JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

Application Note

Cite This: J. Chem. Inf. Model. 2019, 59, 2529−2537

pubs.acs.org/jcim

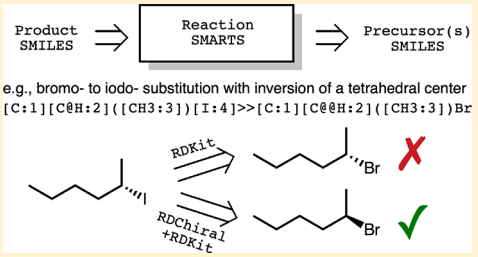
RDChiral: An RDKit Wrapper for Handling Stereochemistry in Retrosynthetic Template Extraction and Application

Connor W. Coley, William H. Green,* and Klavs F. Jensen*

Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States

Supporting Information

ABSTRACT: There is a renewed interest in computer-aided synthesis planning, where the vast majority of approaches require the application of retrosynthetic reaction templates. Here we introduce RDChiral, an open-source Python wrapper for RDKit designed to provide consistent handling of stereochemical information in applying retrosynthetic transformations encoded as SMARTS strings. RDChiral is designed to enforce the introduction, destruction, retention, and inversion of chiral tetrahedral centers as well as the cis/trans configuration of double bonds. We also introduce an open-source implementation of a retrosynthetic template extraction algorithm to generate SMARTS patterns from atom-mapped reaction SMILES strings. In this application note, we describe the implementation of these two pieces of code and illustrate their use through many examples.



INTRODUCTION

The rising availability of reaction corpora, hardware for rapid computing, and algorithms for efficient search have led to a renewed interest in computer-aided synthesis planning (CASP).[1−5] The majority of CASP programs are based on the use of reaction templates—subgraph patterns that describe the changes in connectivity between a product molecule and its corresponding reactant(s)—to generate recommendations for retrosynthetic disconnections.

The earliest CASP programs sought to directly codify expert chemist knowledge about what reactions are allowed.[6] This "expert approach" is reflected by Synthia (formerly Chematica[4]), which now contains around 70 000 hand-encoded reaction transformation rules and has been successfully used to plan synthetic routes to a number of complex products.[4] Significantly smaller rule sets have been made public and popularized,[7,8] like Hartenfeller et al.'s set of 58 reactions popular in medicinal chemistry, while others are integrated into closed-source programs,[9,10] e.g., a set of 2300 highly curated rules for reaction enumeration in the Synthetically Accessible Virtual Inventory (SAVI) project.[11]

As an alternative to manual encoding of allowable transformations, heuristics for algorithmic extraction have been developed.[12−23] These algorithms build generalized rules from known reaction examples. Broadly speaking, they all identify the atoms that change connectivity as the reaction center. Different levels of generalization can then be used to extend that reaction center to include varying numbers of neighbors using either a fixed distance or heuristics that decide which neighboring atoms are *relevant*.

There are additional approaches to computer-aided retrosynthesis that avoid the need for reaction templates entirely. These include sequence-to-sequence models,[24] similarity-based methods,[25] and some graph-based methods that to date have been applied only to the problem of forward prediction.[26]

In this application note, we describe our approach to retrosynthetic template extraction and application. With the exception of a recent open-source implementation in C++ from Watson et al.,[23] there have been no published algorithms for template extraction to enable complete reproducibility. Moreover, this implementation does not explicitly discuss stereochemistry; stereochemical handling is rarely mentioned in the context of SMARTS template application.[4,27] This work makes two specific contributions:

(1) An open-source Python implementation of retrosynthetic template *extraction*, designed to operate on atom-mapped SMILES strings[28] (i.e., where the correspondence between atoms in the product and atoms in the reactants is known) and generate generalized SMARTS patterns.[29] While the procedure we describe also works for reactions in the forward synthetic direction, we focus on the retrosynthetic direction, where templates are applied to product molecules to generate one or more reactant/precursor molecules.

(2) RDChiral, an open-source Python implementation of retrosynthetic template *application*, designed to provide consistent handling of stereochemistry defined by template SMARTS strings.

IMPLEMENTATION

Both software contributions make extensive use of RDKit (version 2018.09.1),[30] which has become one of the most widely used frameworks for cheminformatics research.

Received: April 4, 2019
Published: June 13, 2019

Downloaded via IDAHO STATE UNIV on July 22, 2019 at 03:44:37 (UTC).
See https://pubs.acs.org/sharingguidelines for options on how to legitimately share published articles.

© 2019 American Chemical Society

DOI: 10.1021/acs.jcim.9b00286
J. Chem. Inf. Model. 2019, 59, 2529−2537

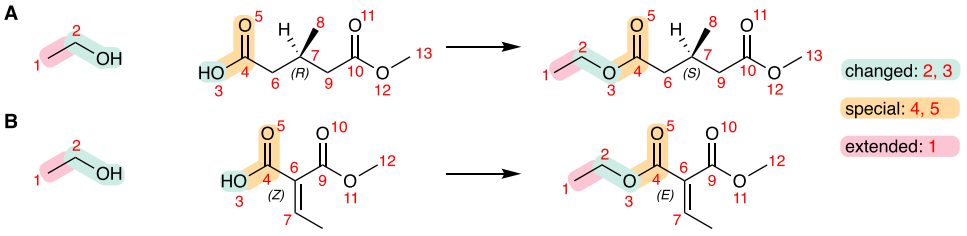


Figure 1. Reactions in which there is a change in absolute stereochemical assignment but the (A) atom or (B) bonds that are affected are not part of the reaction center and thus do not undergo a change in their *local* configuration.

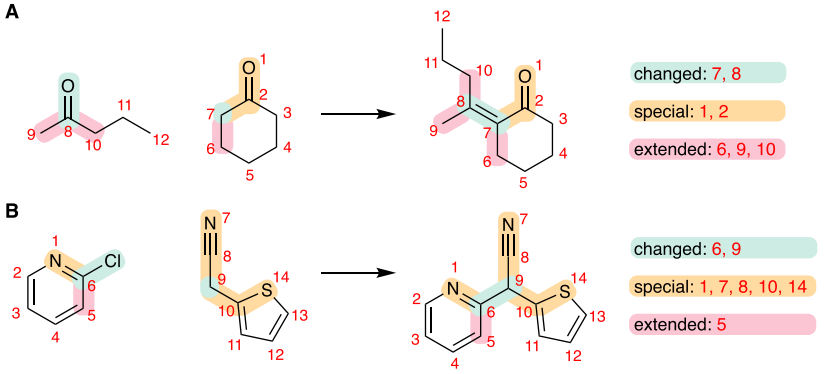


Figure 2. Two example reactions in which atoms belonging to "special" functional groups are added to the reaction template. This helps to ensure that the relevant chemical context for the reaction is present, even if it is more than a single bond away from the reacting atoms.

Template Extraction Procedure. The pipeline for template extraction begins with atom-mapped SMILES strings in the form `reactants≫major_product`. Only molecules that contribute heavy (non-hydrogen) atoms to the major product are required; spectator molecules are discarded. Not all product atoms need to be atom-mapped; it is assumed that small fragments of fewer than five atoms (e.g., halogen, oxygen) may come from unlisted reagents (e.g., Br_2, O_2) that were mistakenly excluded from the data set. Reactant molecules that are only partially mapped are allowed, provided that the unmapped atoms correspond to leaving groups. Each atom map number can appear only once in the product. In this code, we do not provide our own solution to reaction role assignment[31] or atom mapping,[32] both of which would be important if starting from uncleaned, unmapped reaction SMILES strings.

1. Check for Parsing Errors. If any of the reactant or product atoms fail to be parsed and sanitized (by RDKit), they are skipped. This is uncommon but can result from perceived valence rule violations if the source SMILES string is prepared by another program. Molecules are sanitized, which includes perceiving aromaticity and converting structures to their non-Kekulé form.

2. Check for Unmapped Product Atoms. If the number of unmapped product atoms exceeds a maximum allowed threshold (five), the reaction is skipped. Otherwise, the fragment that must be contributed by an unreported reagent is recorded. For example, the reaction `c1ccccc1≫[Br]c1ccccc1` (mapping omitted for brevity) suggests that a bromine atom, [Br], must come from a reagent.

3. Determine Which Atoms, by Map Number, Have Changed. First, the map numbers of the atoms in the reactants and product are identified. For each atom in the product, it is determined whether that atom's local properties are identical to how they were in the reactants. Two atoms are considered identical if they have the same SMARTS pattern, atomic number, total number of hydrogens, formal charge, degree, number of radical electrons, aromaticity, and bond order and atomic number of neighboring atoms. Any reactant atoms that are mapped but do not appear in the product are included in the list of changed atoms as well.

A more detailed version of this analysis is performed for any atoms that are tetrahedral centers that could have specified chirality: quaternary carbons with nonidentical side chains. The chirality of the atom—as determined by its clockwise/counterclockwise orientation and neighboring atoms—is checked. It should be noted that this must be a *local* check, as a chiral center's absolute stereochemical assignment (*R* or *S*) can change as a result of changes to the Cahn Ingold Prelog (CIP)[33] priority of its side chains (Figure 1); the same holds true for absolute cis/trans (*Z*/*E*) assignment. Equivalence of tetrahedral chirality is determined by comparing each atom's clockwise/counterclockwise assignment and the parity[34] of the sequence of neighboring atoms.

4. Define a SMARTS Pattern To Describe the Reactants. a. Define a Strict SMARTS Pattern for Each Reacting Atom. For each atom that changes between the reactants and products, including unmapped atoms belonging to leaving groups, a strict SMARTS pattern is defined. This SMARTS pattern includes atomic number, aromaticity, tetrahedral chirality (if applicable), number of hydrogens, degree, formal charge, and atom map number.

b. Identify Atom Membership in Special Functional Groups. When reacting atoms are part of a specific substructural motif, that whole motif should be included in the template. As an example, if a reacting atom is adjacent to an alkene or carbonyl, then the presence of that alkene or carbonyl is likely important for its reactivity. The two additional atoms belonging to the alkene or carbonyl are added to the list of neighboring atoms (Figure 2).

DOI: 10.1021/acs.jcim.9b00286
J. Chem. Inf. Model. 2019, 59, 2529−2537

We have defined roughly 30 such groups on the basis of conversations with expert synthetic chemists to cover a minimal set of common functional groups (Table S1). This is an extensible list and is not meant to be exhaustive or definitive. The list includes carboxylic acids, amides/sulfamides, boronic acids/esters, common protecting groups, alkenes/imines, alkynes/nitriles, adjacency to alkenes/alkynes/carbonyls, organometallics, diazo groups, adjacency to a heteroatom in a ring, two atoms away from a heteroatom in an aromatic ring, and trifluoromethyls. If reacting atoms belong to an alkene for which the cis/trans configuration is specified, the neighboring atoms required to fully define that local configuration—irrespective of the priority of side chains—are included as well.

c. Define a Generalized SMARTS Pattern for Each Neighboring Atom. For every atom that is a neighbor of a reacting atom (or belongs to a special functional group), a SMARTS atom pattern is defined. For terminal atoms with degree 1, this SMARTS pattern includes its atomic number, number of hydrogens, degree, formal charge, and atom map number (if applicable). For atoms with degree greater than 1, this SMARTS pattern includes only its atomic number, aromaticity, and formal charge.

d. Generate an Overall SMARTS Pattern for the Reactants. A molecular fragment containing only the atoms around the reaction center is converted to a SMILES string. Atoms' SMILES tokens are replaced by their custom SMARTS patterns to produce an overall SMARTS pattern for the reactants. All hydrogens and bonds are included explicitly.

The generated SMARTS pattern is matched back onto the reactant molecules to ensure a match. If there is a mismatch due to tetrahedral chirality, the orientation of chiral centers is flipped until a match can be found.

e. Record Auxiliary Information about the Transformation. If only one reactant molecule has changed, it is recorded that the eventual SMARTS transformation should be applied only as an intramolecular reaction. If two reactant molecules have changed but have identical SMILES strings when not atom-mapped, it is recorded that the SMARTS transformation should be applied only as a dimerization reaction.

5. Define a SMARTS Pattern To Describe the Product. a. Define a Strict SMARTS Pattern for Each Reacting or Unmapped Atom. For each atom that changed between the reactants and products, a strict SMARTS pattern is defined. This SMARTS pattern includes the atomic number, aromaticity, tetrahedral chirality (if applicable), number of hydrogens, degree, formal charge, and atom map number. Any unmapped product atoms are also included with this strict definition.

b. Include Additional Atoms Corresponding to Non-reacting Reactant Atoms. Mapped atoms in the reactants that were added to the SMARTS pattern but did not react (i.e., mapped atoms neighboring the reacting atoms or added as part of special functional groups) are identified. For each, a generalized SMARTS pattern is generated. For a terminal atom with degree 1, this SMARTS pattern includes its atomic number, degree, number of hydrogens, formal charge, and atom map number (if applicable). For an atom with degree greater than 1, this SMARTS pattern includes only its atomic number, aromaticity, and formal charge.

c. Generate an Overall SMARTS Pattern for the Product. A molecular fragment containing only the atoms around the reaction center is converted to a SMILES string. Atoms' SMILES tokens are replaced by their custom SMARTS patterns to produce an overall SMARTS pattern for the product. All hydrogens and bonds are included explicitly.

The generated SMARTS pattern is matched back onto the product molecules to ensure a match. If there is a mismatch due to tetrahedral chirality, the orientation of chiral centers is flipped until a match can be found.

6. Merge the Two Patterns into an Overall Retrosynthetic Reaction SMARTS Pattern. An attempt is made to canonicalize the order of disconnected reactant/product fragments by sorting the SMARTS patterns alphabetically. Atom map numbers are reassigned in the string to replace the original map numbers from the reactant and product molecules with a continuous sequence starting at 1. The final retrosynthetic template is created as a concatenation `product_smarts≫reactants_smarts`.

Template Application Procedure. The initial set of outcomes is generated by applying the template SMARTS string to a version of the product structure without defined stereochemistry. What follows is a list of steps to ensure that (a) a match should have occurred and (b) the stereochemsitry of the resulting precursors is as intended.

1. Initialize the rdchiralReaction Object from the SMARTS Pattern. A reaction is initialized from the SMARTS pattern. Also, auxiliary lists/dictionaries are created for faster processing: a dictionary mapping the atom map number in a template to the atom ID, a list of atoms in the template that could have tetrahedral chirality specified, and a list of double bonds that could have cis/trans configurations specified.

2. Initialize the rdchiralReactants Object from the SMILES String. A molecule is initialized from the input SMILES string. A copy of that molecule is created without stereochemical information. All tetrahedral centers and double bonds are identified, and it is determined whether they have defined stereochemical assignments (or whether that is even possible because of symmetry). Auxiliary lists/dictionaries are created for faster processing, including lists of these directional double bonds and tetrahedral centers.

3. Generate Outcomes Using the Non-stereodefined Input Compound. The initial set of precursors are generated using a copy of the input molecule without defined stereochemistry. This leads to *at least as many* matches as we would like to consider valid.

4. Ensure That the Tetrahedral Center Chiralities of the Input Molecule and the Template Match. If a reaction template has tetrahedral chirality defined in its SMARTS pattern, then the input molecule must also have its chirality defined. When there is only one tetrahedral center, it must be well-defined but need not match. When there are multiple tetrahedral centers, they must all match the template exactly or all be mirror images. That is, the same template will match two compounds that are enantiomers but will distinguish between diastereomers. This check is again done *locally*, as absolute stereocenter assignment can change as a result of transformations far from the stereocenter.

5. Ensure That the Double Bond Configurations of the Input Molecule and the Template Match. For each alkene in the input molecule, a check is done to see whether all of the atoms that would be required to define cis/trans stereochemistry were included in the reaction template. If the reaction template does not have defined stereochemistry but the product molecule does, a match is not allowed. If the reaction template does have defined stereochemistry, a check is done to ensure that the product molecule matches. This check is slightly complicated by

DOI: 10.1021/acs.jcim.9b00286
J. Chem. Inf. Model. 2019, 59, 2529−2537

the fact that, as with checking tetrahedral centers, it must be done on the basis of the local connectivity and not the absolute cis/trans assignment. The implied stereochemistry of double bonds found in aliphatic rings is taken into account during this consistency check.

6. Ensure That Intramolecular Reactions Are Applied Properly. Another quirk of retrosynthetic template application is that ring-opening reactions can sometimes lead to accidental fragmentation when applied intramolecularly (Figure 5B). Duplicate atoms in the reactants are detected, and the two precursor fragments are recombined.

7. Check the Chirality of Tetrahedral Precursor Atoms. If a precursor atom was generated by the reaction SMARTS pattern and not copied over (i.e., if it was part of a leaving group), it will have the correct chirality. If the atom did not match any part of the template, then the chirality is directly copied from the reactants. If the atom matched part of the template, then we must check whether it is possible for that template to have specified the chirality: if the atom in the template could *not* have had its chirality specified, then the chirality of the precursor atom is copied from the input molecule.

There are several cases to consider if the template input atom could have had its chirality specified. If the corresponding template output atom has unspecified chirality, then the generated precursor atom should have its chirality stripped (for a retrosynthetic template, this means that this is a stereoselective reaction in which the selectively is attributable to a reagent or catalyst, e.g., a proline-catalyzed aldol reaction). If the template input atom is unspecified, then the generated precursor atom has its chirality copied from the template output atom. If both the template input and template output atoms have specified chirality, then we must check whether the template describes the *retention* or *inversion* of chirality, again using a local definition. The chirality of the generated precursor atom is copied from the input atom and retained or inverted according to the template.

8. Check the Geometric Isomerism of Precursor Double Bonds. If both atoms across a carbon−carbon or carbon−nitrogen double bond in the generated precursor matched the template and it was possible for that template to specify cis/trans stereochemistry, then they will already have the correct configuration. If both atoms were created by the template (retrosynthetically, were part of a leaving group), then they will also have been instantiated as the correct stereoisomer. Otherwise, as in the case of tetrahedral chirality, the stereochemistry of the generated precursor double bond should be copied from the input molecule. This is again somewhat complex, as there are many ways to locally define the orientation of a double bond in the SMILES language and the absolute assignment of a double bond cannot be used to check for consistency (see Figure 3B).

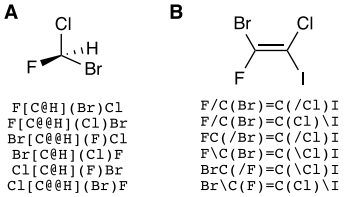


Figure 3. Nonexhaustive set of SMILES strings that all refer to the same molecular structure for (A) a molecule with tetrahedral chirality and (B) a molecule with cis/trans stereochemistry.

9. Merge the Enantiomeric Precursors into a Single Racemic Precursor. In cases where the set of possible precursors includes two enantiomers, we can optionally merge these into a single racemic precursor. This can happen in cases like the elimination of a chiral alcohol to form a racemic alkene; though the retrosynthetic template will generate chiral precursors from the alkene, the elimination product of either enantiomer will be the same alkene, so it is appropriate to combine the precursor suggestions into the racemic alcohol.

For speed, when the same template or same product molecule will be used numerous times (e.g., when a full library of retrosynthetic templates is applied to a library of candidate product molecules), a custom initialization can precompute many required properties to reduce computational cost.

RESULTS

Reaction Data Source. All of the examples in Figures 4 and S1 are from reactions contained in the open-source USPTO reaction data set containing ca. 1.8 million reactions.[35] This data set has been previously mapped using the Indigo toolkit, so all of the heavy atoms in the products are mapped.

Template Extraction. Several example reactions are presented in Figure 4 that show the identification of changed atoms (cyan), adjacent atoms belonging to special functional groups (dark yellow), and then the final extension to any atoms neighboring the changed atoms that were not part of such groups (pink). The resulting templates have varying degrees of specificity.

Figure 4A defines a highly general aromatic bromination with *N*-bromosuccinimide (NBS). Figure 4B defines the double alkylation of a catechol derivative with 1,2-dibromoethane. Figure 4C defines the acid hydrolysis of an alkyl nitrile using sulfuric acid. Figure 4D defines an amidation reaction between an aniline and an aliphatic acid chloride. Figure 4E produces a three-component reaction template between a primary thioamide, formaldehyde, and a disubstituted amine. Figure 4F is a simple deacetylation of a phenyl acetate. Figure 4G defines a template to prepare a 4-chloropyrimidine from a tautomer of 4-hydroxypyrimidine. Figure 4H defines the alkylation of a 2-hydroxypyridine using a tosyl alkylate. Figure 4I defines an epoxide opening using a monosubstituted amine. Figure 4J is a simple ethyl ester hydrolysis. Figure 4K is a Knoevenagel condensation between a benzaldehyde and a cyanoacetate. Figure 4L is another bromination with NBS, but of a methyl group on an aromatic ring. Figure 4M is an organometallic reaction between an aryllithium and a benzaldehyde.

Additional example reactions are presented in Figure S1 that show how the same process applies to reactions involving changes in stereochemistry.

Template Application. Several examples of retrosynthetic template application are shown in Figures 5 and 6. These are designed to showcase a range of situations where we may or may not require different behavior than what RDKit's standard function `RunReactants` can provide. The SMARTS string for each retrosynthetic template is included, as is a short description of the forward reaction. For all of these cases, RDChiral yields the "correct" precursors or lack thereof. The comparisons we show are meant to highlight the role of the RDChiral wrapper that is specifically tailored to retrosynthesis; they are not meant to reflect negatively on RDKit, which makes fewer assumptions about the intended use cases than we do.

Figure 5A is a single example of consistency between RDKit and RDChiral for the case where neither the template nor the

DOI: 10.1021/acs.jcim.9b00286
J. Chem. Inf. Model. 2019, 59, 2529−2537

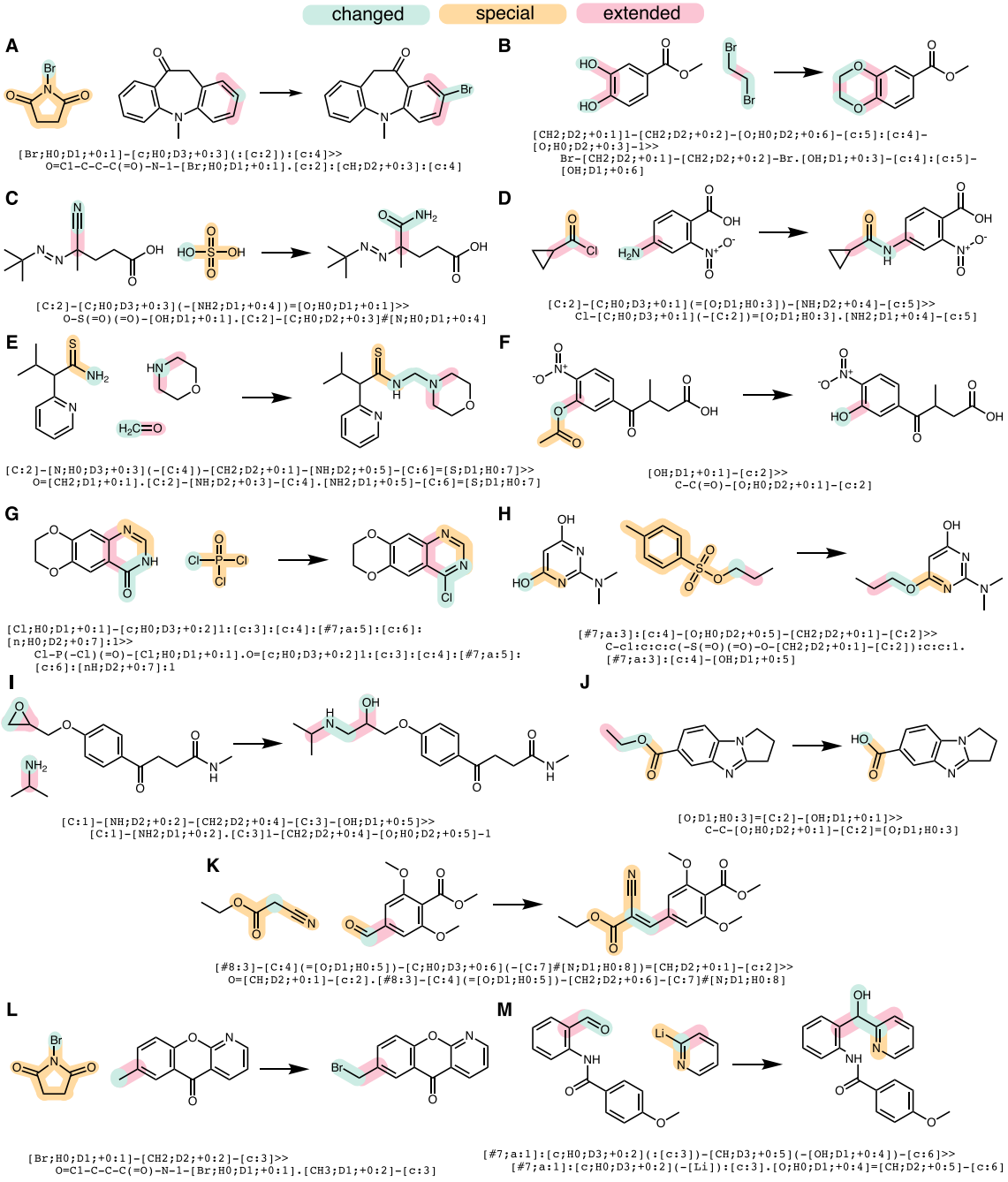


Figure 4. Examples of reactions from the USPTO and the retrosynthetic templates we extract from them. Atom mapping has been omitted for brevity but is unambiguous in all cases shown. Spectator molecules have also been omitted, as they do not contribute heavy atoms to the product and are not included in the resulting reaction template.

input product molecule contains any stereochemical information. Because RDChiral is a wrapper for RDKit, there will almost always be agreement in this non-stereodefined case. Figure 5B shows the one exception where applying a retrosynthetic disconnection to cleave a cyclic ester can result in inadvertent fragmentation of a molecule, which RDChiral can identify and correct.

When the product molecule has stereochemical information that is not part of where the template is matched, the standard behavior of RDKit is what we would like: local configurations are copied to the generated precursors (see the agreement in Figure 5C). When the stereochemical information is partially in the template, there are certain cases where RDChiral must restore that information in the generated precursors (Figure 5D).

Figure 5E demonstrates the most important role of RDChiral: preventing matches of stereodefined products with non-stereodefined templates when the substructure that matches the template could have specified stereochemistry (i.e., the matching substructure fully contains the atoms and bonds required to assign a stereochemical configuration). In these situations, generating precursors would mistakenly suggest that the forward reaction is guaranteed to be stereoselective or stereoretentive, which is *not* implied by the template. For both of the cases shown (defined tetrahedral center and defined double bond directionality), no precursors should be generated, and

DOI: 10.1021/acs.jcim.9b00286
J. Chem. Inf. Model. 2019, 59, 2529−2537

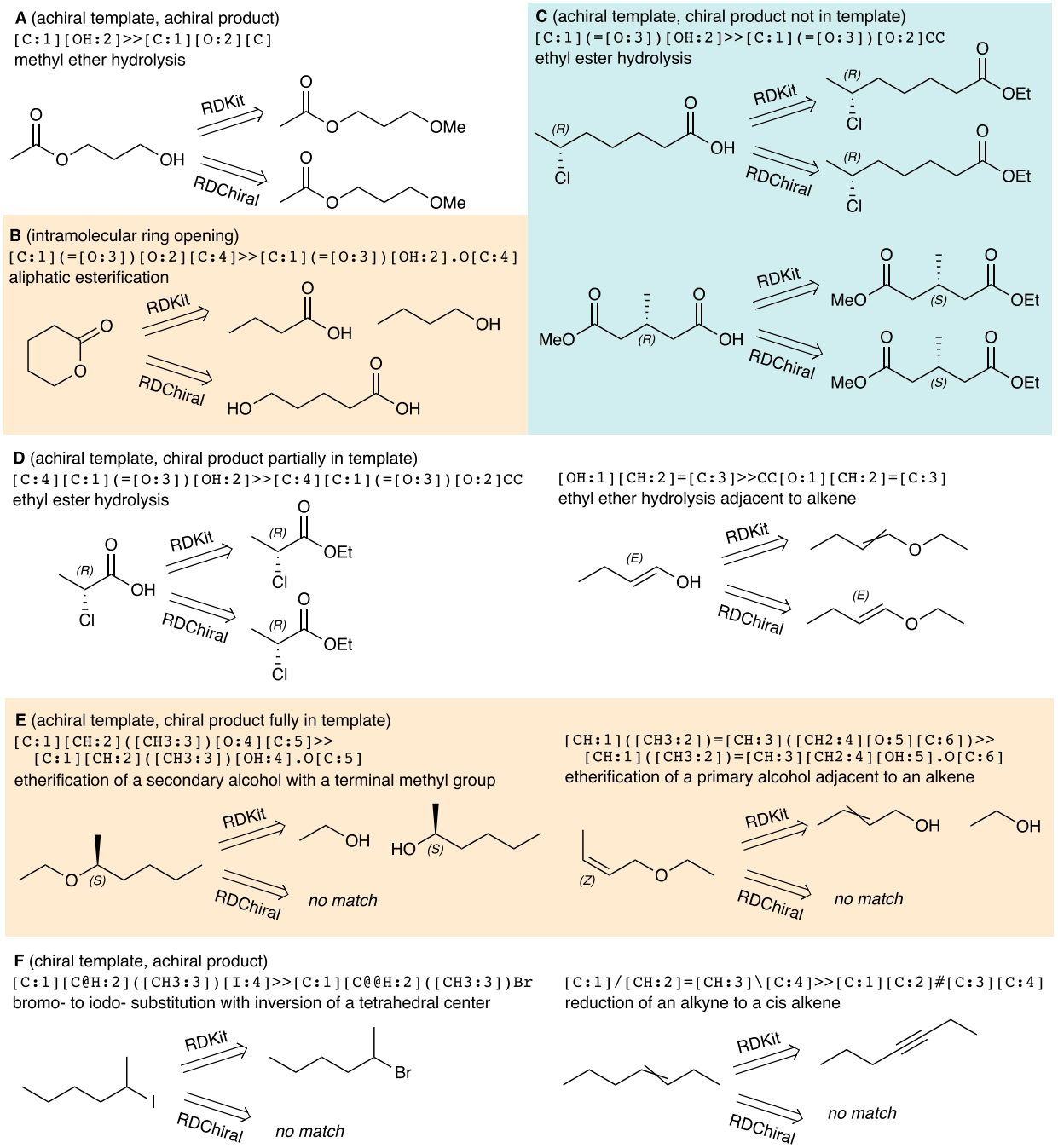


Figure 5. Examples of differences in retrosynthetic template application between the standard RDKit implementation and RDChiral for a variety of cases: (A) achiral template matched to an achiral product; (B) achiral template matched to an achiral product resulting in an intramolecular ring opening; (C) achiral template applied to a chiral product, where the chiral center is not part of the template; (D) achiral template applied to a chiral product, where the chiral center is part of the template but not fully specified; (E) achiral template applied to a chiral product, where the chirality is fully within the template; (F) chiral template applied to an achiral product. For brevity, in this figure "(a)chiral" is used to mean "(non)stereodefined" and describes both tetrahedral chirality and cis/trans stereochemistry interchangeably.

RDChiral ensures that this is the case. The reverse is also true (Figure 5F): matches between chiral templates and achiral products are not allowed.

The examples in Figure 6 show several cases where checking for consistency in retrosynthetic template application is important. Of particular note are the following: (1) Double bonds found in aliphatic rings are implicitly cis and will match templates requiring this double bond configuration (Figure 6A). (2) Inversion and retention of tetrahedral centers, as defined by the template, result in precursors whose chirality is consistent with or the opposite of the chirality in the product (rather than being generated on the basis of the right side of the template SMARTS) (Figure 6C,D). In the cases where each side of the template contains a single stereocenter, the specific clockwise/counterclockwise specification is ignored; the information we parse from the reaction SMARTS pattern is simply whether that stereocenter is inverted or preserved, as evidenced by the multiple SMARTS strings resulting in equivalent transformations. However, when multiple tetrahedral centers are present, the diastereomerism must match (Figure 6G).

DOI: 10.1021/acs.jcim.9b00286
J. Chem. Inf. Model. 2019, 59, 2529−2537

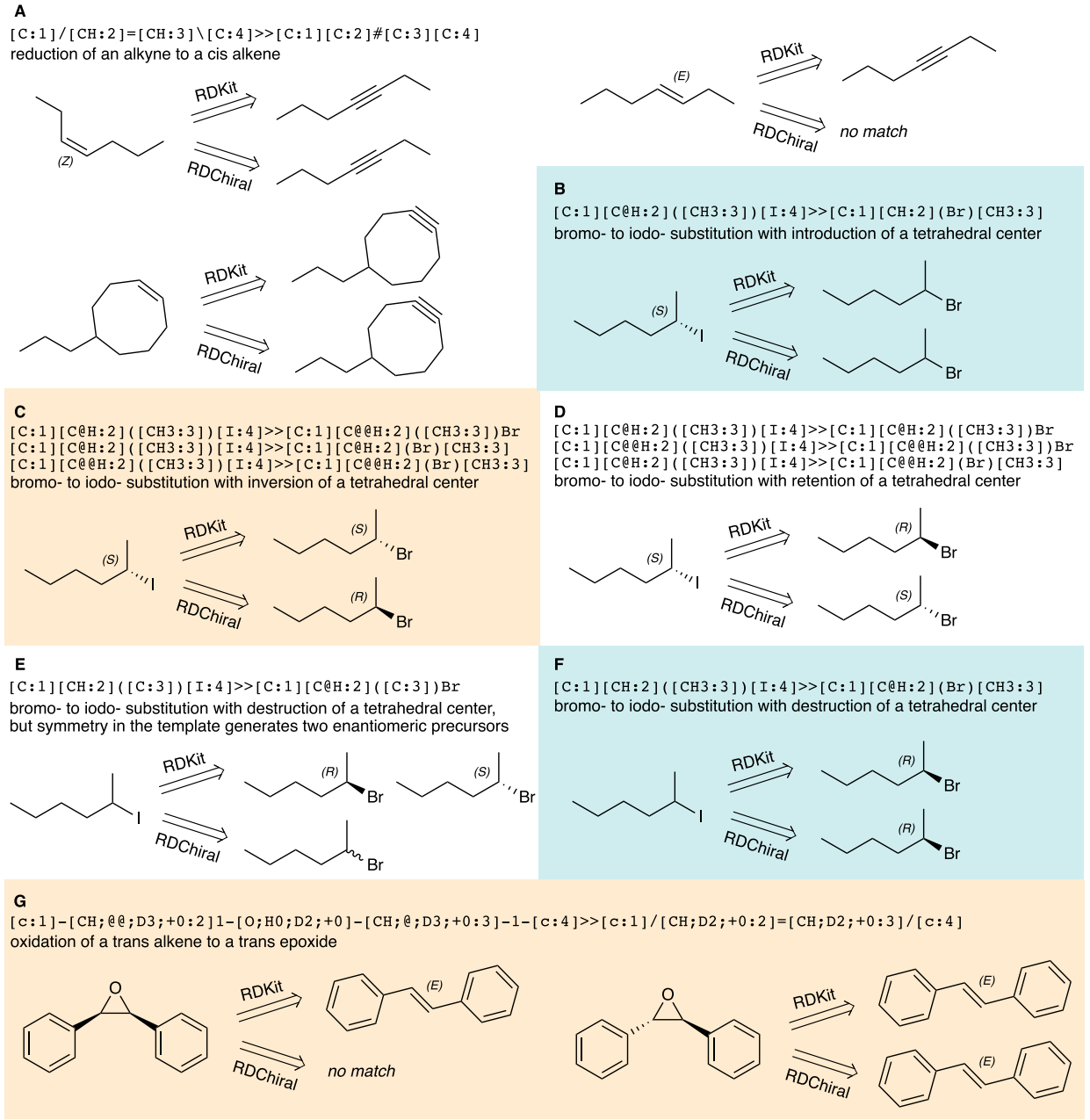


Figure 6. Additional examples of differences in retrosynthetic template application between the standard RDKit implementation and RDChiral for a variety of cases: (A) chiral template to produce a *cis*-alkene requires a cis product but recognizes implicit definitions in ring structures; (B) template that removes a tetrahedral center from the product; (C) template to invert a tetrahedral center, with several equivalent SMARTS strings; (D) template to retain a tetrahedral center, with several equivalent SMARTS strings; (E) template describing the destruction of a chiral center that yields racemic precursors because of symmetry on the right side of the template; (F) template describing the destruction of a chiral center that yields a single enantiomer as specified by the right side of the template; (G) template describing diastereoselective epoxidation that discriminates between diastereometers but not enantiomers.

CONCLUSION

We have described two interrelated pieces of open-source Python software based on RDKit that (A) extract retrosynthetic templates from atom-mapped SMILES strings using a dynamic definition of the relevant context surrounding the reaction center and (B) apply retrosynthetic templates in a manner that is faithful to the definition of stereochemical information (or lack thereof) in the template SMARTS pattern and input product molecule. This software has proved essential for our own synthetic planning workflows[36] in enabling increasingly complex small-molecule targets that necessitate careful consideration of stereochemistry.

ASSOCIATED CONTENT

Supporting Information

List of special functional groups and additional figures. The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jcim.9b00286.

(PDF)

DOI: 10.1021/acs.jcim.9b00286
J. Chem. Inf. Model. 2019, 59, 2529−2537

AUTHOR INFORMATION

Corresponding Authors

*E-mail: whgreen@mit.edu.
*E-mail: kfjensen@mit.edu.

ORCID

Connor W. Coley: 0000-0002-8271-8723
William H. Green: 0000-0003-2603-9694
Klavs F. Jensen: 0000-0001-7192-580X

Notes

The authors declare no competing financial interest.
All of the code used as well as Jupyter notebooks containing all of the examples included herein can be found at https://github.com/connorcoley/rdchiral.

ACKNOWLEDGMENTS

This work was supported by the DARPA Make-It Program under Contract ARO W911NF-16-2-0023. C.W.C. received additional funding from the NSF GRFP under Grant 1122374. We thank Mike Fortunato and Thomas J. Struble for helpful comments on this manuscript and aspects of the code.

REFERENCES

(1) Todd, M. H. Computer-Aided Organic Synthesis. *Chem. Soc. Rev.* **2005**, *34*, 247−266.
(2) Cook, A.; Johnson, A. P.; Law, J.; Mirzazadeh, M.; Ravitz, O.; Simon, A. Computer-Aided Synthesis Design: 40 Years On. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2012**, *2*, 79−107.
(3) Warr, W. A. A. Short Review of Chemical Reaction Database Systems, Computer-Aided Synthesis Design, Reaction Prediction and Synthetic Feasibility. *Mol. Inf.* **2014**, *33*, 469−476.
(4) Szymkuc, S.; Gajewska, E. P.; Klucznik, T.; Molga, K.; Dittwald, P.; Startek, M.; Bajczyk, M.; Grzybowski, B. A. Computer-Assisted Synthetic Planning: The End of the Beginning. *Angew. Chem., Int. Ed.* **2016**, *55*, 5904−5937.
(5) Coley, C. W.; Green, W. H.; Jensen, K. F. Machine Learning in Computer-Aided Synthesis Planning. *Acc. Chem. Res.* **2018**, *51*, 1281−1289.
(6) Corey, E. J.; Jorgensen, W. L. Computer-Assisted Synthetic Analysis. Synthetic Strategies Based on Appendages and the Use of Reconnective Transforms. *J. Am. Chem. Soc.* **1976**, *98*, 189−203.
(7) Hartenfeller, M.; Eberle, M.; Meier, P.; Nieto-Oberhuber, C.; Altmann, K.-H.; Schneider, G.; Jacoby, E.; Renner, S. A Collection of Robust Organic Synthesis Reactions for in Silico Molecule Design. *J. Chem. Inf. Model.* **2011**, *51*, 3093−3098.
(8) Avramova, S.; Kochev, N.; Angelov, P. RetroTransformDB: A Dataset of Generic Transforms for Retrosynthetic Analysis. *Data* **2018**, *3*, 14.
(9) Konze, K.; Bos, P.; Dahlgren, M.; Leswing, K.; Tubert-Brohman, I.; Bortolato, A.; Robbason, B.; Abel, R.; Bhat, S. Reaction-Based Enumeration, Active Learning, and Free Energy Calculations To Rapidly Explore Synthetically Tractable Chemical Space and Optimize Potency of Cyclin Dependent Kinase 2 Inhibitors. https://chemrxiv.org/articles/Reaction-based_Enumeration_Active_Learning_and_Free_Energy_Calculations_to_Rapidly_Explore_Synthetically_Tractable_Chemical_Space_and_Optimize_Potency_of_Cyclin_Dependent_Kinase_2_Inhibitors/7841270 (accessed March 18, 2019).
(10) ChemAxon. Reactor: A high performance virtual synthesis engine. https://chemaxon.com/products/reactor (accessed May 31, 2018).
(11) Synthetically Accessible Virtual Inventory (SAVI) Database. https://cactus.nci.nih.gov/download/savi_download/ (accessed Feb 12, 2019).
(12) Gelernter, H.; Rose, J. R.; Chen, C. Building and Refining a Knowledge Base for Synthetic Organic Chemistry Via the Methodology of Inductive and Deductive Machine Learning. *J. Chem. Inf. Model.* **1990**, *30*, 492−504.
(13) Satoh, H.; Funatsu, K. SOPHIA, a Knowledge Base-guided Reaction Prediction System-utilization of a Knowledge Base Derived From a Reaction Database. *J. Chem. Inf. Model.* **1995**, *35*, 34−44.
(14) Satoh, K.; Funatsu, K. A Novel Approach to Retrosynthetic Analysis Using Knowledge Bases Derived From Reaction Databases. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 316−325.
(15) Law, J.; Zsoldos, Z.; Simon, A.; Reid, D.; Liu, Y.; Khew, S. Y.; Johnson, A. P.; Major, S.; Wade, R. A.; Ando, H. Y. Route Designer: A Retrosynthetic Analysis Tool Utilizing Automated Retrosynthetic Rule Generation. *J. Chem. Inf. Model.* **2009**, *49*, 593−602.
(16) Christ, C. D.; Zentgraf, M.; Kriegl, J. M. Mining Electronic Laboratory Notebooks: Analysis, Retrosynthesis, and Reaction Based Enumeration. *J. Chem. Inf. Model.* **2012**, *52*, 1745−1756.
(17) Kraut, H.; Eiblmaier, J.; Grethe, G.; Löw, P.; Matuszczyk, H.; Saller, H. Algorithm for Reaction Classification. *J. Chem. Inf. Model.* **2013**, *53*, 2884−2895.
(18) Bogevig, A.; Federsel, H.-J.; Huerta, F.; Hutchings, M. G.; Kraut, H.; Langer, T.; Low, P.; Oppawsky, C.; Rein, T.; Saller, H. Route Design in the 21st Century: The ICSYNTH Software Tool as an Idea Generator for Synthesis Prediction. *Org. Process Res. Dev.* **2015**, *19*, 357−368.
(19) Coley, C. W.; Barzilay, R.; Jaakkola, T. S.; Green, W. H.; Jensen, K. F. Prediction of Organic Reaction Outcomes Using Machine Learning. *ACS Cent. Sci.* **2017**, *3*, 434−443.
(20) Segler, M. H. S.; Waller, M. P. Neural-Symbolic Machine Learning for Retrosynthesis and Reaction Prediction. *Chem. - Eur. J.* **2017**, *23*, 5966−5971.
(21) Segler, M. H. S.; Preuss, M.; Waller, M. P. Planning Chemical Syntheses With Deep Neural Networks and Symbolic AI. *Nature* **2018**, *555*, 604−610.
(22) Baylon, J. L.; Cilfone, N. A.; Gulcher, J. R.; Chittenden, T. W. Enhancing Retrosynthetic Reaction Prediction with Deep Learning Using Multiscale Reaction Classification. *J. Chem. Inf. Model.* **2019**, *59*, 673−688.
(23) Watson, I. A.; Wang, J.; Nicolaou, C. A. A Retrosynthetic Analysis Algorithm Implementation. *J. Cheminf.* **2019**, *11*, 1.
(24) Liu, B.; Ramsundar, B.; Kawthekar, P.; Shi, J.; Gomes, J.; Luu Nguyen, Q.; Ho, S.; Sloane, J.; Wender, P.; Pande, V. Retrosynthetic Reaction Prediction Using Neural Sequence-to-sequence Models. *ACS Cent. Sci.* **2017**, *3*, 1103−1113.
(25) Coley, C. W.; Rogers, L.; Green, W. H.; Jensen, K. F. Computer-Assisted Retrosynthesis Based on Molecular Similarity. *ACS Cent. Sci.* **2017**, *3*, 1237−1245.
(26) Coley, C. W.; Jin, W.; Rogers, L.; Jamison, T. F.; Jaakkola, T. S.; Green, W. H.; Barzilay, R.; Jensen, K. F. A Graph-convolutional Neural Network Model for the Prediction of Chemical Reactivity. *Chem. Sci.* **2019**, *10*, 370−377.
(27) Kochev, N.; Avramova, S.; Jeliazkova, N. Ambit-SMIRKS: a software module for reaction representation, reaction search and structure transformation. *J. Cheminf.* **2018**, *10*, 42.
(28) Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Model.* **1988**, *28*, 31−36.
(29) Daylight Chemical Information Systems. SMARTS - A Language for Describing Molecular Patterns. http://www.daylight.com/dayhtml/doc/theory/theory.smarts.html (accessed March 22, 2019).
(30) Landrum, G. RDKit: Open-Source Cheminformatics Software. http://www.rdkit.org (accessed Nov 20, 2016).
(31) Schneider, N.; Stiefl, N.; Landrum, G. A. What's What: The (Nearly) Definitive Guide to Reaction Role Assignment. *J. Chem. Inf. Model.* **2016**, *56*, 2336−2346.
(32) Chen, W. L.; Chen, D. Z.; Taylor, K. T. Automatic Reaction Mapping and Reaction Center Detection. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2013**, *3*, 560−593.
(33) Cahn, R. S.; Ingold, C.; Prelog, V. Specification of Molecular Chirality. *Angew. Chem., Int. Ed. Engl.* **1966**, *5*, 385−415.

DOI: 10.1021/acs.jcim.9b00286
J. Chem. Inf. Model. 2019, 59, 2529−2537

(34) Dalke, A. Faster parity calculation. http://www.dalkescientific.com/writings/diary/archive/2016/08/15/fragment_parity_calculation.html (accessed June 18, 2017).
(35) Lowe, D. M. Patent Reaction Extraction: Downloads. https://bitbucket.org/dan2097/patent-reaction-extraction/downloads (accessed May 31, 2018).
(36) Coley, C. W. ASKCOS. http://askcos.mit.edu/ (accessed Feb 8, 2019).

DOI: 10.1021/acs.jcim.9b00286
J. Chem. Inf. Model. 2019, 59, 2529−2537