

Knowledge-Based Approach to *de Novo* Design Using Reaction Vectors

Hina Patel,[†] Michael J. Bodkin,[§] Beining Chen,[‡] and Valerie J. Gillet^{*†}

Department of Information Studies, University of Sheffield, Regent Court, 211 Portobello Street, Sheffield S1 4DP, U.K., Department of Chemistry, University of Sheffield, Western Bank, Sheffield S10 2TN, U.K., and Eli Lilly U.K., Erl Wood Manor, Windlesham, Surrey GU20 6PH, U.K.

Received November 9, 2008

A knowledge-based approach to the *de novo* design of synthetically feasible molecules is described. The method is based on reaction vectors which represent the structural changes that take place at the reaction center along with the environment in which the reaction occurs. The reaction vectors are derived automatically from a database of reactions which is not restricted by size or reaction complexity. A structure generation algorithm has been developed whereby reaction vectors can be applied to previously unseen starting materials in order to suggest novel syntheses. The approach has been implemented in KNIME and is validated by reproducing known synthetic routes. We then present applications of the method in different drug design scenarios including lead optimization and library enumeration. The method offers great potential for capturing and using the growing body of data on reactions that is becoming available through electronic laboratory notebooks.

INTRODUCTION

de Novo design tools first appeared nearly two decades ago with more than twenty programs developed within a few years;^{1–3} however, despite this intense activity, such tools are rarely commonplace on the medicinal chemist's desktop. Many of the early *de novo* programs were focused on building molecules in three-dimensions in the context of a receptor site and scoring them on fit to the site, for example, LUDI,⁴ GrowMol,⁵ and SPROUT.⁶ However, it soon became abundantly clear that the lack of synthetic feasibility and the overly complex nature of the structures generated in *de novo* design are its Achilles' heel.^{3,7} The difficulty in persuading chemists to adopt such tools is further compounded by the subjective nature of synthetic accessibility as highlighted recently by Lajiness et al., who reported on the inconsistencies of medicinal chemists in assessing the druglikeness and/or synthetic feasibility of compounds.⁸ Furthermore, it is now clear that drug discovery is a multiobjective optimization problem, and multiple design criteria should be taken into account when designing novel molecules, for example, pharmacokinetic properties should be optimized alongside predicted potency.⁹

More recently methods that attempt to account for synthetic accessibility have been developed.^{10–14} A typical approach is to encode a fixed set of known transformations which can then be applied to starting molecules. For example, the Flux program^{10,11} is a fragment-based *de novo* design tool in which a retro-synthesis algorithm based on the earlier RECAP program¹⁵ is used to fragment molecules into building blocks. The same rules, defined as SMIRKS,¹⁶ are then used to link together fragments to generate new

molecules. Schürer et al.¹² also describe an approach based on SMIRKS representations with additional filters being used to indicate compatibility of the different building blocks and a user-interface provided to facilitate the definition of the filters. A common limitation of these approaches is the restricted number of "reactions" that can be carried out. For example, the Flux program is based on ten or eleven of the RECAP bond cleavage rules.¹⁵ Although the number of different reaction types encoded in the SYNOPSIS program¹³ is significantly higher at seventy, the complexity of this program is increased by the requirement to explicitly define rules to indicate the conditions under which a reaction can occur.

An alternative strategy is to score ligands on their synthetic complexity postgeneration. Thus, Boda and Johnson have developed a complexity score that is based on the statistical distribution of fragments in known drugs and commercially available starting materials.⁷ However, such an approach loses the ability to directly iterate and optimize the design in an efficient manner. Somewhat in contrast to this, so-called 'interactive evolution' is used in the Molecule Evuator¹⁴ whereby the user acts as the fitness function, accepting or rejecting mutations in an interactive manner. In terms of synthetic feasibility this must be limited by the user's knowledge of chemistry.

The mimicking of medicinal chemistry schema using reaction-like approaches has also been employed in the data mining and QSAR fields. For example, the 'matched molecular pairs' approach mimics the steps taken by a medicinal chemist in assessing a change in property per single structural change.¹⁷ A similar approach is reported by Sheridan et al.¹⁸ with a focus on local QSAR. In their approach, pairs of molecules are clustered based on descriptor difference vectors which are formed by subtracting the vector representation of one molecule as "product" from the other as "reactant". Thus, the pairs of molecules within a cluster should represent

* Corresponding author phone: +44-1142-222652; fax: +44-1142-780300; e-mail: v.gillet@sheffield.ac.uk.

[†] Department of Information Studies, University of Sheffield.

[§] Eli Lilly U.K.

[‡] Department of Chemistry, University of Sheffield.

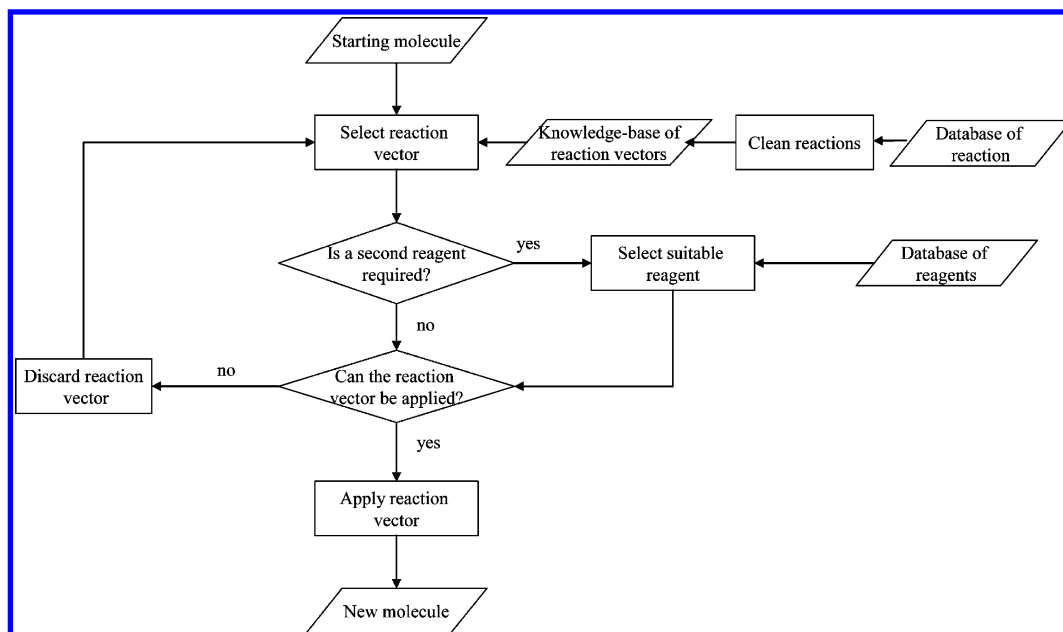


Figure 1. Flowchart of the use of reaction vectors in *de novo* design.

similar transformations. An analysis of the changes in activities associated with the pairs can then indicate whether a particular transformation tends to increase, decrease, or has a mixed effect on activity.

In this paper, we describe a knowledge-based approach to *de novo* design. The changes that take place in reactions are captured in what are known as reaction vectors which are closely related to the descriptor difference vectors described by Sheridan et al.¹⁸ A structure generation algorithm has been developed in which reaction vectors are applied to previously unseen starting materials in order to generate novel molecules. Each transformation that is applied has a precedent in the literature, and thus a high degree of confidence is established in the synthetic feasibility of the resulting molecules. The knowledge-base is easily constructed and can be built from any number of reactions. Thus the method is not limited by reaction type or number and is able to deal with a variety of reaction types from simple functional group interconversions through to complex rearrangements. The approach is well suited to integration with electronic laboratory notebooks (ELNs) which have been widely adopted by the pharmaceutical industry.^{19,20} ELNs provide an unprecedented opportunity to gather information electronically on the successes and failures of reactions undertaken by bench chemists, and our approach to *de novo* design provides an opportunity to mine and learn from this freshly acquired data. Here, we describe how single-step transformations that are derived from a reactions database can be applied to a starting molecule to generate novel molecules. In a subsequent paper, we will describe the incorporation of single step transformations into an iterative multiobjective *de novo* design program.

METHODOLOGY

The heart of the method is the capture of the knowledge contained in a set of reactions into reaction vectors, which essentially describe both the changes at the reaction center together with the environment in which the reaction occurs. The changes encoded in a reaction vector can then be applied

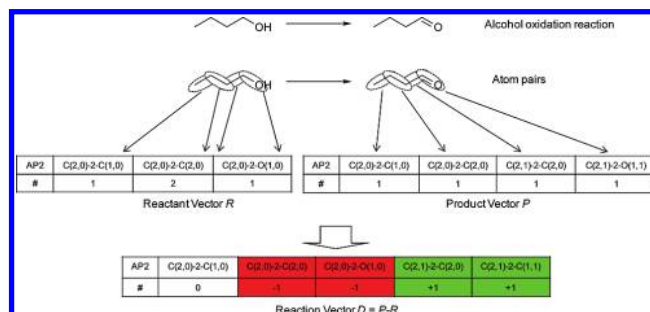


Figure 2. The generation of the reaction vector is shown for a simple transformation using atom pair descriptors.

to input molecules in order to suggest novel molecules for synthesis. A link back to the reaction(s) from which the reaction vector was derived can also be provided to substantiate the suggested transformation. Thus, for a given starting material, reaction vectors are extracted from the knowledge-base, and, if applicable, the changes encoded in the vector are applied to the starting material to generate one or more product molecules. The application of a reaction vector may also require a second reagent for success.

The methodology section is organized as follows. First a brief introduction to reaction vectors is provided, and the reaction vectors used in this work are described. An initial investigation of reaction databases revealed that many reactions are incomplete, and so a reaction cleaning algorithm has been implemented and is described next. The structure generation algorithm is then described in which the changes indicated in a reaction vector can be applied to generate novel molecules. Finally the implementation of the various components into a desktop tool is described. The overall process is illustrated in Figure 1.

Reaction Vectors. Reaction vectors were employed by Broughton et al. for classifying and searching chemical reactions.²¹ A reaction vector encodes the changes that take place during a reaction as the descriptors that are gained in the product(s) and those lost from the reactant(s). A reaction vector is generated as follows: the individual components of a reaction are described using vector-based descriptors

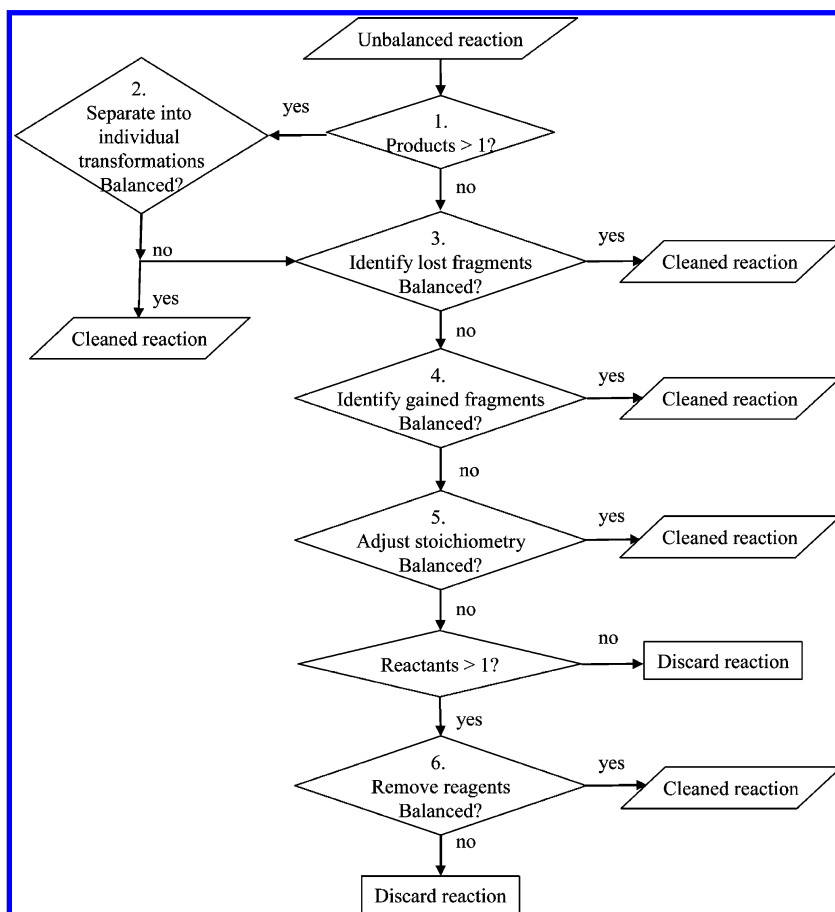


Figure 3. Overview of the reaction “Cleaning” algorithm.

Table 1. Examples of Incomplete Reactions

Incomplete Reaction	Example
A fragment in the reactant is not accounted for in the products	
A fragment in the product is not accounted for in the reactants.	
A reagent is shown in the reactants and is not accounted for in the products	
The reaction is not stoichiometrically balanced	
The reaction contains more than one product (stereoisomer, structural isomer or side product).	

such as atom pairs or topological torsions; a reactant vector, R , is generated by summing the vectors of the individual reactants (e.g., $R = R_1 + R_2$); a product vector, P , is generated by summing the vectors of the individual products (e.g., $P = P_1 + P_2$); and the reaction vector, D , is calculated by subtracting the reactant vector from the product vector as shown in eq 1.

$$D = P - R \quad (1)$$

Figure 2 illustrates the generation of a reaction vector for a simple alcohol dehydration reaction using atom pair descriptors of the form $X1(h,p)-X2(h,p)$ where $X1$ and $X2$ are the atomic symbols, h is the number of non-hydrogen connections, p is the number of π electrons, and the “2” indicates atom pairs at one bond separation. The reactant

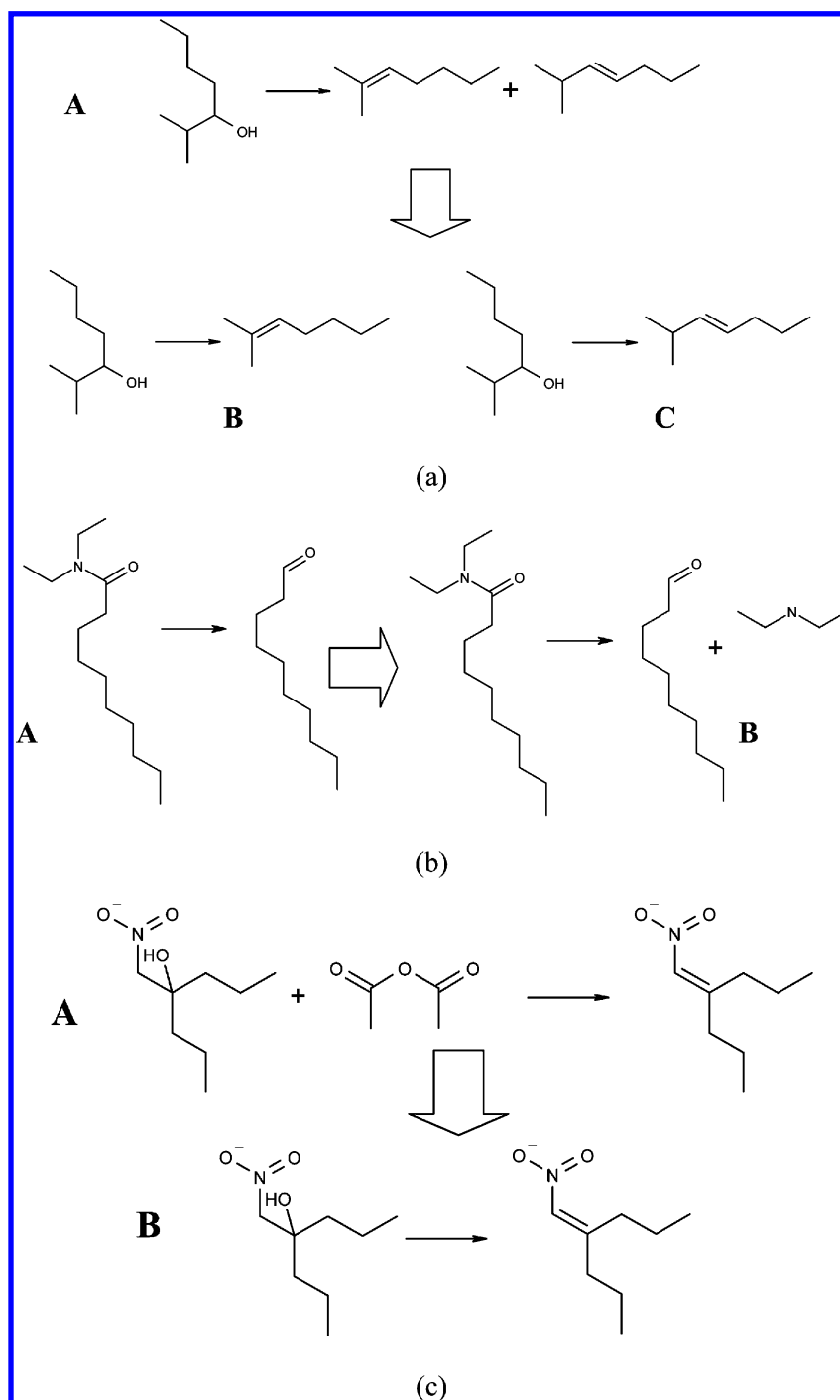


Figure 4. (a) An alcohol dehydration reaction (A) consisting of multiple products which is cleaned by separating it into two individual reactions (B and C). (b) An amide reduction (A) with a missing fragment which is cleaned by adding the missing fragment as a new molecule in the products (B). (c) An alcohol dehydration reaction (A) which is cleaned by removing an unnecessary reactant (B).

and product vectors record the number of occurrences of the constituent atom pairs, and the reaction vector contains positive and negative counts of atom pairs. The negative counts (red) indicate the bonds that are removed from the reactant(s), while the positive counts (green) indicate the bonds that must be added to the reactant(s) to create the product(s).

The reaction from which a reaction vector is derived is referred to here as the *parent reaction*. In general, there is a one-to-many relationship between reaction vectors and parent reactions, that is, multiple reactions may be represented by the same reaction vector. The degree of redundancy is dependent on the descriptors used so that the number of

reactions encoded by a reaction vector will be larger for short-range descriptors than for longer ranging descriptors which encode more of the environment of the reaction. For example, a reaction vector constructed from atom pairs at one bond separation describes only the bonds that are changed during the reaction (more precisely bonds whose incident atoms are directly involved in the reaction) and does not take into account the environment in which the bonds occur. However, the inclusion of atom pairs separated by two bonds ($X1(h,p)-3-X2(h,p)$) will extend the environment of the reaction that is encoded, and the greater the environment the more specific is the reaction vector. At the limit, the complete environment of a reaction can be encoded with

```

Generate the reactant vector from the input molecule.
Generate the product vector from the reactant vector and the reaction
vector.
Map the negative descriptors in the reaction vector to descriptors in
the reactant(s) in all possible ways.
For each mapping,
    Remove the corresponding atoms and bonds from the reactant(s)
    Input the reactant fragment, the reaction vector (positive
    descriptors) and the product vector to the structure generation
    algorithm.
  
```

Figure 5. The steps involved in applying a reaction vector to an input molecule.

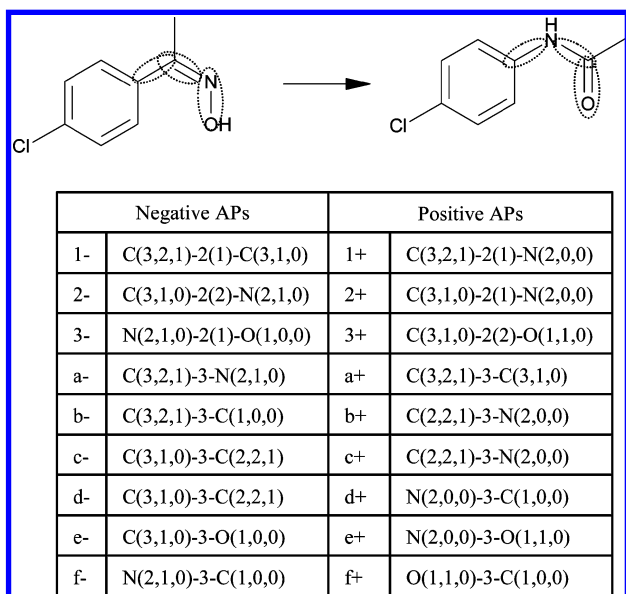


Figure 6. A Beckmann rearrangement reaction together with the reaction vector. The reaction vector is divided into negative and positive descriptors with the AP2s identified by integer labels and the AP3s identified by character labels; a minus symbol (“-”) following a label indicates APs that are lost from the reactant, and the addition symbol (“+”) indicates APs that are gained in the formation of the product. The three bonds lost from the reactant and the three bonds gained in the product are also highlighted in the reaction.

Table 2. Reaction Data Sets Used To Compile Sets of Reaction Vectors for the Experiments Reported Below

	5K data set	26K data set
starting number of rxns	5283	26616
after cleaning	6016	26676
removal of rxns with > 2 reactants and 2 products	5695	24418
unique rxn vectors	2866	16859

the result that each reaction vector is unique (describes a single reaction only).

Reaction Vectors for *de Novo* Design. In *de novo* design, the aim is to use reaction vectors to suggest synthetic routes to *novel* molecules, and it is necessary to achieve a balance in specificity and generality: the reaction vectors should be sufficiently specific to avoid their application in environments which are known to prevent a reaction from occurring, yet encoding of the entire environment will prevent their application to the generation of novel, previously unseen molecules. The combination of atom pairs at one and two

bonds separation was found to provide an effective balance between generalizing the reactions to allow novel molecules to be generated, while maintaining specificity.²² The reaction vectors used in this work are modified versions of the original descriptors developed by Cahart et al.²³ and are defined as follows

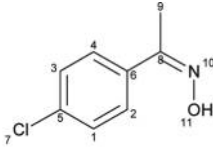
$$\text{AP2: } X1(h, p, r) - 2(\text{BO}) - X2(h, p, r)$$

$$\text{AP3: } X1(h, p, r) - 3 - X2(h, p, r)$$

where $X1$ and $X2$ are the chemical symbols of the two atoms, h is the number of non-hydrogen connections, p is the number of π bonds, r is the number of rings the atom is a member of and BO is the bond order (1 = single bond, 2 = double bond, 3 = triple bond, and 4 = aromatic bond). The inclusion of the ring descriptor r enables a reaction that takes place within a ring environment to be distinguished from one in an acyclic environment. The atom and bond attributes of the reaction vector are derived using the JOELib toolkit.²⁴

Reaction Cleaning Algorithm. An initial investigation of reactions in the Lilly database revealed many of them to be incomplete with the number and types of atoms in the reactants and products unbalanced. Examples are shown in Table 1. While it would still be possible to generate a reaction vector for such a reaction, the resulting vector would not be a true representation of the changes that take place during the reaction and would contain noise, for example, additional negative or positive descriptors that are not part of the true reaction center. Such noise in the reaction vectors will affect the ability to apply them to generate novel, sensible molecules and could result in reactions of the same type being represented by distinct reaction vectors. Therefore, an automated reaction “cleaning” algorithm has been implemented that seeks to balance the number of carbon atoms on each side of the equation and is applied prior to calculating the reaction vectors.

An overview of the reaction cleaning algorithm is given in Figure 3. Incomplete reactions that require cleaning are easily identified using a simple count of the carbons atoms in the reactants and products: if the number of carbon atoms in the reactants does not equal the number in the products, then the reaction is passed to the cleaning algorithm. Step 1 determines the number of reactants and products in the reaction. If there is more than one product, step 2 attempts to separate the reaction into individual reactions. For example, the reaction $R1 + R2 \rightarrow P1 + P2$ would become the following: $R1 + R2 \rightarrow P1$ and $R1 + R2 \rightarrow P2$. The carbon count test is then applied to each reaction, and if all reactions are balanced they are passed as clean and each

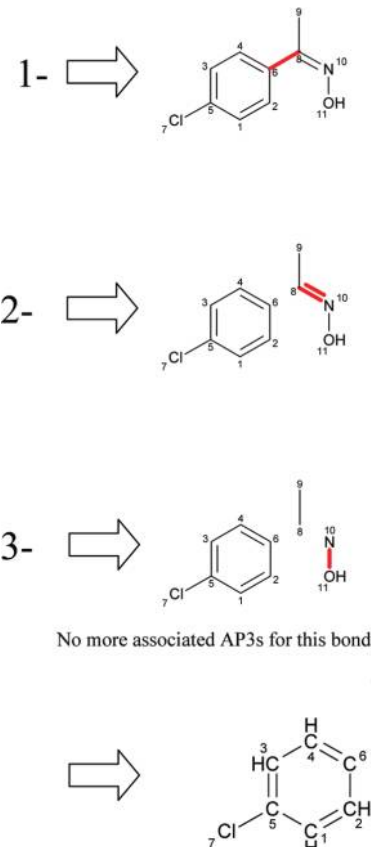


Atom	Type	<i>h</i>	<i>p</i>	<i>r</i>
1	C	2	2	1
2	C	2	2	1
3	C	2	2	1
4	C	2	2	1
5	C	3	2	1
6	C	3	2	1
7	Cl	1	0	0
8	C	3	1	0
9	C	1	0	0
10	N	2	1	0
11	O	1	0	0

AP2	Atom1	Atom2	Bond Order
C(2,2,1)-2(4)-C(2,2,1)	1	2	4
C(2,2,1)-2(4)-C(2,2,1)	3	4	4
C(3,2,1)-2(4)-C(2,2,1)	5	1	4
C(3,2,1)-2(4)-C(2,2,1)	5	3	4
C(3,2,1)-2(4)-C(2,2,1)	6	2	4
C(3,2,1)-2(4)-C(2,2,1)	6	4	4
C(3,2,1)-2(1)-Cl(1,0,0)	5	7	1
C(3,2,1)-2(1)-C(3,1,0)	6	8	1
C(3,1,0)-2(1)-Cl(1,0,0)	8	9	1
C(3,1,0)-2(2)-N(2,1,0)	8	10	2
N(2,1,0)-2(1)-O(1,0,0)	10	11	1

AP3	Atom1	Atom2
C(3,2,1)-3-C(2,2,1)	6	1
C(3,2,1)-3-C(2,2,1)	6	3
C(3,2,1)-3-C(2,2,1)	5	2
C(3,2,1)-3-C(2,2,1)	5	4
C(2,2,1)-3-C(2,2,1)	1	3
C(2,2,1)-3-C(2,2,1)	2	4
C(3,1,0)-3-C(2,2,1)	8	2
C(3,1,0)-3-C(2,2,1)	8	4
C(2,2,1)-3-Cl(1,0,0)	3	7
C(2,2,1)-3-Cl(1,0,0)	1	7
C(3,2,1)-3-Cl(1,0,0)	6	9
C(3,2,1)-3-N(2,1,0)	6	10
C(3,1,0)-3-O(1,0,0)	8	11
N(2,1,0)-3-C(1,0,0)	10	9

Figure 7. The atom table of the reactant is shown on the left (*h* is the number of non-hydrogen connections, *p* is the number of π bonds, and *r* is the number of ring memberships). The bond table is shown in the center, and each bond is associated with an AP2 descriptor. The bonds that map to the reaction vector and are to be removed are highlighted in red.



AP3	Atom1	Atom2	AP3	Atom1	Atom2
C(3,2,1)-3-C(2,2,1)	6	1	C(3,1,0)-3-C(2,2,1)	8	4
C(3,2,1)-3-C(2,2,1)	6	3	C(2,2,1)-3-Cl(1,0,0)	3	7
C(3,2,1)-3-C(2,2,1)	5	2	C(2,2,1)-3-Cl(1,0,0)	1	7
C(3,2,1)-3-C(2,2,1)	5	4	C(3,2,1)-3-Cl(1,0,0)	6	9
C(2,2,1)-3-C(2,2,1)	1	3	C(3,2,1)-3-N(2,1,0)	6	10
C(2,2,1)-3-C(2,2,1)	2	4	C(3,1,0)-3-O(1,0,0)	8	11
C(3,1,0)-3-C(2,2,1)	8	2	N(2,1,0)-3-C(1,0,0)	10	9

AP3	Atom1	Atom2	AP3	Atom1	Atom2
C(3,2,1)-3-C(2,2,1)	6	1			
C(3,2,1)-3-C(2,2,1)	6	3	C(2,2,1)-3-Cl(1,0,0)	3	7
C(3,2,1)-3-C(2,2,1)	5	2	C(2,2,1)-3-Cl(1,0,0)	1	7
C(3,2,1)-3-C(2,2,1)	5	4			
C(2,2,1)-3-C(2,2,1)	1	3			
C(2,2,1)-3-C(2,2,1)	2	4	C(3,1,0)-3-O(1,0,0)	8	11
			N(2,1,0)-3-C(1,0,0)	10	9

AP3	Atom1	Atom2	AP3	Atom1	Atom2
C(3,2,1)-3-C(2,2,1)	6	1			
C(3,2,1)-3-C(2,2,1)	6	3	C(2,2,1)-3-Cl(1,0,0)	3	7
C(3,2,1)-3-C(2,2,1)	5	2	C(2,2,1)-3-Cl(1,0,0)	1	7
C(3,2,1)-3-C(2,2,1)	5	4			
C(2,2,1)-3-C(2,2,1)	1	3			
C(2,2,1)-3-C(2,2,1)	2	4			

No more associated AP3s for this bond

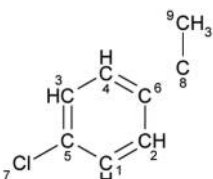


Figure 8. Removal of bonds and associated AP3 descriptors according to the negative AP2s in the reaction vector and the resulting fragment. See text for explanation.

reaction is used to generate a reaction vector, otherwise the reactions are passed to step 3.

Steps 3 and 4 require atom mapping values to be present in the input file (an MDL rxn file²⁵). The rxn file has a structured format with the reactants and products clearly identified, and atom mapping values, when present, are used to indicate the correspondence between atoms in the products and atoms in the reactants. Atom mapping values are usually

generated (either automatically or manually) when a reaction is first stored; however, they are optional, and many rxn files do not contain this information. When atom mapping information is present, atoms with mapping values of zero in the reactants indicate that they are not present in the products and vice versa for atoms assigned mappings of zero in the products. Step 3 attempts to find fragments that are missing from the products by first identifying atoms in the

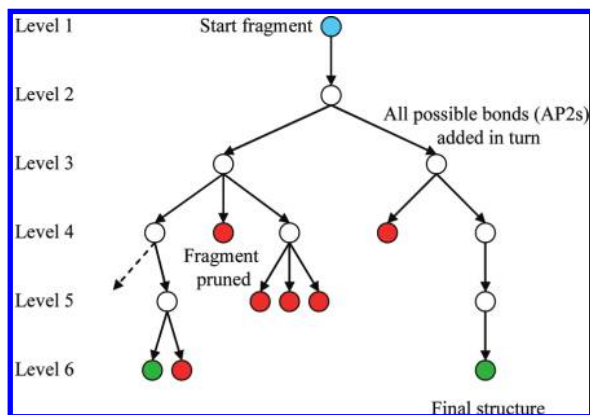


Figure 9. Structure generation proceeds as a breadth-first search from the start fragment. In moving from one level of the tree to the next, each fragment is extended by one bond in all possible ways based on the positive AP2 descriptors in the reaction vector. The addition of a bond may result in a new atom being added, two disconnected fragments being connected, or in a ring closure. A red node represents a fragment that cannot be used any further, for example, it contains AP3 descriptors that are not present in the product vector, and so is pruned from the tree; the green nodes represent complete structures, i.e., all positive descriptors have been used and all valencies in the structures are satisfied.

reactants with mapping values of zero and then combining them to generate a new product molecule. The carbon count test is then applied to check if the reaction is now “clean”. In a similar manner, step 4 attempts to construct missing reactants based on atoms in the product with mapping values of zero.

If a reaction remains unbalanced and reaches step 5, an attempt is made to balance it stoichiometrically by generating additional instances of each reactant and each product incrementally and applying the carbon count test after each addition. Any reaction that fails this step and consists of a single reactant is rejected. Reactions that contain more than one reactant proceed to step 6 which attempts to remove each reactant in turn, and each combination of reactants where there are more than two reactants, before applying the carbon count test. This last step also allows reagents that are not directly involved in the reaction to be identified and discarded. Any reactions that are not clean at this stage are rejected.

Figure 4 shows examples of reactions “cleaned” by the algorithm: Figure 4(a) shows a reaction with more than one product that is cleaned by separating the reaction into individual reactions (step 2); Figure 4b shows a reaction with a missing fragment in the products (side product) that is cleaned by generating a new molecule (step 4) to take account of the atoms and bonds missing from the reaction; Figure 4(c) shows a reactant that is not actually used in the reaction (i.e., it is a reagent) and is removed in step 6.

Applying Reaction Vectors. Given a reaction vector (D) and the reactant vector (R) from which it was derived, the product vector (P) of the reaction is easily calculated by the simple rearrangement of eq 1, as shown in eq 2:

$$P = D + R \quad (2)$$

Assuming the product consists of a single component, then the product molecule can be generated from the reactant, the reaction vector and product vector using the structure generation algorithm detailed below. The procedure described

in eq 2 enables the product of the parent reaction to be generated from a reaction vector and the parent reactant, that is, it allows a known reaction to be reproduced. For *de novo* design applications, the generation of novel molecules can be achieved by applying a reaction vector to a reactant (or starting material) other than that in the parent reaction, as shown in eq 3

$$P' = D + R' \quad (3)$$

where R' represents a reactant other than that used to derive the reaction vector (D) and P' represents a novel product. As discussed earlier, the reaction vector encodes the changes at the reaction center as negative and positive descriptors that represent bonds lost from the reactant and gained in the product, respectively. Since the negative descriptors indicate bonds that are lost from the reactants, in order for a reaction vector to be applicable to a previously unknown reactant, each negative descriptor in the reaction vector must be present in the reactant (with count greater than or equal to the absolute magnitude in the reaction vector).

Structure Generation Algorithm. We first describe the structure generation algorithm to regenerate the product for a known reaction, i.e., the generation of the product of the parent reaction from the reactant and the reaction vector. We then describe the additional steps required in order to apply the algorithm to the generation of novel molecules.

The steps involved in applying a reaction vector are shown in Figure 5 and are illustrated using the Beckmann rearrangement in Figure 6. The reactant vector is generated by analyzing the connection table of the reactant. The product vector is generated by summing the reactant vector and reaction vector, according to eq 2. The next step is to remove atoms and bonds from the reactant according to the negative descriptors in the reaction vector. The resulting fragment consists of atoms and bonds that are unchanged by the reaction. The fragment is then input to the structure generation algorithm together with the positive descriptors in the reaction vector and the product vector.

The reactant is represented by an atom table, a bond table which contains the AP2 descriptors, and a table of AP3 descriptors, as shown in Figure 7. In the example, only one mapping exists between the negative AP2 descriptors in the reaction vector and the AP2 descriptors in the reactant and is indicated by the red shading in the bond table. Each bond (AP2) thus identified is removed from the reactant together with any atom which has lost all of its bonds. The first bond to be removed is between atoms 6 and 8 (corresponding to AP2(1-)). The AP3 descriptors are then used to ensure that the environment of the bond in the reactant corresponds to that encoded in the reaction vector, otherwise this step will fail. Thus, the AP3 descriptors associated with the bond are also identified, as indicated at the top of Figure 8, and a check is made to ensure that these AP3s are also present as negative descriptors in the reaction vector. The next bond to be removed is that between atoms 8 and 10 (corresponding to AP2(2-)). The AP3s that are associated with this bond also match to the reaction vector. The final bond is that between atoms 10 (N) and 11 (C). There are no AP3 descriptors associated with this bond, and both of these atoms have lost all of their bonds; therefore, they are both removed from the connection table. (Atoms 6 (C), 8 (C), and 9 (C) are retained since not all bonds incident on them have been

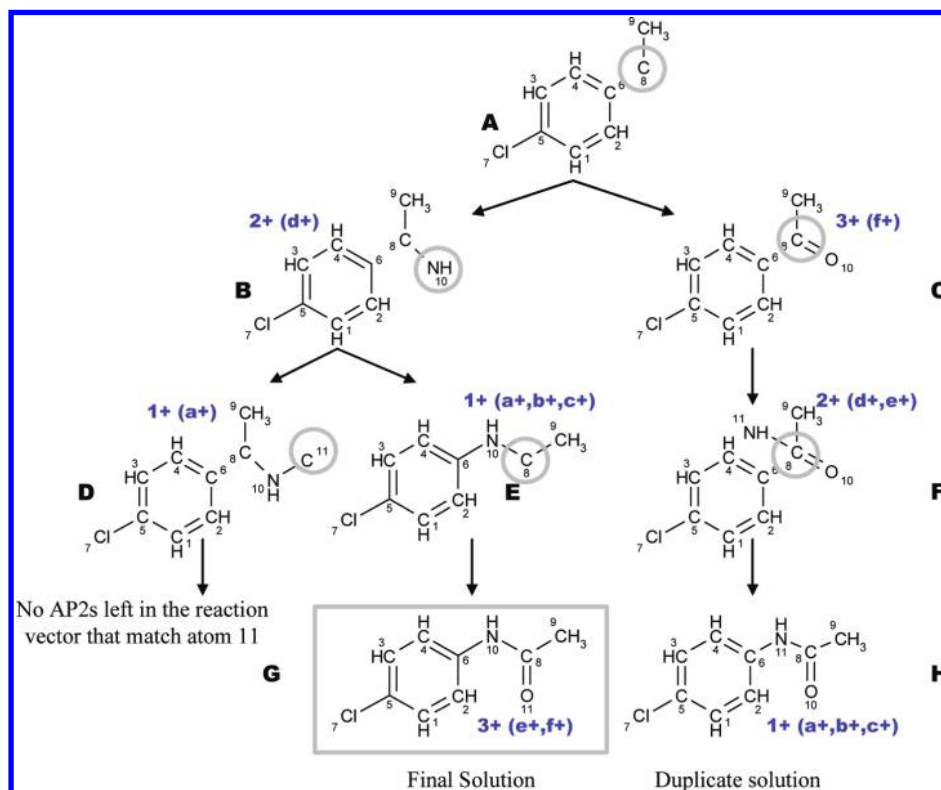


Figure 10. Structure generation applied to the Beckmann rearrangement. The starting fragment, the positive APs from the reaction vector, and the product vector form the input. The AP2s that are considered at each level are shown in blue with the AP3s shown in brackets. The gray circles indicate the atoms in the fragment that match the positive AP2 chosen to extend the fragment.

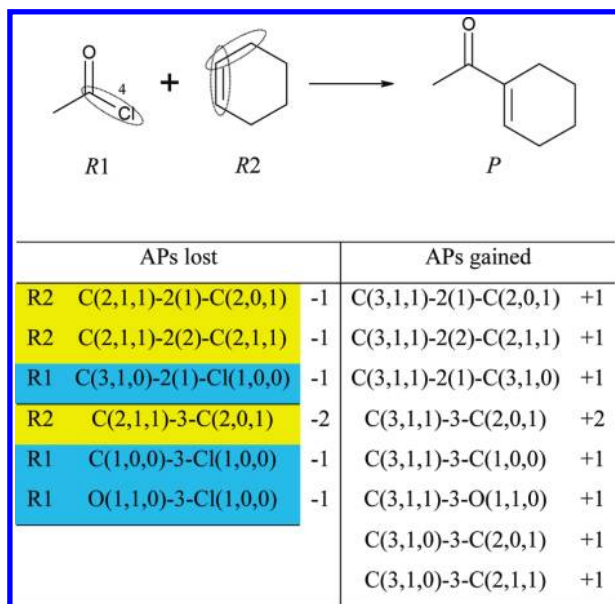


Figure 11. A simple substitution reaction is shown together with its reaction vector in the table. The APs lost highlighted in blue are found in R1, while those in yellow are found in R2.

removed.) At this stage all the negative descriptors (AP2s and AP3s) in the reaction vector have been accounted for, and the fragment generated from the reactant is shown at the bottom of Figure 8.

The fragment is input to the structure generation algorithm along with the positive AP descriptors in the reaction vector and the product vector. Structure generation proceeds in a breadth-first manner whereby the fragment is extended one bond at a time in all possible ways before proceeding to the next level, as illustrated in Figure 9. The highest numbered

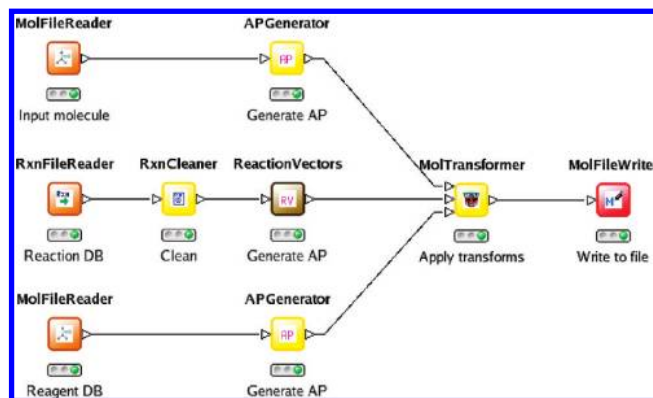


Figure 12. The KNIME *de novo* design workflow.

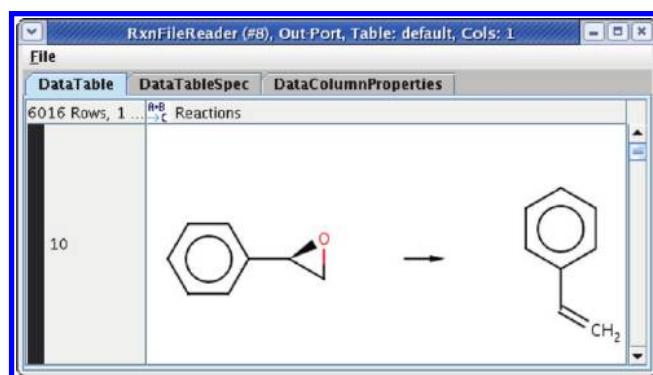


Figure 13. The molecule renderer implemented using Marvin Beans.²⁷

atom in the fragment that has unfilled valencies is selected as the seed atom, and all positive AP2s in the reaction vector that contain a matching atom are identified: a match is based on identical element type, X, and atom attributes. Each such

Table 3. Success Rates for Reproducing Known Reactions by Reaction Type^a

reaction type	total number of reactions	correct product generated	>1 product generated including correct product	incorrect product generated	no product generated
$R \rightarrow P$	3324	3018 (90.8)	183 (5.5)	55 (1.7)	251 (7.5)
$R1 + R2 \rightarrow P$	907	779 (85.9)	82 (9.0)	39 (4.3)	89 (9.8)
$R \rightarrow P1 + P2$	767	546 (71.2)	35 (4.6)	56 (7.3)	165 (21.5)
$R1 + R2 \rightarrow P1 + P2$	697	516 (74.0)	55 (7.9)	25 (3.6)	156 (22.4)

^a The numbers in brackets represent numbers as percentages of the respective reaction type.

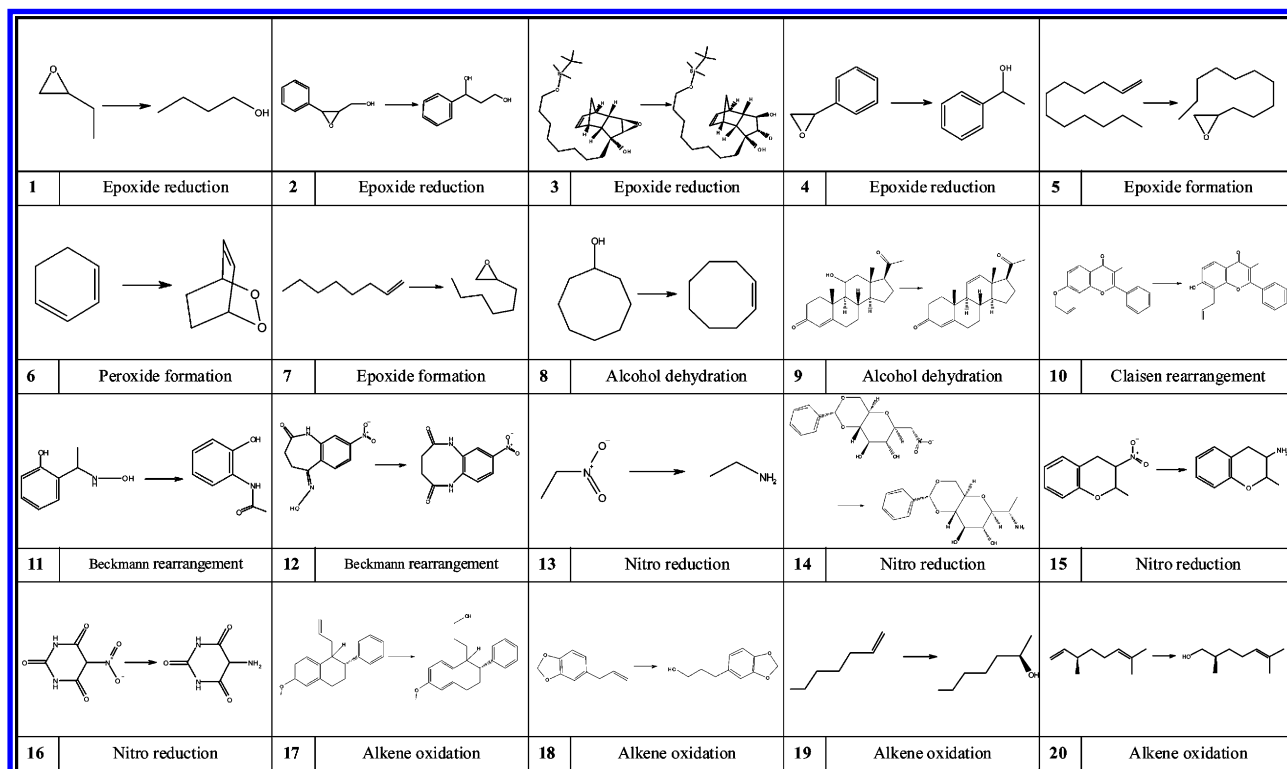


Figure 14. Examples of transformations of type 1 ($R \rightarrow P$ reactions) selected at random from the database that were successfully reproduced by the structure generation algorithm. All reactions produced the correct product only.

AP2 leads to the creation of a new fragment at the next level of the tree that is extended by one bond relative to its parent fragment. The new bond can result in one of the following actions: the fragment is extended by one atom; two disconnected substructures become connected, if there is a second unfilled valency in the fragment that is also consistent with the AP2; or a ring closure (if there is an appropriate second unfilled valency in the fragment). Each new fragment is then validated using the AP3 descriptors: AP3s are generated for the fragment and are compared with the AP3s in the product vector; if the fragment contains AP3s that are not present in the product vector, then the fragment is pruned from the tree. Next all fragments at the current level are compared and duplicates are eliminated. The algorithm then iterates on each fragment at the current level and so on, until all AP2s have been used. When there are no more AP2s left to be matched, a final check is made to ensure that each fragment is fully bonded.

Figure 10 illustrates the structure generation procedure for the Beckmann rearrangement in Figure 6. The highest numbered atom in the starting fragment (A) that has unfilled valences is atom 8 (C), and there are two positive AP2 descriptors in the reaction vector that match this atom (labeled 2+ and 3+). The descriptor AP2(2+) results in the addition of a single bond and the nitrogen atom as shown in

fragment B with the new bond also satisfying the positive AP3 descriptor d+. The descriptor AP2(3+) results in the addition of a double bond and an oxygen atom in fragment C and satisfies the positive AP3 descriptor f+. For fragment B, there are two atoms with unfilled valencies, atom 8 (C) and atom 10 (N). Where two or more such atoms exist, the atom appearing latest in the connection table is expanded, which in this case is atom 10. The AP2 descriptor AP2(1+) matches atom 10 and results in two structures at the next level: fragment D is formed by the addition of a single bond to a new carbon atom, whereas fragment E is formed by the addition of a single bond between the two disconnected fragments. Fragment C is expanded to a single structure at level 3 resulting from the addition of a single bond to a new nitrogen atom based on AP(2+). Proceeding to level 4: there are no AP2s left that match to atom 11 in fragment D, and so this branch of the tree is terminated; in fragment E, AP(3+) matches to atom 8 and results in the addition of a double bond and a new oxygen atom, all atoms are now complete, and all positive descriptors in the reaction vector are accounted for therefore structure G represents a final solution; the addition of AP(1+) to fragment F also results in a final solution H which is identified as a duplicate of G. Thus, in this example, a single product only is generated.

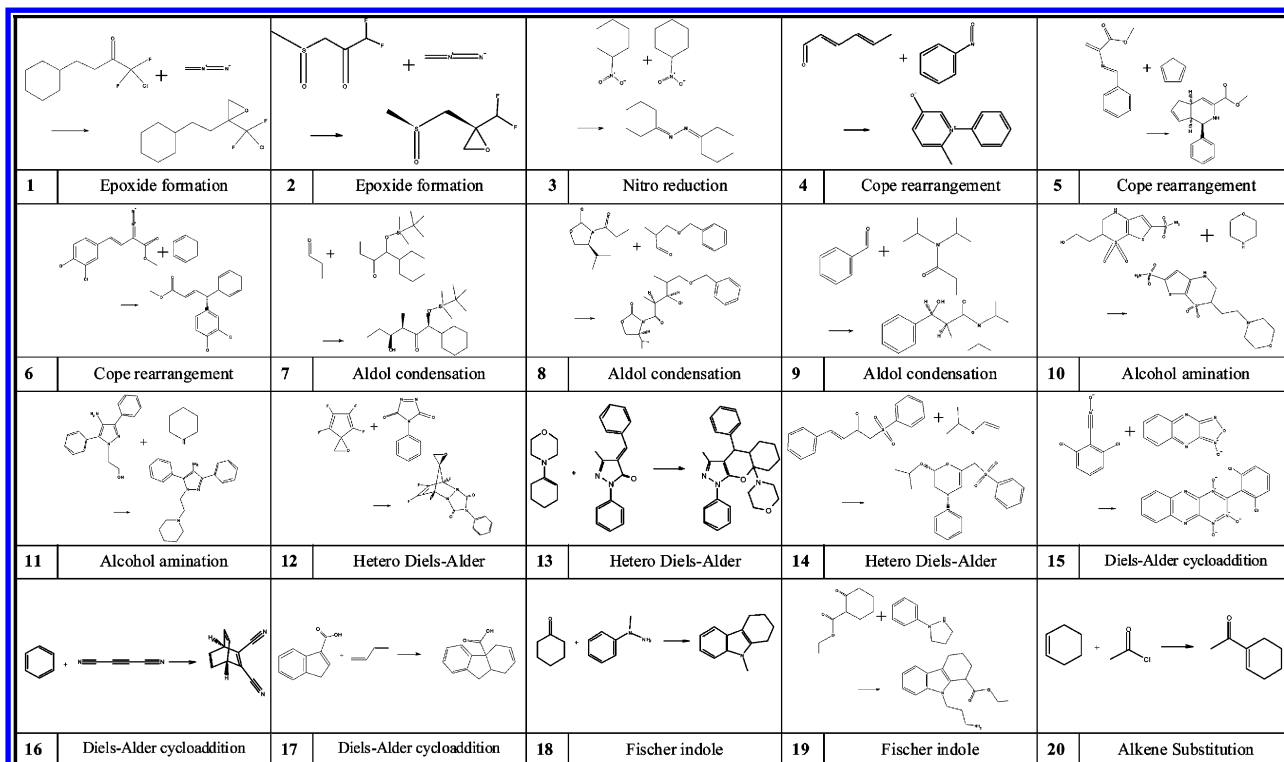


Figure 15. Examples of transformations of type 2 ($R_1 + R_2 \rightarrow P$ reactions) selected at random from the database that were successfully reproduced by the structure generation algorithm. All reactions produced the correct product only.

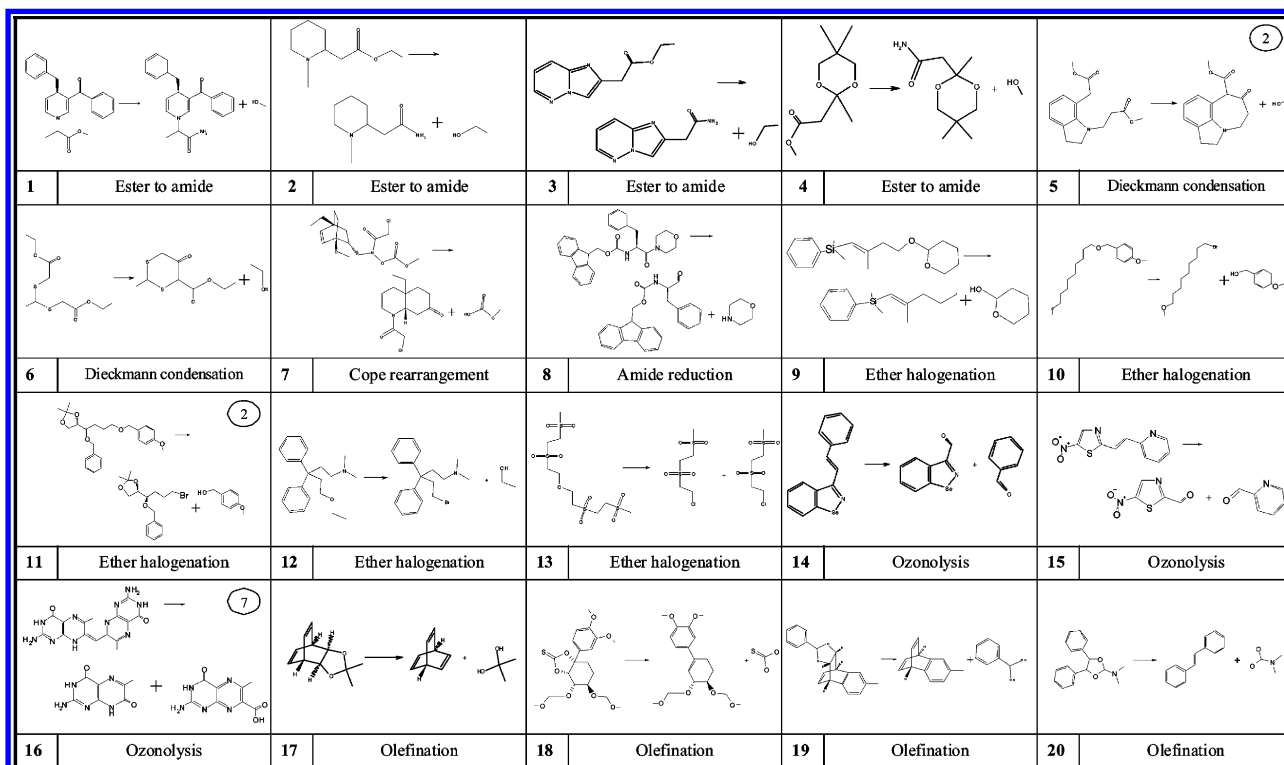


Figure 16. Examples of transformations of type 3 ($R \rightarrow P_1 + P_2$ reactions) selected at random from the database. The value in the circle shows the number of structures generated if more than one.

Using Reaction Vectors To Generate Novel Molecules. In *de novo* design, the goal is to apply a reaction vector to a starting material other than that of the parent reaction in order to generate a novel molecule. For a reaction vector to be applicable to a previously unseen starting material, all the negative descriptors in the reaction vector must be present in the vector representation of the starting material, that is,

all the bonds to be changed must be present in the correct environment as encoded by the atom pair descriptors. If this condition is met and the reaction consists of a single reactant and a single product (a type 1 reaction, see below), then the algorithm proceeds as described above. The algorithm has also been extended to deal with multicomponent reactions of types 2–4 below.

1. $R \rightarrow P$
2. $R1 + R2 \rightarrow P$
3. $R \rightarrow P1 + P2$
4. $R1 + R2 \rightarrow P1 + P2$

Transformations of type 2 ($R1 + R2 \rightarrow P$) consist of two reactants each of which contributes negative descriptors to the reaction vector. In order to apply reaction vectors of this type to a single starting material it is necessary to also identify a reagent that will complete the requirements for the reaction to occur. Thus if some, but not all, of the negative descriptors in the reaction vector are found, a search

is made in the reagents database for a reagent that would offset the remaining negative descriptors. The minimum requirement is that at least one negative AP2 descriptor and one negative AP3 descriptor that contains an atom also found in the AP2 must be present in the starting material with the remaining descriptors (AP2 and AP3) being found in a second reagent. This minimum requirement permits reactions to be applied in which only one bond is removed from the starting material, such as the simple substitution reaction shown in Figure 11. The starting material ($R1$) contains the one AP2 highlighted in the structure diagram and shaded in

1 Epoxide reduction	2 Epoxide formation	3 Friedel-Crafts Acylation	4 Friedel-Crafts Acylation	5 Friedel-Crafts Acylation
6 Friedel-Crafts Acylation	7 Alkene Oxidation	8 Alkene Oxidation	9 Cope rearrangement	10 Hetero Diels-Alder
11 Hetero Diels-Alder	12 Hetero Diels-Alder	13 Hetero Diels-Alder	14 Claisen condensation	15 Olefination
16 Wittig Horner	17 Wittig Horner	18 Wittig Horner	19 Wittig Horner	20 Wittig Horner

Figure 17. Examples of transformations selected at random of type 4 ($R1 + R2 \rightarrow P1 + P2$ reactions) that were successfully reproduced by the structure generation algorithm. The value in the circle shows the number of structures generated if more than one.

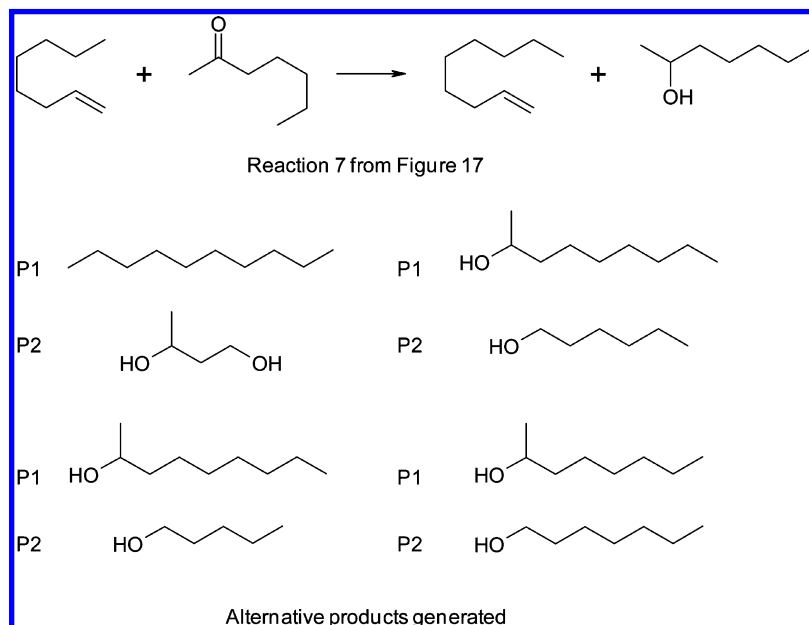


Figure 18. Alternative products generated when attempting to reproduce the parent reaction.

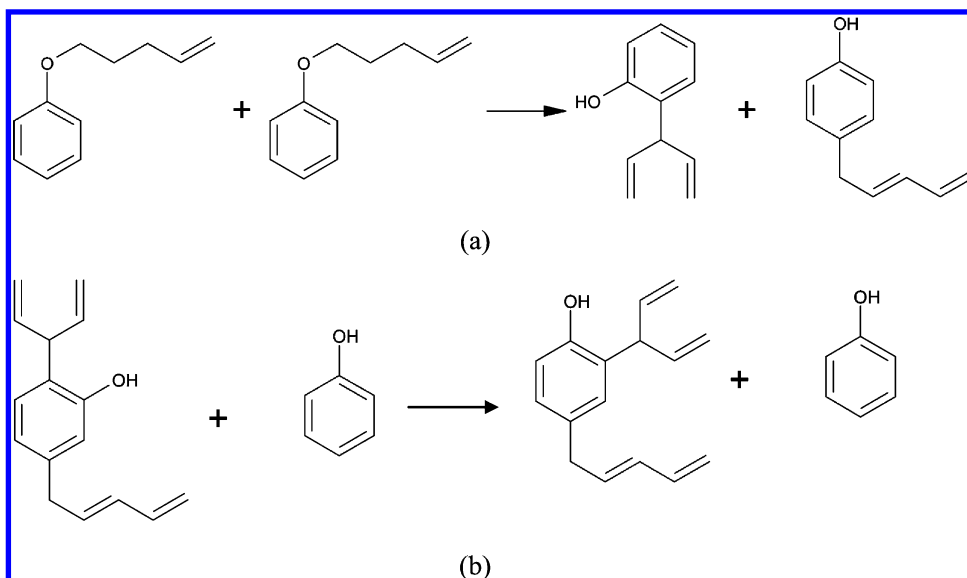


Figure 19. An example where an incorrect solution is generated. (a) The parent reaction and (b) two alternative solutions are generated; however, neither is correct.

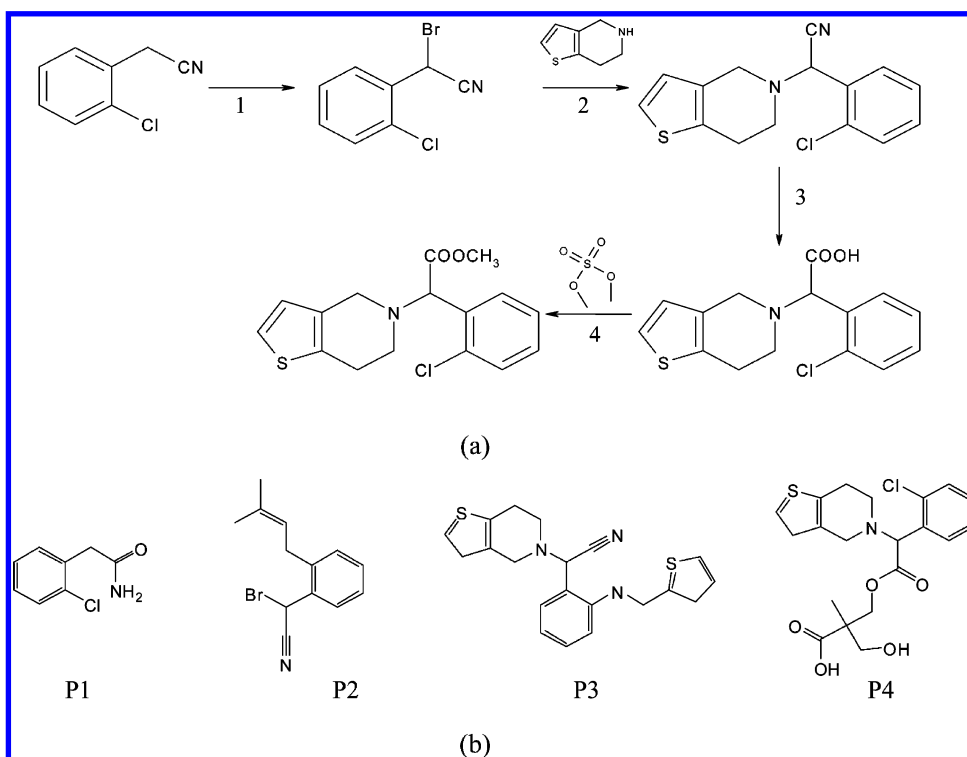


Figure 20. (a) The synthetic route of the intermediate of the antithrombotic agent (S)-(+)-clopidogrel bisulfate as given by Wang et al.²⁹ with the reaction steps labeled under the arrows. (b) Examples of additional product molecules generated for each step.

the table and two AP3s, whereas the reagent (*R*2) contains two AP2s representing the two bonds highlighted in the structure diagram. If such a reagent is found, then the relevant atoms and bonds are removed from both the starting material and the reagent to form two partial structures which are both input to the structure generation algorithm as the input “fragment”. The structure generation algorithm then explores all possible ways of adding the positive atom pairs considering all substructures present. The addition of a bond in moving from one level of the search tree to the next may extend one reactant, join two reactants together, or result in a ring closure as before. Transformations of type 3 involve the generation of two product molecules, either from a single reactant or from two reactants. These are handled by forming

a new molecule once all valences in the starting fragment have been satisfied. Transformations of type 4 (*R*1 + *P*2 → *P*1 + *P*2) are essentially a combination of type 2 and type 3 transformations.

Identifying Applicable Reaction Vectors. A knowledge-base of reaction vectors is derived by first compiling a database of reactions of interest, passing each reaction through the reaction cleaning algorithm, calculating the atom pair descriptors for the reactant(s) and product(s), and then calculating the reaction vector by applying eq 1. The resulting reaction vectors are organized by first removing duplicates and then storing them as an inverted index to speed up the search for applicable reaction vectors. The index is generated by extracting all the unique APs with negative counts from

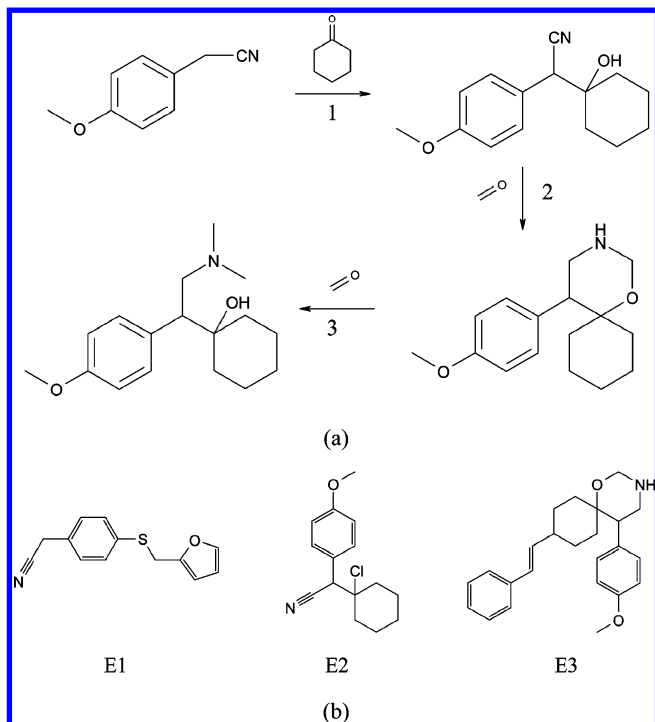


Figure 21. (a) The synthetic route of the antidepressant venlafaxine as given by Kavitha and Rangappa³⁰ with the reaction steps labeled under the arrows. (b) Examples of additional product molecules generated for each step.

the database of reaction vectors and associating each with a list of reaction vectors it is found in. When searching for reaction vectors that are applicable to a given starting material, each AP in the starting material is taken in turn, and the reaction vector lists merged. For reactions of type 1 and type 3, the lists are intersected since all AP descriptors must be present. For reactions of type 2 and type 4, each reaction vector from the merged lists is taken in turn. If it has at least one AP2 and one AP3 that match with the starting material, then it is accepted, and the reagent list is searched as indicated above.

Implementation Details. The algorithms have been written using the JoeLib toolkit²⁴ with the desktop *de novo* design tool implemented in the open source pipelining tool KN-IME²⁶ (Figure 12). KNIME was chosen because of its simple plug and play architecture and because it has a large data mining/model building component which could be used in the scoring of the generated structures. Several KNIME nodes have been written to support *de novo* design. The MolFileReader node is used to read in a starting material and a database of reagents. The APGenerator calculates atom pair descriptors for the input molecules (starting materials and reagents). The RxnFileReader node allows a database of reactions to be read in. The RxnCleaner node passes the

reactions through the reaction cleaning algorithm, and the ReactionVectors node then generates reaction vectors for all reactions that pass the reaction cleaning process. The MolTransformer node applies reaction vectors to the input molecule. This modular implementation allows the tool to be tailored for different design scenarios both through the use of bespoke sets of reactions and reagents and by providing the following run-time options:

1. No reagents permitted: the applicable reaction vectors are restricted to those that do not require an additional reagent, that is, transformations of types 1 and 3.

2. Single reagent permitted: reaction vectors that represent transformations of types 2 and 4 which require a reagent can be applied, and for such reaction vectors the first applicable reagent is selected only.

3. Full enumeration: all reagents applicable to a given reaction vector and starting material are selected and applied in a full enumeration mode. This option is of interest in library design where a user wishes to design multiple products that could be made from the starting material using the same reaction.

The MolFileWriter node writes the product molecule(s) out to file. The input and output nodes also incorporate molecule/reaction renderers for display. These are implemented by converting JOEMol representations to Marvin Beans molecule representations (ChemAxon²⁷) via SMILES strings and using the Marvin Beans paint method to draw the molecule/reaction in a pane, as shown in Figure 13.

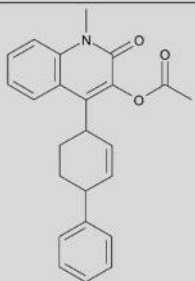
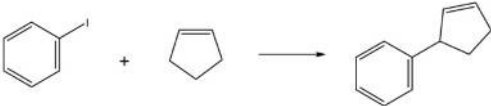
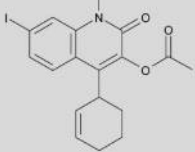

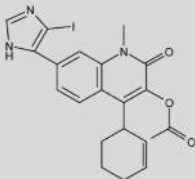
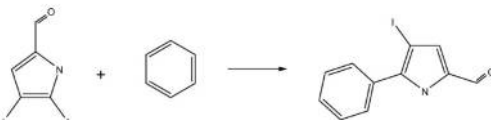
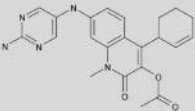
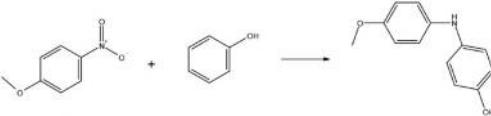
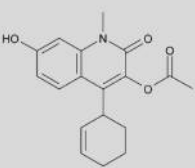
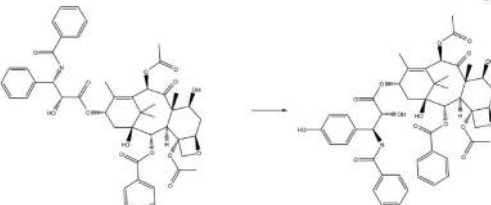
EXPERIMENTAL SECTION

Validation of a *de novo* design tool is difficult and ultimately requires the synthesis and testing of the molecules. During program development, however, it has been common practice to devise retrospective validation experiments. By way of example, many of the 3D *de novo* design programs have been applied to the design of small molecules that could bind to a receptor with the aim of designing known binders found in the Protein Data Bank.²⁸ Here, we have devised two validation experiments: an internal validation that aims to reproduce reactions in the reaction database and an external or retrospective validation in which we attempt to reproduce the known synthetic routes of two drug molecules. We then present three applications of the reaction vector approach to *de novo* design that represent different drug design scenarios. We first investigate the extent to which a reaction vector can be used to generate products which are diverse with respect to the parent reaction. We then apply the method in a typical lead optimization setting in which analogues of a lead compound are generated. Finally we apply the algorithm to library enumeration.

Table 4. Number of Applicable Reaction Vectors and the Total Number of Products Generated for Each Step in the Synthetic Routes of Clopidogrel and Venlafaxine

clopidogrel			venlafaxine		
step	no. applicable reaction vectors	total number of products generated	step	no. applicable reaction vectors	total number of products generated
1	17	158	1	10	155
2	11	123	2	4	6
3	12	124	3	8	12
4	41	386			

Table 6. Selection of Five *de Novo* Products Generated from Molecule 2 in Table 5^a

ID	<i>De novo</i> product	Parent reaction	Sim to parent product
1			0.47
2			0.15
3			0.33
4			0.41
5			0.15

^a The reaction from which the reaction vector was derived (parent reaction) and the similarity of the *de novo* product to the product of the parent reaction are shown.

cleaning (where some reactions were split into more than one reaction) and removal of reactions with more than two reactants or products. This data set is referred to as the 5K data set and resulted in 2866 unique reaction vectors, i.e., on average each reaction vector represents 2.0 reactions. The larger set of 26,616 reactions was compiled by first collecting all reactions with yields of 100%, 75%, and 50% and excluding those consisting of solid-phase chemistry. After cleaning and removal of reactions with more than two reactants or products there were 24,418 reactions, and these resulted in 16,859 unique reaction vectors with each reaction vector representing an average of 1.5 reactions. This data set is referred to as the 26K data set.

Database of Reagents. A data set of 5839 reagents was extracted from the Available Chemicals Directory²⁵ of commercially available reagents consisting of molecules with molecular weight less than 100, containing the element carbon and no metal atoms.

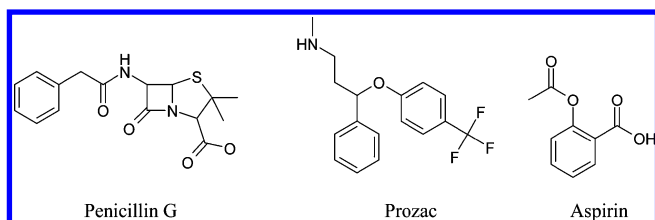
Reproducing Reactions in the Knowledge-Base. The internal validation involved attempting to generate the known

product(s) for each reaction in the 5K reaction database from the known reactant(s) as starting material(s) and the known reaction vector, i.e., attempting to reproduce the product(s) of the parent reaction. The results are summarized in Table 3, and example reactions for each type of transformation are shown in Figures 14–17 with 20 reactions for each transformation type selected at random. As shown by the examples, the algorithm is able to deal with simple transformations through to complex rearrangements. The parent product was generated for over 90% of the type 1 reactions which form the majority of the reactions in the data set. For a small number of these, alternative products were also generated which were structural isomers of the correct product. The failure rates for the more complex multicomponent reactions were slightly higher with the reactions consisting of two products proving more difficult than reactions consisting of a single product, as might be expected. However, these more complex reactions represent a much smaller proportion of the knowledge-base.

Table 7. Seven *de Novo* Products Generated from Molecule 9 in Table 5^a

ID	<i>De novo</i> product	Parent reaction	Similarity to parent product
1			0.16
2			0.28
3			0.08
4			0.07
5			0.25
6			0.12
7			0.08

^a The reaction from which the reaction vector was derived (parent reaction) and the similarity of the *de novo* product to the product of the parent reaction are shown.

**Figure 22.** The drug molecules used in the lead optimization experiments.

An example where alternative products are generated is shown in Figure 18. This illustrates a limitation of the reaction vector as implemented which, at present, does not record which functional group is derived from which product. The failures include a small number where the incorrect product(s) was generated or no product was generated at all. Figure 19 shows an example of incorrect products being generated for a type 4 reaction ($R1 + R2 \rightarrow P1 + P2$). In

this example, the two reactants are identical, and the positive descriptors encoded in the reaction vector are applied to one fragment rather than being applied to both.

Reproducing Known Synthetic Routes. The retrospective validation aimed to reproduce the synthetic routes of two drugs currently on the market. The examples consist of the synthesis of the intermediate of the antithrombotic agent (S)-(+)-clopidogrel bisulfate²⁹ shown in Figure 20(a) and the antidepressant drug venlafaxine³⁰ shown in Figure 21(a). In each case, the starting material for each reaction step was provided as input, and product molecules were generated using the 26K data set of reaction vectors and the 5839 reagents set described above. In both cases, the known synthetic routes were reproduced successfully, even though the reactions themselves were not present in the database.

A secondary aim of the experiment was to determine the number of alternative products molecules that were suggested

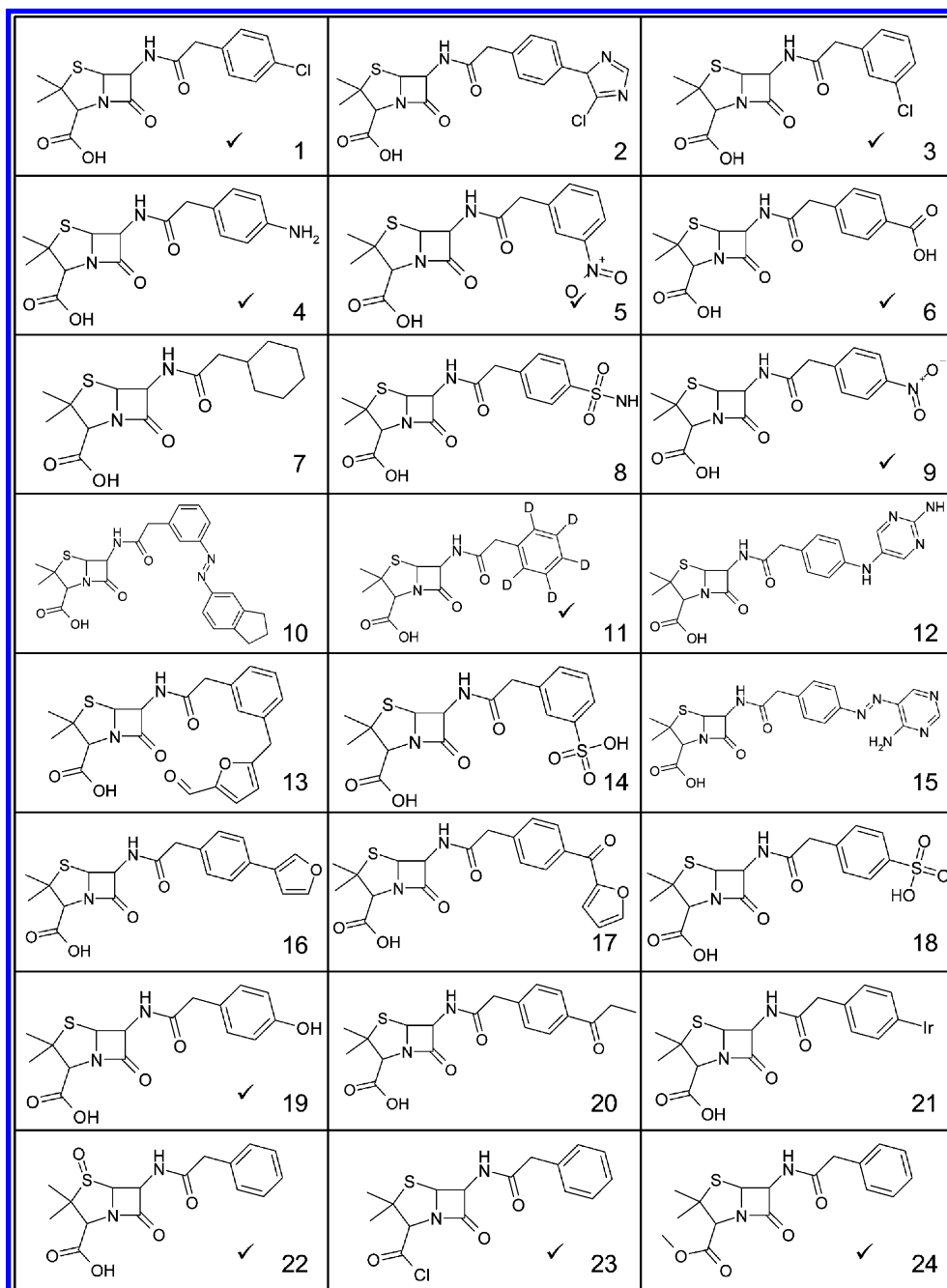


Figure 23. Potential products that could be synthesized in a single step from penicillin G. The tick marks indicate compounds found in SciFinder.

at each reaction step in order to give an idea of the size of the search space in a real *de novo* design application. The number of reaction vectors that are applicable at each reaction step is shown in Table 4 together with the number of products generated in each step. Note that a full combinatorial approach was not followed and that the input to each reaction step consisted of the single molecule shown in the respective schemes. Thus, for clopidogrel there were 17 reaction vectors that could be applied to the starting material, and a total of 158 products were generated for the first reaction step; 11 reaction vectors were applicable at step 2, and 123 products were generated; and so on. Nearly 400 alternative products were generated for the final step of the synthesis of clopidogrel due to the large number of reactions in the database that involve carboxylic acids. Examples of alternative structures generated at each reaction step are shown in

Figure 20(b). In the case of venlafaxine, while a total of 155 alternative products were generated for the initial reaction step, there were considerably fewer alternatives for the next two steps. Again, examples of alternative products generated for each step are shown in Figure 21(b).

This retrospective validation experiment demonstrates that the reaction vector approach is successful in reproducing known synthetic routes. However, it also indicates that a very large number of alternative products could be generated in a true *de novo* design application, especially, if multiple reaction steps are performed and no restrictions are placed on the reaction vectors and reagents that are applied. This suggests the need for scoring methods in order to focus the search on druggable molecules (in addition to ease-of-synthesis which is the primary aim of the work described here). As mentioned earlier this can be implemented easily

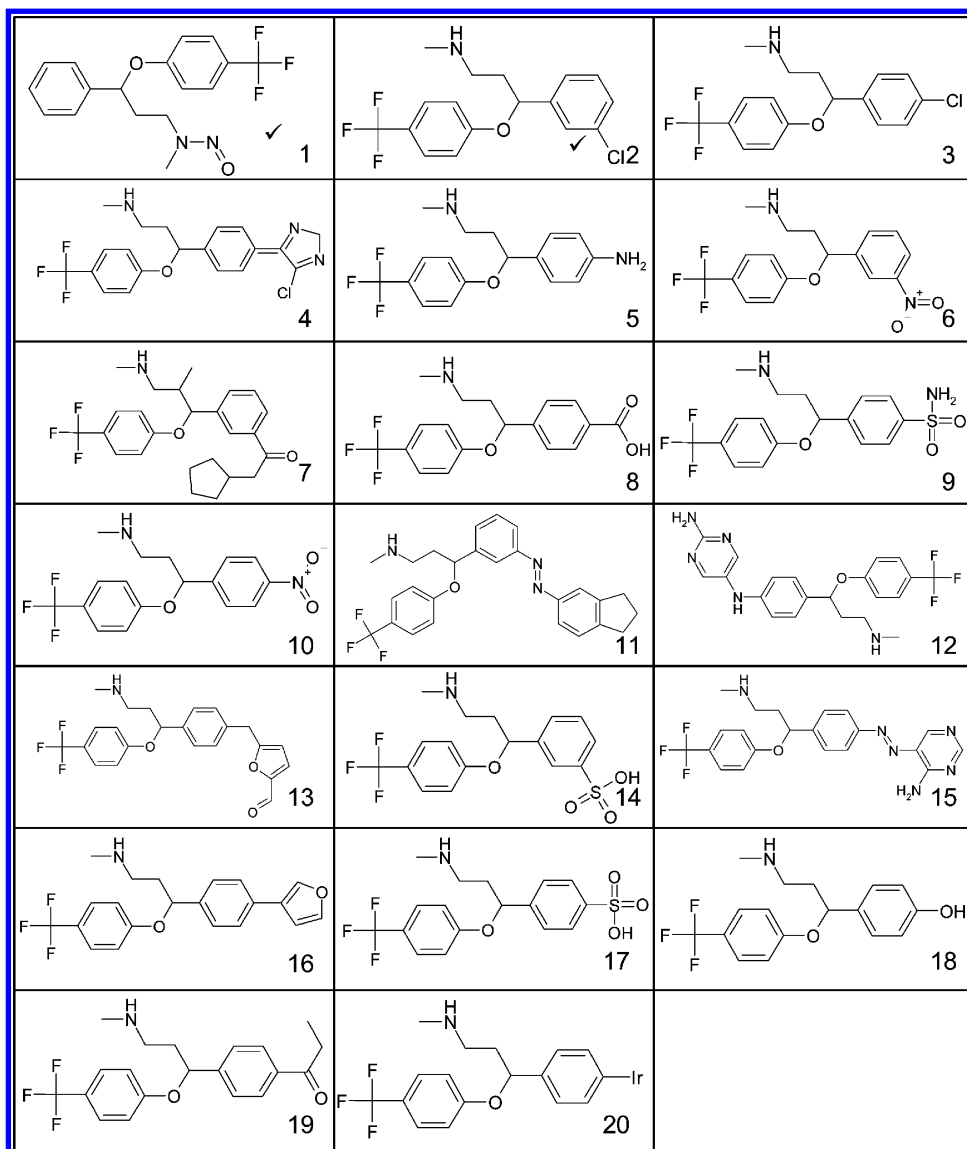


Figure 24. Potential products that could be synthesized in a single step from Prozac. The tick marks indicate compounds found in SciFinder.

using a variety of scoring mechanisms in a workflow tool such as KNIME, and we are currently embedding the reaction vector method into an iterative multiobjective *de novo* design tool which will be described in a future publication. The remaining experiments in this paper describe additional applications of the *de novo* design tool that are based on single-step reactions.

Using Reaction Vectors in *de Novo* Design. Three applications have been chosen to illustrate the use of the structure generation algorithm for *de novo* design. These consist of the following: an evaluation of the algorithm's ability to generate novel compounds; the application of the tool in a lead optimization scenario; and the enumeration of a library of compounds based on a starting material, a single reaction, and a set of reagents.

Evaluation of Product Diversity. Ten reactants were selected at random from the 5K reaction data set. These were input to the *de novo* design tool, and all possible products were generated using the reaction vectors built from the 26K data set. For reactions of type 2 ($R1 + R2 \rightarrow P$) and type 4 ($R1 + R2 \rightarrow P1 + P2$) where a reagent is required, only one reagent was selected (the first identified as applicable).

The similarity of each *de novo* product (i.e., the product generated by the tool) to the product of the parent reaction was calculated using atom pairs up to AP4 (i.e., AP2, AP3 and AP4 descriptors) and the Tanimoto coefficient, and the average value over all reactions is reported. For reactions of type 3 ($R \rightarrow P1 + P2$) and type 4 ($R1 + R2 \rightarrow P1 + P2$), where there are two products, the similarities of the *de novo* product to both $P1$ and $P2$ of the parent reaction were calculated and the largest similarity value reported. The results are shown in Table 5.

The number of products that are generated ranges from 0 to 44, with molecules that contain more functional groups tending to result in more solutions. Furthermore, the average similarities of the products generated to the products of the parent reactions varies widely, from 0.15 to 0.96. These results suggest that the product diversity that can be achieved is a function of both the starting material itself as well as the contents of the reaction database. For example, molecules 4 and 5 are relatively small and of low functionality, and any reaction vectors that are applicable to them are likely to be derived from parent reactions based on similar starting materials and thus result in similar products. Molecule 7 did

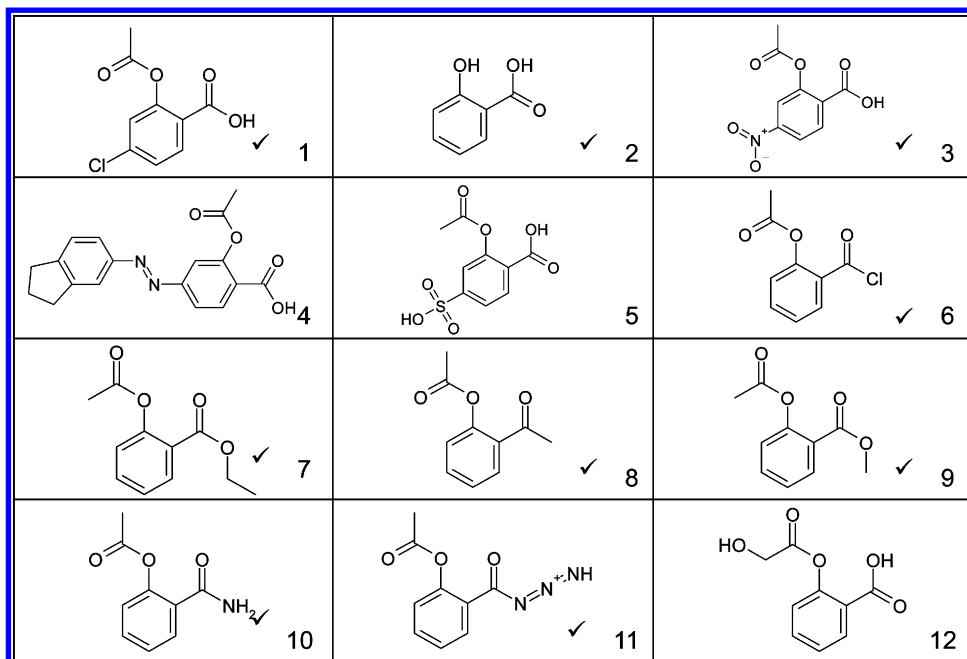


Figure 25. Potential products that could be synthesized in a single step from aspirin. The tick marks indicate compounds found in SciFinder.

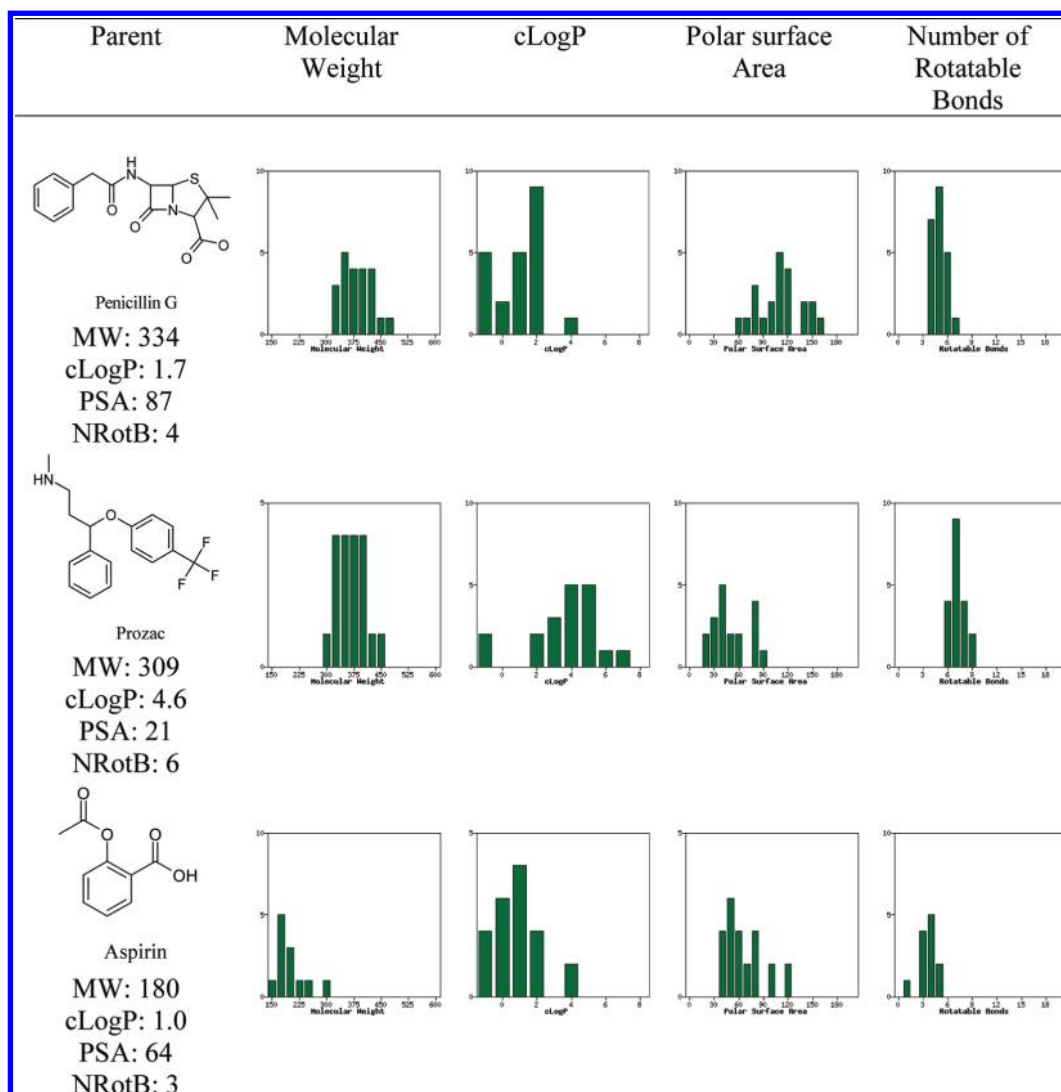


Figure 26. Distributions of physicochemical properties of the analogue compounds in Figures 23–25. Calculations performed using ChemAxon's Marvin Beans.²⁵

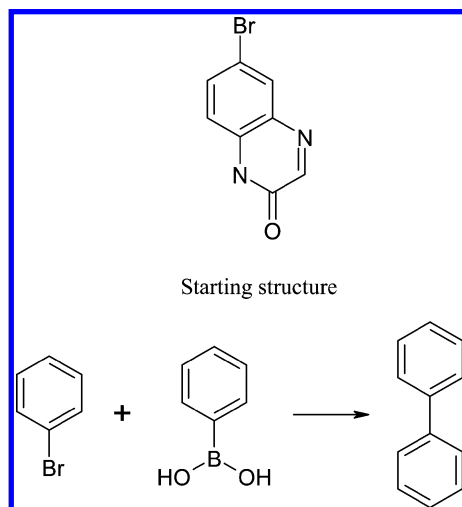


Figure 27. The starting molecule used for library enumeration and the Suzuki reaction from which the reaction vector is calculated.

not generate any solutions, indicating that there were no applicable reactions in the knowledge-base. Apart from molecules 4 and 5, the average similarity values are below 0.35 and therefore suggest that in many cases the reaction vectors are applicable to starting molecules with low similarities to the reactants in the parent reaction. Tables 6 and 7 show examples of the *de novo* products generated from molecules 2 and 9, respectively, together with their similarities to the corresponding products of the parent reactions used to generate them.

Starting molecule 9 in Table 5 contains two ester groups, and the first *de novo* molecule, in Table 7, is generated by hydrolysis of only one of them since there is only one hydrolysis in the parent reaction. In reality, it is likely that such a reaction would result in the hydrolysis of both ester groups, as occurs in product 2. This highlights a current limitation of the method which does not take into account functional groups that are remote from the reaction center identified by the algorithm.

Lead Optimization. Lead optimization aims to improve the biological activity profile of a lead compound, such as its potency and ADMET (absorption, distribution, metabolism, excretion, and toxicity) properties, by making small modifications to its structure to generate structural analogues. The effectiveness of the tool in lead optimization scenarios was investigated by taking known drugs as starting molecules and generating potential products that could be made using single-step transformations. The products were deemed as synthetically accessible if they were found following a search in SciFinder Scholar.³¹ However, it should be noted that presence in SciFinder does not necessarily indicate synthesis via the mechanism suggested by the program. The program was run with the 26K set of reaction vectors and the reagent set described above.

Figures 23–25 show the products suggested for penicillin G, Prozac, and aspirin, respectively; Figure 22. The largest number of potential products (24) was generated for penicillin G which has the most functional groups; 20 were generated for Prozac; and 12 were generated for aspirin. The compounds that are present in SciFinder are indicated by the tick marks; thus 11 of the 24 structures generated for penicillin G, 9 of the 12 aspirin products, and two of the Prozac structures were found in the literature. The low

incidence of the Prozac products using SciFinder may be due to the general nature of the original patent in which many of the analogues are not included as individual structures and have therefore not been encoded in the CAS Registry. Some structures were similar to literature structures, such as structure 10 in Figure 24. Structures have also been generated that would not be very useful in drug design, such as the iridium complexes (structure 21 in Figure 23 and structure 20 in Figure 24) and the deuterium compound (structure 11) in Figure 23. This suggests that the reactions used to generate the reaction vectors should be further filtered, for example, to remove metallic complex and isotope containing reactions. The distributions of various physico-chemical properties of the resulting libraries of compounds are shown in Figure 26 where it can be seen that the suggested compounds sample a druggability space which is obviously an important goal in lead optimization.

The examples given demonstrate that the reaction vector approach to *de novo* design enables the generation of synthetically accessible molecules that are close structural analogues of a parent molecule. Thus, the method could be used to suggest compounds for synthesis that explore the structural space around a known lead compound in order to generate an SAR (structure–activity relationship) profile. Again it should be commented that it would be trivial to filter the potential products through a number of druglike filters to increase the given hit-rate by targeting the compounds toward a particular drug property space.

Library Enumeration. The *de novo* design tool implemented in KNIME allows the user to easily generate all possible solutions from a given starting material by iterating through all reagents for each applicable reaction vector. Thus, the tool can be customized to generate a virtual library of potential products by enumerating all possible products from a single reaction using an appropriate set of available reagents. Such an approach could be used to guide a parallel synthesis experiment based on a single reaction with the reagents being varied. An example is shown in Figure 27 and uses 6-bromoquinoxalin-2-one as starting material and a reaction consisting of the single Suzuki coupling. A set of 628 boronic acids was used as the reagent database extracted from the ACD²⁷ using a simple boronic acid query.

The enumerated library consisted of 292 structures with example products in the library shown in Figure 28. Fewer than half of the boronic acids were applicable as reagents. This is due to the environment encoded within the reaction vector including the 2 and 5 positions on the aromatic ring of the acid; these positions are unsubstituted so that only boronic acids that are similarly unsubstituted are valid reagents for the reaction. The use of substituted acids would require additional Suzuki reactions to be used to generate the appropriate reaction vectors.

This example highlights the way in which the reaction vector approach can capture the environment of the reaction which is a key feature of the method. As such it should provide greater confidence that the reaction will complete as there exists a literature precedence. However, this also means that care should be taken to ensure there are sufficient examples in the reaction database to enable full coverage of a generic reaction.

The advantage of the reaction vector approach over other approaches such as fragment marking and earlier reaction

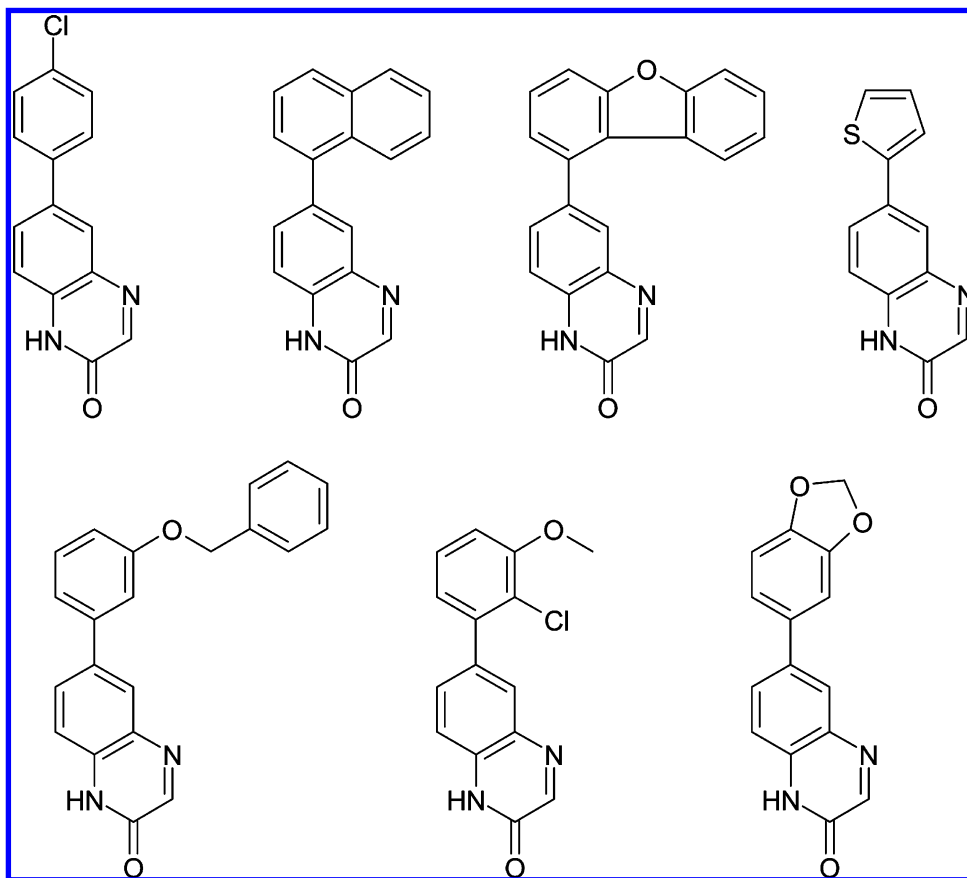


Figure 28. Example products resulting from library enumeration.

transform approaches is the ease with which the enumeration is carried out: once an example of the reaction has been provided and the reagents have been assembled, the process is fully automated. Both of the pre-existing methods require some degree of preprocessing. For example, fragment marking involves the addition of R groups to the core molecule and clipping of the reactive groups in the reagents (and this approach may fail for some types of reactions such as the Diels–Alder reaction); the reaction transform approach requires encoding of the reaction using, for example, a language such as SMIRKS, which can be nontrivial for complex reactions.³²

CONCLUSIONS

A novel approach to the *de novo* design of synthetically feasible molecules has been described. The method is based on reaction vectors which describe the structural changes that take place at the reaction center. The reaction vectors are derived automatically from a collection of reactions which is not restricted by size or reaction complexity (up to the limit of two reactant and two product reactions, currently). Thus no encoding of complex reaction schemes is required. This automated and flexible approach offers great potential for capturing the growing body of data on reactions that is becoming available through electronic laboratory notebooks.

The method has been validated by reproducing the products of the reactions contained in the knowledge-base and also by replicating known syntheses. A number of applications of the tool to the design of novel molecules have also been described that demonstrate the different modes in which the program can be run. These include the following:

an analysis of the diversity of products generated relative to those encoded in the knowledge-base using a random set of starting materials; simulated lead optimization where known drugs are used as starting materials to generate analogues, several of which were found in the literature; and library enumeration in which a single reaction is considered and all possible products generated using a customized set of reagents.

Although the examples emphasize the success and flexibility of the approach, they also highlight some limitations and opportunities for future work. A key limitation at present is the inability to detect remote or multiple functionality in the starting material that may preferentially react with the reagents under the reaction conditions proposed. This, to some extent, can be overcome by using descriptors of greater complexity and path length. However, greater descriptor complexity would increase calculation time and decrease the applicability and generality of the reaction vector. Likewise the product built from a complex descriptor would bear close similarity to the parent and, in a *de novo* design paradigm, may be of little use. Thus a trade-off exists between specificity and novelty in the degree of intricacy encoded in the descriptor. An alternative strategy might be, once the reaction and reagents are identified, to search using those reagents for other suitable reactions involving the atom pairs in the remote part of the molecule. However, this would once again increase the time taken per step and add to the complexity of the algorithm. This cognisance around time per step is highlighted as the reaction databases used were relatively small. Commercial reaction databases can be much larger and of the order of millions of reactions and it is

envisaged will require careful clustering and optimization to make the search time for large databases truly efficient.

The examples reported also suggest that a mechanism to control the combinatorial explosion achievable with *de novo* design is required, and, to this end, the method was implemented within KNIME. This invocation as a pipelining tool allows the knowledge-base of reactions and reagents to be easily modified so that the process can be readily customized for different types of applications. The data modeling package within KNIME provides a workbench to readily 'pipe-in' a variety of scoring functions. As modern drug discovery is a multiobjective optimization problem current work is focused on embedding the structure generation algorithm within an iterative loop to provide populations of molecules that satisfy multiple objectives while also having a high degree of confidence that they are synthetically accessible. This confidence will be gained from using a knowledge-base of reactions whose reaction environments, as encoded in the reaction vectors, bear close resemblance to the reactions proposed for the *de novo* designed molecules. The multiobjective framework should also allow us to take account of additional factors when selecting reaction vectors to apply, such as the reaction conditions and yields of the parent reactions.

ACKNOWLEDGMENT

We gratefully acknowledge Howard Broughton for many helpful discussions throughout this work. We thank the Engineering and Physical Sciences Research Council and Eli Lilly for funding and the Royal Society and the Wolfson Foundation for laboratory support. Programs were written using JOELib (University of Tübingen, Germany), Marvin Beans (ChemAxon), and KNIME (University of Konstanz, Germany), and we gratefully acknowledge the provision of software by these organisations and the support provided.

Supporting Information Available: Data tables detailing the reaction types included in the 5K data set and the reaction cleaning process applied to this data set. This material is available free of charge via the Internet at <http://pubs.acs.org>.

REFERENCES AND NOTES

- Gillet, V. J.; Johnson, A. P. Structure Generation for De Novo Design. In *Designing Bioactive Molecules: Three-Dimensional Techniques and Applications*; Martin, Y. C., Willett, P., Eds.; American Chemical Society: Washington, 1998; pp 149–174.
- Lewis, R. A.; Leach, A. R. Current Methods for Site-Directed Structure Generation. *J. Comput.-Aided Mol. Des.* **1994**, *8*, 467–475.
- Schneider, G.; Fechner, U. Computer-based de novo design of drug-like molecules. *Nat. Rev. Drug Discovery* **2005**, *4*, 649–663.
- Bohm, H. J. Ludi - Rule-Based Automatic Design of New Substituents for Enzyme-Inhibitor Leads. *J. Comput.-Aided Mol. Des.* **1992**, *6*, 593–606.
- Bohacek, R. S.; McMartin, C. Multiple Highly Diverse Structures Complementary to Enzyme Binding-Sites - Results of Extensive Application of a De-Novo Design Method Incorporating Combinatorial Growth. *J. Am. Chem. Soc.* **1994**, *116*, 5560–5571.
- Gillet, V. J.; Myatt, G.; Zsoldos, Z.; Johnson, A. P. SPROUT, HIPPO and CAESA: Tools for de novo structure generation and estimation of synthetic accessibility. *Perspect. Drug Discovery Des.* **1995**, *3*, 34–50.
- Boda, K.; Johnson, A. P. Molecular complexity analysis of de novo designed ligands. *J. Med. Chem.* **2006**, *49*, 5869–5879.
- Lajiness, M. S.; Maggiora, G. M.; Shanmugasundaram, V. Assessment of the consistency of medicinal chemists in reviewing sets of compounds. *J. Med. Chem.* **2004**, *47*, 4891–4896.
- Ekins, S.; Boulanger, B.; Swaan, P. W.; Hupcey, M. A. Z. Towards a new age of virtual ADME/TOX and multidimensional drug discovery. *J. Comput.-Aided Mol. Des.* **2002**, *16*, 381–401.
- Fechner, U.; Schneider, G. Flux (1): A virtual synthesis scheme for fragment-based de novo design. *J. Chem. Inf. Model.* **2006**, *46*, 699–707.
- Fechner, U.; Schneider, G. Flux (2): Comparison of molecular mutation and crossover operators for ligand-based de novo design. *J. Chem. Inf. Model.* **2007**, *47*, 656–667.
- Schürer, S. C.; Tyagi, P.; Musk, S. A. Prospective exploration of synthetically feasible, medically relevant chemical space. *J. Chem. Inf. Model.* **2005**, *45*, 239–248.
- Vinkers, H. M.; de Jonge, M. R.; Daeyaert, F. F. D.; Heeres, J.; Koymans, L. M. H.; van Lenthe, J. H.; Lewi, P. J.; Timmerman, H.; Van Aken, K.; Janssen, P. A. J. SYNOPSIS: SYNthesis and OPTimize system in silico. *J. Med. Chem.* **2003**, *46*, 2765–2773.
- Lameijer, E. W.; Kok, J. N.; Back, T.; Ijzerman, A. P. The molecule evaluator. An interactive evolutionary algorithm for the design of drug-like molecules. *J. Chem. Inf. Model.* **2006**, *46*, 545–552.
- Lewell, X. Q.; Judd, D. B.; Watson, S. P.; Hann, M. M. RECAP - Retrosynthetic combinatorial analysis procedure: A powerful new technique for identifying privileged molecular fragments with useful applications in combinatorial chemistry. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 511–522.
- Daylight Chemical Information Systems, Inc., 120 Vantis - Suite 550, Aliso Viejo, CA 92656, USA. www.daylight.com at <http://www.daylight.com> (accessed Nov 2008).
- Leach, A. G.; Jones, H. D.; Cosgrove, D. A.; Kenny, P. W.; Ruston, L.; MacFaul, P.; Wood, J. M.; Colclough, N.; Law, B. Matched molecular pairs as a guide in the optimization of pharmaceutical properties; a study of aqueous solubility, plasma protein binding and oral exposure. *J. Med. Chem.* **2006**, *49*, 6672–6682.
- Sheridan, R. P.; Hunt, P.; Culberson, J. C. Molecular transformations as a way of finding and exploiting consistent local QSAR. *J. Chem. Inf. Model.* **2006**, *46*, 180–192.
- Taylor, K. T. The status of electronic laboratory notebooks for chemistry and biology. *Curr. Opin. Drug Discovery Dev.* **2006**, *9*, 348–353.
- Drake, D. J. ELN implementation challenges. *Drug Discovery Today* **2007**, *12*, 647–649.
- Broughton, H. B.; Hunt, P. A.; MacKey, M. D. Methods for Classifying and Searching Chemical Reactions. U.S. 2003/0182094 A1, 2003.
- Patel, H.; Gillet, V. J.; Chen, B.; Bodkin, M. J. Development of a de novo design tool using reaction vectors. Poster presented at the 4th Joint Sheffield Conference on Chemoinformatics, Sheffield, UK, 2007.
- Carhart, R. E.; Smith, D. H.; Venkataraghavan, R. Atom Pairs as Molecular-Features in Structure Activity Studies - Definition and Applications. *J. Chem. Inf. Comput. Sci.* **1985**, *25*, 64–73.
- JOELib. A java based computational chemistry package. <http://joelib.sourceforge.net/> (accessed January 31, 2008).
- Symyx, 2440 Camino Ramon, Suite 300, San Ramon, CA 94583.
- Berthold, M. R.; Cebon, N.; Dill, F.; Gabriel, T. R.; Kotter, T.; Meinl, T.; Ohl, P.; Sieb, C.; Thiel, K.; Wiswedel, B. KNIME: The Konstanz Information Miner. In *Data Analysis, Machine Learning and Applications*; Preisach, C., Schmidt-Thieme, L., Eds.; Springer-Verlag: Berlin, 2008; pp 319–326.
- ChemAxon, Máramaros köz 3/a, Budapest, 1037 Hungary.
- Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235–242.
- Wang, L. X.; Shen, J. F.; Tang, Y.; Chen, Y.; Wang, W.; Cai, Z. G.; Du, Z. J. Synthetic improvements in the preparation of clopidogrel. *Org. Process Res. Dev.* **2007**, *11*, 487–489.
- Kavitha, B. C. V.; Rangappa, K. S. Simple and an efficient method for the synthesis of 1-[2-dimethylamino-1-(4-methoxy-phenyl)-ethyl]-cyclohexanol hydrochloride (\pm) venlafaxine racemic mixtures. *Bioorg. Med. Chem. Lett.* **2004**, *14*, 3279–3281.
- SciFinder, Chemical Abstracts Service, P.O. Box 3012, Columbus, Ohio 43210, USA.
- Leach, A. R.; Bradshaw, J.; Green, D. V. S.; Hann, M. M.; Delany III, J. J. Implementation of a System for Reagent Selection and Library Enumeration, profiling and Design. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 1161–1172.

CI800413M