

# CalcUS: An Open-Source Computational Chemistry Web Platform.

Raphaël Robidas\* and Claude Y. Legault\*

University of Sherbrooke, Department of Chemistry, Centre in Green Chemistry and Catalysis,  
2500 boul. de l'Université, Sherbrooke (Québec) J1K 2R1, CANADA.

---

**ABSTRACT:** Computational chemistry is an increasingly active field due to the improvement of computing resources and theoretical tools. However, its use remains usually limited to technically-inclined users due to the technical challenges of preparing, launching and analyzing calculations. In this context, we have developed CalcUS, an open-source platform to streamline computational chemistry studies. Its objective is to democratize access to computational chemistry by providing a user-friendly web interface to simplify running and analyzing quantum mechanical calculations. It is freely available, expandable and customizable. It promotes connectivity to multiple software packages and algorithms, thus providing state-of-the-art techniques to all practitioners. We propose CalcUS as a standalone tool and infrastructure to support other open-source packages.

---

## INTRODUCTION

Computational chemistry is a major field in chemical sciences and is rapidly gaining popularity as a central tool to provide insights and predictions on experimental systems. New computational methods are being developed to improve accuracy or increase computational efficiency.<sup>1</sup> Combined with the improvements in the computing power of modern processors, these advances make computational modelling of realistic chemical systems increasingly valuable and common. For example, molecular dynamics (MD) have found numerous applications in the simulation of large organic systems, such as proteins<sup>2</sup> and polymers.<sup>3,4</sup> Quantum mechanical (QM) calculations are used for smaller (typically up to 200 atoms) yet extremely diverse systems. By computing the approximate wave function or electron density, one gains access to a whole range of valuable fundamental insights. Structural, electronic, and energetic properties are of great interest in fundamental and mechanistic studies. Moreover, these calculations can provide properties such as NMR shifts,<sup>5</sup> absorption spectra,<sup>6-8</sup> EPR spectra<sup>9</sup> and Mössbauer isomer shifts.<sup>10</sup> Accordingly, a large number of researchers can benefit from chemical modelling.

Many software packages exist to carry out QM calculations. They all possess their idiosyncratic format, syntax and output files. This can be quite daunting to new practitioners, as they have to learn the particularities of the packages they are using in addition to the numerous concepts of computational chemistry. For more advanced users, these differences hinder productivity and mandate additional efforts to develop an efficient workflow (e.g. by scripting) with multiple packages. Thus, users tend to restrain themselves to software suites they are familiar with and may miss on other packages' features.

We envisioned creating an interface to perform, automate, and manage QM calculations with multiple software packages to solve these challenges. Several projects aim to automate computational chemistry, but either for a specific purpose<sup>11-13</sup> or through a form of scripting.<sup>13-19</sup> We had specific objectives for the interface: 1) Be user-friendly and intuitive; 2) Be open-

source, customizable and expandable; 3) Be easy to use on multiple operating systems (OS); 4) Provide a suitable environment to learn computational chemistry; 5) Provide all the necessary tools to manage and analyze QM calculations; 6) Provide advanced functionalities for tasks which would otherwise be cumbersome to carry out. One software that approaches this vision is WebMO,<sup>20</sup> although it is not open-source and cannot be modified freely. To the best of our knowledge, there is no open-source project that embodied our vision and suited our needs. We report herein the details of CalcUS, our open-source, web-based computational chemistry platform.

## CONCEPTION

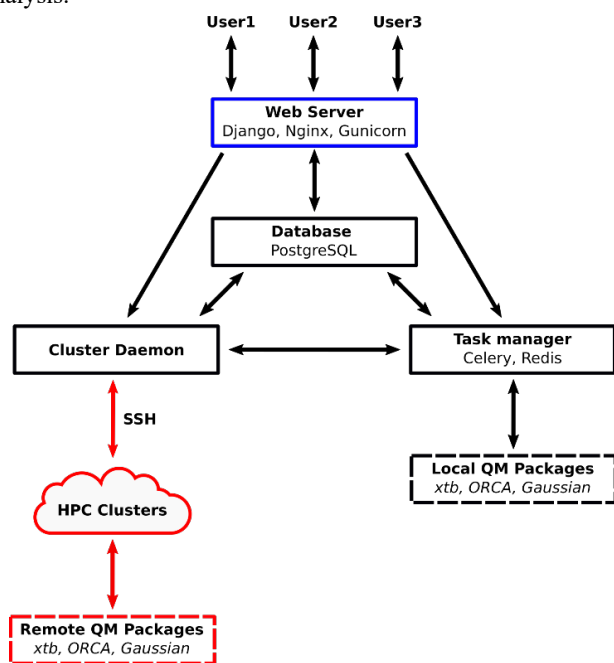
In the field of computational chemistry, it is common to run QM software packages on a Linux distribution, the usual OS for high-performance computing (HPC) clusters. Its use tends to be different from GUI-centered OS such as Windows and MacOS. Consequently, an investment of time and effort is required to become proficient in its use.

To benefit from the broad compatibility of Linux with QM software while avoiding its steep learning curve, we elected to develop CalcUS as a web-based platform based on a Linux server. This makes the tool OS independent for the end-user and eliminates the need to download and install software to be used. We thus designed a stack of services to power the required functionalities. The entire configuration has been automated using Docker,<sup>21</sup> an open-source containerization platform. Indeed, the configuration files provided in CalcUS contain the procedures to ensure that the different services are set up correctly. Docker uses these files to create the CalcUS virtual environment. In practice, this means that CalcUS requires minimal configuration during the installation. Moreover, Docker makes running a Linux virtual machine on any OS possible. This creates a stable and constant Linux environment for CalcUS while still being available on all systems.

We designed CalcUS as a web platform that entire research groups can use. The end-users each possess individual accounts

and can form groups. They can view each other's calculations in a group, which promotes collaboration and mutual assistance. This offers an attractive alternative to manual file transfers and instead allows users to browse through the information freely. Due to the easy installation, anyone could also install a local platform instance on their personal computer instead. Indeed, the software infrastructure that manages calculations in CalcUS requires little computing power and memory. Even a small single-board computer (SBC), such as a Raspberry Pi, can be sufficient to host the server. Consequently, research groups that do not want to invest in a powerful workstation can nevertheless run a centralized instance of CalcUS on nearly any functioning computer. In contrast, using a powerful workstation, with the required QM software installation, provides access to practical local QM calculations.

The backend of CalcUS is written in the programming language Python, which is widely used in science. As such, many Python packages can be utilized directly to provide useful features. The general software infrastructure is presented in Scheme 1. Every software component and module used are open-source and available freely under various permissive licenses. The program is centered around a Django web application, which powers the interface. The web server handles the requests and manages the data stored in a PostgreSQL database. Tasks which have long runtimes, such as local calculations, are sent to a queue and processed by the task manager. Remote connections to computation clusters are handled by a separate daemon, which controls file upload, task monitoring and result analysis.



**Scheme 1. Simplified structure of CalcUS. Dashed nodes represent optional third-party packages.**

## MAIN FUNCTIONALITIES

CalcUS makes it easier for beginners to learn practical computational chemistry by providing all the tools required to launch, monitor and analyze calculations in a simple web interface. For example, input structures can directly be drawn in a 2D sketcher. The result of the conversion to 3D can be inspected in

the same interface, before launching the calculation. This eliminates the need for a 3D molecular editor and accelerates the input creation in many cases. In cases where the 2D sketcher does not do justice to the desired input structure, molecular structures can be uploaded in any of the standard formats, such as XYZ, MOL, MOL2 or SDF formats. Raw Gaussian output files can also be uploaded directly; CalcUS will parse the last structure of the files and use them as input.

The interactive interface replaces manual calculation keyword specification. The type of calculation (*e.g.* geometrical optimization, frequency calculation, *etc.*) and level of theory are simply chosen for the desired software package configured. Fields for additional parameters appear in the interface when required. For example, constrained optimizations require the end-user to enter the constraints based on the atomic numbering of the input structure. Consequently, this structure is displayed in the 3D molecular viewer integrated in the interface, and numbering appears when a calculation that requires atom numbering is selected. Finally, custom keywords can also be appended to the input file to not limit the possibilities to what is explicit in the interface and allow more advanced users to customize the input for a given QM software suite fully. Settings can then be saved as preset and later reused. It is important to note that the interface does not change depending on the software package used, making it effortless for the users to use multiple ones. CalcUS currently interfaces with Gaussian,<sup>22</sup> ORCA<sup>23–25</sup> and the xTB suite.<sup>26–39</sup> These packages are of course *not* provided with CalcUS due to the licensing requirements and must be installed on either local machine that will perform the calculations, or the remote HPC resources used.

## HPC RESOURCES MANAGEMENT

One of the challenges that beginners in computational chemistry face is the use of high-performance computing (HPC) clusters, as it commonly requires functional knowledge of shell (*i.e.* command line) environments. This technical proficiency will often restrict some users to particular GUI-based QM software. CalcUS enables the use of the full potential of remote HPC resources (*i.e.* clusters) in the same user-friendly interface developed for the local resources. When launching new calculations, the user can simply specify a remote cluster (previously configured) to be used. CalcUS then manages remote job submission of calculations, as the user would through a shell. It handles file upload and job submission to SLURM scheduling software, which is widely used on HPC clusters. The user simply needs to provide a submission script template for each HPC cluster. This approach provides good flexibility for choosing the parameters of the jobs to be submitted and allows support for a variety of cluster configurations.

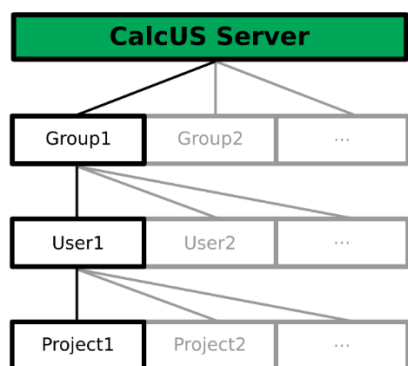
Local and remote calculations can seamlessly be used simultaneously. For example, one could find useful to perform quick conformational sampling locally using free DFTB software such as the xTB package, then submit the resulting structures for high-level DFT optimization on HPC clusters. All these steps are performed within the same interface without the need for the end-user to perform any of the usual file transfer/job preparation steps.

Of course, the use of HPC clusters requires additional configuration, which is relevant to explain further. The key element of this process is the initiation of a secure shell (SSH) connection to the remote HPC cluster. When adding a new cluster

access in CalcUS, a public-private key pair will be generated using standard cryptographic protocols. The public key must be added manually by the user in his list of allowed keys on the HPC cluster. The private key can then act as a surrogate for the user password, allowing the platform to access the user account on the remote HPC cluster. For security reasons, this private key is encrypted by a password chosen by the user, which is not saved anywhere. Thus, the platform can only connect to the HPC cluster when the user enters this password and initiates the connection. This SSH connection can be closed by the user at any moment. All these measures ensure that the actual password of the user account on the HPC cluster is never used or saved in the CalcUS infrastructure, and that only the user can make use of the private key and allow connection to his account.

## USERS MANAGEMENT

As mentioned before, the interface supports multiple users and groups. Thus, a single server can allow many researchers to perform calculations from their web browser, without having to install any software on their personal computer. Moreover, users in the same group can have access to other member's projects (unless marked as "private", in which case they are hidden and inaccessible). In this way, users can collaborate, provide assistance, and learn from each other and leverage others' calculations as starting point for their own. Calculations of users in a group cannot be viewed by users outside of the group, in order to protect potentially confidential or sensitive data. This structure enables a single CalcUS server, if installed on a computer accessible network-wide, to be used by as many research groups and users as desired (Scheme 2). If many calculations are submitted locally to be performed on the server, they will be queued and performed sequentially through the task manager. The execution of calculations submitted to remote HPC resources will be managed by each cluster's queueing system. Consequently, the number of end-users is only limited by the power of the server host and by the amount of computational power required by them. If the server is only used to manage remote calculations on HPC clusters, however, a very large number of end-users can be supported.



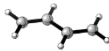
Scheme 2. User structure.

## DATA MANAGEMENT

One key aspect of computational chemistry is data management. Indeed, the generation and analysis of large amount of output files is a continuous challenge in this field. In line with our main objective with CalcUS, we developed a data

management scheme that aims at simplifying the technical work for the end-user. Consequently, a conceptual organization was introduced in the database, with the entry point being a project (Scheme 3). Each user can define multiple projects, enabling a simple organization.

All calculations are then organized in a chemically logical fashion. Firstly, we define a *structure* as a molecular object with uniquely defined 3D coordinates. For example, each conformer of a species will be stored in a separate structure. All calculation types can be separated into structure-creating calculations (e.g. geometrical optimization, transition structure search, conformer search) and non-structure-creating calculations (e.g. frequency calculation, NMR prediction, single-point energy). As the non-structure-creating calculations depend on these initial 3D coordinates, their results are linked to their corresponding structure. The parsed data from a calculation is stored in the *properties* class and also point to the *parameters* class used to describe its specific conditions (e.g. theory model, basis set).

Project	<b>Name:</b> Diels-Alder Reaction
Molecule	<b>Name:</b> Butadiene <b>InChI:</b> 1S/C4H6/c1-3-4-2/h3-4H,1-2H2
Ensemble	<b>Name:</b> Butadiene Conformers
Structure	<b>Structure:</b> 
Properties	<b>Energy:</b> -155.79080812 Ha <b>Free energy:</b> -155.72879922 Ha
Parameters	<b>Method:</b> M062X <b>Basis set:</b> Def2SVP

Scheme 3. Example of data organization in the database.

Structures are grouped into *ensembles*. An ensemble will contain a set of closely related structures. The most common case is an ensemble containing all conformers of a given species. This is especially powerful, as it allows the user to perform actions on sets of structures. Any calculation that can be launched from a structure can also be launched from an ensemble. Thus, for the user, it is as quick to start a geometry optimization on an ensemble of 10 structures than on a single structure. Moreover, ensembles can be filtered by energetic parameters (cutoff by relative energy or Boltzmann weight) prior to submission. Additionally, it is possible to automatically obtain the Boltzmann-weighted properties computed for each ensemble and each theory level, which reduces further the overhead in dealing with many conformers and model parameters.

In some cases, an ensemble does not have thermochemical significance and simply serve as a convenient container for related structures. This is the case for example with relaxed geometrical scans. In these cases, the ensemble object simply allows the user to browse through the results and launch subsequent calculations on chosen structures.

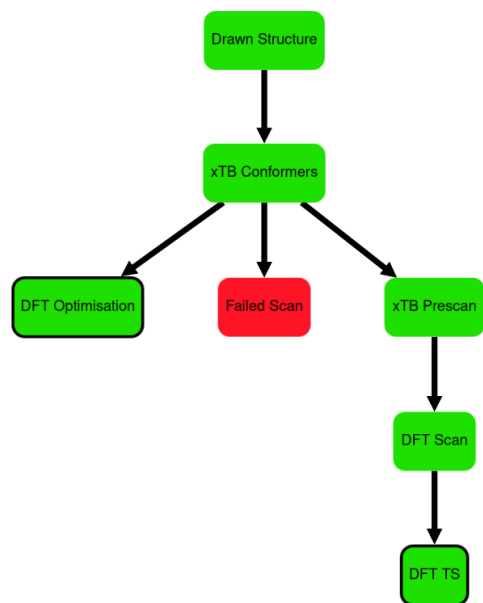
Ensembles are further grouped into *molecules*. In this context, a molecule refers to a species or system with a unique chemical identity, independent of its precise geometry or properties. Thus, each molecule can be characterized by its InChI key,<sup>40</sup> which is automatically computed. While this definition is not restrictive (e.g. when modelling transition states and

reaction products within the same "molecule" object), it nevertheless serves as a useful organizational tool. In this way, large numbers of calculations are classified based on their chemical relevance.

All the molecule instances are grouped in the user-defined *projects*, which separate unrelated molecules. For these projects, CalcUS can generate summaries in CSV files containing all the required information for analysis. In particular, the Boltzmann-weighted energies, enthalpies and free energies are computed for each ensemble of each molecule. The individual properties of each structure can be consulted. A summary section presents the Boltzmann-weighted free energies per ensemble and per theory level, which is generally the desired information. This CSV file can be viewed directly on the platform or downloaded and opened with any spreadsheet software.

## WORKFLOW MANAGEMENT

In complex projects, it is common to perform multi-step sequences of calculations which use the structure of a previous calculation as input. This is typically called a computational workflow. Furthermore, branching can occur in these processes, thus forming a complex tree of linked calculations. This can be confusing and lead to errors and time loss. To address this issue, CalcUS keeps track of the origin of each ensemble and constructs a visual, interactive ensemble tree for each molecule (Figure 1). The user can provide custom names to ensembles for clarity. Clicking on any node redirects the user to the corresponding ensemble page.

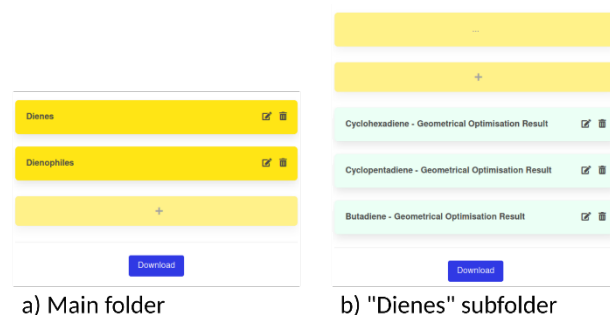


**Figure 1. Automatically generated interactive ensemble map.**

One advantage of using a relational database like PostgreSQL is the ability to link objects together in a non-linear fashion. For example, this is what allows the creation of ensemble trees, such as the one illustrated in Figure 1. Moreover, the ensemble properties (e.g. Boltzmann-weighted energies) can be calculated dynamically by querying all the structures belonging to that ensemble and performing the appropriate weighting. These relationships do not need to follow a restrictive top-down

organization, such as in folders and subfolders. As such, data can be organized simultaneously in different ways and efficiently retrieved in multiple ways.

We used this flexibility to facilitate the management of complex projects. Indeed, not all generated ensembles will be useful for final analysis, as many of them will be preoptimizations, scans or failed attempts at obtaining the desired structure. Nonetheless, it might be desirable to keep them for future consultation. Consequently, a priority flag has been introduced to differentiate ensembles used for final analysis. Visually, this adds a black border around the corresponding nodes on the ensemble map (Figure 1). With respect to the data analysis, flagged ensembles gain a priority over unflagged ensembles. Consequently, when generating the CSV file, the user can choose to include only the flagged ensembles, providing only the relevant information in the summary. To enable further organization of the data, the interface offers a classification system based on folders, which is familiar to all users (Figure 2). Flagged ensembles can thus be organized as the user desires, which facilitates results management, in particular for complex projects. A CSV file which respects this structure can then be generated.



**Figure 2. Illustration of the online folders system.**

The complete output files of each calculation of a project can also be downloaded, either in a "molecule-based" file structure or a "folder-based" file structure. While CalcUS aims to provide the tools necessary to manage and analyze computational chemistry projects, one can use the interface as an initial calculation creator and manager prior to further local analysis.

## ADVANCED FUNCTIONALITIES

CalcUS was developed with the aim to democratize access to computational chemistry by providing a user-friendly interface to beginners. Nonetheless, advanced users can also benefit from multiple time-saving features integrated in CalcUS:

1) The basis set for each individual element can be easily chosen in the interface. The Basis Set Exchange Python library<sup>41</sup> offers an extensive choice of basis sets which can be used in the calculations. Similar to the project's site, our interface automatically filters the available basis sets for the selected atoms (and *vice versa*).

2) Minimal energy path calculations (MEP) can also be performed by combining the rapid GFN2-xTB method<sup>27</sup> from the xTB package and the nudged elastic band (NEB) algorithms implemented in ORCA.<sup>42-44</sup> This is useful to obtain transition states with multiple reaction coordinates, for example. This approach takes as input one structure on either side of the transition state and constructs a path between them. The use of GFN2-xTB significantly reduces the computational cost of this method and makes it viable for relatively large systems.



3) Conformational sampling is greatly simplified by the interface. Using the software package CREST,<sup>31,32</sup> conformational sampling can be performed like any other calculation. The result is an ensemble of conformers which can be inspected or filtered for further calculations. Moreover, constrained conformation searches can be performed, for example for transition states. To do so, CREST requires a specific input file which specifies the constraints as well as the unconstrained atoms. While it would be time-consuming and error-prone to create this file manually, CalcUS makes that process much faster: as with constrained optimizations, only the constrained distances/angles/dihedrals need to be specified.

4) NMR calculations can be useful for structure assignment and mechanistic studies.<sup>5</sup> One challenge that render the process cumbersome is that QM software provide unique isotropic values for each atom of interest, while there are usually numerous chemically equivalent atoms. The values of these atoms need to be averaged to compare to the actual experimentally obtained chemical shift values. This step is done automatically by CalcUS, using Morgan's algorithm<sup>45</sup> for the identification of chemically-equivalent atoms, saving time to the user. Furthermore, the computed isotropic values are not directly related to the experimental chemical shifts; they must be corrected using regression parameters (available for many theory models<sup>5,46-48</sup>). This correction can be performed directly in the interface by specifying the appropriate parameters.

## FUTURE DEVELOPMENT AND PUBLIC CONTRIBUTIONS

While the CalcUS platform currently contains a selected set of tools for computational chemistry, it also constitutes a solid foundation for developers to expand on. Through the use of a user-friendly web interface and a stable virtual environment, it constitutes a convenient way to distribute software packages in a "plug-and-play" fashion. Indeed, any required configuration or dependency can be automated through Docker. This greatly reduces the friction that a user might encounter when trying new software packages for the first time. We aim to constantly improve the usefulness of CalcUS for the scientific community, especially with the help of the latter. Consequently, while it is possible to copy the code and perform modifications on local versions, we encourage users to submit improvements on the Github repository.

## CONCLUSION

We introduced CalcUS, an open-source project aiming to make computational chemistry more accessible as well as to save time to more experienced practitioners. It reduces the cognitive load through multiple convenient features and automated processes. We propose CalcUS as a tool in itself, but also as a means of simplifying the use of new software packages and algorithms. For developers, this means a greater accessibility to their tool and more visibility. For end-users, it will result in a seamless integration of state-of-the-art modelling tools into their workflow.

## SOFTWARE AVAILABILITY

CalcUS is freely available under the GPL-v3 license on Github: <https://github.com/cyllab/CalcUS>. The repository contains or links to all relevant documentation.

## AUTHOR INFORMATION

### Corresponding Authors

E-mail: [raphael.robidas@usherbrooke.ca](mailto:raphael.robidas@usherbrooke.ca)

E-mail: [claudel.legault@usherbrooke.ca](mailto:claudel.legault@usherbrooke.ca)

### Author Contributions

The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript.

## ACKNOWLEDGMENT

C. Y. L. and R. R. thank the National Science and Engineering Research Council (NSERC) of Canada, the Canada Foundation for Innovation (CFI), the FRQNT Centre in Green Chemistry and Catalysis (CGCC), and the Université de Sherbrooke for support. Computational resources were provided by Calcul Québec and Compute Canada. R. R. is grateful to FRQNT and NSERC for a Graduate Scholarship (B1X/B2X) and a Canada Graduate Scholarship (CGS M), respectively.

## REFERENCES

- (1) Grimme, S.; Schreiner, P. R. Computational Chemistry: The Fate of Current Methods and Future Challenges. *Angew. Chem. Int. Ed.* **2018**, 57 (16), 4170–4176.
- (2) Adcock, S. A.; McCammon, J. A. Molecular Dynamics: Survey of Methods for Simulating the Activity of Proteins. *Chem. Rev.* **2006**, 106 (5), 1589–1615.
- (3) Krishna, S.; Sreedhar, I.; Patel, C. M. Molecular Dynamics Simulation of Polyamide-Based Materials – A Review. *Comput. Mater. Sci.* **2021**, 200, 110853.
- (4) Joshi, S. Y.; Deshmukh, S. A. A Review of Advancements in Coarse-Grained Molecular Dynamics Simulations. *Mol. Simul.* **2021**, 47 (10–11), 786–803.
- (5) Lodewyk, M. W.; Siebert, M. R.; Tantillo, D. J. Computational Prediction of <sup>1</sup>H and <sup>13</sup>C Chemical Shifts: A Useful Tool for Natural Product, Mechanistic, and Synthetic Organic Chemistry. *Chem. Rev.* **2012**, 112 (3), 1839–1862.
- (6) Laurent, A. D.; Jacquemin, D. TD-DFT Benchmarks: A Review. *Int. J. Quantum Chem.* **2013**, 113 (17), 2019–2039.
- (7) Zuehlsdorff, T. J.; Isborn, C. M. Modeling Absorption Spectra of Molecules in Solution. *Int. J. Quantum Chem.* **2019**, 119 (1), e25719.
- (8) Makkar, P.; Ghosh, N. N. A Review on the Use of DFT for the Prediction of the Properties of Nanomaterials. *RSC Adv.* **2021**, 11 (45), 27897–27924.
- (9) Neese, F. Quantum Chemistry and EPR Parameters. *EPR Spectrosc. Fundam. Methods* **2018**, 115.
- (10) Filatov, M. First Principles Calculation of Mössbauer Isomer Shift. *Coord. Chem. Rev.* **2009**, 253 (5–6), 594–605.
- (11) Yesiltepe, Y.; Nuñez, J. R.; Colby, S. M.; Thomas, D. G.; Borkum, M. I.; Reardon, P. N.; Washton, N. M.; Metz, T. O.; Teeguarden, J. G.; Govind, N.; Renslow, R. S. An Automated Framework for NMR Chemical Shift Calculations of Small Organic Molecules. *J. Cheminformatics* **2018**, 10 (1), 52.

- (12) Ferrari, B. AutoGAMESS: A Python Package for Automation of GAMESS(US) Raman Calculations. *J. Open Source Softw.* **2019**, *4* (41), 1612.
- (13) Sun, X.; Soini, T. M.; Poater, J.; Hamlin, T. A.; Bickelhaupt, F. M. PyFrag 2019—Automating the Exploration and Analysis of Reaction Mechanisms. *J. Comput. Chem.* **2019**, *40* (25), 2227–2233.
- (14) Jacob, C. R.; Beyhan, S. M.; Bulo, 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* (10), 2328–2338.
- (15) Krogel, J. T. Nexus: A Modular Workflow Management System for Quantum Simulation Codes. *Comput. Phys. Commun.* **2016**, *198*, 154–168.
- (16) Řezáč, J. Cuby: An Integrative Framework for Computational Chemistry. *J. Comput. Chem.* **2016**, *37* (13), 1230–1237.
- (17) Richard, R. M.; Bertoni, C.; Boschen, J. S.; Keipert, K.; Pritchard, B.; Valeev, E. F.; Harrison, R. J.; de Jong, W. A.; Windus, T. L. Developing a Computational Chemistry Framework for the Exascale Era. *Comput. Sci. Eng.* **2019**, *21* (2), 48–58.
- (18) 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* (7), 3191–3197.
- (19) 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* (4), e1510.
- (20) Polik, W. F.; Schmidt, J. R. WebMO: Web-Based Computational Chemistry Calculations in Education and Research. *WIREs Comput. Mol. Sci.* e1554.
- (21) Merkel, D. Docker: Lightweight Linux Containers for Consistent Development and Deployment. *Linux J.* **2014**, *2014* (239), 2.
- (22) 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-Young, D.; 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, J. A., Jr.; 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 Revision C.01. 2016.
- (23) Neese, F. The ORCA Program System. *WIREs Comput. Mol. Sci.* **2012**, *2*, 73–78.
- (24) Neese, F. Software Update: The ORCA Program System, Version 4.0. *WIREs Comput. Mol. Sci.* **2018**, *8*, e1327.
- (25) Neese, F.; Wennmohs, F.; Becker, U.; Riplinger, C. The ORCA Quantum Chemistry Program Package. *J. Chem. Phys.* **2020**, *152* (22), 224108.
- (26) 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* (5), 1989–2009.
- (27) Bannwarth, C.; Ehlert, S.; Grimme, S. GFN2-XTB—An Accurate and Broadly Parametrized Self-Consistent Tight-Binding Quantum Chemical Method with Multipole Electrostatics and Density-Dependent Dispersion Contributions. *J. Chem. Theory Comput.* **2019**, *15* (3), 1652–1671.
- (28) Spicher, S.; Grimme, S. Robust Atomistic Modeling of Materials, Organometallic, and Biochemical Systems. *Angew. Chem. Int. Ed.* **2020**, *59* (36), 15665–15673.
- (29) Bannwarth, C.; Caldeweyher, E.; Ehlert, S.; Hansen, A.; Pracht, P.; Seibert, J.; Spicher, S.; Grimme, S. Extended Tight-Binding Quantum Chemistry Methods. *WIREs Comput. Mol. Sci.* **2021**, *11* (2), e1493.
- (30) Pracht, P.; Caldeweyher, E.; Ehlert, S.; Grimme, S. A Robust Non-Self-Consistent Tight-Binding Quantum Chemistry Method for Large Molecules. *ChemRxiv*. DOI: 10.26434/chemrxiv.8326202.v1
- (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) Pracht, P.; Bohle, F.; Grimme, S. Automated Exploration of the Low-Energy Chemical Space with Fast Quantum Chemical Methods. *Phys. Chem. Chem. Phys.* **2020**, *22* (14), 7169–7192.
- (33) Grimme, S. A Simplified Tamm-Dancoff Density Functional Approach for the Electronic Excitation Spectra of Very Large Molecules. *J. Chem. Phys.* **2013**, *138* (24), 244104.
- (34) Bannwarth, C.; Grimme, S. A Simplified Time-Dependent Density Functional Theory Approach for Electronic Ultraviolet and Circular Dichroism Spectra of Very Large Molecules. *Comput. Theor. Chem.* **2014**, *1040–1041*, 45–53.
- (35) Grimme, S.; Bannwarth, C. Ultra-Fast Computation of Electronic Spectra for Large Systems by Tight-Binding Based Simplified Tamm-Dancoff Approximation (STDA-XTB). *J. Chem. Phys.* **2016**, *145* (5), 054103.
- (36) de Wergifosse, M.; Grimme, S. Nonlinear-Response Properties in a Simplified Time-Dependent Density Functional Theory (STD-DFT) Framework: Evaluation of the First Hyperpolarizability. *J. Chem. Phys.* **2018**, *149* (2), 024108.
- (37) de Wergifosse, M.; Bannwarth, C.; Grimme, S. A Simplified Spin-Flip Time-Dependent Density Functional Theory Approach for the Electronic Excitation Spectra of Very Large Diradicals. *J. Phys. Chem. A* **2019**, *123* (27), 5815–5825.
- (38) de Wergifosse, M.; Grimme, S. Nonlinear-Response Properties in a Simplified Time-Dependent Density Functional Theory (STD-DFT) Framework: Evaluation of Excited-State Absorption Spectra. *J. Chem. Phys.* **2019**, *150* (9), 094112.

- (39) de Wergifosse, M.; Seibert, J.; Grimme, S. Simplified Time-Dependent Density Functional Theory (STD-DFT) for Molecular Optical Rotation. *J. Chem. Phys.* **2020**, *153* (8), 084116.
- (40) Heller, S. R.; McNaught, A.; Pletnev, I.; Stein, S.; Tchekhovskoi, D. InChI, the IUPAC International Chemical Identifier. *J. Cheminformatics* **2015**, *7* (1), 23.
- (41) Pritchard, B. P.; Altarawy, D.; Didier, B.; Gibson, T. D.; Windus, T. L. New Basis Set Exchange: An Open, Up-to-Date Resource for the Molecular Sciences Community. *J. Chem. Inf. Model.* **2019**, *59* (11), 4814–4820.
- (42) Mills, G.; Jónsson, H.; Schenter, G. K. Reversible Work Transition State Theory: Application to Dissociative Adsorption of Hydrogen. *Surf. Sci.* **1995**, *324* (2–3), 305–337.
- (43) Jónsson, H.; Mills, G.; Jacobsen, K. W. Nudged Elastic Band Method for Finding Minimum Energy Paths of Transitions. In *Classical and Quantum Dynamics in Condensed Phase Simulation*; World Scientific: Singapore, 1998; pp 384–404.
- (44) Ásgeirsson, V.; Birgisson, B. O.; Björnsson, R.; Becker, U.; Neese, F.; Riplinger, C.; Jónsson, H. Nudged Elastic Band Method for Molecular Reactions Using Energy-Weighted Springs Combined with Eigenvector Following. *J. Chem. Theory Comput.* **2021**, *17* (8), 4929–4945.
- (45) Morgan, H. L. The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **1965**, *5* (2), 107–113.
- (46) Rablen, P. R.; Pearlman, S. A.; Finkbiner, J. A Comparison of Density Functional Methods for the Estimation of Proton Chemical Shifts with Chemical Accuracy. *J. Phys. Chem. A* **1999**, *103* (36), 7357–7363.
- (47) Jain, R.; Bally, T.; Rablen, P. R. Calculating Accurate Proton Chemical Shifts of Organic Molecules with Density Functional Methods and Modest Basis Sets. *J. Org. Chem.* **2009**, *74* (11), 4017–4023.
- (48) Bally, T.; Rablen, P. R. Quantum-Chemical Simulation of <sup>1</sup>H NMR Spectra. 2. Comparison of DFT-Based Procedures for Computing Proton–Proton Coupling Constants in Organic Molecules. *J. Org. Chem.* **2011**, *76* (12), 4818–4830.