molli: A General-Purpose Python Toolkit for Combinatorial Small Molecule Library Generation, Manipulation, and Feature Extraction.

Alexander S. Shved,* Blake E. Ocampo, Elena S. Burlova, Casey L. Olen, N. Ian Rinehart, and Scott E. Denmark*

Department of Chemistry, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, United States

* shvedalx@illinois.edu, sdenmark@illinois.edu

ABSTRACT: The construction, management and analysis of large *in silico* molecular libraries is critical in many areas of modern chemistry. Herein, we introduce the MOLecular LIibrary toolkit, "molli", which is a Python 3 cheminformatics module that provides a streamlined interface for manipulating large *in silico* libraries. Three-dimensional, combinatorial molecule libraries can be expanded directly from two-dimensional chemical structure fragments stored in CDXML files with high stereochemical fidelity. Geometry optimization, property calculation, and conformer generation are executed by interfacing with widely used computational chemistry programs such as OpenBabel, RDKit, ORCA, NWChem, and xTB/CREST. Conformer-dependent grid-based feature calculators provide numerical representation, and interface to robust three-dimensional visualization tools that provide comprehensive images to enhance human understanding of libraries with thousands of members. The package includes a command-line interface in addition to Python classes to streamline frequently used workflows. Parallel performance is benchmarked on various hardware platforms, and common workflows are demonstrated for different tasks ranging from optimized grid-based descriptor calculation on catalyst libraries to an NMR chemical shift prediction workflow from CDXML files.

KEYWORDS: cheminformatics, Python, molecular formats, descriptors, parallel computations

1 INTRODUCTION

Modern synthetic chemistry increasingly incorporates theoretical and empirical data-oriented approaches for designing functional small molecules, understanding reaction pathways, and predicting and optimizing reaction outcomes.[1–5] In recent years, medium- to high-throughput experimentation techniques have provided access to large data sets suitable for subsequent statistical analysis and predictive modeling.[6–10] Critically, encoding molecules in a machine-readable format is essential before any computational analysis of the physical molecular entities can commence.[11,12] Although a variety of different software tools for the enumeration and encoding of in silico libraries exists,[13] we have found a lack of suitably general, open-source tools to support accurate generation of libraries containing complex stereochemical information directly from the two-dimensional depictions familiar to all chemists.

Representations of molecules with calculated features range from computationally simple to highly complex. In general, feature extraction from a molecule can be accomplished by considering, in order of increasing computational complexity: (1) only the atoms and bonds encoded in the molecular graph, (2) the three-dimensional (3D) shape, and (3) the full electronic structure of the molecule.[14] Molecular graph-based feature extraction methods, such as topological fingerprinting,[15] are fast to calculate but lack 3D information that is critical for certain optimization problems. Indeed, the low-energy conformers of a molecule play an essential role in determining its chemical properties. Consequentially, 3D fingerprinting methods have been developed[16,17] and recent interest in incorporating 3D information into molecular graph objects has led to a variety of feature extraction methods employing graph neural networks.[18–20] More challenges in representation arise when considering conformational flexibility, solvation, non-covalent interaction, and other molecular features that can only be described by explicit 3D molecular encoding.

Our interest in molecular representation stems from our attempts at modelling quantitative structure-(enantio)selectivity relationships (QSSR) in enantioselective chemical reactions using chiral, small molecule catalysts.[21] Our group and others have designed a variety of alignment-dependent, molecular interaction and indicator field (MIF) descriptors intending to capture the relevant features of a chiral catalyst that lead to high enantioselectivity.[22–24] A particular catalyst scaffold typically offers numerous options for analogue synthesis at well-defined positions on the structure, and each analogue then has potentially many possible conformers. Therefore, our workflow required the ability to write custom code to manipulate large collections of 3D molecular structures and perform high-throughput computations on combinatorially constructed libraries of compounds. In 2019, this laboratory released the ccheminfolib toolkit,[22] an early iteration of a software package designed to handle combinatorial construction of large *in silico* libraries. One of the main motivations for the creation of a new software package was *to establish a modern, convenient and extensible interface* that would allow rapid prototyping of chemical library-oriented workflows. Since the release of ccheminfolib, we sought to address the following problems:

1. Generation of molecule and conformer libraries directly from ChemDraw™ .CDXML files with stereochemical fidelity.

2. Parallelization mechanisms capable of processing chemical libraries with external computational software.
3. Rapid input/output of molecular entities from the disk-based storage
4. Optimized calculations of the grid-based descriptors

As a result, we began the project to create the MOLecular LIbrary toolkit Python 3 package we have dubbed "molli".

2 COMBINATORIAL LIBRARY GENERATION PIPELINE

2.1 CDXML File Parsing

Most computational workflows start with either 1D representations (SMILES) or 3D representations (.xyz or .mol files). We frequently faced challenges associated with the 1D representations. Axial and planar chirality cannot be encoded in SMILES strings and the stereochemical information is therefore lost upon the library generation. Although extensions to SMILES and other string-based representation methods have been developed to address these issues,[25–28] 3D structures are naturally devoid of such limitations in encoding chirality. We believe that one of the most desirable ways to generate large libraries of 3D structures is by correctly interpreting their 2D chemical depictions.

Our contribution to the process of 2D to 3D structure conversion was in the realization of algorithmically deterministic *z*-coordinate (out-of-plane) displacement, guided by the 2D stereobonds. This process mimics the thought process that chemists use to interpret the 2D-structures. For all acyclic stereobonds[29] leading from an atom, the connected fragment (determined by the breadth-first graph traversal) was rotated out of plane depending on the number of adjacent atoms out of the plane of drawing (Figure 1A). We chose ±60° for when the substituent was attached to an atom with two adjacent atoms, and ±90° for three adjacent atoms (Figure 1B) (see Figure 1A, B, E). Since no deterministic rotation of endocyclic bonds could be devised, the atoms in respective bonds are subjected to simple out-of-plane displacement of the participating atoms (Figure 1C). The endpoints of a wedged bond are shifted out of plane only if the wedge points towards them, whereas both atoms are displaced in bold/hash style bonds. The z-coordinate adjustments by rotations or translations allowed the displacement of the coordinates in the correct direction toward the desired minimum after a geometry optimization (Figure 1E). This practically eliminated the cases of unanticipated configurational inversion upon a forcefield minimization (See the Supporting Information 3.2). It proved useful in the context of axial and planar chirality interpretation into 3D representation wherein no simple designators can typically be assigned and enforced by ChemDraw™ or related packages (Figure 1F). We purposefully avoided the use of any stereochemical designators (e.g. *R*/*S*), so that all stereogenic elements can be encoded in the same way, regardless of designator availability.

The present parser has only two hard rules enforced in the CDXML file parsing: (1) the label used for dictionary-like lookup of structures must be bold-faced, not chemically interpreted, and placed below the structure and (2) the parsing is deterministic with respect to the drawing, however different ways of depicting the same configuration may result in different parsing results; It is the responsibility of the user to verify that the minimization after parsing yields expected results. For example, changing the directionality of stereobonds may produce results that are not identical (see Figure 1E).

In addition to the enhanced coordinate perception, the parser recognizes most other elements, which are available from the structures (Figure 1D). Labeling the atoms proves useful for subsequent direct referencing of atoms in the code (see Section 3). Specifying attachment points provides a convenient handle for 3D combinatorial expansion. Abbreviated functional groups are allowed, so long as ChemDraw™ can expand them into valid structures. Isotopic notations and multi-center attachments are interpreted by the parser. Although some of this additional functionality is available in other codes, the convenience of a Python backend used by molli not only enhances the transparency of the workflow, but also ensures easy customizations by the users. Parsing CDXML files to `Molecule` objects can be executed directly from the command line with the `molli parse` command, or by using the `molli.load` interface (Figure 1F).

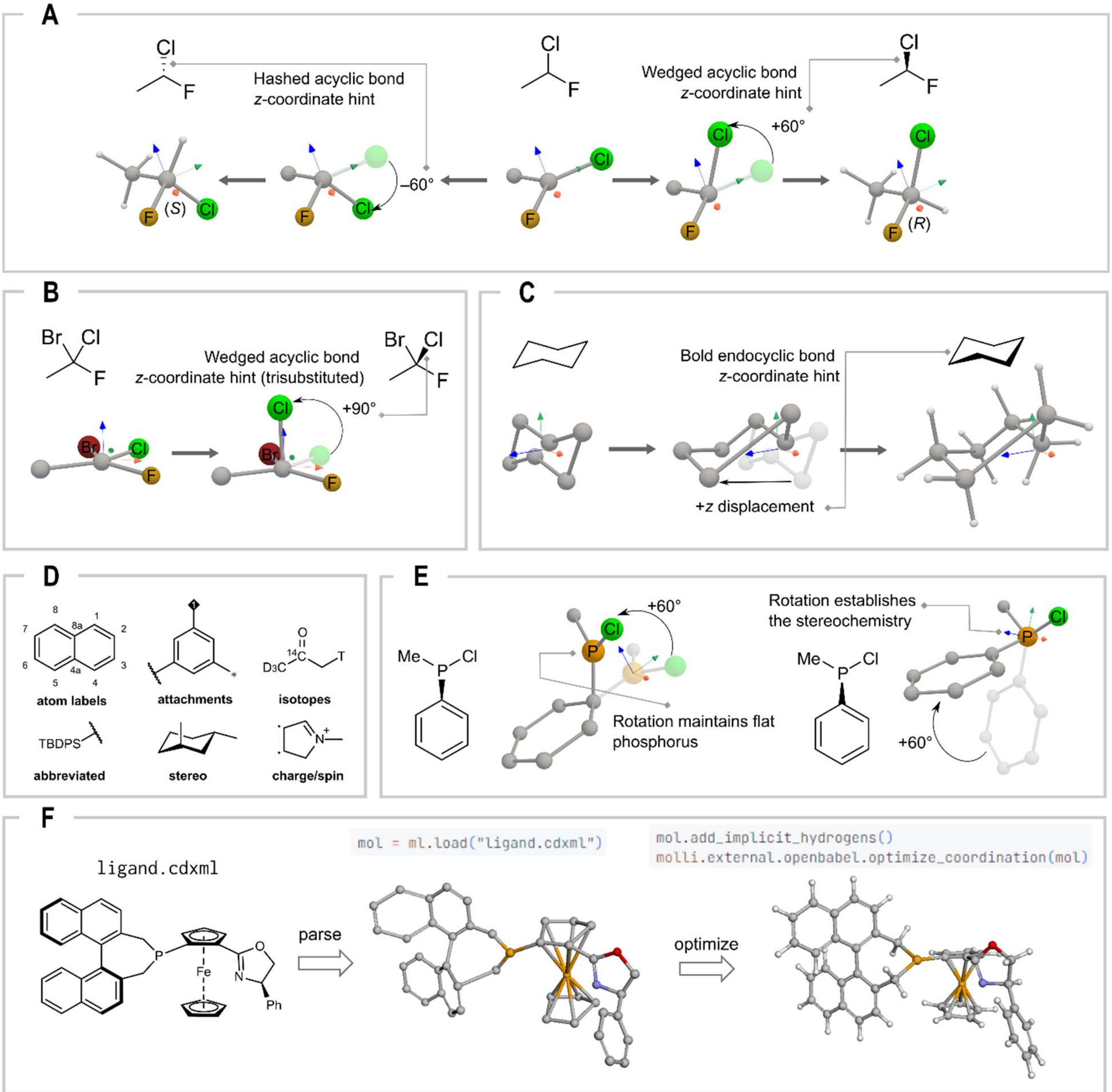


Figure 1. Molli CDXML parsing workflow. (A) Out-of-plane rotation of substituent results in a structure with defined stereochemistry after hydrogen addition. (B) Illustration of +90° rotation in the trisubstituted case (C) Out-of-plane displacement of endocyclic atoms (D) Additional elements recognized by molli (E) Stereochemical depiction ambiguity (F) Molli preserves stereochemical information in cases of point, axial and planar chirogenic elements.

2.2 Combinatorial Library Expansion from CDXML Files

Combinatorial library expansion can be performed programmatically in Python or directly from the command line with the `molli combine` command. Because CDXML parsing stores the atom labels and native CDXML attachment point markup (see Section 2.1), these can be accessed directly to specify the rules of combinatorial expansion. The user parses `MoleculeLibrary` objects containing cores, with labelled attachment points, and substituents with attachment points.

The `molli combine` command then takes in both core and substituent library objects and joins the substituents to the user-specified attachment points on the cores (see the Supporting Information, Figure S9A). Substituent sets are selected based on the number of user-specified attachment points and a selection rule, which can be the same substituents attached at all core attachment points, permutations of the substituents, combinations of the substituents, or combinations with replacement. The result of this command then produces the enumerated combinatorial library as a `MoleculeLibrary` object (Figure S9B). We have previously reported the generation of a bis(oxazoline) (BOX) combinatorial library (Figure S9C) comprising a total of 96,120 members, with 267 options for 4,4'-oxazoline substitution, nine options for 5,5'-oxazoline substitution including stereochemical analogues relative to the 4,4'-positions, and 40 options for substitution at the methylene group bridging the two oxazoline rings.[30] With the streamlined workflow described in Figure S9, we successfully obviated manual creation of the full expanded .CDXML file shown in Figure S9C.

2.3 Molecular Object Collections

Modern cheminformatics tools offer a multitude of ways of storing chemical information for single molecules or small collections. We identified a need to access molecules or conformer ensembles from large collections without the necessity to create a full-fledged database. A binary molecule and conformer serialization strategy was implemented through a disk-based dictionary-like structure of MessagePack-serialized data that we refer to as uKV file (see the Supporting Information for details). This form of storage offers the flexibility of storing molli objects in large random-access files, with optimized read/write performance.

To demonstrate the broader applicability of the proposed molecular storage toward medicinally relevant datasets, we provide examples imported from the literature. The data from the MoleculeNet[31] subset of the GEOM[32] dataset was reimported as a molli .uKV file (see the Supporting Information, Section 1.2.3). The same operation was performed on the `drugs_crude` subset (Supporting Information, Section 1.2.4), providing the largest collection, containing 292,028 discrete molecules and 31,223,451 conformers.

3 PARALLEL CALCULATION PIPELINE

In a typical workflow, tasks such as geometry optimizations, conformer generations, and property calculations are done in parallel. Typically, these calculations are carried out with external software[33] by a unified process in which: (1) a set of input files is prepared, (2) a worker process receives said input files and shell commands to execute, (3) the commands are run, and the output is captured, and (4) the necessary files are subsequently transferred to permanent storage and are analyzed. Molli implements a parallel job pipeline that allows computation of molecular properties with external software such as RDKit,[34,35] XTB,[36] CREST,[37] NWChem[38] and ORCA,[39] and it can be easily extended to any other package (see Supporting Information Section 7.1 and 7.2 for more details). The two workflows shown below demonstrate the flexibility that a molli library can offer.

3.1 KRAS inhibitor rotational barrier estimation.

Hindered rotation around single bonds resulting in axial chirality is an important structural motif in catalysts and pharmaceuticals.[40,41] The barrier height may not always be straightforward to estimate experimentally and doing so computationally in a high throughput sense with minimal human involvement may significantly facilitate pre-screening of synthetic candidates before their experimental evaluation. The workflow started with the CDXML file containing the necessary molecular fragments which was deliberately constructed to mimic the original figure[42] as closely as possible (Figure 2). Parsing the CDXML files with the help of molli results in the `MoleculeCollection` file that was subsequently subjected to the computational pipeline. Coarse structure minimization with MMFF94,[43] as implemented in OpenBabel,[44] yielded the initial guess structures. An XTB[36] relaxed surface scan was then used to explore the potential energy surface with respect to rotation around the C–N bond by constraining the appropriate dihedral atoms. Parsing and serialization of atom labels allowed quick identification of specific atoms for the dihedral angle constraints within the scripts. When XTB relaxed surface scan maxima and minima were used as guess structures, we were unable to locate transition states **23**, **27**, and **29**. The inability to converge to transition states may be challenging to rectify manually for large libraries. Molli molecular building capability allows the construction of better transition state guesses by joining the distorted core from a successfully located transition state with an optimized aryl substituent. These structures converged smoothly to the corresponding transition states. The computed barriers were generally close to the experimentally observed ones (Table 1), except for **25** and **27**. Despite structural similarities between **24** and **25**, the latter was overpredicted by 25 kJ mol^{-1} compared to experimental measurement. The barrier for **27**, on the other hand, was underpredicted by 16 kJ mol^{-1}. We cannot offer a supportable explanation for these outliers. We cautiously speculate that the barrier could potentially exhibit a significant dependence on explicit solvation and/or proton transfer effects.

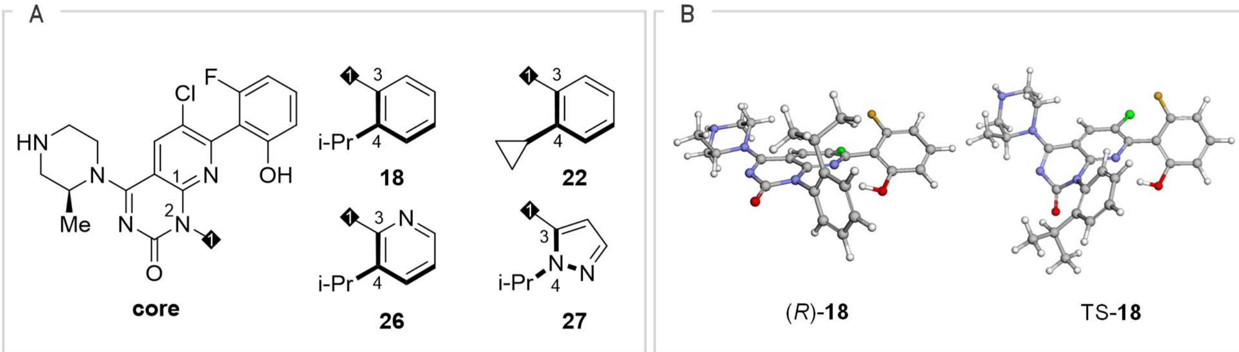


Figure 2. KRAS inhibitor rotational barrier estimation workflow. (A) The fragment of the CDXML file that was used for parsing and library assembly. For a full list of structures see the Supporting Information Section 6.3. (B) Representative equilibrium geometries of *R*-isomers and transition states. Of note is the remarkable distortion of the 2-pyrimidinone ring away from planarity in the transition states owing to severe strain (Figures S36–S44), consistent with the previous report.[45]

Table 1. Summary of Predicted vs. Observed Rotational Barriers at B97-3c Level of Theory (in kJ mol^{-1}). For a Full List of Structures see the Supporting Information, Section 6.3.

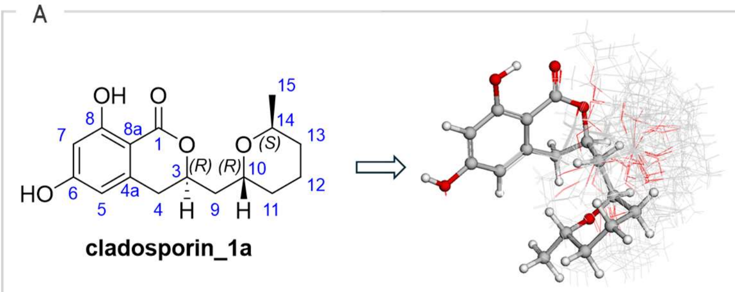
Compound*	Exp.	Pred.
18	108.8	108.3
22	104.6	103.2
23	>125.5	141.0
24	>125.5	150.0
25	121.3	146.1
26	98.3	92.6
27	90.0	73.7
28	73.2	69.9
29	107.9	101.3

3.2 GIAO-DFT NMR prediction workflow.

Prediction of NMR spectra, particularly ^{13}C NMR spectra is a common task encountered in structural elucidation and revision.[46,47] Although modern computational tools allow fast GIAO-DFT NMR prediction, a complete cycle workflow that automates the task to start with a ChemDraw™ file and orchestrates the required computations, is not generally available using open-source tools. A major advance toward this goal is the CENSO program, which enables direct processing of CREST conformer ensembles and plotting of Boltzmann-weighted spectra.[48]

* The compound labels throughout the manuscript were chosen to be non-standard on purpose. This decision is to demonstrate that the source CDXML files can be constructed with the compounds labeled arbitrarily. We chose to label ours the way they were labeled in the original publications.

The workflow starts with parsing the 3D structures from the .CDXML file to yield a `MoleculeCollection` (Figure 3A). Basic minimization with the MMFF94[43] force field as implemented in OpenBabel followed by conformer generation with the CREST v4 workflow[49] created the desired conformer ensembles. These ensembles were subjected to geometry evaluation with the B97-3c method as implemented in ORCA. Upon conformer generation, the NMR isotropic shieldings were calculated with RIJCOSX-PBE0 / pcSseg-2[50] + CPCM(chloroform).[51,52] Molli features simple syntax that is used to compute the NMR shieldings (Figure 3B). Molli implements a parser of output files, which was used to scrape thermochemical and magnetic properties and stores them within the molecule objects. Boltzmann weights were computed, and the resulting weighted average NMR chemical shifts were subsequently compared to the experimental data showing close correspondence (Tables S6-S13). The average errors in the range [1.2, 2.0] ppm with maximum errors in the range [3.1, 4.0] ppm are consistent with the general expectations of DFT prediction methods.[46]



B

```
ml.pipeline.jobmap(
    orca.giao_nmr_ens,
    source=orca_conf_dft,
    destination=orca_conf_nmr,
    cache_dir="_04_dft_nmr",
    kwargs={
        "keywords": "rks pbe0 pcSseg-2
verytightscf nmr cpcm(chloroform)",
        "elements": ("C",),
    },
    n_workers=16,
)
```

Figure 3. (A) CDXML parsing and conformer generation workflow results for cladosporin. (B) Minimal code example for GIAO-DFT NMR chemical shielding calculations.

4 GRID-BASED DESCRIPTORS

4.1 Efficiency Optimization

Grid-based, conformer-averaged (GBCA) indicator field descriptors, such as the average steric occupancy descriptor (ASO) and the average electronic indicator field (AEIF), were useful in the enantioselectivity prediction workflow developed in this laboratory. A naïve implementation of the GBCA descriptors suffers from significant, unfavorable scaling dependencies with respect to the grid size. This step was very computationally expensive to carry out on libraries of tens of thousands of molecules, requiring high performance computational hardware. To eliminate the slow process of descriptor computation, we performed an optimization. Molli employs two levels of optimization of the computing process. The optimization of the GBCA descriptors began by outsourcing numerically intensive arrayed calculations to a more efficient C implementation of the numpy package (Table 2). A 25-40-fold acceleration was observed; however, the processing time was still high for libraries of >1M conformers. Thus, an auxiliary C++ sublibrary (called `molli_xt`) was created through the use of pybind11.[53] Two functions were implemented that reproduced the behavior of SciPy's[54] cdist function that computes the distance matrix (and an analogous function was made that would compute a higher dimensional analog of the distance tensor). This process provided a considerable speed enhancement owing to elimination of slower Python code overhead. In 64-bit floats, molli achieved 1.96±0.05 acceleration by elimination of the extra for-loop in the distance matrix computation. A further 1.34±0.04 fold increase in performance was gained by computing the distance matrix in 32-bit floats, giving a total acceleration of 2.62±0.08. Relative errors in squared Euclidean distance did not exceed 2×10^{-7}, and the resulting ASO mean absolute errors were less than 2×10^{-9} for 99% of the data (see the Supporting Information for details). For a selected small number of samples, this error was larger because of cases wherein grid points were located close to the van der Waals boundary. To further reduce the size of the problem, the grid was pruned to eliminate the points that lie far away from any atoms for which the values could be assigned as zeros (see Table 2 grid sparsity). To enable this process, we employed the SciPy implementation of the *k*-d tree[55,56] data structure. Pruning the grid for ASO computations reduced the grid size by 80 to 90%, therefore providing an average of 5-fold acceleration. Overall, combining these optimizations achieved a 1,700× acceleration of the process compared to a naïve python implementation, and a 50× acceleration as compared to naïve numpy approach.

Table 2: Benchmarking Results of GBCA Descriptor Calculation.[a]

Grid point spacing, Å	**1.5**	**1.0**	**0.7**
Number of grid points	3510	11362	32832
Descriptor vector sparsity (mean ± stdev)	92.0±4.4%	91.6±4.6%	91.5±4.7%
Pruned grid sparsity (mean ± stdev)	86.7±6.5%	86.0±6.7%	85.9±6.8%
Naïve python ASO, s	175.4	580.8	1686.5
Naïve numpy ASO, s	5.0	14.3	67.1
Scipy cdist optimized ASO, s	0.8	2.6	7.3
molli cdist ASO, s	0.5	1.8	4.9
KDTree & molli cdist optimized ASO, s	0.1	0.5	1.2

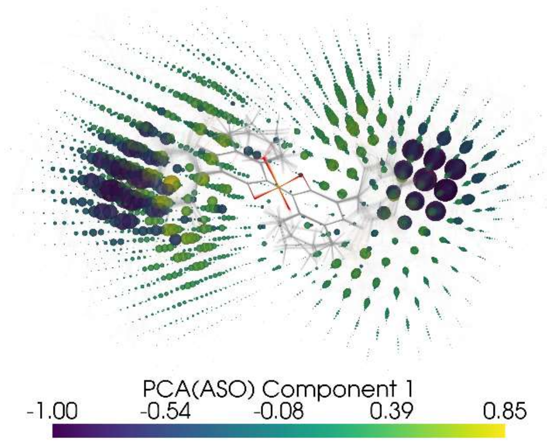
[a] Timings are reported on the BPA catalyst **65_vi** (88 atoms, 215 conformers). Benchmarks reported on system 3 (see the Supporting Information, sections 1.2.1 and 1.1, respectively).

With the optimized GBCA calculation protocol in hand, the benchmark calculations were performed on the binol phosphoric acid (BPA) dataset[23] consisting of 806 entries and a total of 99,680 conformers, as well as on a subset of BOX dataset (Supporting Information, Section 1.2.2)[30] consisting of 72,542 entries and 4,662,551 conformers. The calculations on the BPA dataset could be performed on a laptop computer (system 3) within two minutes. Computing the BOX dataset under identical conditions took ca 1.5 h, which could be sped up considerably by employing more parallel processes on a workstation. Employing 64 processes in parallel, ASO computation for the BOX dataset was complete in under five minutes. This result represents a marked enhancement in speed and enables the calculation of descriptors with chemical resolution (0.75 Å spacing or below).

4.2 Molecule, Ensemble and Descriptor Visualization

By virtue of being a pure Python library, molli can be easily interfaced with a few different visualization libraries. Molli uses two different engines for visualization purposes, the first is 3DMol.js,[57] which is used for simpler molecular renderings inside Jupyter notebooks. This implementation allows a very simple, in-place visualization that helps the end user understand the contents of their molecular or conformer libraries much better without the need to transfer the data to a third-party program for rendering.

The second is the pyvista package which is a convenient set of wrapping functions over the VTK (Visualization ToolKit).[58,59] This engine can be employed for molecular rendering and it performs particularly well for visualizing high-dimensional, grid-based descriptors in context of conformer ensembles (Figure 4). Highly dimensional grid-based descriptors are particularly hard to interpret by a chemist without relying on the visual representation. Figure 4 illustrates the directions of the maximal variance in the ASO and average electronic indicator field (AEIF) descriptors, corresponding to the locations of largest steric and charge distribution diversity in the BPA catalyst library (see also Figures S14–S28).



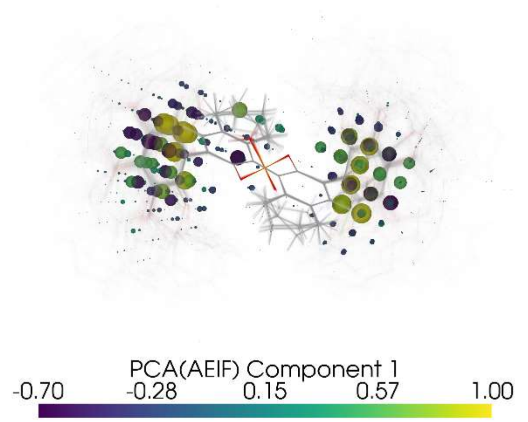


Figure 4. Normalized PCA1 loadings of ASO (left) and AEIF (right) descriptors of the BPA dataset overlayed with the conformer ensemble visualization. A 1.0 Å spacing grid was chosen for the visualization.

5 CONCLUSIONS

Molli comprises a powerful chemoinformatics toolkit that specializes in the creation of large combinatorial libraries of small molecules and parallel computations. A pure Python interface enables a seamless transition between a plain chemical drawing to a large *in silico* molecular dataset with preservation of stereochemical integrity. Combinatorial library creation can be performed with ease through both the command line interface as well as by writing custom scripts. Optimized GBCA descriptor calculations can now easily reproduce the existing ASO and AEIF calculations as well as visualize their corresponding results. Lastly, one can employ the parallelized computational pipeline to compute the properties of isolated molecules and their conformer ensembles with external software; examples of workflows for XTB, CREST, ORCA and NWChem are provided.

6 ASSOCIATED CONTENT

6.1 Data Availability Statement

Source code for the project can be found at https://github.com/SEDenmarkLab/molli. The project is available for quick installation Python package index and conda channels. Up-to-date documentation detailing the installation procedure and package usage examples can be found on the documentation portal, https://molli.readthedocs.io. Datasets and the code for workflows discussed in the present manuscript can be downloaded from the Zenodo repository (https://zenodo.org/records/10719791, doi 10.5281/zenodo.10719790).

6.2 Supporting Information

Description of the hardware, additional information about implementation details, results from the computational pipeline workflows (including atomic coordinates), and plots of PCA components can be found in the attached pdf file.

7 AUTHOR INFORMATION

7.1 Corresponding Authors

*Email: shvedalx@illinois.edu, sdenmark@illinois.edu

7.2 ORCID

Alexander S. Shved: 0000-0001-5979-179X

Blake E. Ocampo: 0000-0002-1987-2576

Elena S. Burlova: 0009-0005-0250-3749

Casey L. Olen: 0000-0002-8621-2973

N. Ian Rinehart: 0000-0002-3106-5208

Scott E. Denmark: 0000-0002-1099-9765

7.3 Notes

The authors declare no competing financial interest.

8 ACKNOWLEDGMENTS

We are grateful to the National Science Foundation for financial support (NSF CHE 2154237) as well as for the Molecule Maker Laboratory Institute (NSF CHE 2019897). We thank the W. M. Keck Foundation for contributing to the purchase of system 4 and system 5, and Merck & Co. for contribution to the purchase of system 1. Blake Ocampo thanks the Alfred P. Sloan Foundation's Minority Ph.D. Program for funding. Alexander Shved, Blake Ocampo, Casey Olen, and Ian Rinehart thank the University of Illinois for graduate fellowships. We thank Sara Lambert and Matthew Berry (UIUC NCSA) for their assistance with the GitHub workflow and insightful discussions. The authors acknowledge Austin Douglas for assistance with establishing the documentation system and Ethan G. M. Mattson for prototyping some of the conformer generation code. We thank Mark Hewitt for his assistance with the cluster computing resources. Finally, we thank Dr. Jeremy J. Henle and Dr. Andrew F. Zahrt for the design of ccheminfolib library that inspired the creation of molli.

9 REFERENCES

(1) Taylor, C. J.; Pomberger, A.; Felton, K. C.; Grainger, R.; Barecka, M.; Chamberlain, T. W.; Bourne, R. A.; Johnson, C. N.; Lapkin, A. A. A Brief Introduction to Chemical Reaction Optimization. *Chem. Rev.* **2023**, *123* (6), 3089–3126. https://doi.org/10.1021/acs.chemrev.2c00798.

(2) Wang, H.; Fu, T.; Du, Y.; Gao, W.; Huang, K.; Liu, Z.; Chandak, P.; Liu, S.; Van Katwyk, P.; Deac, A.; Anandkumar, A.; Bergen, K.; Gomes, C. P.; Ho, S.; Kohli, P.; Lasenby, J.; Leskovec, J.; Liu, T.-Y.; Manrai, A.; Marks, D.; Ramsundar, B.; Song, L.; Sun, J.; Tang, J.; Veličković, P.; Welling, M.; Zhang, L.; Coley, C. W.; Bengio, Y.; Zitnik, M. Scientific Discovery in the Age of Artificial Intelligence. *Nature* **2023**, *620* (7972), 47–60. https://doi.org/10.1038/s41586-023-06221-2.

(3) W. Coley, C.; Jin, W.; Rogers, L.; F. Jamison, T.; S. Jaakkola, T.; H. Green, W.; Barzilay, R.; F. Jensen, K. A Graph-Convolutional Neural Network Model for the Prediction of Chemical Reactivity. *Chemical Science* **2019**, *10* (2), 370–377. https://doi.org/10.1039/C8SC04228D.

(4) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Cent. Sci.* **2018**, *4* (2), 268–276. https://doi.org/10.1021/acscentsci.7b00572.

(5) Gao, H.; Struble, T. J.; Coley, C. W.; Wang, Y.; Green, W. H.; Jensen, K. F. Using Machine Learning To Predict Suitable Conditions for Organic Reactions. *ACS Cent. Sci.* **2018**, *4* (11), 1465–1476. https://doi.org/10.1021/acscentsci.8b00357.

(6) Mennen, S. M.; Alhambra, C.; Allen, C. L.; Barberis, M.; Berritt, S.; Brandt, T. A.; Campbell, A. D.; Castañón, J.; Cherney, A. H.; Christensen, M.; Damon, D. B.; Diego, J. E. D.; García-Cerrada, S.; García-Losada, P.; Haro, R.; Janey, J.; Leitch, D. C.; Li, L.; Liu, F.; Lobben, P. C.; Macmillan, D. W. C.; Magano, J.; McInturff, E.; Monfette, S.; Post, R. J.; Schultz, D.; Sitter, B. J.; Stevens, J. M.; Strambeanu, I. I.; Twilton, J.; Wang, K.; Zajac, M. A. The Evolution of High-Throughput Experimentation in Pharmaceutical Development and Perspectives on the Future. *Organic Process Research and Development* **2019**, *23* (6), 1213–1242. https://doi.org/10.1021/acs.oprd.9b00140.

(7) Krska, S. W.; DiRocco, D. A.; Dreher, S. D.; Shevlin, M. The Evolution of Chemical High-Throughput Experimentation To Address Challenging Problems in Pharmaceutical Synthesis. *Acc. Chem. Res.* **2017**, *50* (12), 2976–2985. https://doi.org/10.1021/acs.accounts.7b00428.

(8) Collins, K. D.; Gensch, T.; Glorius, F. Contemporary Screening Approaches to Reaction Discovery and Development. *Nature Chem* **2014**, *6* (10), 859–871. https://doi.org/10.1038/nchem.2062.

(9) Isbrandt, E. S.; Sullivan, R. J.; Newman, S. G. High Throughput Strategies for the Discovery and Optimization of Catalytic Reactions. *Angewandte Chemie International Edition* **2019**, *58* (22), 7180–7191. https://doi.org/10.1002/anie.201812534.

(10) Shen, Y.; Borowski, J. E.; Hardy, M. A.; Sarpong, R.; Doyle, A. G.; Cernak, T. Automation and Computer-Assisted Planning for Chemical Synthesis. *Nat Rev Methods Primers* **2021**, *1* (1), 1–23. https://doi.org/10.1038/s43586-021-00022-5.
(11) Wigh, D. S.; Goodman, J. M.; Lapkin, A. A. A Review of Molecular Representation in the Age of Machine Learning. *WIREs Computational Molecular Science* **2022**, *12* (5), e1603. https://doi.org/10.1002/wcms.1603.
(12) Gensch, T.; dos Passos Gomes, G.; Friederich, P.; Peters, E.; Gaudin, T.; Pollice, R.; Jorner, K.; Nigam, A.; Lindner-D'Addario, M.; Sigman, M. S.; Aspuru-Guzik, A. A Comprehensive Discovery Platform for Organophosphorus Ligands for Catalysis. *J. Am. Chem. Soc.* **2022**, *144* (3), 1205–1217. https://doi.org/10.1021/jacs.1c09718.
(13) Saldívar-González, F. I.; Huerta-García, C. S.; Medina-Franco, J. L. Chemoinformatics-Based Enumeration of Chemical Libraries: A Tutorial. *J Cheminformatics* **2020**, *12* (1), 64. https://doi.org/10.1186/s13321-020-00466-z.
(14) Bender, A.; C. Glen, R. Molecular Similarity: A Key Technique in Molecular Informatics. *Organic & Biomolecular Chemistry* **2004**, *2* (22), 3204–3218. https://doi.org/10.1039/B409813G.
(15) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.* **2010**, *50* (5), 742–754. https://doi.org/10.1021/ci100050t.
(16) Axen, S. D.; Huang, X.-P.; Cáceres, E. L.; Gendelev, L.; Roth, B. L.; Keiser, M. J. A Simple Representation of Three-Dimensional Molecular Structure. *J. Med. Chem.* **2017**, *60* (17), 7393–7409. https://doi.org/10.1021/acs.jmedchem.7b00696.
(17) Wang, Y.; Hu, J.; Lai, J.; Li, Y.; Jin, H.; Zhang, L.; Zhang, L.-R.; Liu, Z. TF3P: Three-Dimensional Force Fields Fingerprint Learned by Deep Capsular Network. *J. Chem. Inf. Model.* **2020**, *60* (6), 2754–2765. https://doi.org/10.1021/acs.jcim.0c00005.
(18) Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular Graph Convolutions: Moving beyond Fingerprints. *J Comput Aided Mol Des* **2016**, *30* (8), 595–608. https://doi.org/10.1007/s10822-016-9938-8.
(19) Cho, H.; Choi, I. S. Enhanced Deep-Learning Prediction of Molecular Properties via Augmentation of Bond Topology. *ChemMedChem* **2019**, *14* (17), 1604–1609. https://doi.org/10.1002/cmdc.201900458.
(20) Ishida, S.; Miyazaki, T.; Sugaya, Y.; Omachi, S. Graph Neural Networks with Multiple Feature Extraction Paths for Chemical Property Estimation. *Molecules* **2021**, *26* (11), 3125. https://doi.org/10.3390/molecules26113125.
(21) Zahrt, A. F.; Athavale, S. V.; Denmark, S. E. Quantitative Structure–Selectivity Relationships in Enantioselective Catalysis: Past, Present, and Future. *Chem. Rev.* **2020**, *120* (3), 1620–1689. https://doi.org/10.1021/acs.chemrev.9b00425.
(22) Henle, J. J.; Zahrt, A. F.; Rose, B. T.; Darrow, W. T.; Wang, Y.; Denmark, S. E. Development of a Computer-Guided Workflow for Catalyst Optimization. Descriptor Validation, Subset Selection, and Training Set Analysis. *J. Am. Chem. Soc.* **2020**, *142* (26), 11578–11592. https://doi.org/10.1021/jacs.0c04715.
(23) Zahrt, A. F.; Henle, J. J.; Rose, B. T.; Wang, Y.; Darrow, W. T.; Denmark, S. E. Prediction of Higher-Selectivity Catalysts by Computer-Driven Workflow and Machine Learning. *Science* **2019**, *363* (6424), eaau5631. https://doi.org/10.1126/science.aau5631.
(24) Yamaguchi, S. Molecular Field Analysis for Data-Driven Molecular Design in Asymmetric Catalysis. *Organic & Biomolecular Chemistry* **2022**, *20* (31), 6057–6071. https://doi.org/10.1039/D2OB00228K.
(25) Staalduinen, N. van; Bannwarth, C. MolBar: A Molecular Identifier for Inorganic and Organic Molecules with Full Support of Stereoisomerism. ChemRxiv July 1, 2024. https://doi.org/10.26434/chemrxiv-2024-k40v5-v2.

(26) Krenn, M.; Ai, Q.; Barthel, S.; Carson, N.; Frei, A.; Frey, N. C.; Friederich, P.; Gaudin, T.; Gayle, A. A.; Jablonka, K. M.; Lameiro, R. F.; Lemm, D.; Lo, A.; Moosavi, S. M.; Nápoles-Duarte, J. M.; Nigam, A.; Pollice, R.; Rajan, K.; Schatzschneider, U.; Schwaller, P.; Skreta, M.; Smit, B.; Strieth-Kalthoff, F.; Sun, C.; Tom, G.; Falk von Rudorff, G.; Wang, A.; White, A. D.; Young, A.; Yu, R.; Aspuru-Guzik, A. SELFIES and the Future of Molecular String Representations. *Patterns* **2022**, *3* (10), 100588. https://doi.org/10.1016/j.patter.2022.100588.
(27) Cheng, A. H.; Cai, A.; Miret, S.; Malkomes, G.; Phielipp, M.; Aspuru-Guzik, A. Group SELFIES: A Robust Fragment-Based Molecular String Representation. *Digital Discovery* **2023**, *2* (3), 748–758. https://doi.org/10.1039/D3DD00012E.
(28) *Chemaxon Extended SMILES and SMARTS - CXSMILES and CXSMARTS | Chemaxon Docs*. https://docs.chemaxon.com/display/docs/formats_chemaxon-extended-smiles-and-smarts-cxsmiles-and-cxsmarts.md (accessed 2024-07-19).
(29) Brecher, J. Graphical Representation of Stereochemical Configuration (IUPAC Recommendations 2006). *Pure and Applied Chemistry* **2006**, *78* (10), 1897–1970. https://doi.org/10.1351/pac200678101897.
(30) Olen, C. L.; Zahrt, A. F.; Reilly, S. W.; Schultz, D.; Emerson, K.; Candito, D.; Wang, X.; Strotman, N. A.; Denmark, S. E. Chemoinformatic Catalyst Selection Methods for the Optimization of Copper–Bis(Oxazoline)-Mediated, Asymmetric, Vinylogous Mukaiyama Aldol Reactions. *ACS Catal.* **2024**, 2642–2655. https://doi.org/10.1021/acscatal.3c05903.
(31) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: A Benchmark for Molecular Machine Learning. *Chem. Sci.* **2018**, *9* (2), 513–530. https://doi.org/10.1039/C7SC02664A.
(32) Axelrod, S.; Gómez-Bombarelli, R. GEOM, Energy-Annotated Molecular Conformations for Property Prediction and Molecular Generation. *Sci Data* **2022**, *9* (1), 185. https://doi.org/10.1038/s41597-022-01288-4.
(33) Alegre-Requena, J. V.; Sowndarya S. V., S.; Pérez-Soto, R.; Alturaifi, T. M.; Paton, R. S. AQME: Automated Quantum Mechanical Environments for Researchers and Educators. *WIREs Computational Molecular Science* **2023**, *13* (5), e1663. https://doi.org/10.1002/wcms.1663.
(34) Greg Landrum; Paolo Tosco; Brian Kelley; Ric; David Cosgrove; sriniker; Riccardo Vianello; gedeck; NadineSchneider; Gareth Jones; Eisuke Kawashima; Dan Nealschneider; Andrew Dalke; Brian Cole; Matt Swain; Samo Turk; Aleksandr Savelev; Alain Vaucher; Maciej Wójcikowski; Ichiru Take; Vincent F. Scalfani; Daniel Probst; Kazuya Ujihara; Rachel Walker; guillaume godin; Axel Pahl; Juuso Lehtivarjo; Francois Berenger; strets123; jasondbiggs. Rdkit/Rdkit: 2023_09_6 (Q3 2023) Release, 2024. https://doi.org/10.5281/ZENODO.591637.
(35) *RDKit*. RDKit: Open-source cheminformatics. https://rdkit.org/ (accessed 2023-10-27).
(36) 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. *Journal of Chemical Theory and Computation* **2019**, *15* (3), 1652–1671. https://doi.org/10.1021/acs.jctc.8b01176.
(37) 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* (5), 2847–2862. https://doi.org/10.1021/acs.jctc.9b00143.
(38) Aprà, E.; Bylaska, E. J.; De Jong, W. A.; Govind, N.; Kowalski, K.; Straatsma, T. P.; Valiev, M.; Van Dam, H. J. J.; Alexeev, Y.; Anchell, J.; Anisimov, V.; Aquino, F. W.; Atta-Fynn, R.; Autschbach, J.; Bauman, N. P.; Becca, J. C.; Bernholdt, D. E.; Bhaskaran-Nair, K.; Bogatko, S.; Borowski, P.; Boschen, J.; Brabec, J.; Bruner, A.; Cauët, E.; Chen, Y.; Chuev, G. N.; Cramer, C. J.; Daily, J.; Deegan, M. J. O.; Dunning, T. H.; Dupuis, M.;

Dyall, K. G.; Fann, G. I.; Fischer, S. A.; Fonari, A.; Früchtl, H.; Gagliardi, L.; Garza, J.; Gawande, N.; Ghosh, S.; Glaesemann, K.; Götz, A. W.; Hammond, J.; Helms, V.; Hermes, E. D.; Hirao, K.; Hirata, S.; Jacquelin, M.; Jensen, L.; Johnson, B. G.; Jónsson, H.; Kendall, R. A.; Klemm, M.; Kobayashi, R.; Konkov, V.; Krishnamoorthy, S.; Krishnan, M.; Lin, Z.; Lins, R. D.; Littlefield, R. J.; Logsdail, A. J.; Lopata, K.; Ma, W.; Marenich, A. V.; Martin Del Campo, J.; Mejia-Rodriguez, D.; Moore, J. E.; Mullin, J. M.; Nakajima, T.; Nascimento, D. R.; Nichols, J. A.; Nichols, P. J.; Nieplocha, J.; Otero-de-la-Roza, A.; Palmer, B.; Panyala, A.; Pirojsirikul, T.; Peng, B.; Peverati, R.; Pittner, J.; Pollack, L.; Richard, R. M.; Sadayappan, P.; Schatz, G. C.; Shelton, W. A.; Silverstein, D. W.; Smith, D. M. A.; Soares, T. A.; Song, D.; Swart, M.; Taylor, H. L.; Thomas, G. S.; Tipparaju, V.; Truhlar, D. G.; Tsemekhman, K.; Van Voorhis, T.; Vázquez-Mayagoitia, Á.; Verma, P.; Villa, O.; Vishnu, A.; Vogiatzis, K. D.; Wang, D.; Weare, J. H.; Williamson, M. J.; Windus, T. L.; Woliński, K.; Wong, A. T.; Wu, Q.; Yang, C.; Yu, Q.; Zacharias, M.; Zhang, Z.; Zhao, Y.; Harrison, R. J. NWChem: Past, Present, and Future. *The Journal of Chemical Physics* **2020**, *152* (18), 184102. https://doi.org/10.1063/5.0004997.

(39) Neese, F. Software Update: The ORCA Program System—Version 5.0. *WIREs Computational Molecular Science* **2022**, *12* (5), e1606. https://doi.org/10.1002/wcms.1606.

(40) LaPlante, S. R.; Fader, L. D.; Fandrick, K. R.; Fandrick, D. R.; Hucke, O.; Kemper, R.; Miller, S. P. F.; Edwards, P. J. Assessing Atropisomer Axial Chirality in Drug Discovery and Development. *J. Med. Chem.* **2011**, *54* (20), 7005–7022. https://doi.org/10.1021/jm200584g.

(41) Basilaia, M.; Chen, M. H.; Secka, J.; Gustafson, J. L. Atropisomerism in the Pharmaceutically Relevant Realm. *Acc. Chem. Res.* **2022**, *55* (20), 2904–2919. https://doi.org/10.1021/acs.accounts.2c00500.

(42) *Discovery of a Covalent Inhibitor of KRASG12C (AMG 510) for the Treatment of Solid Tumors | Journal of Medicinal Chemistry*. https://pubs.acs.org/doi/10.1021/acs.jmedchem.9b01180 (accessed 2024-02-17).

(43) *Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94 - Halgren - 1996 - Journal of Computational Chemistry - Wiley Online Library*. https://onlinelibrary.wiley.com/doi/10.1002/(SICI)1096-987X(199604)17:5/6%3C490::AID-JCC1%3E3.0.CO;2-P (accessed 2024-02-17).

(44) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An Open Chemical Toolbox. *Journal of Cheminformatics* **2011**, *3* (1), 33. https://doi.org/10.1186/1758-2946-3-33.

(45) Beaver, M. G.; Brown, D. B.; Campbell, K.; Fang, Y.-Q.; Ford, D. D.; Mardirossian, N.; Nagy, K. D.; Rötheli, A. R.; Sheeran, J. W.; Telmesani, R.; Parsons, A. T. Axial Chirality in the Sotorasib Drug Substance, Part 2: Leveraging a High-Temperature Thermal Racemization to Recycle the Classical Resolution Waste Stream. *Org. Process Res. Dev.* **2022**, *26* (9), 2636–2645. https://doi.org/10.1021/acs.oprd.2c00177.

(46) Lodewyk, M. W.; Siebert, M. R.; Tantillo, D. J. Computational Prediction of 1 H and 13 C Chemical Shifts: A Useful Tool for Natural Product, Mechanistic, and Synthetic Organic Chemistry. *Chemical Reviews* **2012**, *112* (3), 1839–1862. https://doi.org/10.1021/cr200106v.

(47) Kutateladze, A. G.; Reddy, D. S. High-Throughput in Silico Structure Validation and Revision of Halogenated Natural Products Is Enabled by Parametric Corrections to DFT-Computed 13C NMR Chemical Shifts and Spin–Spin Coupling Constants. *Journal of Organic Chemistry* **2017**, *82* (7), 3368–3381. https://doi.org/10.1021/ACS.JOC.7B00188.

(48) *Efficient Quantum Chemical Calculation of Structure Ensembles and Free Energies for Nonrigid Molecules | The Journal of Physical Chemistry A*. https://pubs.acs.org/doi/10.1021/acs.jpca.1c00971 (accessed 2024-02-17).

(49) Pracht, P.; Grimme, S. Calculation of Absolute Molecular Entropies and Heat Capacities Made Simple. *Chem. Sci.* **2021**, *12* (19), 6551–6568. https://doi.org/10.1039/D1SC00621E.
(50) Jensen, F. Segmented Contracted Basis Sets Optimized for Nuclear Magnetic Shielding. *Journal of Chemical Theory and Computation* **2015**, *11* (1), 132–138. https://doi.org/10.1021/ct5009526.
(51) Boyko, Y. D.; Huck, C. J.; Ning, S.; Shved, A. S.; Yang, C.; Chu, T.; Tonogai, E. J.; Hergenrother, P. J.; Sarlah, D. Synthetic Studies on Selective, Proapoptotic Isomalabaricane Triterpenoids Aided by Computational Techniques. *Journal of the American Chemical Society* **2021**, *143* (4), 2138–2155. https://doi.org/10.1021/jacs.0c12569.
(52) Stoychev, G. L.; Auer, A. A.; Neese, F. Efficient and Accurate Prediction of Nuclear Magnetic Resonance Shielding Tensors with Double-Hybrid Density Functional Theory. *Journal of Chemical Theory and Computation* **2018**, *14* (9), 4756–4771. https://doi.org/10.1021/acs.jctc.8b00624.
(53) Pybind/Pybind11, 2023. https://github.com/pybind/pybind11 (accessed 2023-10-28).
(54) Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; van der Walt, S. J.; Brett, M.; Wilson, J.; Millman, K. J.; Mayorov, N.; Nelson, A. R. J.; Jones, E.; Kern, R.; Larson, E.; Carey, C. J.; Polat, İ.; Feng, Y.; Moore, E. W.; VanderPlas, J.; Laxalde, D.; Perktold, J.; Cimrman, R.; Henriksen, I.; Quintero, E. A.; Harris, C. R.; Archibald, A. M.; Ribeiro, A. H.; Pedregosa, F.; van Mulbregt, P.; SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **2020**, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2.
(55) Bentley, J. L. Multidimensional Binary Search Trees Used for Associative Searching. *Commun. ACM* **1975**, *18* (9), 509–517. https://doi.org/10.1145/361002.361007.
(56) *scipy.spatial.KDTree — SciPy v1.11.4 Manual*. https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.KDTree.html (accessed 2023-12-22).
(57) Rego, N.; Koes, D. 3Dmol.Js: Molecular Visualization with WebGL. *Bioinformatics* **2015**, *31* (8), 1322–1324. https://doi.org/10.1093/bioinformatics/btu829.
(58) Sullivan, C. B.; Kaszynski, A. A. PyVista: 3D Plotting and Mesh Analysis through a Streamlined Interface for the Visualization Toolkit (VTK). *Journal of Open Source Software* **2019**, *4* (37), 1450. https://doi.org/10.21105/joss.01450.
(59) Schroeder, W.; Martin, K.; Lorensen, B. *The Visualization Toolkit (4th Ed.)*; Kitware, 2006.

TOC Graphic