	json

qchem_descriptors	
id	int
molecule_id	int
descriptors	json
atom_descriptors	json
modes	json
mode_vectors	json
transitions	json

tags	
id	int
molecule_id	int
tag	char

Fig. 2 Collection schema of Auto-QChem database.

Molecules are indexed such that a particular molecule along with its metadata must be unique, thus disallowing repeated calculations of one molecule with the same calculation configurations. However, calculations of the same molecule with a different configuration (*e.g.*, different solvents, different basis sets) are allowed. Prior to generation of DFT jobs, Auto-QChem warns users if the requested calculation has already been performed and exists in the database.

Queries and data retrieval

Data can be viewed and retrieved from the web interface hosted at https://autoqchem.org. There are two views available:

- **Query view**: a view that allows for web queries of the database and downloads of descriptor sets. The query form contains the following filters: dataset name tags, solvents, functionals, basis sets, SMARTS substructure and SMILES strings.
- **Molecule view**: an interactive display of the structures of all calculated conformers for one molecule, as well as tabulated numeric descriptors (an example is shown in Fig. 3).

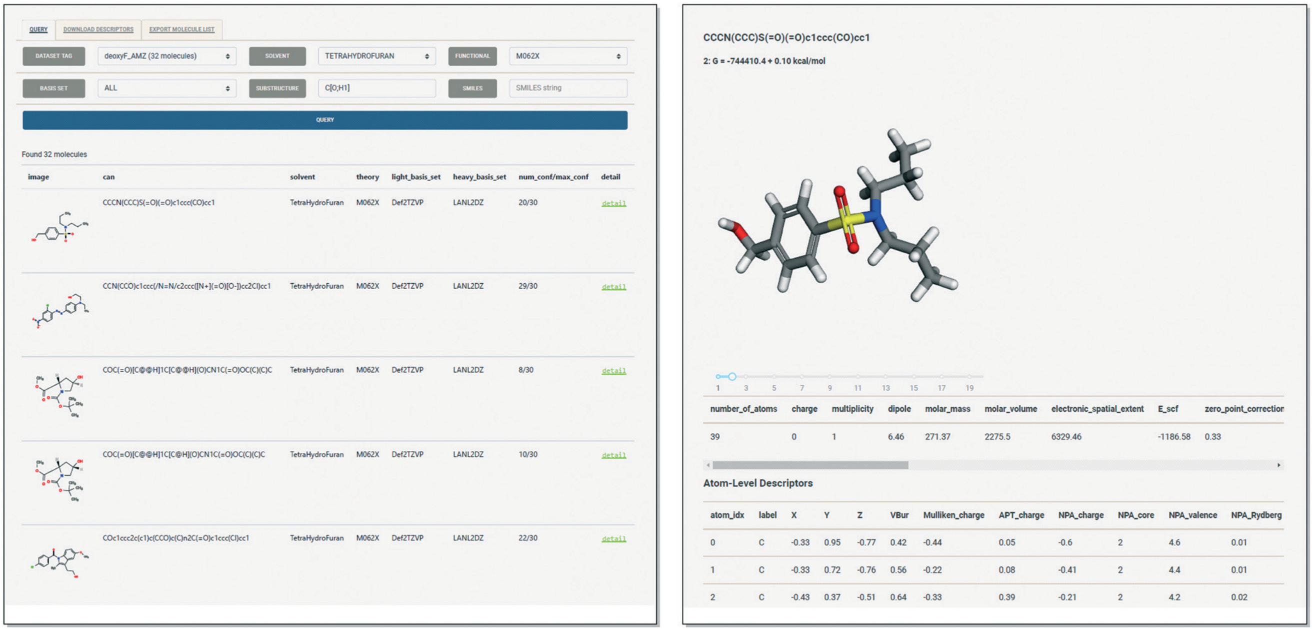
After a successful query, a selection of numeric descriptor sets can be downloaded with the following configurations:

- **Global**: molecular descriptors, such as HOMO/LUMO energy, dipole moment and molecular weight.
- **Substructure atomic**: atomic descriptors from substructure searches. When a substructure is used for the query, atoms from substructure matches are identified in a consistent order and their atomic descriptors (*e.g.*, NMR shifts, partial charges, buried volume) are extracted.
- **Common core atomic**: atomic descriptors for the maximum common substructure within a dataset of molecules. The common core is determined using the FMCS (Find Maximum Common Substructure) algorithm[34] implemented in RDKit.[35]
- **Min max atomic**: minimum and maximum for each atomic descriptor over all atoms.
- **Transitions**: top 10 excited state transitions ordered by oscillation strength.

By default, Boltzmann-weighted average of all conformers is calculated for each numeric descriptor and treated as feature vectors for each molecule. Different weighting options can be specified when exporting descriptors, for example, arithmetic average, lowest energy conformer only, or highest energy conformer only.

Use case 1: substrate scope design in Ni/photoredox methodology development

In a recent example,[36] we developed a Ni/photoredox catalyzed alkylation reaction of aryl halides using acetals as alcohol-derived aliphatic radical sources.[37] To evaluate the



DATASET TAG: deoxyF_AMZ (32 molecules) | SOLVENT: TETRAHYDROFURAN | FUNCTIONAL: M062X

BASIS SET: ALL | SUBSTRUCTURE: C[O;H1] | SMILES: SMILES string

Found 32 molecules

image	can	solvent	theory	light_basis_set	heavy_basis_set	num_conf/max_conf	detail
	CCCN(CCC)S(=O)(=O)c1ccc(CO)cc1	TetraHydroFuran	M062X	Def2TZVP	LANL2DZ	20/30	detail
	CCN(CCO)c1ccc(/N=N/c2ccc([N+](=O)[O-])cc2Cl)cc1	TetraHydroFuran	M062X	Def2TZVP	LANL2DZ	29/30	detail
	COC(=O)[C@@H]1C[C@@H](O)CN1C(=O)OC(C)(C)C	TetraHydroFuran	M062X	Def2TZVP	LANL2DZ	8/30	detail
	COC(=O)[C@@H]1C[C@H](O)CN1C(=O)OC(C)(C)C	TetraHydroFuran	M062X	Def2TZVP	LANL2DZ	10/30	detail
	COc1ccc2c(c1)c(CCO)c(C)n2C(=O)c1ccc(Cl)cc1	TetraHydroFuran	M062X	Def2TZVP	LANL2DZ	22/30	detail

CCCN(CCC)S(=O)(=O)c1ccc(CO)cc1

2: G = -744410.4 + 0.10 kcal/mol

number_of_atoms	charge	multiplicity	dipole	molar_mass	molar_volume	electronic_spatial_extent	E_scf	zero_point_correction
39	0	1	6.46	271.37	2275.5	6329.46	-1186.58	0.33

Atom-Level Descriptors

atom_idx	label	X	Y	Z	VBur	Mulliken_charge	APT_charge	NPA_charge	NPA_core	NPA_valence	NPA_Rydberg
0	C	-0.33	0.95	-0.77	0.42	-0.44	0.05	-0.6	2	4.6	0.01
1	C	-0.33	0.72	-0.76	0.56	-0.22	0.08	-0.41	2	4.4	0.01
2	C	-0.43	0.37	-0.51	0.64	-0.33	0.39	-0.21	2	4.2	0.02

Fig. 3 Query view (left) and the molecule view (right) of the web interface. The molecule view is a snapshot while viewing the second lowest energy conformation in 3D.

generalizability of this methodology, we designed a representative, diverse, and unbiased aryl bromide substrate scope through an unsupervised learning approach with DFT-derived featurization. An initial set of aryl bromides (molecular weight < 400) was generated through a Reaxys® search, which yielded around 290 000 candidates. After applying additional filters, such as commercial availability, spectroscopic data availability and functional group compatibility, we selected 2683 aryl bromides for DFT calculation. Our preliminary studies suggest that common featurization approaches, such as molecular fingerprints and cheminformatics descriptors, are often insufficient to represent electronic and steric features of substrates relevant to reactivity sites, necessitating the use of DFT-derived featurization. With Auto-QChem, low-energy conformers were generated from SMILES strings for all aryl bromides. Gaussian jobs of generated conformers were then submitted to a connected HPC cluster. Successful calculations were logged and uploaded to the Auto-QChem database, along with 168 electronic and steric features (HOMO/LUMO energy, dipole moments, atomic volume, *etc.*) extracted from Gaussian log files. It is worth noting that, using Auto-QChem, DFT calculations of this size can be completed within a few days with minimal human intervention.

After feature preprocessing,[45] we used the remaining 95 features for hierarchical clustering to generate 15 clusters[44] and chose the molecules closest to the center of each cluster as our substrate scope (Fig. 4b). The final substrate scope includes a wide array of functional groups (such as esters, nitriles, chlorides), substitution patterns (mono-, di- and tri-substitution) and steric features (ortho-, meta- and para-substitution). We also surveyed 116 Ni/photoredox methodology papers and compiled a complete set of 50 aryl bromide substrates used in this literature. By comparing substrates from Ni/photoredox literature with our selected substrate scope, we discovered that most aryl bromides substrates from literature examples are only present in a few clusters, while others (primarily clusters possessing multi-substituted aryl bromides) are significantly unexplored (Fig. 4a). This approach allows for study of chemical space coverage in the literature and identification of areas where high *versus* low yields are generally obtained. Unlike traditional substrate scopes in the literature, where selection usually happens in an arbitrary and subjective fashion, our machine learning-designed substrate scope is better suited for evaluating the generality of a reaction without human bias (Fig. 4c). A systematic selection of substrates also enabled us to train regression models without selection bias and formulate predictive generalizations from DFT-derived features. We discovered that electronegativity of the aryl bromides was highly correlated with yield. Using electronegativity as a predictive feature, a generalized additive model (GAM) was trained with 15 aryl bromides and validated with 37 additional substrates. Similar models trained with 22 literature substrates were less accurate and did not generalize well during validation.[38] This analysis demonstrated that a systematically designed substrate scope can effectively evaluate the generality of a reaction, as well as reveal reactivity trends for a larger population of substrates.

Use case 2: ligand parametrization and enantioselectivity prediction in nickel catalysis

In another example, we developed a Ni/photoredox-catalyzed enantioselective cross-electrophile coupling of aryl iodides

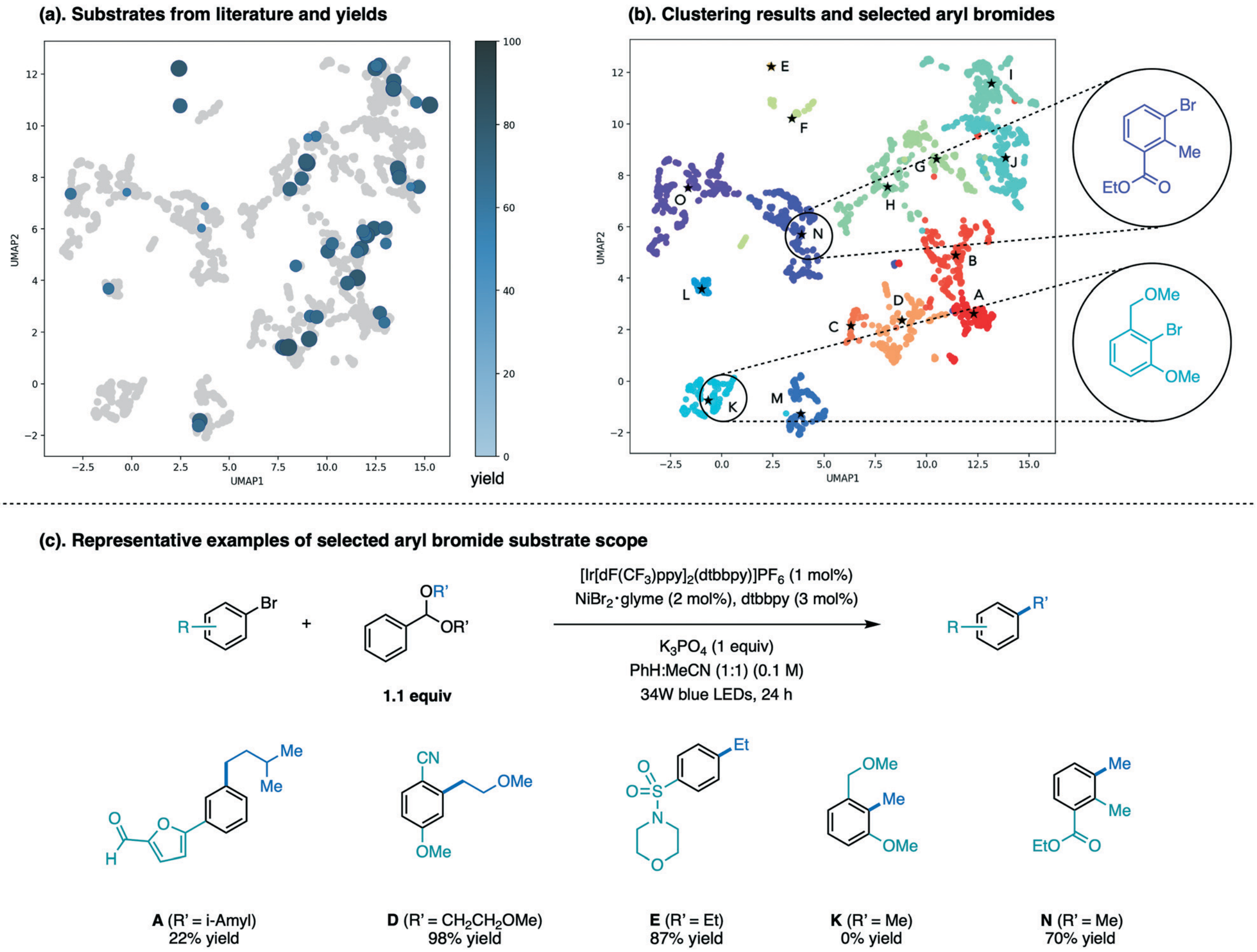


Fig. 4 Use case 1: substrate scope design in a Ni/photoredox methodology development.

and styrene oxides.[39] The optimal ligand, a chiral biimidazoline (BiIm) ligand, was discovered only after extensive screening of common chiral amine bidentate ligands. Bioxazoline (BiOx) ligands previously used in our asymmetric reductive coupling of aziridines[40] resulted in good enantioselectivity but low to moderate yield of the product. To understand the key features of BiIm ligands that affect reactivity and enantioselectivity of this reaction, we sought to use statistical modeling with physical and chemical descriptors from DFT calculations.

We selected a total of 20 BiOx and 9 BiIm ligands and collected enantioselectivity data under standard reaction condition with a model substrate (Fig. 5a). Under the hypothesis that ligand environments will likely affect the computed features, we performed DFT calculations for all the ligands under three different environments: free ligand, ligand bound to a tetrahedral nickel difluoride complexes and ligand bound to a square planar nickel oxidative addition complex (Fig. 5b). As a potential limitation, Auto-QChem (and most conformer-generating software) cannot reliably generate conformers for transition metal complexes,[46] especially for group 10 metals like nickel. As a result, all the initial conformers for nickel-bound ligand were manually generated and submitted for DFT calculation. Auto-QChem was still used to extract electronic and atomic volume features from output files. Importantly, our multivariate linear regression analysis showed that, although they give a worse fit for the data, features derived from free ligands were sufficient for a descriptive linear regression model. From our regression model, NBO_{C4}, NBO_{N1} (ref. 47) and polarizability independently affect $\Delta\Delta G^{\ddagger}$, suggesting that electronic, rather than steric attributes of BiIm ligands govern the enantioselectivity of this reaction (Fig. 5c). This study demonstrated how insights from regression modeling with DFT-derived features can afford a mechanistic probe of complex catalytic reactions.

Use case 3: reaction condition optimization *via* Bayesian optimization

The optimization of reaction conditions is often tedious and time-consuming in methodology development campaigns. In the pursuit of conditions that provide the highest yield for reactions of interest, chemists often rely on empirical

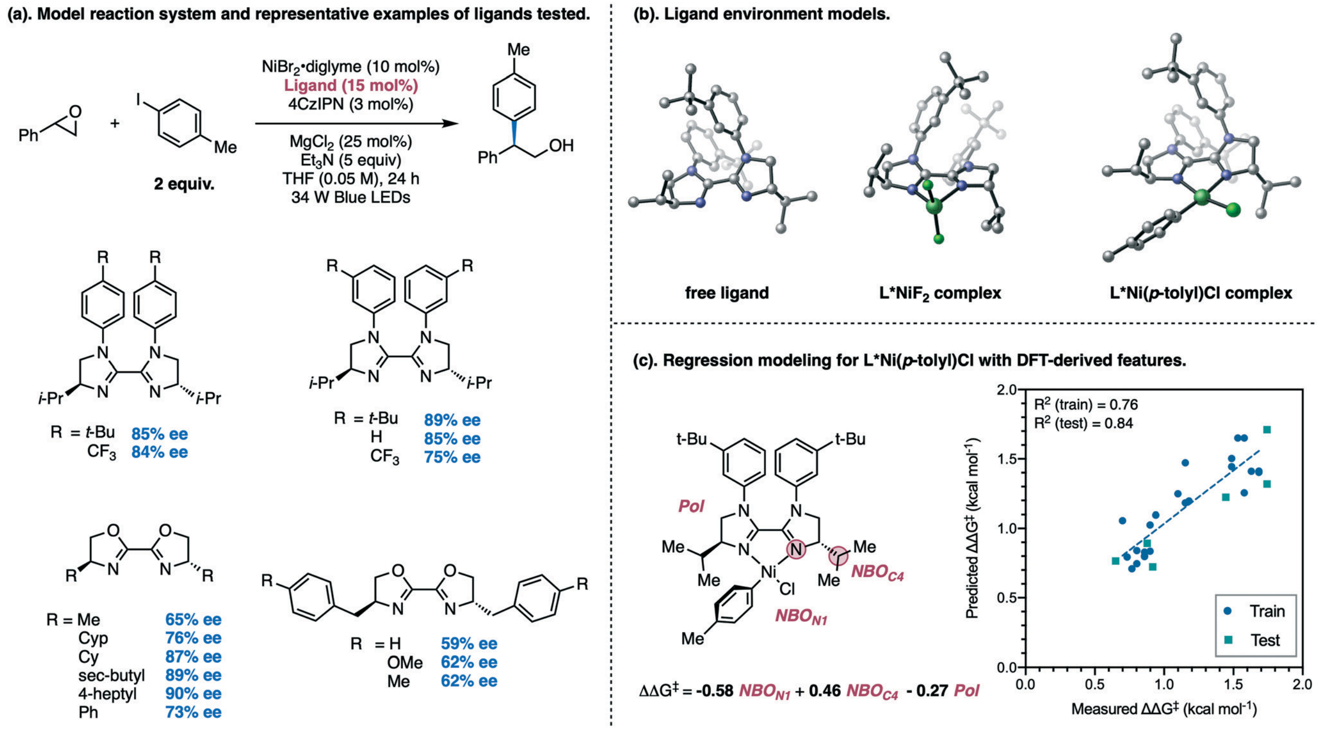


Fig. 5 Use case 2: ligand parametrization and enantioselectivity prediction in nickel catalysis.

knowledge and qualitative understandings of the current optimization progress to design the next experiment. Typical approaches include the adoption of known conditions from literature, design of experiments (DoE), or more time- and resource-intensive methods such as high-throughput experimentations (HTE) and in-depth mechanistic studies. For individual reaction components, the lack of quantitative assessment of their effects on reaction yield usually requires running many combinations of the conditions, which in turn limits the size of chemical space explored during optimization.

In our recent study,[41] we demonstrated the application of Bayesian optimization, a sequential design algorithm for

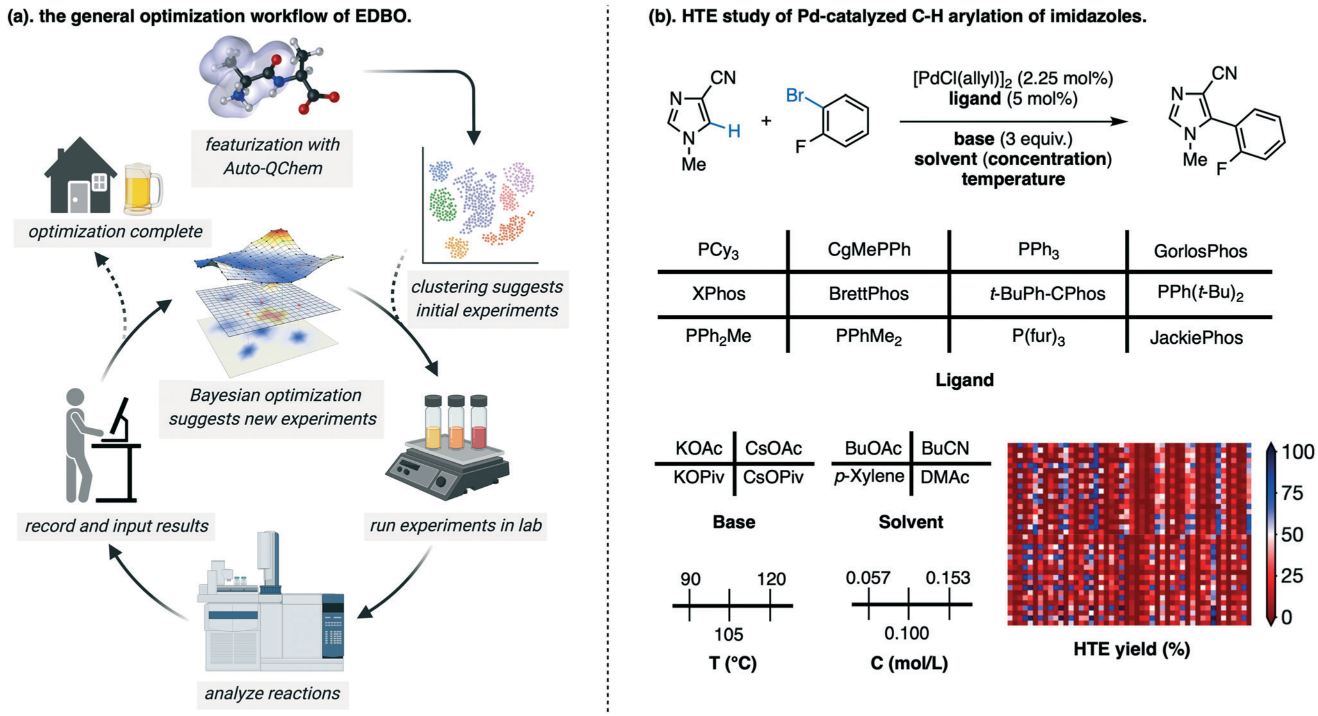


Fig. 6 Use case 3: reaction condition optimization *via* Bayesian optimization.

global optimization of black-box functions, in efficient reaction condition optimization. We developed a software framework, EDBO (Experimental Design *via* Bayesian Optimization), where a Bayesian optimization algorithm was integrated into real-time laboratory experimentations (Fig. 6). After a reaction space is defined, initial experiments are selected *via* clustering or other sampling approaches. Chemists run the suggested reactions in lab, analyze the results when reactions finish and input reaction yield into the system. Bayesian optimization algorithms use new results to update the prior and form a new posterior distribution over the objective function. An acquisition function is constructed with the new posterior to determine new query points (new reactions to run). This optimization loop is repeated until the desired yield or resource limit is reached.

During the development of the Bayesian optimization framework, we evaluated its performance by comparing simulation results to human decision-making benchmarks obtained with large HTE reaction datasets. Bayesian optimization requires each reaction component to be translated into a suitable numeric representation. We tested the effects of different featurizations (DFT-derived features, molecular descriptors such as Mordred,[42] and one-hot encoding) on optimization convergence. DFT calculations for hundreds of molecules contained in these reaction datasets were completed with an early version of Auto-QChem,[43] which greatly simplified our workflow. Compared to other featurizations, DFT features offer improved learning curves and more consistent performance in terms of worst-case loss.

To statistically test the performance of our framework in a new reaction space, we collected reactivity data for a palladium-catalyzed C–H arylation reaction. Using high-throughput experimentation, we evaluated this reaction with 12 phosphine ligands, 4 bases, 4 solvents, 3 temperatures and 3 concentrations (1728 possible conditions in total). Through a web game which simulates the process of choosing conditions and running reactions, we established a human decision-making baseline by inviting 50 expert chemists to optimize this reaction and recording their optimization progress within an imaginary experimental budget (100 experiments). Using DFT-derived features, our Bayesian optimization framework (simulated 50 times) achieved a higher average performance within the first 15 experiments even with random initialization and found conditions with >99% yield 100% of the time. Most human chemists either ended the optimization prematurely or failed to identify the highest-yielding conditions, which had not been previously reported for this type of reaction. The performance benefits obtained with DFT-derived features further validate the necessity of high-throughput DFT featurization frameworks like Auto-QChem.

Limitations and future directions

We would also like to highlight some limitations of Auto-QChem at the present stage and outline some future directions. First, as mentioned in use case 2, Auto-QChem lacks the ability to generate accurate conformers for transition metal complexes and molecules with non-canonical bonds. Such problems are not unique to Auto-QChem as we leverage external programs such as RDKit to handle conformational searches. We are actively seeking improvement and experiment with other conformational search software that can alleviate such problems.

Another important functionality of Auto-QChem is the ability to manage jobs on HPC clusters. Currently, Auto-QChem only supports Slurm scheduler. Integration of other cluster job schedulers will require significant modifications to existing code. We plan to support Univa Grid Engine (UGE) in the near future, and we welcome experienced users to integrate Auto-QChem into their own HPC clusters.

We also plan to expand certain functionalities of Auto-QChem. For example, we will include external packages and automate the calculation of additional electronic and steric features that are not currently supported by Auto-QChem. Barring any quality control issues, we also intend to invite other users to upload data to Auto-QChem. With enough data on hand, we would also like to train machine learning models with existing data to predict DFT-level features for similar molecules, which will address the speed bottleneck of DFT calculations in our workflow.

Conclusions

Herein, we reported Auto-QChem, an automated, high-throughput and end-to-end DFT calculation workflow. The implementation and workflow of Auto-QChem are discussed in detail. Designed to facilitate the increasing applications of machine learning models in organic chemistry, Auto-QChem generates DFT-derived molecular and atomic features starting from simple string representations of the molecules. After initial conformational searches, each conformer is submitted to a local computer cluster for DFT calculations with user-specified configurations. Cluster jobs are managed directly through Auto-QChem with error-correcting mechanisms. Successful calculation results and extracted DFT features are then uploaded to a database. A web interface (https://autoqchem.org) is also available for convenient data access. We also present three distinct studies from our group where Auto-QChem was used to featurize a large set of molecules and greatly simplified the workflow. Current limitations and potential areas of improvement are also discussed to provide an outlook for the future of Auto-QChem.

Data availability

The code and usage examples for Auto-QChem can be found at: https://github.com/PrincetonUniversity/auto-qchem. API and functional documentation for Auto-QChem can be found at: https://princetonuniversity.github.io/auto-qchem. The web interface and data currently deposited in Auto-QChem can be accessed at: https://autoqchem.org.

Conflicts of interest

There are no conflicts of interest to declare.

Acknowledgements

Financial support is provided by Bristol-Myers Squibb, the Princeton Innovation Fund, the Princeton Catalysis Initiative, NIH-NIGMS (R35 GM126986), the NSF under the CCI Center for Computer Assisted Synthesis (CHE-1925607) and the Dreyfus Program for Machine Learning in the Chemical Sciences and Engineering. A. M. Z. gratefully acknowledge financial support from the Schmidt DataX Fund at Princeton University made possible through a major gift from the Schmidt Futures Foundation. The authors would also like to thank Stavros Kariofillis, Will Sii Hong Lau, Jose Garrido Torres and Daniel Seungwook Min for their comments on the manuscript and assistance with figures. Fig. 6 was created with BioRender.com.

Notes and references

1 A. F. Zahrt, J. J. Henle, B. T. Rose, Y. Wang, W. T. Darrow and S. E. Denmark, *Science*, 2019, **363**, 1134–1140.

2 D. T. Ahneman, J. G. Estrada, S. Lin, S. D. Dreher and A. G. Doyle, *Science*, 2018, **360**, 186–190.

3 M. H. S. Segler, M. Preuss and M. P. Waller, *Nature*, 2018, **555**, 604–610.

4 S. Zhao, T. Gensch, B. Murray, Z. L. Niemeyer, M. S. Sigman and M. R. Biscoe, *Science*, 2018, **362**, 670–674.

5 L. David, A. Thakkar, R. Mercado and O. Engkvist, *J. Cheminf.*, 2020, **12**, 56.

6 S. Jaeger, S. Fulle and S. Turk, *J. Chem. Inf. Model.*, 2018, **58**, 27–35.

7 S. Kearnes, K. McCloskey, M. Berndl, V. Pande and P. Riley, *J. Comput.-Aided Mol. Des.*, 2016, **30**, 595–608.

8 C. W. Coley, R. Barzilay, W. H. Green, T. S. Jaakkola and K. F. Jensen, *J. Chem. Inf. Model.*, 2017, **57**, 1757–1772.

9 R. D. Hull, S. B. Singh, R. B. Nachbar, R. P. Sheridan, S. K. Kearsley and E. M. Fluder, *J. Med. Chem.*, 2001, **44**, 1177–1184.

10 M. Olivecrona, T. Blaschke, O. Engkvist and H. Chen, *J. Cheminf.*, 2017, **9**, 48.

11 S. Curtarolo, W. Setyawan, G. L. W. Hart, M. Jahnatek, R. V. Chepulskii, R. H. Taylor, S. Wang, J. Xue, K. Yang, O. Levy, M. J. Mehl, H. T. Stokes, D. O. Demchenko and D. Morgan, *Comput. Mater. Sci.*, 2012, **58**, 218–226.

12 S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson and G. Ceder, *Comput. Mater. Sci.*, 2013, **68**, 314–319.

13 T. Mayeshiba, H. Wu, T. Angsten, A. Kaczmarowski, Z. Song, G. Jenness, W. Xie and D. Morgan, *Comput. Mater. Sci.*, 2017, **126**, 90–102.

14 K. Mathew, J. H. Montoya, A. Faghaninia, S. Dwarakanath, M. Aykol, H. Tang, I. Chu, T. Smidt, B. Bocklund, M. Horton, J. Dagdelen, B. Wood, Z.-K. Liu, J. Neaton, S. P. Ong, K. Persson and A. Jain, *Comput. Mater. Sci.*, 2017, **139**, 140–152.

15 F. Zapata, L. Ridder, J. Hidding, C. R. Jacob, I. Infante and L. Visscher, *J. Chem. Inf. Model.*, 2019, **59**, 3191–3197.

16 J. T. Krogel, *Comput. Phys. Commun.*, 2016, **198**, 154–168.

17 S. P. Huber, S. Zoupanos, M. Uhrin, L. Talirz, L. Kahle, R. Häuselmann, D. Gresch, T. Müller, A. V. Yakutovich, C. W. Andersen, F. F. Ramirez, C. S. Adorf, F. Gargiulo, S. Kumbhar, E. Passaro, C. Johnston, A. Merkys, A. Cepellotti, N. Mounet, N. Marzari, B. Kozinsky and G. Pizzi, *Sci. Data*, 2020, **7**, 300.

18 M. Uhrin, S. P. Huber, J. Yu, N. Marzari and G. Pizzi, *Comput. Mater. Sci.*, 2021, **187**, 110086.

19 S. Kirklin, J. E. Saal, B. Meredig, A. Thompson, J. W. Doak, M. Aykol, S. Rühl and C. Wolverton, *npj Comput. Mater.*, 2015, **1**, 15010.

20 K. Choudhary, K. F. Garrity, A. C. E. Reid, B. DeCost, A. J. Biacchi, A. R. H. Walker, Z. Trautt, J. Hattrick-Simpers, A. G. Kusne, A. Centrone, A. Davydov, J. Jiang, R. Pachter, G. Cheon, E. Reed, A. Agrawal, X. Qian, V. Sharma, H. Zhuang, S. V. Kalinin, B. G. Sumpter, G. Pilania, P. Acar, S. Mandal, K. Haule, D. Vanderbilt, K. Rabe and F. Tavazza, *npj Comput. Mater.*, 2020, **6**, 173.

21 D. G. A. Smith, D. Altarawy, L. A. Burns, M. Welborn, L. N. Naden, L. Ward, S. Ellis, B. P. Pritchard and T. D. Crawford, *WIREs Comput. Mol. Sci.*, 2021, **11**, e1491.

22 B. G. Abreha, S. Agarwal, I. Foster, B. Blaiszik and S. A. Lopez, *J. Phys. Chem. Lett.*, 2019, **10**, 6835–6841.

23 D. Weininger, *J. Chem. Inf. Model.*, 1988, **28**, 31–36.

24 Python Software Foundation, https://www.python.org, (accessed January 2022).

25 M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin, V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. P. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman and D. J. Fox, *Gaussian 16*, Gaussian, Inc., Wallingford CT, 2016.

26 MongoDB, https://www.mongodb.com, (accessed January 2022).

27 Dash Python User Guide, https://dash.plotly.com, (accessed January 2022).

28 Amazon Web Services, https://aws.amazon.com, (accessed January 2022).

29 T. Kluyver, in *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, ed. F. Loizides and B. Schmidt, IOS Press, Amsterdam, 2016, pp. 87–90.

30 RDKit: Open-source cheminformatics, https://www.rdkit.org/, (accessed January 2022).

31 N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch and G. R. Hutchison, *J. Cheminf.*, 2011, **3**, 33.

32 S. Riniker and G. A. Landrum, *J. Chem. Inf. Model.*, 2015, **55**, 2562–2574.

33 Slurm workload manager, https://slurm.schedmd.com, (accessed January 2022).

34 A. Dalke and J. Hastings, *J. Cheminf.*, 2013, **5**, O6.

35 rdkit.Chem.fmcs.fmcs module, https://www.rdkit.org/docs/source/rdkit.Chem.fmcs.fmcs.html, (accessed January 2022).

36 S. K. Kariofillis, S. Jiang, A. M. Żurański, S. S. Gandhi, J. I. Martinez Alvarado and A. G. Doyle, *J. Am. Chem. Soc.*, 2022, **144**, 1045–1055.

37 S. K. Kariofillis, B. J. Shields, M. A. Tekle-Smith, M. J. Zacuto and A. G. Doyle, *J. Am. Chem. Soc.*, 2020, **142**, 7683–7689.

38 See original publication for details on regression models.

39 S. H. Lau, M. A. Borden, T. J. Steiman, L. S. Wang, M. Parasram and A. G. Doyle, *J. Am. Chem. Soc.*, 2021, **143**, 15873–15881.

40 B. P. Woods, M. Orlandi, C.-Y. Huang, M. S. Sigman and A. G. Doyle, *J. Am. Chem. Soc.*, 2017, **139**, 5688–5691.

41 B. J. Shields, J. Stevens, J. Li, M. Parasram, F. Damani, J. I. M. Alvarado, J. M. Janey, R. P. Adams and A. G. Doyle, *Nature*, 2021, **590**, 89–96.

42 H. Moriwaki, Y.-S. Tian, N. Kawashita and T. Takagi, *J. Cheminf.*, 2018, **10**, 4.

43 Auto-QChem, https://github.com/b-shields/auto-QChem, (accessed January 2022).

44 15 is the number of clusters at which the maximum and stable Silhouette score was reached.

45 Preprocessing includes scaling, outlier removal, removal of features with low variance and correlation analysis.

46 Software that specifically focuses on conformer generation for transition-metal complexes do exist, such as molSimplify: E. I. Ioannidis, T. Z. H. Gani and H. J. Kulik, *J. Comput. Chem.*, 2016, **37**, 2106–2117.

47 In the cases of BiIm and BiOx ligands, it is possible to align all the molecules with common substructure and generate consistent indexing for atoms (e.g., N1, C4). For molecules with distinct structures, additional processing might be required to extract features for atoms of interest.