Parallelized tools for the preparation and curation of large libraries for virtual screening

Andreas Vitalis*, David Parker, Fabian Radler, Pablo A. Vargas-Rosales, Yang Zhang, Jean-Rémy Marchand,[1] and Amedeo Caflisch*

Department of Biochemistry, University of Zurich, Winterthurerstr. 190, 8057 Zurich, Switzerland

*E-mail: a.vitalis@bioc.uzh.ch , caflisch@bioc.uzh.ch

Abstract

We introduce a software package, ParaLig, that provides solutions for several workflows occurring repeatedly in computational drug discovery: parameterization of small molecules with partial charges and free energies of solvation, generation of conformers, virtual chemical reactions, creating combinatorial libraries, and molecule editing tasks. Throughout, we emphasize the maintenance/creation of 3D coordinates and better interoperability. The latter is achieved by stable embedding of meta-information into output files and by improving the mutual compatibility of molecular representations (*e.g.*, aromaticity perception). ParaLig wraps around core functionalities provided by a variety of existing software: the two popular cheminformatics packages OpenBabel and RDKit, and the general force field providers CGenFF and AmberTools. Our workflows emphasize scalable bulk processing of large libraries of molecules, and we provide an MPI-based wrapper to simplify deployment to high-performance computing resources. Along with salient descriptions of the methods we benchmark performance both in terms of throughput and in terms of the quality of some of the results.

[1] Current address: *Novartis Institute for Biomedical Research Informatics, Fabrikstrasse 2, Novartis Campus, 4056 Basel, Switzerland*

Introduction

Cheminformatics deals with applying computational tools to process, modify, or make predictions for molecules.[1] While the umbrella term is widely applicable, most often it refers to applications focused on organic small molecules in a biomedical or pharmaceutical context.[2] The applications frequently require predictions of physicochemical properties, for example, solubility, pKa values, or binding free energies.[3] Many simplified heuristics have been in place in the field for decades, such as scoring functions[4] or filters for drug-like properties,[5-6] and their deficiencies are well-known.[7] Simplifications are necessary because cheminformatics is often applied to large libraries of molecules.[8-10] Machine learning applications have, inspired by early work to replace calculations at the quantum level, been deployed to predict many different properties of molecules[11-12] and to create molecules *de novo*.[13-14] It must be noted, however, that experimental data on small molecules are relatively sparse, limiting the power of machine learning. The most striking example are so-called on-demand molecules which have never even been synthesized let alone measured.[9, 15-17] This caveat on data availability is the preeminent concern for contextual properties, such as binding affinities, because we lack (affordable) *ab initio* predictions of these properties.[18]

Irrespective of application and methodology, representation is of universal importance. String-based representations, such as SMILES[19] or InChI,[20] are used very frequently.[21] They encode a graph of covalent bonds and basic element and hybridization information but might be insufficient to explicitly cover all aspects of stereochemistry and tautomerism. They also do not encode spatial information, neither covalently, in terms of bond lengths, angles, etc., nor absolute, as a so-called conformer. Complete file formats encoding a complete graph and explicit protonation states and isomers include the Tripos mol2 or Structured Data Files (sdf), and these are most commonly chosen when using and manipulating molecules in applications like structure-based drug discovery.[22] Seemingly trivial tasks like matching a specific molecule across representations are surprisingly error-prone in practice.[23] Common procedures affecting this are canonicalization,[19, 24] aromaticity perception,[25] or protonation.[26-27] This means that different software packages cannot automatically interact seamlessly, even if they support the same file formats for input and output.

Most cheminformatics tasks ultimately deal with physicochemical properties. Unless a methodology is purely data-driven or truly *ab initio*, atom-based or sometimes group-based parameters will be required. These can include partial charges, Lennard-Jones radii, torsional potentials, solvation-related parameters such as hydropathy, *etc*. There are few centralized hubs to compute such parameters.[28] The closest are so-called general force fields, which include earlier models such as the universal force field (UFF)[29] or Merck's MMFF94.[30] In the context of classical, molecular dynamics simulations, the GAFF[31] and CGenFF[32] force fields stand out for the seamless integration with their parent force fields (AMBER and CHARMM, respectively). Along with the curation of molecules in terms of stereochemistry or protonation, parameters play the largest role in determining the systematic success, or lack thereof, of empirical methodologies. For both curation and parameterization, which are somewhat overlapping tasks, a particular problem is encountered in dealing with very large libraries: systematic errors are very likely to lead to the affected molecules being enriched at either end of a results spectrum, for example in high-throughput virtual screening.[15] It is very tedious to curate large libraries of molecules,[33-34] which is why resources like Zinc[9] tend to be more valuable in cheminformatics than, *e.g.*, vendor catalogs of molecules as string representations. A more nuanced view emerges if purchasability and price are the primary criteria for downstream applications. Here, the best-case

scenario is an embedding of metadata information, *e.g.*, by strongly associating different identifiers used across different platforms in a relational database. In our experience, this information is unfortunately tedious to obtain and curate, which poses considerable difficulties for automation of these tasks. The root cause of systematic errors affecting parameter- or curation-dependent applications is the same throughout: the diversity of chemical space means that whether a molecule is represented correctly and unequivocally at a given pH in water is not generally knowable, even by experts. It is not surprising that many hybrid approaches consult data to aid the curation, for example, crystal structure information of matching substructures.[35]

Here, we present a repository of tools named ParaLig that provides a mix of original and wrapper functionality. It deals with software interoperability and the stringent embedding of meta-information, such as parameters, in mol2 and sdf formats. It facilitates the generation and embedding of partial charges and type information using standalone tools and gives access to the ABSINTH GFF.[36] The tool wraps around native RDKit[37] functionality to offer conformer generation and virtual chemistry, much of which is an original implementation focused on robustness and maintaining 3D coordinates as much as possible. All tasks are smoothly joined in a single invocation, as per user request. We provide an explicit software solution implementing a master-worker scheme to provide HPC-ready parallelizability with built-in load balancing. For convenience, a second-level Python wrapper is offered to provide a standardized command-line interface with expressively named arguments.

Methods

Software requirements

With the exception of OpenBabel,[38] the Python tool has no stringent requirements on exact versions of the required software packages, CGenFF[32], AmberTools[39] (to provide Antechamber), and RDKit,[37] and we have used different versions in conjunction with it. The GFF tools are passed directly as paths to the callable executables, and resolving their dependencies on the host system(s) is sufficient. RDKit and Pybel[40] are imported, with the former providing all of the library-level functionality except for input file parsing and partial charge generation. Setting these up is thus a matter of installing the packages and managing the Python environment. The OpenBabel version we require is tied to minimal modifications we made to its thread safety, and we offer a fork of version 3.1.1 for this purpose (https://github.com/miemiemmmm/openbabel).

The HPC wrapper relies on Fortran MPI code: this ensures that the resource management on supercomputers is as flexible as it can be, and no abstraction framework in Python is required. It relies on the ForPy module https://github.com/ylikx/forpy) to provide the Fortran-Python interface and on libpqxx (https://pqxx.org/development/libpqxx) to interface directly with a PostgreSQL database.

File formats and compatibility

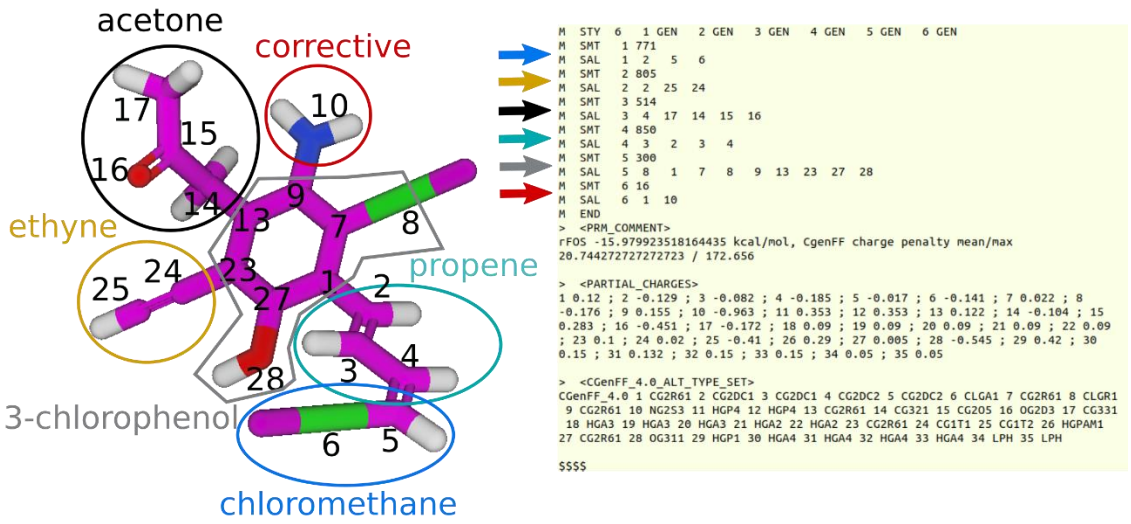
Small molecules are commonly represented in different human-readable files: pdb files, such as ligand blocks excised from structures deposited at the RCSB, are converted with the help of OpenBabel's built-in pdb parser. No pdb output option is provided since the format is ill-suited and contains several fields of no relevance for the task at hand. The Tripos mol2 format is a common standard in the field: it encodes the molecular graph including bond orders, has no intrinsic precision or field-width limits, and contains a standardized set of atomic types, Sybyl types, that can be useful in perception and parameterization tasks. It also encodes some mandatory meta-information along with various optional

sections, such as vectors of alternative types. This is, for example, how the software SEED[41] sets atom types. The main alternative is a variant of Structural Data Files (sdf), which is an evolution of the "molfile" standard to allow collections of molecules to be stored in a single file. These two types are quite similar in scope, but often disfavored for distinct reasons: mol2-files encode bond orders including aromaticity, but there is no universal way of computing aromaticity. Similarly, the extra sections and even some of the standard fields are poorly standardized (as of today) and not consistently understood by parsers. Conversely, sdf-files are less flexible: they contain restrictions on field widths, do not encode aromaticity at all, and have no standard field for storing partial charges. Our tool uses OpenBabel parsers to set up a code-internal object, irrespective of input format. As part of this object, RDKit attempts to read the molecule, either in sdf (mol) format (when given as pdb or sdf) or in mol2 format. The processed RDKit molecule, which is a binary object, is passed to all functionalities that rely on RDKit library functions, but not required for operations that do not.

As alluded to, this step is made complicated by frequent compatibility issues between mol2-input and RDKit's internal standards (mostly related to aromaticity and specific Sybyl types). Any mol2 block obtained from file input is thus filtered through a custom routine that solves the majority of these issues so that the molecules are read, as intended, by RDKit. It is very important in structural workflows that there are no needless conversions that jeopardize the 3D coordinates and protonation states. Because of this, we did not and do not deem it acceptable to perform any conversions that involve coordinate-free representations as intermediates (such as SMILES). A comparable issue is encountered with CGenFF, which, for all halogens except fluorine, adds dummy particles as part of its parameterization (since version 3).[42] These are not handled well as input in general, not even by CGenFF itself, and we implemented a protocol to dynamically remove and re-add them as part of parsing and writing operations, respectively. All output files written by the tool rely on custom write routines to make sure they contain consistent (independent of the input format) and persistent (preserved upon rereading) information.

Lastly, the SQL standard for small molecules is supported by the tool as an output option but not part of this manuscript. It will be described in full elsewhere.

Molecule parameterization




```
M  STY  6   1 GEN   2 GEN   3 GEN   4 GEN   5 GEN   6 GEN
M  SMT   1 771
M  SAL   1  2   5   6
M  SMT   2 805
M  SAL   2  2  25  24
M  SMT   3 514
M  SAL   3  4  17  14  15  16
M  SMT   4 850
M  SAL   4  3   2   3   4
M  SMT   5 300
M  SAL   5  8   1   7   8   9  13  23  27  28
M  SMT   6 16
M  SAL   6  1  10
M  END
>  <PRM_COMMENT>
rFOS -15.979923518164435 kcal/mol, CgenFF charge penalty mean/max
20.744272727272723 / 172.656

>  <PARTIAL_CHARGES>
1 0.12 ; 2 -0.129 ; 3 -0.082 ; 4 -0.185 ; 5 -0.017 ; 6 -0.141 ; 7 0.022 ; 8
-0.176 ; 9 0.155 ; 10 -0.963 ; 11 0.353 ; 12 0.353 ; 13 0.122 ; 14 -0.104 ; 15
0.283 ; 16 -0.451 ; 17 -0.172 ; 18 0.09 ; 19 0.09 ; 20 0.09 ; 21 0.09 ; 22 0.09
; 23 0.1 ; 24 0.02 ; 25 -0.41 ; 26 0.29 ; 27 0.005 ; 28 -0.545 ; 29 0.42 ; 30
0.15 ; 31 0.132 ; 32 0.15 ; 33 0.15 ; 34 0.05 ; 35 0.05

>  <CGenFF_4.0_ALT_TYPE_SET>
CGenFF_4.0 1 CG2R61 2 CG2DC1 3 CG2DC1 4 CG2DC2 5 CG2DC2 6 CLGA1 7 CG2R61 8 CLGR1
 9 CG2R61 10 NG2S3 11 HGP4 12 HGP4 13 CG2R61 14 CG321 15 CG2O5 16 OG2D3 17 CG331
 18 HGA3 19 HGA3 20 HGA3 21 HGA2 22 HGA2 23 CG2R61 24 CG1T1 25 CG1T2 26 HGPAM1
27 CG2R61 28 OG311 29 HGP1 30 HGA4 31 HGA4 32 HGA4 33 HGA4 34 LPH 35 LPH

$$$$
```

Figure 1: **Extensions of the sdf file format to accommodate meta-information**. **Left**. An example (virtual) molecule is shown in 3D that was generated using the tool itself. Heavy atoms carry index labels (starting at 1). A group decomposition is indicated by the continuous enveloping lines and corresponding text labels. **Right**. The corresponding section of the sdf file. Partial charges and types are defined as list-like fields. Here, we show the nonstandard version but, by default, ParaLig will print RDKit-compliant versions of these two sections (see text). Substructure definitions for solvation group decomposition are given as suggested by the sdf standard: for example, the largest substructure is 3-chlorophenol where the code ("300") refers to a corresponding database. A comment is added to document the lump free energy of solvation (sum of all group values) and indicate the origin of the charges. The extra particles on the chlorine atoms are CGenFF-specific lone pair particles (see CGenFF type column, "LPH").

Tripos mol2 files not only contain a field, per atom, to hold that atom's partial charge, but also a metadata line specifying their origin. In contrast, sdf files only have space for net (integer) charge annotations. RDKit supports an extension of sdf files in the block of non-structural data to read and maintain atom-based annotations of different data types, such as "atom.dprop.PartialCharge." We support this standard, also used by the OpenFF[43] toolkit, to write both type and partial charge information per atom. For internal use, we alternatively support a similar extension that makes it easier to deal with missing values, and this version is shown in Fig. 1. Both versions support the carrying over of "ALT_TYPE" information found in mol2 files across conversion between the two formats.

The partial charges are obtained in different formats, and our tool does not interfere with this process, except converting the input, as described in the previous section, to a homogenized standard. When feeding inputs to either CGenFF or, most prominently, Antechamber, it can be mandatory to remove some auxiliary sections of the input mol2-strings, and this is handled automatically whenever necessary. Importantly, the charges are not regularized, which implies that for some models they might not add up to integer values exactly, or that different models predict different net charge values for the same molecule (see Results). For downstream usability, the tool adds corresponding "ALT_TYPE" sections to the output files. For example, if a ligand is used in SEED or the intent is to perform molecular dynamics simulations, these types will be required to set additional parameters, such as Lennard-Jones or bonded potential parameters.

In Marchand *et al*.,[36] we introduced a generalization of the ABSINTH implicit solvent model and force field[44] for small molecules. This extension relies on a careful decomposition of arbitrary molecules into building blocks, as described previously, and we provide the reference implementation here. The algorithm has been reimplemented to be independent of bulky Python packages like networkx and now relies on a manual implementation of the Bron-Kerbosch algorithm[45] to recursively find possible decompositions given a database of fragments and corrective values, each with a measured or inferred reference free energy of solvation. For example, the molecule in Fig. 1 has been decomposed into five small molecule groups plus one corrective value (the NH_2), and this is annotated in the substructure information. The same data are encoded, per atom, in the dedicated fields that Tripos mol2 files offer for this purpose. This parameterization can also be used to predict free energies of solvation or detect inconsistencies with partial charges (see Results).

Free (unrestrained) conformer generation

The task of conformer generation is to propose plausible arrangements of the atoms in a molecule, specified by its bond graph, in 3D space.[46] This requires respecting covalent geometries, primarily bond lengths, angles, and planarity, as well as noncovalent interactions, predominantly excluded volume.

The procedure we employ has two steps: in the first, an initial random attempt or embedding is deduced that satisfies the bond graph geometrically. In the second, all candidate solutions are refined individually, if possible, using a simplified force field, with RDKit specifically offering support for UFF and MMFF. For unrestrained conformer generation, the workflow is largely identical to the one proposed in Riniker *et al*.[35] We expose some relevant parameters to the user: the target number of conformers can either be dealt with rigidly or heuristically (to avoid forcing the generation of redundant conformers for quasi-rigid molecules), and thresholds for pruning based on clustering can be set. These techniques are not always sufficient to avoid redundancy due to the frequent occurrence of symmetries in small molecules. The aforementioned heuristic works by interpolating the upper bound between the target request or, if it is significantly smaller, a value proportional to 3^R. Here, R is the number of rotatable bonds, and the minimum value the heuristic can attain is 1 when R=0. Unrestrained conformer generation uses only UFF at the moment. We do not evaluate the exhaustiveness of the procedure here.

Virtual chemistry

The string-based implementation of chemical reactions for, *e.g*., the creation of combinatorial libraries is well-established.[47] The SMARTS language[48] is commonly used to explicitly encode chemical transformations, which requires that the molecules are expressible and that the reactive patterns can be safely matched. The second aspect is often somewhat imperfect: while negations, wildcards, *etc*. are all part of the SMARTS standard, it remains tricky to avoid both false positives and false negatives when relying on string matching to identify reactive patterns. This concern is predominantly relevant when this type of virtual chemistry is meant to predict synthesizable molecules, although for this, additional factors almost always come into play (competing sites, (de)activating groups, *etc*). These issues are not addressed by our tool beyond what is encoded in the reaction SMARTS we provide (see Table I).

Instead, we are concerned with the propagation of spatial information through the workflow. In both ChemAxon's reactor and in RDKit, reactions require taking a detour through a string-based representation, *i.e*., coordinates are lost. However, in structure-based drug design, it is of interest to maintain this information when developing a hit compound further. This is made complicated by the fact that the preserved part is i) variable, depending on the reaction; ii) can be intrinsically flexible; iii) might require subtle geometry changes (for example, when a reaction reduces a double bond). Thus, we have devised and implemented a restrained conformer generation scheme to specifically address the problem that the new molecule should be generated in a way such that the coordinates of the part that is equivalent to the source molecule are maximally preserved.

The virtual chemistry workflow requires five major steps. First, the site of interest on the root (source) molecule must be tagged at the input level. This comes either from an explicit user selection or from parsing the molecule. A basic SMARTS representation of the intended reaction needs to be supplied. Both pieces of information go into a simple input file. For explicit chemical reactions, a library of reaction partners is needed as well. Second, at the code level, the reaction sites must be labeled, and, within RDKit, we use both chemical element masking and isotope labeling to do this. This is so the original fragment can be identified in the product molecules, which, generally speaking, do not offer any universal way of mapping back substructures. For example, if an ether is formed through coupling with phenol, any reactant containing one or more phenoxy substituents will create a product that is

now ambiguous in terms of this mapping. The tagging ensures site-specific, fully trackable reactions. Third, the reaction is carried out. The reaction SMARTS might be edited preemptively to deal with changes in protonation states. There can be more than one product if there is a partner molecule containing multiple sites matched by the SMARTS. Fourth, the SMILES of the product molecules are back-transformed into RDKit molecules and sanity-checked. This implies that they are valid but also that some of their properties (protonation, aromaticity perception in particular) might be volatile. We try to keep as much of this information as possible, but, clearly, this must be done with care as reactions can in fact alter hybridization states of atoms that are present both before and after. Fifth, the candidate molecules, one by one, are subject to a restrained conformer generation procedure, which ensures that the absolute coordinates of the substructure corresponding to the remaining part of the root molecule are preserved. This is explained in detail in the next section.

We use this workflow for two primary purposes. The first is to perform one of a number of explicit chemical reactions in conjunction with a library of reactant molecules. Most of the reactions available in the initial release are listed in Table I, and they include common synthesis reactions used in organic, particularly medicinal chemistry. The list can easily be extended by users since every reaction is stored in its own file. The target application is, naturally, to be able to make predictions for exploring new chemical space while maintaining a prediction of synthesizability.[49] Furthermore, the choice of reactant library can directly control cost or even be matched to an in-house library of molecules in storage.

Name	Functional groups in educts	Product
Buchwald-Hartwig[50-51]	Aryl halides with 1° or 2° amines	Aryl amine
Suzuki[52]	Non-alkyl halides with boronic acids	R-C-C-R’
Negishi (indirect)[53]	Non-alkyl halides with organohalides	R-C-C-R’
Williamson ether[54]	1° alkyl halides with alcohols	Ether
Reductive amination (ketones)[55]	Ketones with 1° or 2° amines	Amine
Amide condensation[56-57]	Carboxylates with 1° or 2° amines	Amide
Ester formation [58]	Carboxylates with alcohols	Ester
Reductive amination (aldehydes)[55]	Aldehydes with 1° or 2° amines	Amine
Amine sulfonation	Sulfonates with 1° or 2° amines	Sulfonamide
Diels-Alder cycloaddition	Conjugated diene with alkene	Cyclic alkene
Alkyne coupling (Eglinton) [59]	Term. alkyne with term. alkyne	R-C≡C-C≡C-R’

Table I. **SMARTS reactions**. Many reactions use educts generated *in situ*, and sometimes it is more useful to consider a preliminary educt. Here, for example, for Negishi coupling, the creation of the organozinc compound is presumed to proceed through a halide itself. For the Williamson reaction, the activation of the alcohol is assumed possible. Some specificity is in place: halogen leaving groups normally exclude fluorine, the alkene in Diels-Alder must be activated, amines exclude some delocalized cases such as guanidino groups, *etc*. Similarly, generalizations are sometimes available: carboxylates can be acyl chlorides, triflate leaving groups are included for Suzuki, *etc*. That said, it is not a goal of the software to assess whether the synthetic pathways would need

further modifications (protecting and activating groups) in practice. One particular difficulty are conformational requirements, here that the diene for Diels-Alder cannot be in *trans*, and we use a custom encoding for this.

The second purpose is to explore chemical space without explicit considerations for synthesizability. In hit-to-lead optimization, it is fairly common to consider simple "decorations" of an initial hit to optimize its properties.[60] In this process, researchers might want to iterate a certain number of sites of interest with a certain number of spatially small modifications.[61-62] This is often done, synthetically speaking, by changing the source materials rather than the reaction itself. Such combinatorial enumerations are not well-suited to be represented as explicit, bimolecular chemical reactions. Instead, our tool allows the user to give a list of sites and a list of SMILES/SMARTS, along with a depth setting, to quickly produce such libraries, again ensuring that the coordinates of the unmodified part stay intact. The depth is for enumerating also doubly or higher-substituted versions, and the size of the library is $\sum_{d=1}^{D}\binom{S}{d}G^d$, where *S* is the number of sites, *G* is the number of modifications, and *D* is the depth. For ease of use, many common modifications can also be encoded by proxy strings such as "Methyl" or "Alcohol."

Restrained conformer generation

Because the virtual chemistry procedures described above have to pass through a stage in which molecules are represented as strings only, it is surprisingly challenging to maintain a strict mapping between atoms from reactants to products. Substructures can be ambiguous and expensive to search and match for large molecules, which means that the reconstruction of such a map is fraught with errors and needless cost. Instead, we use a combination of labels, both isotopes and elements, afforded in RDKit, to intermittently tag atoms and maintain their identities throughout the workflow.

With these labels in place, the first step of restrained conformer generation is the straightforward and explicit identification of the relevant substructure in the product molecule. If this fails or is unavailable, substructure search and matching is implemented as a fallback solution, which is slow for large molecules. With mappable atoms and coordinates in place, RDKit offers a facility to perform constrained embeddings (AllChem.EmbedMolecule). However, we found that the function offers too little control and thus reimplemented it using the same, more basic RDKit functionalities as follows. The heavy atoms in the mapped set are used for measuring coordinate deviations. We first customize an RDKit force field (UFF or MMFF) to include distance restraints: specifically, the full list of unique, interatomic distances of the reference substructure is used and applied to the product. A similar procedure adds absolute position restraints (10 times stronger), which are implementable with the help of fixed dummy particles. To avoid an overly rugged potential energy surface, restraints are buffered by 0.1Å (flat-bottom).

With this setup in place, we run through a fixed number of attempts to generate a conformer with a geometrically similar substructure. Both the number of attempts and the threshold for "similar" are parameters. In each attempt, we first construct a "blind" embedding of the molecule (up to 1000 tries), using standard RDKit methodology and unaware of the customized FF. If this is unsuccessful, an error is returned immediately. If it is successful, the conformer is checked for unusual bond angles (the check is designed to catch occasional failures in constructing groups like R-COO^- or R-NO_2 where two terminal atoms overlap). If the check fails, the next attempt begins; if not, the conformer is aligned, based on

the matched substructure, to the root molecule. For a rigid substructure this would be enough. Since this is not generally the case, in the second step of every attempt, the conformer is relaxed in the customized force field, using RDKit's optimization routine (AllChem.optimizeMoleculeConfs, 1000 steps maximum, one thread), before a final alignment is performed. If such an attempt yields a conformer meeting the similarity threshold for the matched substructure and passes the aforementioned sanity check, the procedure returns the relevant information and completes. If not, it is repeated with the similarity threshold doubled. If this still fails, free conformer generation (see above) is activated as a fallback action. We emphasize that our derivatized function is only rewriting/adding the components necessary for the extra steps outlined above, but generally defers to RDKit.

Other editing tasks

Aside from the virtual chemistry tasks outlined above, the tool allows additional simple editing mechanisms, always with a focus on the preservation of 3D coordinates. First, the indices of a subset of heavy (non-terminal) atoms can be provided to have the molecule be truncated by this subset. This will cause an error if it splits the molecule but it does allow opening rings or removing large fractions of a molecule. Second, atoms can be substituted, which is encoded as a SMARTS (pseudo-)reaction and therefore uses the same strategy outlined above. This also holds for modifications of bond orders, which are tricky, in particular when aromatic groups are involved. Here, the tool tries to ensure that the request, which consists of a bond (defined by atom indices) and a target order (single, double, triple, or aromatic), is carefully translated into a valid reaction SMARTS, taking advantage of the fact that the SMARTS language encodes bond order explicitly. This is nontrivial because bond orders cannot be defined independently of hybridization states, entailing necessary changes to numbers of hydrogen atoms and local geometries. Lastly, conformational editing is possible via rotatable dihedral angles. A rotatable bond is picked, directionally, via atom indices, and the terminal end is rotated by a specified increment. This is mostly meant for the interactive optimization of conformers, usually in the context of a binding site. All requests can cover multiple atoms or bonds simultaneously, and preferences for isomeric preferences at new stereocenters or double bonds can be included.

Results

Software versions

All of the tests performed with our software rely on our custom fork of OpenBabel 3.1.1,[38] RDKit 2022_3,[63] Python3.9 (bindings for RDKit in Python 3.5), AmberTools (2019),[39] CGenFF 2.5.1 (using CGenFF parameters version 4.6),[32] and CAMPARIv5.[64] Data and plots have been generated using R 4.3.2 and standard shell tools.

Virtual libraries

To demonstrate the performance of the tool, we focus on three different test libraries:

VL1. VL1 is a random subset of tranches of smaller on-demand molecules in Zinc.[9] Molecules were selected to not include halogen atoms other than fluorine: these are not problematic *per se*, except that CGenFF creates extra dummy atoms for them,[42] which would have complicated some of the analyses below. These 12710 molecules cover common Sybyl types well: some of the rarer ones include S.o (found in 56 molecules), C.cat (in 92), or S.2 (in 68). Phosphorous is basically absent. Quartiles for

numbers of heavy atoms are 8 (minimum), 15, 16, 17, and 19 (maximum); the same for the number of rings is 0, 1, 2, 2, and 4, while the means are 16.2 and 1.7, respectively. Using smaller molecules limits the computational cost and is more representative of virtual screening campaigns. It must be noted that the likelihood of failure in any step increases, on average, with molecule size.

VL2. On a molecule with 20 heavy atoms, 6 carbon atoms were tagged for combinatorial modification. The 5 groups replacing one of the hydrogen atoms were chlorine, ethyl, pyrimidine, pyrrole, and sulfonate. By enumerating up to combinatorial depth 4, a library of no more than 12280 molecules is obtained (see Methods, Virtual chemistry). Naturally, these molecules are of limited diversity: they share a common scaffold and the decorations are repeated throughout. Normally, only up to 10 molecules fail to be produced, highlighting the robustness of the creation pipeline. VL2 contains all 12280 molecules, and they have quartiles for the number of heavy atoms of 21 (minimum), 31, 33, 36, and 44 (maximum).

VL3. We created a small molecule with a number of different functional groups, see Fig. 2. One after the other, this molecule was subject to SMARTS-based chemical reactions, each time selecting all eligible candidates from a small library of 1550 reaction partners. This library of reactants is specifically tailored for this purpose and could be found, in physical form, in an organic chemistry lab. As one example, it includes boronic acids (Suzuki reagents), which would not be findable in Zinc. The list of reactions performed includes all in Table I except for the reductive amination of aldehydes, each time using the first educt listed from the root molecule and the second from the library, as indicated in Fig. 2. The 6676 resultant molecules in VL3 have quartiles for the number of heavy atoms of 42 (minimum), 48, 51, 54, and 72 (maximum) and 2, 3, 3, 4, and 8 for the number of rings (means 51.3 and 3.4). Unlike in the case of VL2, this final number reflects a small fraction of sporadic failures as discussed below.

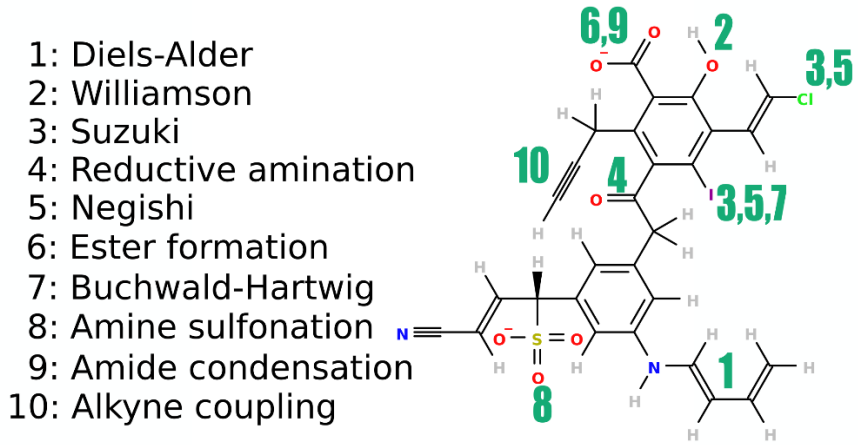


Figure 2. **Root molecule for VL3 in 2D**.The list of reactions is on the left with the sites at which they are performed labeled, by number, in the diagram on the right. The molecule is not meant to be synthesizable or stable but simply to provide a platform for testing the virtual chemical reactions in a single, challenging environment. The ketone moiety (4) is the most occluded.

Speed of processing

Since the parallelization is trivial, the goal here is to demonstrate that it meets expectations, *i.e.*, that the processing rate increases linearly with the number of worker processes employed. Fig. 3A shows that the time per molecule is roughly constant, as expected. The time reflects the amortized cost of

partial charge generation (here, CGenFF), solvation group decomposition, conformer generation (a total of 5-6 conformers are produced and written per molecule), sanitization, and I/O. Parallel efficiency can only be hampered if communication or I/O become limiting, which is an effect apparent from the nonzero slope of the fitted line and, especially, from the last data point. In this regime, the net cost per molecule is down to a few ms. Our software centralizes all file I/O through the master process, and it starts to be saturated for this number of processes. The master-worker scheme is chosen for guaranteeing load balancing. Note that, unlike the cost of parameterization tasks, the cost of communication and I/O grows at worst linearly with molecule size, which means that the test performed here is stringent.

In Fig. 3B, we show a similar analysis but for the creation of a combinatorial library based on only a single root molecule. This means that there is no significant file reading cost. Instead, the master process systematically enumerates and communicates different combinations of sites and decorating groups to the workers, again, one-by-one, to ensure load balancing. The cost shown here encompasses only the steps for the chemical modification, which includes everything described in Methods, Virtual Chemistry, including restrained conformer generation, but no other operations, such as partial charge generation. Only a single conformer per molecule is written, and, unlike in A, there is no drop off in scaling for the largest number of processes evaluated. We emphasize that our software also makes it possible to, in one call, perform any of these other operations on the molecules being produced combinatorially "*in situ*."

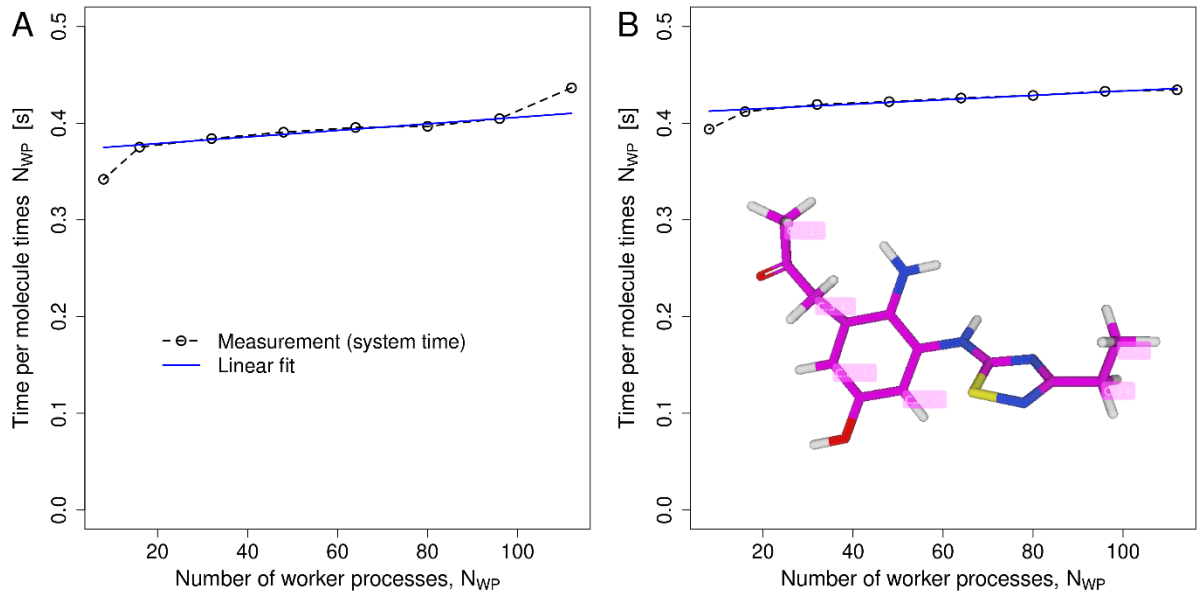


Figure 3. **Analysis of parallel performance**. All tests are performed on a small compute cluster running Slurm. Each node is equipped with two 16-core Intel Xeon Gold 6130 CPUs and 64GB of memory. The system was otherwise empty for the purpose of testing. **A**. Strong scaling is demonstrated by the near-constant average time cost per molecule when multiplied with the number of worker processes, N_{WP}. The blue line is a linear fit for all data points excluding the first and last (slope 0.3ms/N_{WP}). The first data point falls of the line because the available memory size and bandwidth per core (matters primarily for cache) becomes larger once a CPU is underutilized. Data are obtained for a set of operations performed on VL1 (see main text). **B**. The same but for the creation of VL2. The root molecule is shown as an inset (carbon atoms in magenta) with the 6 target sites labeled. The blue line is a linear fit for all data points excluding the first (slope 0.2ms/N_{WP}).

Sanity of molecules

The primary design goal of our workflow is to ensure that the resultant molecules are valid in a software sense, in a chemical sense (valency, *etc*), and, especially, in a 3D spatial sense: they should not contain clashes and/or violations of covalent geometries. For the virtual chemistry tasks, it is a point of particular emphasis that the coordinates of atoms that are present in both the root molecule and the products are preserved as much as possible.

To quantify software compatibility, we first note that, as described in "File formats and compatibility," the annotation of aromatic bonds and the treatment of certain Sybyl types is a prominent problem in interoperability across different software packages. Our solution is essential, depending on how the input file is generated: with it in place, the 12710 molecules of VL1 are all processable, irrespective of underlying conventions used in the input file. If, on the other hand, the sanitization step is skipped, RDKit fails to process 5123 of these from a mol2-file written by OpenBabel, which re-perceives aromaticity. This is by far the largest effect we observe. In comparison, only 47 of the 12710 molecules are not processed by CGenFF, 433 cannot be decomposed into solvation groups, and only 4 cannot be read by RDKit when the input file is an sdf-file written by OpenBabel and sanitization is skipped. This highlights the aforementioned compatibility issues between the mol2 format and RDKit, attributable to aromaticity perception and Sybyl types. In contrast, the use of sdf files entails the cost of a larger disconnect between file and internal representations.

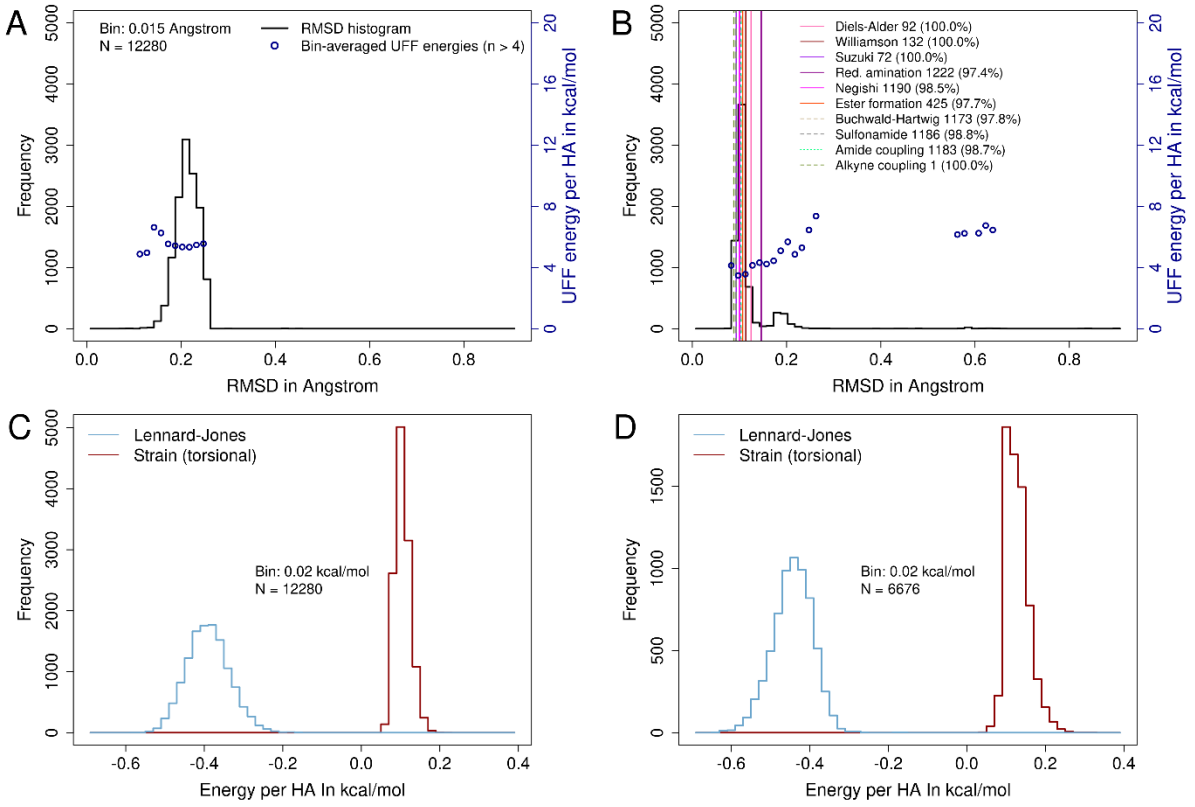


Figure 4. **Sanity of molecules produced through virtual chemistry tasks**. **A**. For VL2, the histogram of RMSD values for the preserved molecular core is plotted along with UFF energies reported by RDKit and averaged per bin (right y-axis). The latter data are only shown for bins with a count of 5 or more. UFF energies include the position and distance restraint terms. **B**. The same as **A** for VL3. The vertical lines are average RMSD values per reaction, with the number of products given. The percent value in parentheses is the fraction of possible, unique

products for the reaction in question that is represented in VL3. The two reactions with the highest, mean RMSD values also appear sterically to be the most complicated (cycloaddition and modifying the ketone that links the two rings in the root molecule, see Fig. 2). **C**. CAMPARI force field energies split into Lennard-Jones and torsional (strain-like) contributions for VL2. Bond angle strain, if any, is not captured by this. **D**. The same as C for VL3.

Chemical sanity is comfortably checked by CAMPARIv5 (https://campari.sourceforge.net/V5).[65] Upon reading the molecules, CAMPARI will try to construct a Z-matrix representation from the input coordinates: this involves manifold assumptions about reasonable bond lengths, angles, planarity, valency, etc. If this fails, an error is produced. In addition, CAMPARI produces warnings about types of chemical bonds it does not understand. These can indicate a rare functional group but also unexpected geometries (such as non-planar amides). Here, all molecules in both VL2 (combinatorial library) and VL3 (explicit chemical reactions) are regarded as sane, which is remarkable. It is beyond the scope of this work to try to parse warnings about unusual bonds systematically except to say that they are virtually absent in VL2 and slightly more common in VL3, most prominently for the reductive amination where ~0.5% of all bonds are tagged in this manner.

The spatial sanity is analyzed twofold. First, for VL2 and VL3, we can plot the distribution of root mean-square deviations (RMSD) of the coordinates of the preserved heavy atoms between the root molecule and the products, which are the atoms not appearing in any SMARTS and not cleaved off. For VL2, these are between 16 and 19 heavy atoms (more for fewer substitutions) while for VL3 there are between 35 and 38. Figs. 4A-B show that the vast majority of products comply with the original substructures very well. There are minor outliers visible only for VL3 while the distribution for VL2 is homogeneous. Note that we used a similarity threshold of 0.25Å in RMSD in the first pass, which is why there is a hard cut discernible in Fig. 4A. For VL3, the molecules are generally larger and have lower RMSD on average. The protocol of using a fixed number of attempts with a similarity threshold (see Methods, Restrained conformer generation) was chosen because straightforward changes to the protocol (more steps, weaker or stronger restraints, using only one type of restraint, *etc*) were not able to resolve an initially faulty embedding. In cases where this still failed, a second round was performed at 0.5Å, before free conformers generation kicked in as a fallback. Based on the minimal presence of outliers, neither mechanism was resorted to frequently. Lastly, the UFF energies increase, on average, with RMSD, marking this as an optimization problem.

To further corroborate the sanity of the conformers, Figs. 4C-D show distributions of components of force field energies per heavy atom obtained using the automatic small molecule parameterization in CAMPARI, which is a detailed, all-atom force field. More importantly, it does not contain the optimization restraints added to the UFF terms shown in A-B, which dominate the UFF energies. For both VL2 (C) and VL3 (D), the distributions look normal with maybe a minimal skew visible. The Lennard-Jones energies per atom are more negative for VL3 because these molecules are much larger. There is not a single molecule for which they are positive, confirming the steric feasibility of the reaction products. Strain energy is much harder to quantify[66] but the proxy we employ here does not suggest major problems. Coupled to the fact that we obtained all the requested molecules for VL2 and that nearly all expected reaction products are obtained for VL3 (see legend in Fig. 4B), we can assert the robustness of the virtual chemistry pipelines.

Parameterization

In the final part, we highlight the two main aspects of parameterization that our tool provides. First, we focus on partial charges with a particular emphasis on gross differences at the net charge level. This is meant to aid users of the tool to gauge possible errors. We emphasize that these are properties of downstream models, and it is clearly out of scope for this work to rank or criticize them. Below, we also omit many other aspects of partial charge sets, *e.g.*, whether they form compact groups with integer net charges, whether they are conformation-dependent, or how costly they are to compute. We focus here only on VL1 since VL2 and VL3, by definition, contain limited chemical diversity. Fig. 5A shows that, at the level of individual atoms, all charge sets investigated are correlated.[18] MMFF94[30] is somewhat of an outlier in terms of correlation (upper left), which is most likely a direct consequence of all partial charges on aliphatic hydrogen atoms being 0.0. The Gasteiger[67] model is also special in that it fails to produce any Spearman cross-correlation above 0.64, by far the lowest maximum. In terms of mean absolute differences (bottom right), only MMFF94 stands out, for the same reason. The fraction of atoms with large partial charge differences (>0.5) is surprising and lowest (0.3%) for MMFF94 and GAFF. Throughout, hydrogen and oxygen are massively depleted as outliers while carbon features roughly as expected based on overall prevalence. Nitrogen is clearly enriched (factors of 5-10) whenever Gasteiger or CGenFF are involved in a cross-pair. Lastly, for all pairs involving EEM, sulfur is massively enriched (so much that >90% of all sulfur atoms are outliers). Sulfur is also enriched when CGenFF is involved, particularly with respect to GAFF. Overall, this suggests that features such as sulfonamides or N-rich heteroaromatics will cause a high sensitivity to the partial charge set in use.

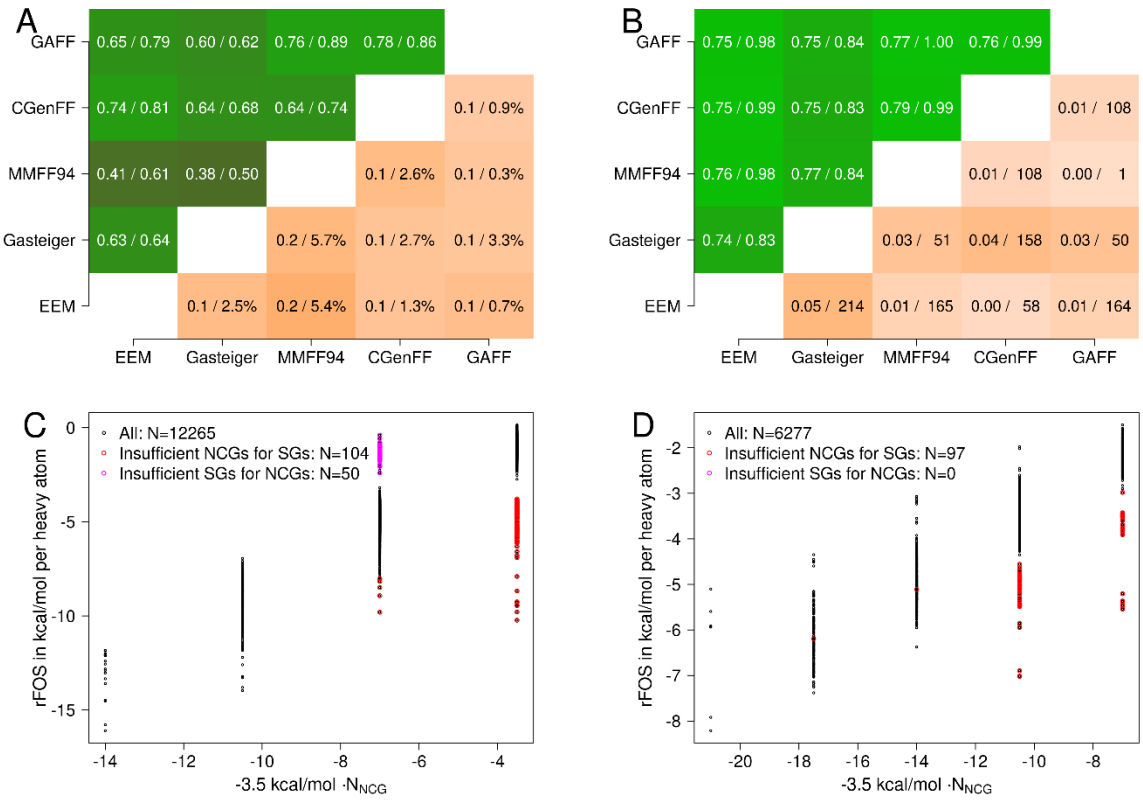


Figure 5. **Consistency of assigned parameters**. **A**. For VL1, a cross-relation matrix between atomic partial charges is shown (upper left). The numbers give the Spearman and Pearson correlation coefficients, respectively. The bottom right matrix is filled with MAD (mean absolute deviation, first number) values. The second number is the

fraction of atoms with absolute deviations exceeding 0.5 units. The color is just to guide the eye (maps Spearman correlation and MAD). **B**. The same as A but at the level of whole molecules. For every molecule, the partial charges were simply summed up. The Spearman correlations are lower because ranks can be tied but also because the charges often do not add up exactly to an integer value. The second number in the bottom right matrix (gross deviations) is given as the total instead. Color tracks Pearson correlation and MAD. **C**. For VL1, actual rFOS values normalized by the number of heavy atoms are plotted against a heuristic prediction based on the number of NCGs. Outliers are detected as cases where the assigned rFOS value is more negative than $\left[-80\left(N_{NCG}+0.5\right)-N_{HA}\right]kcal/mol$ (red) or more positive than $\left(-50N_{NCG}+N_{HA}\right)kcal/mol$ (magenta). **D**. The same as C for VL3. Because the sulfonate and (neutral) sulfonamide moieties are absent or exceedingly rare in experimental data, we added them as empirical groups for the ABSINTH GFF parameterization (rFOS values of -81.3 and -5.98kcal/mol, respectively).

At the whole molecule level, gross differences can appear as well (Fig. 5B). This often has to do with the perception of specific chemical moieties. Here, MMFF94, GAFF, and Gasteiger seem to form one group that is mutually consistent, which for MMFF94 and GAFF is inevitable: for the BCC-AM1[68] algorithm to work for odd-numbered net charge values in Antechamber, the net charge must be passed as an argument, and we use a guess derived from MMFF94 for this purpose. EEM[69-70] and CGenFF form the other group. From visual inspection, the ~50 discrepancies between Gasteiger and either MMFF94 or GAFF are random: they seem to be from misperceptions of the Gasteiger algorithm for relatively simple molecules featuring amides, esters, *etc*. The differences between EEM and CGenFF are dominated by protonated amidine/guanidine-like moieties marked as aromatic (37/58 matches with only 45/12710 molecules matching the pattern across the entire VL1). Conversely, the differences between GAFF or MMFF94 and CGenFF are more distributed, but tetrazolide anions (19/108 vs. 19/12710), 5-rings with 3 heteroatoms and at least one carbon atom double-bonded to O or N outside of the ring (20/108 vs. 46/12710), and amides with an unusual O-R substitution at the nitrogen, such as Weinreb amides (38/108 vs. 153/12710), are clearly systematic problems and sufficient to explain the bulk of differences. Not only is such information useful to be aware of for screening combinatorial designs that all share a problematic functional group, but it is also fair to classify these moieties as difficult to parameterize in general. Another function that has low prevalence but seems systematically inconsistent are deprotonated sulfonamides. Importantly, the molecules in VL1 are so small that they usually contain only one of the aforementioned functionalities (only 2 molecules have more than one match).

Second, we highlight the properties of the solvation group decomposition algorithm, which appeared in a developmental version in Marchand *et al*.[36] The goal of this algorithm is to allow an ABSINTH-based GFF: the ABSINTH implicit solvent model describes solvation as two processes: dielectric screening and direct mean-field interactions (DMFI).[44] To compute the DMFI, each entity must be decomposed into a set of groups, which should correspond to small molecules with experimentally known free energies of solvation. This provides group contributions to a reference free energy of solvation (rFOS), which are then scaled by solvent accessibility. Using it on small molecules is mostly a data problem: available experimental data are sparse, necessitating the inclusion of less accurate, single-atom corrective values.[36, 71] A second issue is the dramatic difference in scale for molecules carrying a net charge over neutral ones and the uncertainty they carry.[72]

Here, we have recast the algorithm, extended the library of small molecules with known free energies of solvation, and changed some of the exception handling. In principle, molecules can be decomposed

in several different ways: to pick a particular decomposition, we maintain the original heuristic, which is to choose, amongst the decompositions requiring the fewest groups, the one with the median net rFOS value. For VL1 (Fig. 5C), 96.5% of molecules were parameterized successfully. Since the experimental free energies of solvation are unknown, the main sanity check is to avoid gross errors. Gross errors arise when there is a mismatch between moieties carrying a net charge (NCGs) and the rFOS values for the solvation groups (SGs) encompassing them.[73] rFOS values depend astutely on charge with values near -70kcal/mol for small, monovalent, organic ions but only -5 to -10kcal/mol for their neutral counterparts. The correlation between the number of NCGs and the net rFOS is evident in Fig. 5C, and it can be used to derive a heuristic to identify outliers (colored dots). These can be failures of either the decomposition or the partial charge generation algorithm, here MMFF94.

Finding NCGs requires grouping charges into topologically compact sets accounting for a net integer charge, which we do in CAMPARI as explained in Marchand *et al*. (tolerance of 0.05 units). There are 104 problematic molecules where insufficient NCGs are detected to explain the assigned rFOS value. These are massively enriched in protonated pyridine and pyrimidine moieties (65/104 matches with 83/12710 molecules matching this pattern across all of VL1). They could give rise to false negatives because the desolvation penalty (which is somewhat proportional to the rFOS value) is massively inflated.[74-76] Conversely, the 50 inverse errors are more of a concern as they are heavily predisposed to be false positives. The erroneous parameterization of molecules is a concern that is increasingly recognized in virtual screens, in particular on ultralarge libraries.[10] Here, the dominant patterns are tetrazolide anions (12/50 vs. 19/12710) and 5-rings with 3 heteroatoms and at least one carbon atom double-bonded to O or N outside of the ring (17/50 vs. 46/12710). The second are probably also N-based anions, at least in the perception of MMFF94. Since the decomposition approach lacks an experimental reference value for this type of ion, it constructs the rings from atom-based corrective values, which fail to detect that one of them must be assigned as an ion. There are numerous ways in which this can be improved in the future, but the heuristic used in Figs. 5C-D is a good diagnostic tool to begin with. For VL3 (94.0% of molecules parameterized, Fig. 5D), there are only false-negative candidates, and many of these involve imidazolium ions (34/97 vs. 34/6676), thiazolium ions (N-alkylated, 10/97 vs. 14/6676), or fusion nitrogen atoms, all of which contain mismatches in positive charge perception. The molecules in VL3 are much larger on average, which is why the y-axis scale differs relative to C.

Discussion & Conclusions

In this work, we have presented and tested ParaLig, a solution to efficiently perform a variety of cheminformatics tasks, in parallel, with a particular emphasis on maintaining 3D coordinates. As part of the results, we have also analyzed common discrepancies in parameter assignments, which indirectly uncover chemical moieties that are “difficult.” One aspect that is common to many of them are nitrogen-based charges in delocalized systems,[77-79] including cations but also anions such as tetrazolides or deprotonated sulfonamides. By streamlining various interwoven workflows, our software simplifies and standardizes these tasks considerably and, thus, allows meta-analyses of this kind. It is also used as a core component of an integrated molecular modelling platform, ACGui, which will contribute favorably toward ensuring active maintenance and future extensions.

Ensuring an accurate treatment of molecules in terms of curation of conformers,[80] protonation states,[81-82] electron delocalization,[25] parameters, *etc.*, is crucial, in particular when it comes to avoiding

gross errors.[10] Avoiding these errors requires active and sustained curation at the source library level, and the Zinc library,[9] with its intricate preparation workflow, is a good example for this. Even when working with libraries of only a few 100s of molecules, human, visual supervision would be impractical. More importantly, it is possible only for relatively simple features like steric clashes or bad bond angles, but not for more intricate problems like isomeric preferences or difficult protonation states. Further still, chemical space is vast, and frequently predictions for molecules that do not exist yet are desired.[16, 49] Thus, one is inevitably forced to defer to automation where tools like the one presented here come into play. Through a variety of diagnostic tests, we have demonstrated the high success rate of this automation, in particular for molecules obtained through the virtual chemistry pipeline.

In drug discovery tasks, which is where our application interests lie, both false negatives and positives can result from aberrant molecule preparation (primarily conformers, parameters, protonation).[10, 80, 83] ParaLig does not address the issue of protonation directly, except for the helpful heuristics to flag inconsistencies shown in Fig. 5. In our experience, current practice in this area has not evolved dramatically, and experimental data remain sparse. There are excellent tools for enumerating reasonable tautomers,[84] but the prediction of pKa values to high accuracy remains challenging.[85] In particular, simple approaches based on SMARTS-matching and tabulated reference values are unlikely to be useful. Thus, we envision future extensions of ParaLig to defer to external software for these tasks, similar to the solution with CGenFF and AmberTools. Aside from preparation and parameterization issues, false negatives can also result from failures to obtain valid molecules to begin with. Thus, the main avenues for future work, aside from improving generality, will be to embed various heuristic sanity checks into the workflow and deposit their results as metadata, per molecule, in the output files. This would allow application scientists to flag possible errors of this kind more easily.

Finally, we want to highlight that ParaLig can be embedded into surrounding workflows. Visual guidance is fundamental in many aspects of computational, molecular science,[16, 86-89] and we have developed an integrated solution for dealing with small molecules in the context of drug discovery. This tool, accessible through a browser, will be presented in full elsewhere and combines a molecular viewer with an application software-like graphical user interface to conduct, amongst others, drug discovery tasks. Such an environment enables application scientists to select, edit, or screen small molecules even if they possess little to no expertise in the underlying software solution. A demo server is available at https://acgui.bioc.uzh.ch/acgui that also offers rich embedded help (button “Videos”) to illustrate some of the workflows discussed in this manuscript.

Software & Data Availability

The source code of the software tools described herein is publicly available at https://gitlab.com/CaflischLab/paraLig. In addition, this repository contains all the required input files to recreate the results presented above. All the auxiliary tools except CGenFF are available freely as well.

Acknowledgments

We are grateful to Laurent Batiste for being a codeveloper during early stages, to Cassiano Langini for his many indirect contributions to this work, and to Francesco Errani for helpful discussions. This work was supported by grants 189363 and 212195 from the Swiss National Science Foundation to AC.

References

1. Varnek, A.; Baskin, I. I., Chemoinformatics as a Theoretical Chemistry Discipline. *Mol. Inf.* **2011,** *30* (1), 20-32.
2. Gasteiger, J., Chemoinformatics: Achievements and Challenges, a Personal View. *Molecules* **2016,** *21* (2), 151.
3. Hughes, L. D.; Palmer, D. S.; Nigsch, F.; Mitchell, J. B. O., Why Are Some Properties More Difficult To Predict than Others? A Study of QSPR Models of Solubility, Melting Point, and Log P. *J. Chem. Inf. Model.* **2008,** *48* (1), 220-232.
4. Huang, S.-Y.; Grinter, S. Z.; Zou, X., Scoring functions and their evaluation methods for protein–ligand docking: recent advances and future directions. *Phys. chem. chem. phys.* **2010,** *12* (40), 12899-12908.
5. Hann, M. M.; Oprea, T. I., Pursuing the leadlikeness concept in pharmaceutical research. *Curr. Opin. Chem. Biol.* **2004,** *8* (3), 255-263.
6. Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J., Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug Deliv. Rev.* **2001,** *46* (1), 3-26.
7. Plewczynski, D.; Łaźniewski, M.; Augustyniak, R.; Ginalski, K., Can we trust docking results? Evaluation of seven commonly used programs on PDBbind database. *J. Comput. Chem.* **2011,** *32* (4), 742-755.
8. Chevillard, F.; Kolb, P., SCUBIDOO: A Large yet Screenable and Easily Searchable Database of Computationally Created Chemical Compounds Optimized toward High Likelihood of Synthetic Tractability. *J. Chem. Inf. Model.* **2015,** *55* (9), 1824-1835.
9. Tingle, B. I.; Tang, K. G.; Castanon, M.; Gutierrez, J. J.; Khurelbaatar, M.; Dandarchuluun, C.; Moroz, Y. S.; Irwin, J. J., ZINC-22─A Free Multi-Billion-Scale Database of Tangible Compounds for Ligand Discovery. *J. Chem. Inf. Model.* **2023,** *63* (4), 1166-1176.
10. Irwin, J. J.; Shoichet, B. K., Docking Screens for Novel Ligands Conferring New Biology. *J. Med. Chem.* **2016,** *59* (9), 4103-4120.
11. Wang, Y.; Xing, J.; Xu, Y.; Zhou, N.; Peng, J.; Xiong, Z.; Liu, X.; Luo, X.; Luo, C.; Chen, K.; Zheng, M.; Jiang, H., In silico ADME/T modelling for rational drug design. *Quart. Rev. Biophys.* **2015,** *48* (4), 488-515.
12. Faber, F. A.; Hutchison, L.; Huang, B.; Gilmer, J.; Schoenholz, S. S.; Dahl, G. E.; Vinyals, O.; Kearnes, S.; Riley, P. F.; von Lilienfeld, O. A., Prediction Errors of Molecular Machine Learning Models Lower than Hybrid DFT Error. *J. Chem. Theor. Comput.* **2017,** *13* (11), 5255-5264.
13. Lo, Y.-C.; Rensi, S. E.; Torng, W.; Altman, R. B., Machine learning in chemoinformatics and drug discovery. *Drug Discov. Today* **2018,** *23* (8), 1538-1546.
14. Wale, N., Machine learning in drug discovery and development. *Drug Devel. Res.* **2011,** *72* (1), 112-119.
15. Lyu, J.; Irwin, J. J.; Shoichet, B. K., Modeling the expansion of virtual screening libraries. *Nat. Chem. Biol.* **2023,** *19* (6), 712-718.
16. Lyu, J.; Wang, S.; Balius, T. E.; Singh, I.; Levit, A.; Moroz, Y. S.; O'Meara, M. J.; Che, T.; Algaa, E.; Tolmachova, K.; Tolmachev, A. A.; Shoichet, B. K.; Roth, B. L.; Irwin, J. J., Ultra-large library docking for discovering new chemotypes. *Nature* **2019,** *566* (7743), 224-229.
17. Grygorenko, O. O.; Radchenko, D. S.; Dziuba, I.; Chuprina, A.; Gubina, K. E.; Moroz, Y. S., Generating Multibillion Chemical Space of Readily Accessible Screening Compounds. *iScience* **2020,** *23* (11), 101681.
18. Chakravorty, A.; Hussain, A.; Cervantes, L. F.; Lai, T. T.; Brooks, C. L., III, Exploring the Limits of the Generalized CHARMM and AMBER Force Fields through Predictions of Hydration Free Energy of Small Molecules. *J. Chem. Inf. Model.* **2024,** *64* (10), 4089-4101.
19. Weininger, D.; Weininger, A.; Weininger, J. L., SMILES. 2. Algorithm for generation of unique SMILES notation. *J. Chem. Inf. Computer Sci.* **1989,** *29* (2), 97-101.
20. Heller, S.; McNaught, A.; Stein, S.; Tchekhovskoi, D.; Pletnev, I., InChI - the worldwide chemical structure identifier standard. *J. Cheminform.* **2013,** *5* (1), 7.
21. Chuang, K. V.; Gunsalus, L. M.; Keiser, M. J., Learning Molecular Representations for Medicinal Chemistry. *J. Med. Chem.* **2020,** *63* (16), 8705-8722.
22. Schymanski, E. L.; Bolton, E. E., FAIR chemical structures in the Journal of Cheminformatics. *J. Cheminform.* **2021,** *13* (1), 50.
23. Clark, A. M., On the myth of chemical structure format conversion. In *Cheminformatics 2.0*, wordpress.com, 2014; Vol. 2024.

24. Schneider, N.; Sayle, R. A.; Landrum, G. A., Get Your Atoms in Order—An Open-Source Implementation of a Novel and Robust Molecular Canonicalization Algorithm. *J. Chem. Inf. Model.* **2015,** *55* (10), 2111-2120.
25. Merino, G.; Solà, M.; Fernández, I.; Foroutan-Nejad, C.; Lazzeretti, P.; Frenking, G.; Anderson, H. L.; Sundholm, D.; Cossío, F. P.; Petrukhina, M. A.; Wu, J.; Wu, J. I.; Restrepo, A., Aromaticity: Quo Vadis. *Chem. Sci.* **2023,** *14* (21), 5569-5576.
26. Bietz, S.; Urbaczek, S.; Schulz, B.; Rarey, M., Protoss: a holistic approach to predict tautomers and protonation states in protein-ligand complexes. *J. Cheminform.* **2014,** *6* (1), 12.
27. Navo, C. D.; Jiménez-Osés, G., Computer Prediction of pKa Values in Small Molecules and Proteins. *ACS Med. Chem. Lett.* **2021,** *12* (11), 1624-1628.
28. Bietz, S.; Inhester, T.; Lauck, F.; Sommer, K.; von Behren, M. M.; Fährrolfes, R.; Flachsenberg, F.; Meyder, A.; Nittinger, E.; Otto, T.; Hilbig, M.; Schomburg, K. T.; Volkamer, A.; Rarey, M., From cheminformatics to structure-based design: Web services and desktop applications based on the NAOMI library. *J. Biotech*, **2017,** *261*, 207-214.
29. Rappe, A. K.; Casewit, C. J.; Colwell, K. S.; Goddard, W. A., III; Skiff, W. M., UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *J. Am. Chem. Soc.* **1992,** *114* (25), 10024-10035.
30. Halgren, T. A., Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94. *J. Comput. Chem.* **1996,** *17* (5-6), 490-519.
31. Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A., Development and testing of a general amber force field. *J. Comput. Chem.* **2004,** *25* (9), 1157-1174.
32. Vanommeslaeghe, K.; Hatcher, E.; Acharya, C.; Kundu, S.; Zhong, S.; Shim, J.; Darian, E.; Guvench, O.; Lopes, P.; Vorobyov, I.; Mackerell Jr, A. D., CHARMM general force field: A force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields. *J. Comput. Chem.* **2010,** *31* (4), 671-690.
33. Fourches, D.; Muratov, E.; Tropsha, A., Trust, But Verify: On the Importance of Chemical Structure Curation in Cheminformatics and QSAR Modeling Research. *J. Chem. Inf. Model.* **2010,** *50* (7), 1189-1204.
34. Sitzmann, M.; Ihlenfeldt, W.-D.; Nicklaus, M. C., Tautomerism in large databases. *J. Comput. Aided Mol. Des.* **2010,** *24* (6), 521-551.
35. Riniker, S.; Landrum, G. A., Better Informed Distance Geometry: Using What We Know To Improve Conformation Generation. *J. Chem. Inf. Model.* **2015,** *55* (12), 2562-2574.
36. Marchand, J.-R.; Knehans, T.; Caflisch, A.; Vitalis, A., An ABSINTH-Based Protocol for Predicting Binding Affinities between Proteins and Small Molecules. *J. Chem. Inf. Model.* **2020,** *60* (10), 5188-5202.
37. RDKit: Open-source cheminformatics. https://www.rdkit.org/ (accessed 10/24/2024).
38. O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R., Open Babel: An open chemical toolbox. *J. Cheminform.* **2011,** *3* (1), 33.
39. Case, D. A.; Aktulga, H. M.; Belfon, K.; Cerutti, D. S.; Cisneros, G. A.; Cruzeiro, V. W. D.; Forouzesh, N.; Giese, T. J.; Götz, A. W.; Gohlke, H.; Izadi, S.; Kasavajhala, K.; Kaymak, M. C.; King, E.; Kurtzman, T.; Lee, T.-S.; Li, P.; Liu, J.; Luchko, T.; Luo, R.; Manathunga, M.; Machado, M. R.; Nguyen, H. M.; O'Hearn, K. A.; Onufriev, A. V.; Pan, F.; Pantano, S.; Qi, R.; Rahnamoun, A.; Risheh, A.; Schott-Verdugo, S.; Shajan, A.; Swails, J.; Wang, J.; Wei, H.; Wu, X.; Wu, Y.; Zhang, S.; Zhao, S.; Zhu, Q.; Cheatham, T. E., III; Roe, D. R.; Roitberg, A.; Simmerling, C.; York, D. M.; Nagan, M. C.; Merz, K. M., Jr., AmberTools. *J. Chem. Inf. Model.* **2023,** *63* (20), 6183-6191.
40. O'Boyle, N. M.; Morley, C.; Hutchison, G. R., Pybel: a Python wrapper for the OpenBabel cheminformatics toolkit. *Chem. Cent. J.* **2008,** *2* (1), 5.
41. Majeux, N.; Scarsi, M.; Apostolakis, J.; Ehrhardt, C.; Caflisch, A., Exhaustive docking of molecular fragments with electrostatic solvation. *Prot. Struct. Func. Bioinf.* **1999,** *37* (1), 88-105.
42. Soteras Gutiérrez, I.; Lin, F.-Y.; Vanommeslaeghe, K.; Lemkul, J. A.; Armacost, K. A.; Brooks, C. L.; MacKerell, A. D., Parametrization of halogen bonds in the CHARMM general force field: Improved treatment of ligand–protein interactions. *Bioorg. Med. Chem.* **2016,** *24* (20), 4812-4825.
43. Boothroyd, S.; Behara, P. K.; Madin, O. C.; Hahn, D. F.; Jang, H.; Gapsys, V.; Wagner, J. R.; Horton, J. T.; Dotson, D. L.; Thompson, M. W.; Maat, J.; Gokey, T.; Wang, L.-P.; Cole, D. J.; Gilson, M. K.; Chodera, J. D.; Bayly, C. I.; Shirts, M. R.; Mobley, D. L., Development and Benchmarking of Open Force Field 2.0.0: The Sage Small Molecule Force Field. *J. Chem. Theor. Comput.* **2023,** *19* (11), 3251-3275.
44. Vitalis, A.; Pappu, R. V., ABSINTH: A new continuum solvation model for simulations of polypeptides in aqueous solutions. *J. Comput. Chem.* **2009,** *30* (5), 673-699.
45. Bron, C.; Kerbosch, J., Algorithm 457: finding all cliques of an undirected graph. *Commun. ACM* **1973,** *16* (9), 575-577.
46. Hawkins, P. C. D., Conformation Generation: The State of the Art. *J. Chem. Inf. Model.* **2017,** *57* (8), 1747-1756.

47. Batiste, L.; Unzue, A.; Dolbois, A.; Hassler, F.; Wang, X.; Deerain, N.; Zhu, J.; Spiliotopoulos, D.; Nevado, C.; Caflisch, A., Chemical Space Expansion of Bromodomain Ligands Guided by in Silico Virtual Couplings (AutoCouple). *ACS Cent. Sci.* **2018,** *4* (2), 180-188.
48. Daylight Chemical Information Systems, I., SMARTS-A Language for Describing Molecular Patterns. 2007.
49. Beroza, P.; Crawford, J. J.; Ganichkin, O.; Gendelev, L.; Harris, S. F.; Klein, R.; Miu, A.; Steinbacher, S.; Klingler, F.-M.; Lemmen, C., Chemical space docking enables large-scale structure-based virtual screening to discover ROCK1 kinase inhibitors. *Nat. Commun.* **2022,** *13* (1), 6447.
50. Guram, A. S.; Rennels, R. A.; Buchwald, S. L., A Simple Catalytic Method for the Conversion of Aryl Bromides to Arylamines. *Angew. Chem. Intl. Ed. Engl.* **1995,** *34* (12), 1348-1350.
51. Louie, J.; Hartwig, J. F., Palladium-catalyzed synthesis of arylamines from aryl halides. Mechanistic studies lead to coupling in the absence of tin reagents. *Tetrahedron Lett.* **1995,** *36* (21), 3609-3612.
52. Miyaura, N.; Yamada, K.; Suzuki, A., A new stereospecific cross-coupling by the palladium-catalyzed reaction of 1-alkenylboranes with 1-alkenyl or 1-alkynyl halides. *Tetrahedron Lett.* **1979,** *20* (36), 3437-3440.
53. King, A. O.; Okukado, N.; Negishi, E.-i., Highly general stereo-, regio-, and chemo-selective synthesis of terminal and internal conjugated enynes by the Pd-catalysed reaction of alkynylzinc reagents with alkenyl halides. *J. Chem. Soc. Chem. Commun.* **1977,** (19), 683-684.
54. Williamson, A., XLV. Theory of ætherification. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **1850,** *37* (251), 350-356.
55. Abdel-Magid, A. F.; Carson, K. G.; Harris, B. D.; Maryanoff, C. A.; Shah, R. D., Reductive Amination of Aldehydes and Ketones with Sodium Triacetoxyborohydride. Studies on Direct and Indirect Reductive Amination Procedures1. *J. Org. Chem.* **1996,** *61* (11), 3849-3862.
56. Lanigan, R. M.; Starkov, P.; Sheppard, T. D., Direct Synthesis of Amides from Carboxylic Acids and Amines Using B(OCH2CF3)3. *J. Org. Chem.* **2013,** *78* (9), 4512-4523.
57. Fu, X.; Liao, Y.; Glein, C. R.; Jamison, M.; Hayes, K.; Zaporski, J.; Yang, Z., Direct Synthesis of Amides from Amines and Carboxylic Acids under Hydrothermal Conditions. *ACS Earth Space Chem.* **2020,** *4* (5), 722-729.
58. Neises, B.; Steglich, W., Esterification of carboxylic acids with dicyclohexylcabodiimide/4-dimethylaminopyridine: tert-butyl ethyl fumarate. *Org. Synth.* **1985,** *63*, 183.
59. Eglinton, G.; Galbraith, A., 182. Macrocyclic acetylenic compounds. Part I. Cyclo tetradeca-1: 3-diyne and related compounds. *J. Am. Chem. Soc.* **1959**, 889-896.
60. Hoffer, L.; Muller, C.; Roche, P.; Morelli, X., Chemistry-driven Hit-to-lead Optimization Guided by Structure-based Approaches. *Mol. Inf.* **2018,** *37* (9-10), 1800059.
61. Southey, M. W. Y.; Brunavs, M., Introduction to small molecule drug discovery and preclinical development. *Front. Drug Discov.* **2023,** *3*.
62. Ertl, P.; Altmann, E.; McKenna, J. M., The Most Common Functional Groups in Bioactive Molecules and How Their Popularity Has Evolved over Time. *J. Med. Chem.* **2020,** *63* (15), 8408-8418.
63. RDKit: Open-source cheminformatics.
64. Vitalis, A.; Pappu, R. V., Chapter 3 Methods for Monte Carlo Simulations of Biomacromolecules. In *Annual Reports in Computational Chemistry*, Wheeler, R. A., Ed. Elsevier: 2009; Vol. 5, pp 49-76.
65. CAMPARIv5. https://campari.sourceforgenet/V5 (accessed 10/24/2024).
66. Gu, S.; Smith, M. S.; Yang, Y.; Irwin, J. J.; Shoichet, B. K., Ligand Strain Energy in Large Library Docking. *J. Chem. Inf. Model.* **2021,** *61* (9), 4331-4341.
67. Gasteiger, J.; Marsili, M., Iterative partial equalization of orbital electronegativity—a rapid access to atomic charges. *Tetrahedron* **1980,** *36* (22), 3219-3228.
68. Jakalian, A.; Bush, B. L.; Jack, D. B.; Bayly, C. I., Fast, efficient generation of high-quality atomic charges. AM1-BCC model: I. Method. *J. Comput. Chem.* **2000,** *21* (2), 132-146.
69. Geidl, S.; Bouchal, T.; Raček, T.; Svobodová Vařeková, R.; Hejret, V.; Křenek, A.; Abagyan, R.; Koča, J., High-quality and universal empirical atomic charges for chemoinformatics applications. *J. Cheminform.* **2015,** *7* (1), 59.
70. Bultinck, P.; Langenaeker, W.; Lahorte, P.; De Proft, F.; Geerlings, P.; Waroquier, M.; Tollenaere, J. P., The Electronegativity Equalization Method I: Parametrization and Validation for Atomic Charge Calculations. *J. Phys. Chem. A* **2002,** *106* (34), 7887-7894.
71. Cabani, S.; Gianni, P.; Mollica, V.; Lepori, L., Group contributions to the thermodynamic properties of non-ionic organic solutes in dilute aqueous solution. *J. Solut. Chem.* **1981,** *10* (8), 563-595.
72. Kelly, C. P.; Cramer, C. J.; Truhlar, D. G., Aqueous Solvation Free Energies of Ions and Ion−Water Clusters Based on an Accurate Value for the Absolute Aqueous Solvation Free Energy of the Proton. *J. Phys. Chem. B* **2006,** *110* (32), 16066-16081.
73. Mysinger, M. M.; Shoichet, B. K., Rapid Context-Dependent Ligand Desolvation in Molecular Docking. *J. Chem. Inf. Model.* **2010,** *50* (9), 1561-1573.

74. Ferreira, R. S.; Simeonov, A.; Jadhav, A.; Eidam, O.; Mott, B. T.; Keiser, M. J.; McKerrow, J. H.; Maloney, D. J.; Irwin, J. J.; Shoichet, B. K., Complementarity Between a Docking and a High-Throughput Screen in Discovering New Cruzain Inhibitors. *J. Med. Chem.* **2010,** *53* (13), 4891-4905.
75. Deng, N.; Forli, S.; He, P.; Perryman, A.; Wickstrom, L.; Vijayan, R. S. K.; Tiefenbrunn, T.; Stout, D.; Gallicchio, E.; Olson, A. J.; Levy, R. M., Distinguishing Binders from False Positives by Free Energy Calculations: Fragment Screening Against the Flap Site of HIV Protease. *J. Phys. Chem. B* **2015,** *119* (3), 976-988.
76. Graves, A. P.; Shivakumar, D. M.; Boyce, S. E.; Jacobson, M. P.; Case, D. A.; Shoichet, B. K., Rescoring Docking Hit Lists for Model Cavity Sites: Predictions and Experimental Testing. *J. Mol. Biol.* **2008,** *377* (3), 914-934.
77. Maclagan, R. G. A. R.; Gronert, S.; Meot-Ner, M., Protonated Polycyclic Aromatic Nitrogen Heterocyclics: Proton Affinities, Polarizabilities, and Atomic and Ring Charges of 1–5-Ring Ions. *J. Phys. Chem. A* **2015,** *119* (1), 127-139.
78. Melin, J.; Singh, R. K.; Mishra, M. K.; Ortiz, J. V., Tautomeric Forms of Azolide Anions: Vertical Electron Detachment Energies and Dyson Orbitals. *J. Phys. Chem. A* **2007,** *111* (50), 13069-13074.
79. Remko, M.; von der Lieth, C.-W., Theoretical study of gas-phase acidity, pKa, lipophilicity, and solubility of some biologically active sulfonamides. *Bioorg. Med. Chem.* **2004,** *12* (20), 5395-5403.
80. Tirado-Rives, J.; Jorgensen, W. L., Contribution of Conformer Focusing to the Uncertainty in Predicting Free Energies for Protein−Ligand Binding. *J. Med. Chem.* **2006,** *49* (20), 5880-5884.
81. Wei, B. Q.; Baase, W. A.; Weaver, L. H.; Matthews, B. W.; Shoichet, B. K., A Model Binding Site for Testing Scoring Functions in Molecular Docking. *J. Mol. Biol.* **2002,** *322* (2), 339-355.
82. Madhavi Sastry, G.; Adzhigirey, M.; Day, T.; Annabhimoju, R.; Sherman, W., Protein and ligand preparation: parameters, protocols, and influence on virtual screening enrichments. *J. Comput. Aided Mol. Des.* **2013,** *27* (3), 221-234.
83. ten Brink, T.; Exner, T. E., pKa based protonation states and microspecies for protein–ligand docking. *J. Comput. Aided Mol. Des.* **2010,** *24* (11), 935-942.
84. Sommer, K.; Friedrich, N.-O.; Bietz, S.; Hilbig, M.; Inhester, T.; Rarey, M., UNICON: A powerful and easy-to-use compound library converter. *J. Chem. Inf. Model.* **2016,** *56* (6), 1105-1111.
85. Işık, M.; Rustenburg, A. S.; Rizzi, A.; Gunner, M. R.; Mobley, D. L.; Chodera, J. D., Overview of the SAMPL6 pKa challenge: evaluating small molecule microscopic and macroscopic pKa predictions. *J. Comput. Aided Mol. Des.* **2021,** *35* (2), 131-166.
86. Fischer, A.; Smieško, M.; Sellner, M.; Lill, M. A., Decision Making in Structure-Based Drug Discovery: Visual Inspection of Docking Results. *J. Med. Chem.* **2021,** *64* (5), 2489-2500.
87. Dries, D. R.; Dean, D. M.; Listenberger, L. L.; Novak, W. R. P.; Franzen, M. A.; Craig, P. A., An expanded framework for biomolecular visualization in the classroom: Learning goals and competencies. *Biochem. Mol. Biol. Ed.* **2017,** *45* (1), 69-75.
88. Kouřil, D.; Strnad, O.; Mindek, P.; Halladjian, S.; Isenberg, T.; Gröller, M. E.; Viola, I., Molecumentary: Adaptable Narrated Documentaries Using Molecular Visualization. *IEEE Trans. Vis. Comput. Graph.* **2023,** *29* (3), 1733-1747.
89. Pantsar, T.; Poso, A., Binding Affinity via Docking: Fact and Fiction. *Molecules* **2018,** *23* (8).