ELSEVIER

Efficient combinatorial filtering for desired molecular properties of reaction products

Shenghua Shi, Zhengwei Peng, Jaroslaw Kostrowicki, Genevieve Paderes, and Atsuo Kuki

Alanex Research Division, Agouron Pharmaceuticals, Inc., La Jolla, California, USA

Two combinatorial filtering methods for efficiently selecting reaction products with desired properties are presented. The first, "direct reactants" method is applicable only to those molecular properties that are strictly additive or approximately additive, with relatively small interference between neighboring fragments. This method uses only the molecular properties of reactants. The second, "basis products" method can be used to filter not only the strictly additive properties but also the approximately additive molecular properties where a certain degree of mutual influence occurs between neighboring fragments. This method requires the molecular properties of the "basis products," which are the products formed by combining all the reactants for a given reaction component with the simplest set of complementary reactant partners. There is a one-to-one correspondence between the reactants and the "basis products." The latter is a product representation of the former. High efficiency of both methods is enhanced further by a tree-sorting and hierarchical selection algorithm, which is performed on the reaction components in a limited space determined systematically from the filtering criteria. The methods are illustrated with product logPs, van der Waals volumes, solvent accessible surface areas, and other product properties. Good results are obtained when filtering for a number of important molecular properties in a virtual library of 1.5 billion.

Keywords: combinatorial, library design, filtering, molecular properties, reaction, reactants, products

INTRODUCTION

Modern-day combinatorial chemistry provides the opportunity to quickly synthesize extremely large numbers of compounds. The limiting step in high-throughput technology no longer is the synthesis or screening of compounds, but rather the analysis and decisions required before synthesis as well as after screening.[1-3] The potential number of compounds that can be synthesized from readily available components is enormous (it can easily be on the order of billions in a single structural family) and, therefore, choosing the desired subset might involve extensive calculations. In a design of a focused (targeted) library, one seeks to identify the products that are functionally similar to a known active compound and which simultaneously satisfy certain constraints on their molecular properties[4,5] from the consideration of, for example, oral bioavailability.[5] Computational screening of candidates for synthesis might be particularly time consuming when it involves the examination of conformation-dependent 3D properties relevant for binding to an enzyme or receptor. Therefore, there is a strong need for a fast and efficient filtering method that would eliminate inappropriate products before proceeding to more expensive computational screening steps.[6] The magnitude (10^6–10^{11}) of the initial, total number of virtual products demands that this initial filtering be very fast. Because we are dealing with virtual products, which have a natural combinatorial structure, the search across all virtual products can and should be guided by the components used in synthesis. The simplest implementation might be to filter the reaction components independently. This is perfectly legitimate, for example, in the case of filtering for the absence of a certain toxic fragment: if it is not in any component, it will not be in the products. For general properties, the independent filtering approach might be too restrictive. For example, if we want products with molecular weight <500 amu, then to be sure that it is fulfilled, we might require that both reactants of a two-component reaction have molecular weight <250 amu (if we were to disregard the reaction core mass for simplicity). This eliminates perfectly acceptable pairs of masses, e.g., 100 and 400 amu.

The simple independent reactant filtering method illustrates one of four main categories of virtual product filtering algorithms. These algorithms can be classified into four categories based on whether the *input* is a set of reactant properties or a

Corresponding author: Atsuo Kuki, Alanex Research Division, Agouron Pharmaceuticals, Inc., La Jolla, CA 92121, USA. Tel.: 858-455-3256; fax: 858-455-3201. *E-mail address*: atsuo.kuki@agouron.com (A. Kuki).

Journal of Molecular Graphics and Modelling 18, 478–496, 2000

655 Avenue of the Americas, New York, NY 10010

1093-3263/00/$–see front matter
PII S1093-3263(00)00069-3

set of properties of enumerated products and whether the *output* of selected products forms a fully combinatorial superarray or a sparse matrix.

In contrast with the direct preparation of an enumerated virtual product database and direct mining, which, obviously, belongs to the category of enumerated product-in and individual product-out, the independent filtering method has reactant properties as input and has as output a set of products forming a fully combinatorial superarray. We are interested in the algorithms that outperform either of these two extremes of filtering methods. Speed and efficiency relative to the full virtual database mining can be achieved either by sampling and on-the-fly enumeration algorithms[7] prior to product property evaluation, or by systematically exploiting the known properties of the reaction components.

The systematic examination of reactant properties to construct a library consisting of only suitable products without an explicit evaluation of these products is the subject of this article. However, before we discuss the more detailed description of our method, we briefly characterize other techniques that evaluate the products directly.

An efficient sampling of the virtual product set can be performed in two different ways, depending on the aforementioned output format of the selected enumerated products. In the first, a fully combinatorial sublibrary is selected; in the second, single virtual products are examined and individually selected, thus forming a sparse matrix library design. Several probabilistic optimization techniques have been adopted to solve this sampling problem.

When the optimization is performed on the basis of fully combinatorial sublibraries, a genetic algorithm can be effectively applied that scores the set of all virtual products in a given sublibrary, and the population of sublibraries evolves according to the rules of mutation and crossover.[8,9] Mutation replaces the whole column or whole row of products with another randomly selected alternative; analogously, the crossover of two different sublibraries within the current pool draws columns and rows from both of them to create a new sublibrary.

In the optimization of single virtual products,[10] mutation involves the random replacement of some of the components within a given product and the crossover of two virtual products in the current pool draws reaction components from both of them to produce the offspring, as a new virtual product. Another technique used for optimization at the level of single product is simulated annealing,[7] in which the mutations are accepted or rejected by virtue of the Metropolis criterion.

Optimization of individual virtual products generates designs of sparse matrices that do not have fully combinatorial format and therefore might need to be followed with an additional reduction step to compact the array. This task can be achieved readily by calculating the frequencies of occurrence of various components among the best selected virtual products and then identifying the most frequent components as the final design.[10] The generation of combinatorial or nearly combinatorial subsets from a larger sparse set of selected products also can be done by a Monte Carlo technique.[11]

It should be stressed that the probabilistic methods might not necessarily generate all the products within the prescribed range of product property values. Moreover, optimization in the space of virtual products involves explicit formation of product structures and pairwise comparisons of products that might significantly slow down the method.

The need for a rapid and systematic method is clear: it is highly desirable for a medicinal chemist to be able to quickly select compounds that can be made with known chemistry without forming the product structures explicitly and without searching through the resulting huge number of virtual products. To meet this need, two methods for efficiently calculating product molecular properties and filtering for reaction products with desired molecular properties are presented in the present work. Instead of independent or random choices of reactants, we make the choice of one component systematically dependent on the choice of another. For example, similar to the algorithm[12] used for product structure enumeration based on a molecular weight-driven constraint, we start with a given reactant choice for one reaction component, and then the constraint on the molecular weight of the product can be easily translated into the constraint on the molecular weight of the second component. This feature and its generalization for other properties are developed into the filtering algorithm described in the present work.

The methods described here use properties derived from the reactants as input and returns multiple combinatorial subarrays of various sizes. The output is a certain kind of product array that ranges from variants of block diagonal to sparse arrays, depending on the input. The present hierarchical algorithm selects components of the desired products in a way that eliminates the need for the actual formation of product structures and expensive direct evaluation of their properties.

In the Methodology section, the derivation of the methods for calculating and filtering the molecular properties of reaction products is introduced. In addition, the applicability and limitations of the methods are discussed. The hierarchical combinatorial filtering and the tree-sorting algorithms are described for both multicomponent and multiproperty filtering. In the following section, illustrative calculations are presented for determining as well as filtering of molecular properties of reaction products such as molecular weight, number of hydrogen bond donors and hydrogen bond acceptors, LogP, etc., as listed in Table 1. A brief discussion and remarks are included in the final section.

METHODOLOGY

Combinatorial Calculation of Property Values of Reaction Products

Consider a K-component reaction v:

$$\sum_{i=1}^{K} A_i \rightarrow P. \tag{1}$$

If there are N_i reactants for the reaction component A_i, then reaction 1 would generate N_v products:

$$N_v \equiv \prod_{i=1}^{K} N_i. \tag{2}$$

If N_i is on the order of thousands, the virtual products, which would be on the order of 10^{3K}, could easily be billions (K = 3) and trillions (K = 4).

We are interested in filtering reaction products according to certain molecular properties. One of the simple ways is to

Table 1. Summary of molecular properties and their applicability in Equations 6 and 11

Molecular property	Level of additivity	Model applicability: "Direct reactants" Equation 6	Model applicability: "Basis products" Equation 11	Comments
Molecular weight	Exact	Yes	Yes	
Number of HBD/HBA	Exact	Yes	Yes	
Number of positive/negative ionizable groups	Exact	Yes	Yes	
Number of aromatic rings	Exact	Yes	Yes	
Number of N and O atoms	Exact	Yes	Yes	
vdW volume (VWVOL)	Excellent	Yes	Yes	
SLOGP	Good	Yes		
CLOGP	Good	No	Yes	Not good for zwitterions formed by reactions
Solvent accessible volume (SAVOL)	Good	~Yes	Yes	
Solvent accessible surface (SA)	Fair	~Yes	~Yes	Able to identify low/medium/high
Polar solvent accessible surface (PSA)	Fair	~Yes	~Yes	Able to identify low/medium/high
MLOGP	Poor	No	No	

enumerate all the virtual products, determine the property values of all these products, and then filter for the desired products that pass the specified thresholds. This procedure would amount to formation of N_v products plus property determination of the order of N_v and ~ N_v comparisons against the filtering criteria. The effort involved would be enormous. To take advantage of the fact that all the virtual products are formed from a total of N_{tot} reactants:

$$N_{tot} \equiv \sum_{i=1}^{K} N_i, \qquad N_{tot} \ll N_v \tag{3}$$

we propose a combinatorial procedure for product filtering. Reaction 1 as displayed in Figure 1 can always be rewritten as:

$$\sum_{i=1}^{K} (R_i a_i) \rightarrow R_1 R_2 \ldots R_K p. \tag{4}$$

Whereas a_i is the reactive and constant fragment (substructure) of the component A_i, R_i is its remaining and varying fragment. That is, the N_i different choices for the reaction component A_i have the same a_i, but N_i different R_i. The product core, which is the same for all products from a given reaction, is denoted by p in Equation 4. It should be noted that a_i, R_i, and the product core p are not necessarily connected substructures. They may consist of several disconnected pieces of substructures. For example, R_i for the ketone reaction component A_i in Figure 2 consists of two R-groups (R and R'), and the product core p for the product in Figure 3 includes the disjointed substructures at the sulfur atom and the amide group.

Suppose that a molecular property Q is additive (such as molecular weight, or the number of hydrogen bond donors and hydrogen bond acceptors), i.e., the value of the property for the whole molecule is equal to the sum of the properties of its

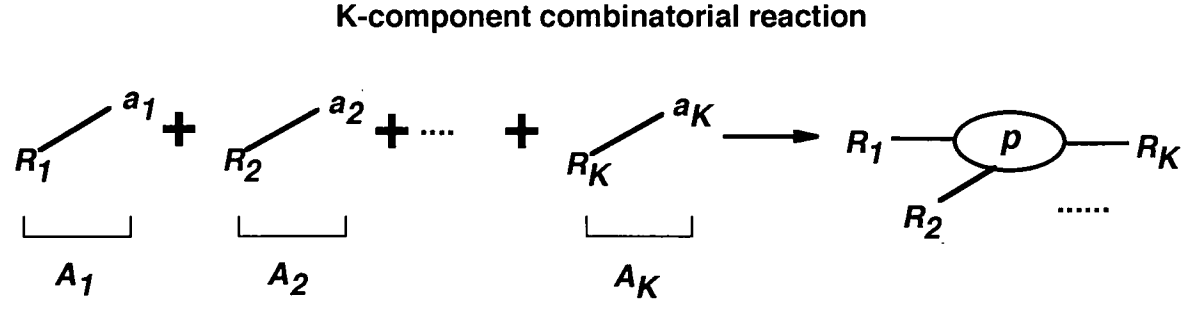


Figure 1. Simple reaction scheme for a K-component combinatorial reaction. As defined in the text, a_1 *to* a_K *are the reactive and constant fragments (substructures) of their corresponding reaction components.* R_1 *to* R_K *are the remaining and varying fragments (R-groups) of the reactants which become the R-groups of the product. The common product core for all products of this combinatorial reaction is denoted here as* p.

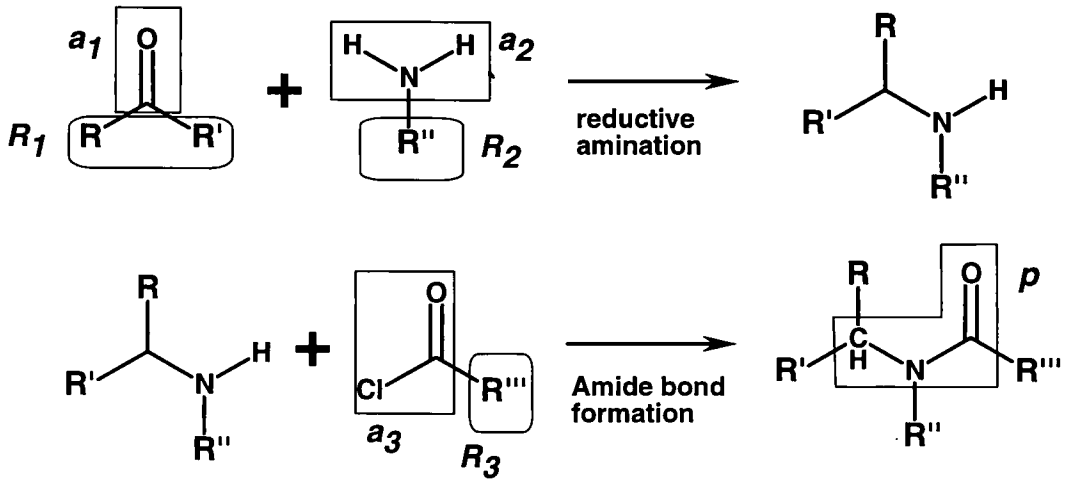


Figure 2. Reaction scheme for a three-component, two-step combinatorial reaction. It consists of one reductive amination step followed by an amide bond formation step. R, R′, R″, *and* R‴ *are the R-groups from the three reactants. The scheme exemplifies the star-shaped topology depicted in Figure 1.*

fragments, given that the fragments are carefully defined. Then for any given product molecule P, we have:

$$Q(P) = \sum_{i=1}^{K} Q(R_i) + Q(p) = \sum_{i=1}^{K} Q(A_i) + Q(p) - \sum_{i=1}^{K} Q(a_i). \quad (5)$$

Because the reactive fragment a_i of component A_i and the product core p are constant for a given reaction, Equation 5 for the property Q of a product molecule P becomes:

$$Q(P) = \sum_{i=1}^{K} Q(A_i) + \Delta Q(\text{rxn}) \quad (6)$$

with

$$\Delta Q(\text{rxn}) \equiv Q(p) - \sum_{i=1}^{K} Q(a_i), \quad (7)$$

where $\Delta Q(\text{rxn})$ is a reaction-dependent constant. Equations 6 and 7 state that the property of a product can be easily calculated by adding a reaction-dependent constant to the sum of the corresponding property values of the reaction components. As an example, for the reaction shown in Figure 2, the reaction-dependent constant for molecular weight, ΔMW, can be calculated as:

$$\Delta MW = MW(p) - (MW(a_1) + MW(a_2) + MW(a_3))$$

$$= -52.47.$$

Because we can calculate and store the property values of all ($\sim N_{tot}$) of the available reagents, on the order of thousands to hundreds of thousands (e.g., from the MDL ACD database[13]), the properties of all (N_v) of the virtual products on the order of billions to trillions can be obtained very efficiently without

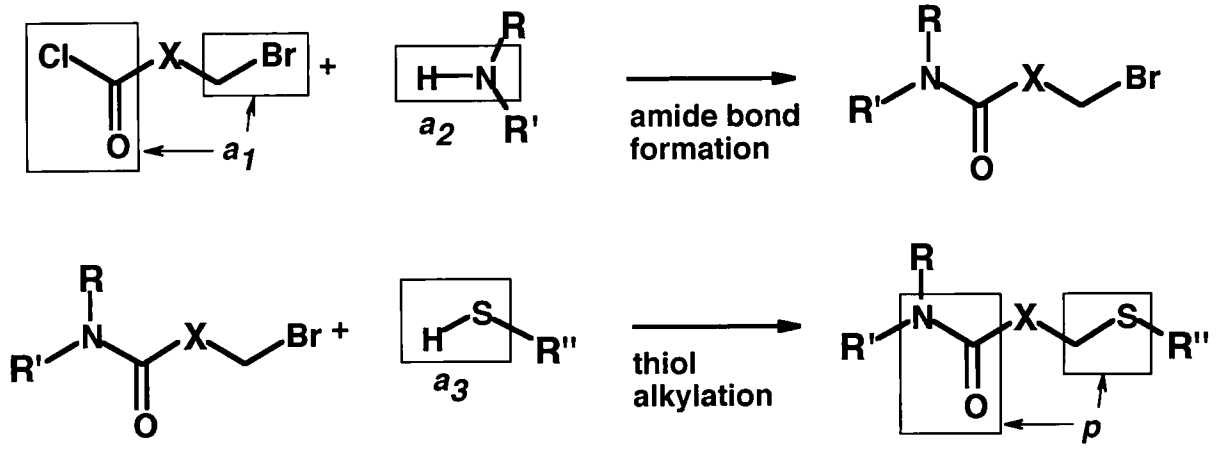


Figure 3. Reaction scheme for another three-component, two-step combinatorial reaction. It consists of one amide bond formation step followed by a thiol alkylation step. The X stands for a linking group, and R, R′, *and* R″ *are the R-groups from the second and third reactants. In this reaction, the product core* (p) *consists of two disjointed pieces, an amide group and a sulfur atom separated by the variable X bridging substructure. In turn, the reactive fragment* (a_1) *of the first reactant contains two disjointed pieces. In this method, the linking group X becomes* R_1 *of the first reactant (in Eq. 4).*

forming the product structures explicitly and without performing expensive property calculations.

Some molecular properties, strictly speaking, are not additive, but are approximately additive in the sense that the property (e.g., van der Waals volume[14]) is calculated with some model (or empirical formula) by summing up all the contributions from its fragments. If they are (or approximately are) independent of each other, then Equation 6 for these molecular properties will still hold.

It should be noted that Equation 7 indicates that the reaction-dependent ΔQ(rxn) can be calculated with the property values, $Q(p)$ and $Q(a_i)$, of molecular fragments. However, for certain molecular properties such as LogP, the property of interest has no exact meaning for molecular fragments. Moreover, the empirical formula used for calculating the property may involve a constant term. Notwithstanding, because ΔQ(rxn) is a constant for a given reaction, it can alternatively be determined from the property values of a particular set $\mathbf{A}^*$ of reaction components $\{A_i^* = (R_i^* a_i^*), i = 1,\ldots,K)\}$ and the particular product P^* formed from them. That is, Equation 7 may be replaced by:

$$\Delta Q(rxn) \equiv Q(P^*) - \sum_{i=1}^{K} (Q(A_i^*)). \qquad (8)$$

Note that, with this formulation, all the molecular property values in Equations 6 and 8 are calculated on complete molecules. An explicit example is shown in Figure 4, where both molecular weight and LogP, calculated as SLOGP,[15] are considered. It is seen that, for molecular weight, either Equation 8 or 7 can be used to calculate ΔMW. However, for SLOGP only Equation 8 can be utilized for determination of ΔSLOGP.

It should be emphasized that Equations 6 and 7 (or 8) are valid only for the additive properties (or calculated properties with additive features). It is assumed that for both reaction components and products, the varying and constant fragments have no influence on each other's contributions to the overall property. The key requirement is that the property values (or contribution to the property values) of the fragments R_i in the reaction component A_i would be the same as that for R_i in the reaction product P. In other words, the influence of a_i on R_i must be small or, less restrictively, the influence of the a_i on R_i should be *very similar* to the influence of p on R_i. Furthermore, it also is assumed that the property (or the contribution to the property) of the reactive fragment a_i would be the same for all the reactants in the reaction component A_i despite the changes in R_i. As we will see, this is exact for molecular weight and is a very good approximation for the number of hydrogen bond donors and acceptors, and van der Waals volume in most cases.

The property calculations in Equations 6 and 8 are performed on complete molecules A_i and P^*. In the next subsection, we will pursue this direction one step further to base the combinatorial product property calculation on a "basis" set of complete molecules that are all in the P product chemical family.

Improved Combinatorial Calculations of Product Properties Based on Properties of "Basis Products"[16]

In chemical reactions, whereas the change from the original reactive functional groups to the product core usually is large, the variation in the core-linkage of R_i may be quite limited. For example, in the reaction depicted in Figure 2, the reactive fragment NH_2 in the second reaction component, which is positively charged under normal pH conditions, is changed to the N of the amide fragment in the product, which is neutral under normal pH conditions. This change from basic sp^3 N to nonbasic sp^2 N may have a significant nearest neighbor influ-

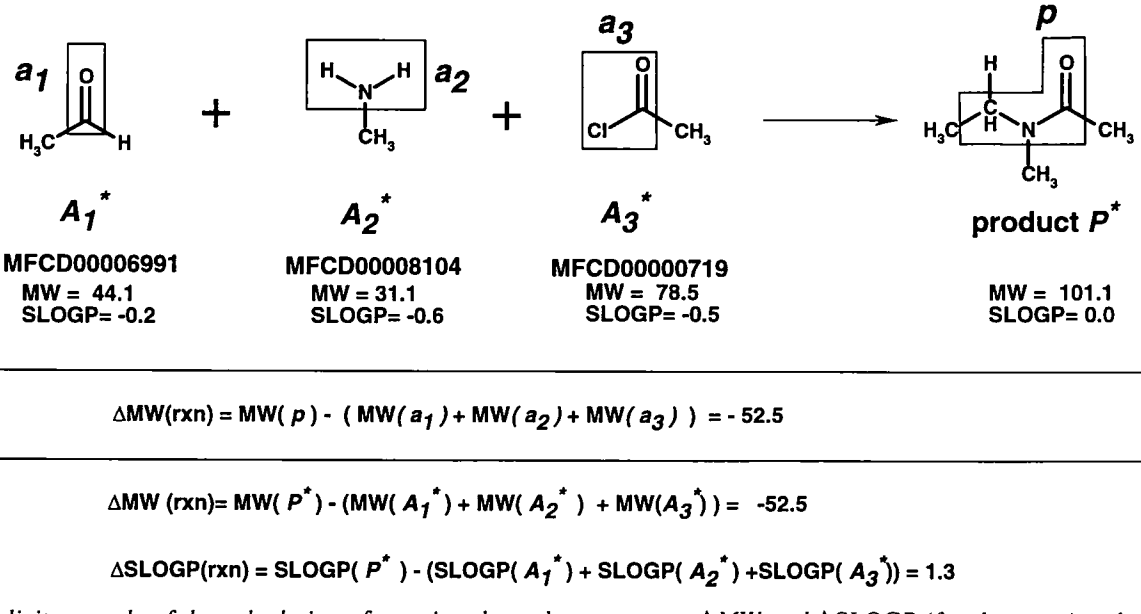


Figure 4. Explicit example of the calculation of reaction-dependent constants ΔMW and ΔSLOGP (for the reaction depicted in Figure 2) according to Equations 7 and 8, respectively. Equation 7 uses atomic properties, whereas Equation 8 is the appropriate form for properties whose calculation requires complete molecules.

ence on the contribution to the molecular property from the appended fragment R″. This would be a case of molecular electron density redistribution or "charge transfer." As a result, while the assumption made for Equation 6 that the property (or the contribution to the property) of the fragment R_i remains unchanged from reactant to product may not be valid, the assumption that the property (or the contribution to the property) of the product core remains constant for all the products (despite the changes in R_i) could be a good approximation. To take advantage of this observation, we formulate the calculation a little differently. Suppose that a set of simplest reaction components is selected as the particular set $\mathbf{A}^*$ for a given reaction v. For this specially selected set $\mathbf{A}^*$ of reaction components, Equation 5 becomes:

$$Q(P^*) = \sum_{i=1}^{K} Q(R_i^*) + Q(p), \tag{9}$$

where P^* is the particular product formed from this selected set of reactants $\mathbf{A}^*$, a 'capped product core.' Next, introducing 'basis products' P_j^*, which are the products formed from all the reactants in a given reaction component A_j with all (K − 1) other components selected from the particular set $\mathbf{A}^*$, then from Equation 5 one has:

$$Q(P_j^*) = \sum_{i=1}^{j-1} Q(R_i^*) + Q(R_j) + \sum_{i=j+1}^{K} Q(R_i^*) + Q(p). \tag{10}$$

Simple algebra gives for the property Q of a virtual product:

$$Q(P) = \sum_{i=1}^{K} Q(P_i^*) - (K-1)Q(P^*). \tag{11}$$

An example of the particular set $\mathbf{A}^*$ and the "basis products" P_j^* for the reaction shown in Figure 2 is given in Figure 5.

If there are N_i choices for the reaction component A_i, then Equation 11 tells us that instead of forming and calculating the property for ΠN_i products, one only needs to form the structures and perform the property calculation for ΣN_i "basis products" P_j^* and one special product P^*. The gain in efficiency is obvious. Moreover, as we will see in the next section, Equation 11, which assumes only that the contribution from the product core remains a constant for all the products, presents a much better approximation than Equation 6 for certain calculated and approximately additive properties.

In chemical terms, the advantage of Equation 11 is that all the computed inputs are calculated on complete molecules that are all within the P product chemical family, where the nearest neighbor influences occurring across the bonds formed in the reaction are already fully incorporated.

Database-Free Virtual Product Mining: Combinatorial Hierarchical Filtering Method to Locate Desired Virtual Products by a Hierarchical Retrieval Operating on Reactants or "Basis Products"

First let us consider filtering the virtual products according to a positive property value Q for a given reaction v. (Note that any property value can be converted to a positive one by a constant shift for filtering purposes.) That is, we would like to select the virtual products **P** such that:

$$w_1 \leq Q(\mathbf{P}) \leq w_2, \tag{12}$$

where both w_1 and w_2 are non-negative values. The output of this filter will be individual combinations of reactants specifying products P in a sparse matrix that satisfy the filtering criteria in Equation 12. According to Equation 6, it is easy to show that for any virtual products to satisfy Equation 12, the reaction component A_j with j< K has to satisfy the requirement:

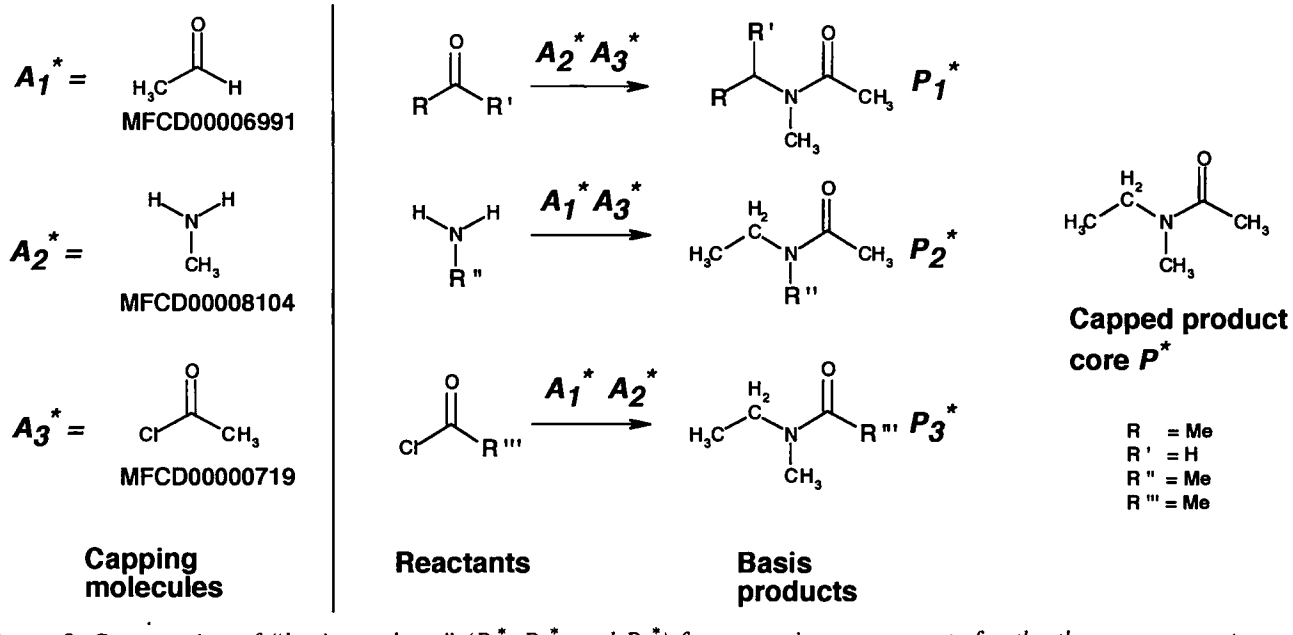


Figure 5. Construction of "basis products" (P_i^*, P_2^*, and P_3^*) from reaction components for the three-component combinatorial reaction depicted in Figure 2. The three capping molecules A_1^*, A_2^*, and A_3^* are given explicitly. The product formed by those three capping molecules is the associated "capped product core" P^*.

$$0 \leq Q(A_j) \leq w_2 - \Delta Q(\nu) - \sum_{i=1}^{j-1} Q(A_i), \quad (13)$$

and for the reaction component K the requirement is:

$$Max\,[0,(w_1 - \Delta Q(\nu) - \sum_{i=1}^{K-1} Q(A_i))] \leq Q(A_K) \leq w_2 - \Delta Q(\nu) - \sum_{i=1}^{K-1} Q(A_i). \quad (14)$$

Equation 13 states that for the first reaction component, the reactants to be used have to satisfy Equation 15:

$$0 \leq Q(A_1) \leq w_2 - \Delta Q(\nu). \quad (15)$$

With the component A_1 being selected in accordance with Equation 15, the reaction component A_2 has to comply with Equation 16:

$$0 \leq Q(A_2) \leq w_2 - \Delta Q(\nu) - Q(A_1). \quad (16)$$

The component A_3 has to be a solution of Equation 17:

$$0 \leq Q(A_3) \leq w_2 - \Delta Q(\nu) - Q(A_1) - Q(A_2); \quad (17)$$

and so on until j = K where Equation 14 applies.

Equations 13 and 14 can be readily generalized to the case involving simultaneous multiple property filtering by simply replacing the scalar values with vectors:

$$\mathbf{0} \leq \mathbf{Q}(A_j) \leq \mathbf{w}_2 - \Delta\mathbf{Q}(\nu) - \sum_{i=1}^{j-1}\mathbf{Q}(A_i), \text{ with } j < K; \quad (18)$$

and

$$Max\,[\mathbf{0},(\mathbf{w}_1 - \Delta\mathbf{Q}(\nu) - \sum_{i=1}^{K-1} \mathbf{Q}(A_i))] \leq \mathbf{Q}(A_K) \leq \mathbf{w}_2 - \Delta Q(\nu) - \sum_{i=1}^{K-1} \mathbf{Q}(A_i). \quad (19)$$

It should be noted that Equations 18 and 19 represent a set of simultaneous equations, one for each of the multiple properties, that the reaction components A_j, j = 1, . . . , K, have to satisfy.

For the given filtering criteria (specified $\mathbf{w}_1$ and $\mathbf{w}_2$), by presorting each of the reaction components according to their properties prior to selection, Equation 18 and 19 can be used to efficiently filter for specific combinations of reactants A_i, i = 1,...,K that would give rise to all the products whose properties satisfy Equation 12. Because no property values need to be calculated for any of the virtual products, the time saving is tremendous with Equations 18 and 19.

Because there is a one-to-one correspondence between P_i^* and A_i, for a better approximation, Equation 11 can be used in the hierarchical filter to yield all the products P whose property $Q(P)$ satisfies the desired criteria, e.g., Equation 12. To meet the filter criterion, Equation 12, the property values of the "basis products" P_j with j < K have to be within the bounds:

$$0 \leq Q(P_j^*) \leq w_2 + (K-1)Q(P^*) - \sum_{i=1}^{j-1} Q(P_i^*), \quad (20)$$

and for j = K, one has:

$$Max\,[0,(w_1 + (K-1)Q(P^*) - \sum_{i=1}^{K-1} Q(P_i^*))] \leq Q(P_K^*) \leq w_2 + (K-1)Q(P^*) - \sum_{i=1}^{K-1} Q(P_i^*). \quad (21)$$

Similarly, for multiple property filtering, one has:

$$\mathbf{0} \leq \mathbf{Q}(P_j^*) \leq \mathbf{w}_2 + (K-1)\mathbf{Q}(P^*) - \sum_{i=1}^{j-1} \mathbf{Q}(P_i^*), \text{ with } j < K; \quad (22)$$

and

$$Max\,[\mathbf{0},(\mathbf{w}_1 + (K-1)\mathbf{Q}(P^*) - \sum_{i=1}^{K-1} \mathbf{Q}(P_i))] \leq \mathbf{Q}(P_K^*) \leq \mathbf{w}_2 + (K-1)\mathbf{Q}(P^*) - \sum_{i=1}^{K-1} \mathbf{Q}(P_i). \quad (23)$$

For any given reaction, if we calculate and store all the property values for the "basis products" P_j^* and "capped product core" P^*, Equations 22 and 23 can be readily utilized to identify efficiently the virtual products with the desired property values. (For details, see Appendix A).

The solutions to Equations 22 and 23 (or Equations 18 and 19) do not form a complete combinatorial superarray. They form many individual combinatorial arrays of various sizes as depicted, e.g., in Figure 6. The efficiency of this filtering method stems from the fact that it does not require calculation of any properties of the numerous virtual products and only searches through a limited number of blocks of reactants (or "basis products") in a hierarchical way. For example, for a two-component reaction, if we are only interested in the reaction products with two or three hydrogen bond acceptors, then, as shown in Figure 6, only the shaded blocks would be considered instead of the whole space. For a multicomponent reaction and for multiproperty filtering, the reduction of search space and, thus, the saving on computation time would be tremendous.

Pan-Reaction Combinatorial Filtering to Locate Products with Desired Molecular Properties

Thus far, we have only been concerned with a given reaction. However, by simply looping through all the reactions in a reaction knowledge base, it is trivial to use the algorithm developed to find from any of these reactions all the products with the desired molecular property values without actually searching through the entire multireaction virtual product space. This is a powerful feature that will be explored in a separate study.[17]

An Efficient Tree Algorithm for Filtering Discrete or Continuous Properties of Products

Some of the molecular properties of interest are integers, such as the number of hydrogen bond donors and/or hydrogen bond acceptors. However, the majority of the molecular properties for a product molecule are real valued. Solving the simultaneous Equations 18 and 19 (or 22 and 23) with real-valued properties is not very efficient unless some care is taken. Moreover, the filtering criteria in Equation 12 often are not determined accurately, but are

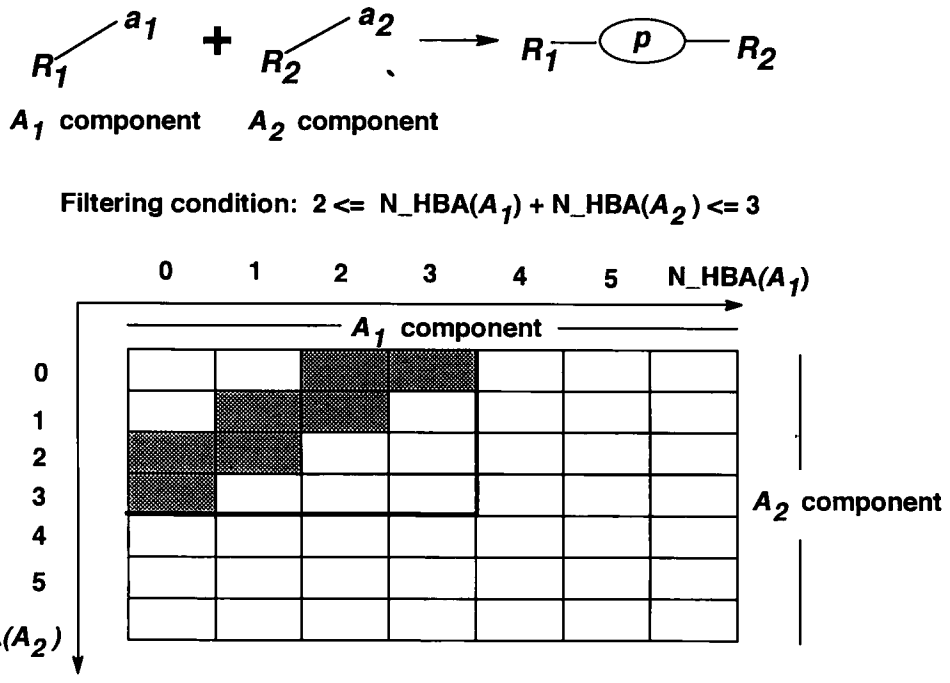


Figure 6. Illustration of the efficiency of the filtering method. Here, a two-component reaction is used as an example. A_1 *and* A_2 *reaction components are presorted and clustered based on the number of hydrogen bond acceptors (N_HBA) they contain. When asked to find all product compounds with two or three hydrogen bond acceptors (assuming the product core* p *does not have any), the filtering algorithm will examine only* $4 \times 4 = 16$ *blocks to find those seven shaded blocks satisfying the filtering condition. This is the first part of the savings. Also, each block may contain many individual products. Instead of making pass-or-fail decisions for each individual product, the algorithm makes those decisions at the block level, one simple decision per block, and thereby gains the second part of the speed advantage. This can be an enormous factor, because each block may contain thousands or even millions of virtual products. The shaded area also depicts the four combinatorial subarrays formed by the selected products as the output of the filter.*

instead just statistically desired property ranges. To improve efficiency, the real-valued molecular properties are first sorted, and then grouped into bins with bin sizes comparable to the inherent accuracy of the calculated properties.

For better efficiency in solving the M simultaneous Equations 18 and 19 (or 22 and 23), a tree-sorting algorithm can be used. The algorithm is a simple recurrence sorting algorithm where at generation m, $1 \leq m \leq M$, all the compounds have the same (within the bin size for real-valued property) property values for properties 1 to $(m - 1)$ and the value for property m is used for sorting. The tree sorting is first performed against M properties used in filtering for reactants (or "basis products") of each of reaction components. Then, to obtain the solution to Equations 18 and 19 (or 22 and 23), one just simply counts the population of the M-th generation.

For example, let us consider a case of filtering by three molecular properties, molecular weight (MW), number of hydrogen bond acceptors (N_HBA), and number of hydrogen bond donors (N_HBD). We assume that the molecular weights for all the reactants are <500 amu, and the number of hydrogen acceptors and the number of hydrogen donors are both <4. As shown in Figure 7, being a root node, the reactants for each of the reaction components are first sorted by, e.g., molecular weight and are grouped into 5 bins with a bin size of 100 amu (just for illustration). The reactants in each of the five bins, which serve as the nodes for next generation, are sorted further by the number of hydrogen bond acceptors. The reactants with the same number of hydrogen bond acceptors are assigned into bins that then become the nodes for next generation. Note that the reactants in each of the nodes have the same bin-valued molecular weight and the same number of hydrogen bond acceptors. They again are sorted by the next property, the number of hydrogen bond donors. The compounds with the same number of hydrogen bond donors form new nodes, which are the leaf nodes in the present case. The reactants in each of the leaf nodes have the same set of bin values for all three molecular properties. Therefore, for a set of specified molecular properties, we just simply follow the tree, as indicated in Figure 7, from the root node to the leaf nodes, and the reactants in the final resulting leaf nodes are the ones with desired molecular properties.

One should bear in mind that although the binning procedure for real-valued properties enhances the efficiency of filtering significantly, it introduces errors that would generate false positives as well as false negatives (for detail, see Appendix B). However, the user can always adjust the bin sizes and the lower and upper bounds to achieve the desired filtering objectives. For example, for purposes of filtering, one usually wants to avoid false negatives. To eliminate false negatives that result from binning, one may broaden the filter window at a price that more false positives would be included. In order to improve both effi-

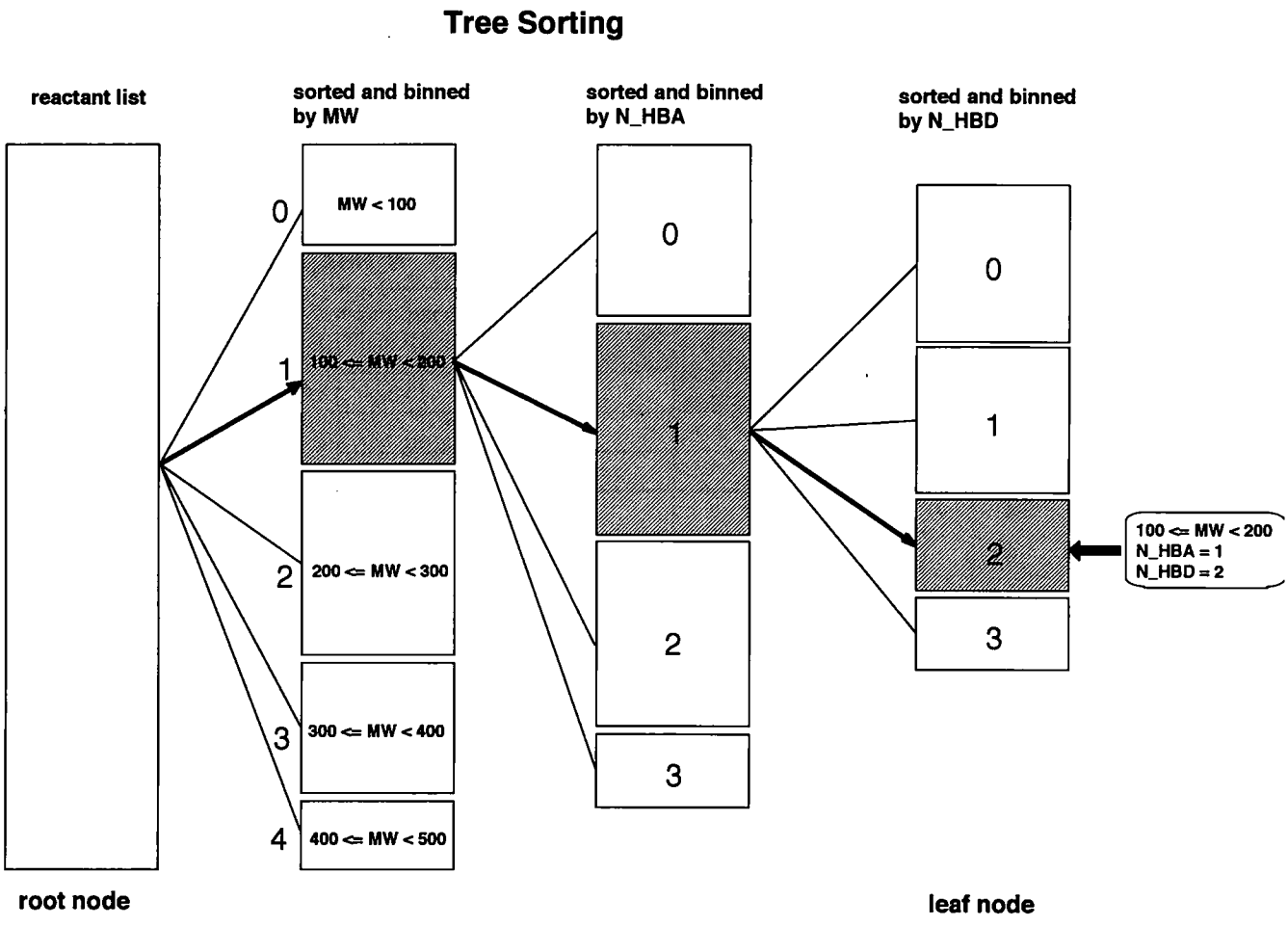


Figure 7. Simple illustration of the tree-sorting scheme. Starting with a given reactant list at the root node, one creates a tree structure by sorting and binning reactant lists based on molecular properties (such as MW, N_HBA, N_HBD, etc.) that one wants to use for product filtering. Each block in the figure represents a list of reactants. Once this tree structure has been constructed, one can find the group of reactants that satisfies the filtering condition {100 ≤ MW < 200, N_HBA = 1, and N_HBD = 2} very efficiently by traversing the tree from the root node to the correct leaf node in just three steps. The product filtering is achieved by applying this tree sorting to each component within Eqs. 18 and 19, or within Eqs. 22 and 23.

ciency and accuracy, one may do the filtering in stages. In practice, two-stage filtering is usually sufficient. At the initial stage, a relatively large bin size and a broad filter window may be applied. With the reduced size of the resulting set as the new domain, a smaller bin size and the tighter desired filter upper bound and lower bound can then be utilized.

ILLUSTRATIVE CALCULATIONS

To illustrate the use of Equations 6 and 11 for estimating molecular properties of reaction products and Equations 18, 19, 22, and 23 for product filtering based on those estimated molecular properties, we used the two-step, three-component combinatorial reaction sequence consisting of a reductive amination step followed by an acylation step, as depicted in Figure 2. As mentioned in the Methodology section, the present methods can be used with any known reaction. The choice of this particular reaction is simply due to the fact that this is a published,[18] well-known reaction. In addition, this reaction is genuinely capable of generating a very large library based on suitable commercially available reagents from MDL ACD991.[13] Even with 500-amu molecular weight cut-off and a salt-free constraint for each reagent, a conservative estimation still leads to 3,023 (aldhydes/ketones) × 1,132 (1°-amines) × 430(acid chlorides) = 1.47 billion products. It is too expensive (and not feasible) to directly calculate molecular properties of all 1.47 billion virtual products for property-based product filtering and selection.

Combinatorial Property Calculation of Reaction Products

In order to probe the accuracy of the two product property approximation methods described by Equations 6 and 11, we constructed a much smaller 21 (aldehydes/ketones) × 21 (1°-amines) × 22 (acid chlorides) fully combinatorial library of

Table 2. Results of linear fitting: Property (product) = k * Property (estimated) + c

Item	k (fitting error)	c (fitting error)	R	RMS error	Mean absolute error
MLOGP: Equation 6	0.724 (0.002)	−0.38 (0.02)	0.955	0.46	0.35
MLOGP: Equation 11	0.794 (0.002)	0.04 (0.01)	0.978	0.32	0.25
SLOGP: Equation 6	0.867 (0.004)	0.47 (0.03)	0.918	0.92	0.70
SLOGP: Equation 11	1.0020 (0.0003)	−0.021 (0.002)	0.9996	0.06	0.01
CLOGP: Equation 6	0.758 (0.004)	1.71 (0.03)	0.902	0.96	0.76
CLOGP: Equation 11	1.006 (0.002)	−0.03 (0.02)	0.978	0.47	0.25
SA: Equation 6	0.859 (0.002)	57.8 (2.8)	0.952	48.1	38.1
SA: Equation 11	0.943 (0.002)	17.3 (1.7)	0.986	31.5	24.4
PSA: Equation 6	0.879 (0.002)	19.9 (0.2)	0.969	16.2	12.5
PSA: Equation 11	0.911 (0.002)	7.5 (0.2)	0.986	10.5	7.6
SAVOL: Equation 6	0.929 (0.001)	213.2 (2.8)	0.986	50.8	39.6
SAVOL: Equation 11	0.970 (0.001)	12.8 (1.8)	0.994	35.2	26.8
VWVOL: Equation 6	0.9825 (0.0005)	1.6 (0.3)	0.999	6.3	4.6
VWVOL: Equation 11	0.9950 (0.0004)	0.0 (2.4)	0.999	4.9	3.4

9,702 product molecules using randomly selected reagents from their corresponding suitable reagent sets. A single 3D conformation for each reagent, "basis product," and product molecule is generated by CORINA[19] from a 2D coordinate file in MDL SDF format. The "basis products" of this three-component reaction are depicted in Figure 5. The following molecular properties are calculated for all reagents, their corresponding "basis products," and their 9,702 combinatorial library products. These properties are molecular weight, number of hydrogen bond acceptors,[14] MLOGP,[20] SLOGP,[15] CLOGP,[21] solvent accessible surface area (SA),[14] polar solvent accessible surface area (PSA),[14] solvent accessible volume (SAVOL),[14] and van der Waals volume (VWVOL).[14] Here, we assume that molecular surfaces and volumes calculated based on a single 3D conformation could still be used as an approximation for filtering purposes by a library designer in the initial stages of his or her library design work.

To test the validity of the "direct reactants" Equation 6 and the "basis products" Equation 11, we computed the property values for 9,702 product molecules according to Equations 6 and 11, and then compared them against the exact values calculated directly based on whole product molecules. For each exact molecular property, we fitted the corresponding approximate values from the "direct reactants" Equation 6 and the "basis products" Equation 11 to a linear regression equation [Property(exact) = k * Property(approximate) + c, where k is the linear scaling coefficient and c the constant shift]. Their corresponding correlation coefficients R, root mean square deviations (RMS), and means of absolute deviations also are calculated to characterize the qualities of those linear fittings.

RESULT AND DISCUSSION

Scatter plots between exact property values (as y-axis) directly calculated based on whole product molecules and their corresponding fast approximation values (as x-axis) from the "direct reactants" Equation 6 and the "basis products" Equation 11 are presented in Figures 8–10 and Figures 12–15. All parameters obtained from the least square linear fittings are also summarized in Table 2.

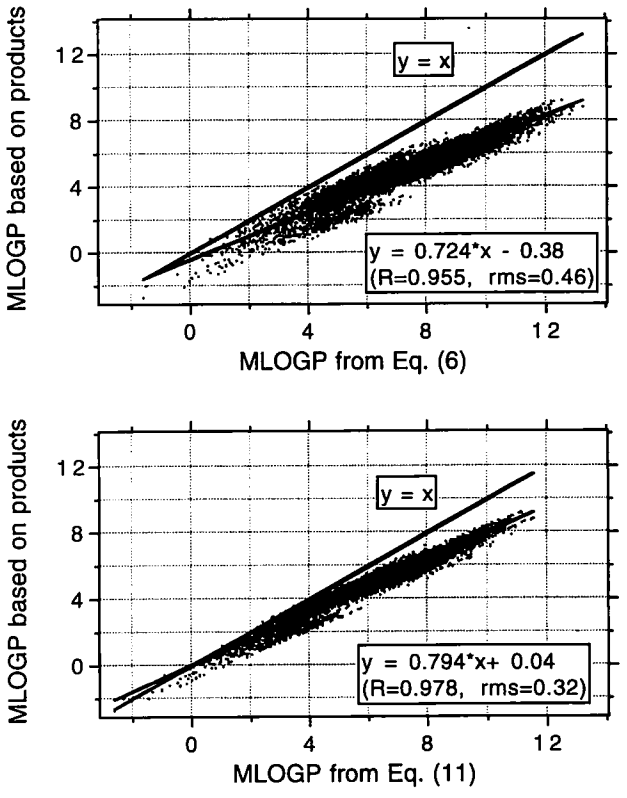


Figure 8. Scatter plots of exact MLOGP values vs the fast approximation from the "direct reactants" Equation 6 and the "basis products" Equation 11 for all 9,702 product molecules generated by the 21 × 21 × 22 test library. The y-axis is the exact property calculated based on product structures. The x-axis is the approximate one either from Equation 6 or Equation 11. The dots represent the data points. The y = x line represents the ideal case, and the other line represents the result of a linear fitting. For almost all products, the estimated MLOGP values from Equations 6 and 11 are larger than the exact ones.

MLOGP: The first observation from Figure 8 is that both the "direct reactants" Equation 6 and the "basis products" Equation 11 give systematically larger LogP values when compared with exact values directly calculated from whole product molecules. This phenomenon has been observed and discussed in previous studies.[5,22] Examination of the QSAR equation used by MLOGP reveals that it contains terms that do not scale linearly with molecular size.[20] Therefore, by definition MLOGP is not additive, and straightforward applications of both the "direct reactants" Equation 6 and the "basis products" Equation 11 would give errors too large for any practical design work. The second observation is that the exact product MLOGP correlates linearly, although not with unity slope, with both approximate ones from the "direct reactants" Equation 6 and the "basis products" Equation 11 with high correlation coefficients (0.955 and 0.978, respectively). One may be tempted to utilize those regression equations to better approximate exact product MLOGP. Still, the RMS errors (0.46 and 0.32 LogP units, respectively) of those linear equations seem too large for practical library design work.

SLOGP: Figure 9 shows that the "direct reactants" Equation 6 gives a very poor estimation for product SLOGP. However, an almost perfect linear fit (k = 1.002, c = −0.03, R = 0.9996, and RMS = 0.06) indicates that the "basis products" Equation 11 is a very good estimation for product SLOGP. This is the desired result, that direct use of the fast approximation, Equation 11, with no adjustment (k = 1), gives suitably accurate product property prediction.

CLOGP: Figure 10 leads to a conclusion for CLOGP similar to that for SLOGP. Equation 6 gives a fairly rough CLOGP estimation, whereas the "basis products" Equation 11 gives a reasonable estimation. The linear regression parameters (k = 1.006, c = −0.03, R = 0.978) reconfirm the observation that the "basis products" Equation 11 provides a very good CLOGP estimation for most product molecules.

However, there is a set of product molecules for which the "basis products" Equation 11 gives approximate values about 2 LogP units higher than product CLOGP values. Analysis revealed that this set consists of zwitterionic product molecules formed by the reaction. In the example given in Figure 11, a zwitterionic product molecule contains one positively ionizable amine group from the component A_2 and one negatively ionizable COOH group from the component A_1. These two ionizable functional groups are considered to be neutral when CLOGP values of their corresponding "basis products," P_1^*

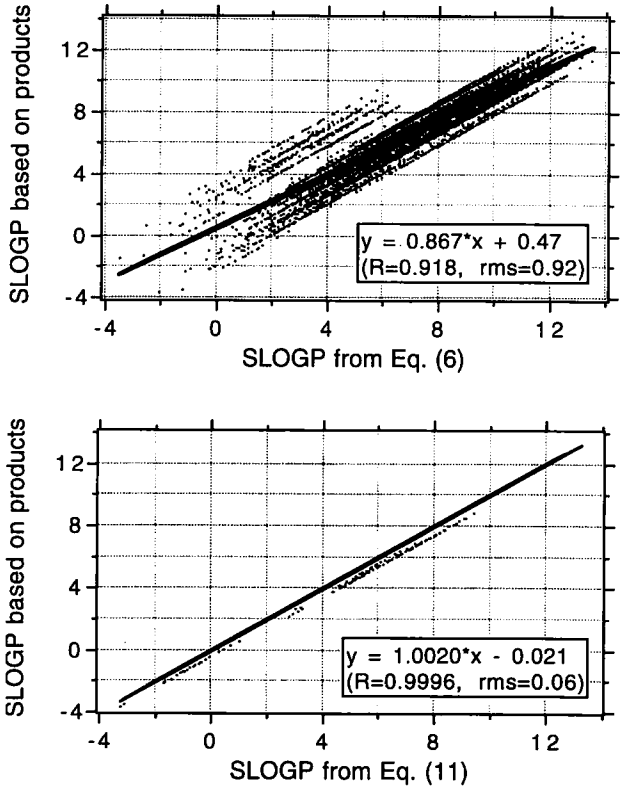


Figure 9. Scatter plots of exact SLOGP values vs the fast approximations from the "direct reactants" Equation 6 and the "basis products" Equation 11 for all 9,702 product molecules generated by the 21 × 21 × 22 test library. See Figure 8 caption for details. Equation 6 leads to large estimation error, whereas Equation 11 gives a good estimation of product SLOGP values.

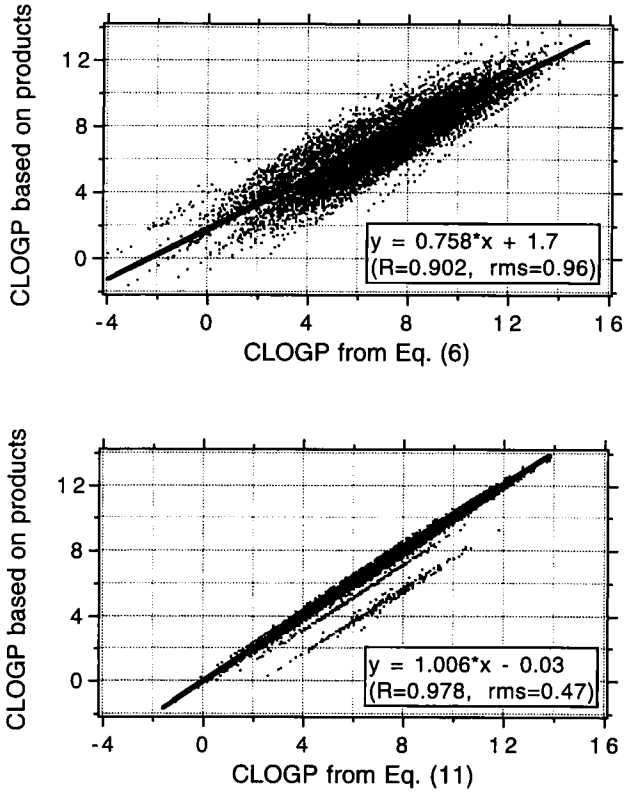


Figure 10. Scatter plots of exact CLOGP values vs the fast approximations from the "direct reactants" Equation 6 and the "basis products" Equation 11 for all 9,702 product molecules generated by the 21 × 21 × 22 test library. See Figure 8 caption for details. The "direct reactants" Equation 6 leads to large estimation error, whereas the "basis products" Equation 11 gives a good estimation of product CLOGP values for most product molecules. The data points with large errors of 2 LogP unit are all from zwitterionic product molecules formed by the reaction. See text and Figure 11 for further discussion.

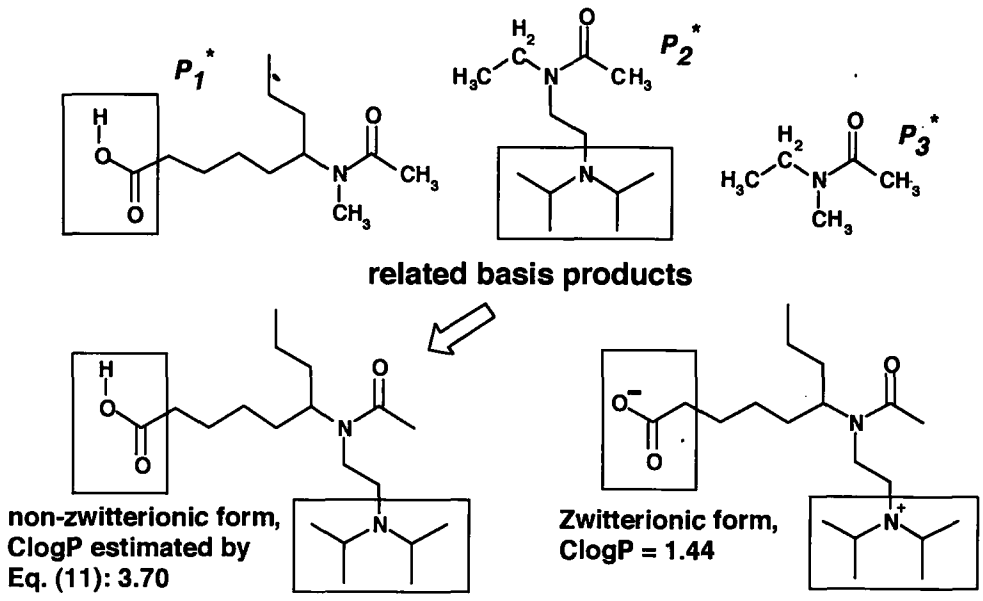


Figure 11. The "basis products" Equation 11 overestimates CLOGP value of a potentially zwitterionic product when compared to the value calculated directly from the zwitterionic product structure.

and P_2^*, are calculated, according to the overall neutrality in the definition of LogP. However, they are both considered ionized (in a zwitterionic form) when the CLOGP value of the whole product molecule is calculated. That is, in CLOGP calculation, the same fragment (amine group or COOH group) is treated differently depending on whether it is isolated or occurs in combination with an oppositely ionizable group. As a result, for zwitterionic product molecules formed by a reaction, the "basis products" Equation 11 is not a good approximation for CLOGP. For comparison, we also checked the SLOGP calculations for the same set of zwitterionic product molecules. By default, SLOGP does not treat them as zwitterions.[23] For the example product molecule in Figure 11, both the exact calculation and approximation by the "basis products" Equation 11 give the same SLOGP value of 3.6.

Solvent Accessible Surface Area: Figure 12 shows that both the "direct reactants" Equation 6 and the "basis products" Equation 11 overestimate SA by about 200 and 100 Å^2, respectively. Because SA does sensitively depend on molecular conformations, it is not surprising that the "direct reactants" Equation 6 and the "basis products" Equation 11 are approximations with relatively larger imprecision. However, given that different conformations of a product intrinsically lead to a range of product SA for a single structure, this level of accuracy still may be adequate for filtering down millions or billions of virtual products. To be cautious in the application of this filtering method, one could adjust the filter criteria w_1 and w_2 to compensate for the systematic errors in SA approximations by Equations 6 and 11.

Polar Solvent Accessible Surface Area: PSA is the polar part of the solvent accessible surface. In the literature, the areas of polar molecular surfaces, such as polar solvent accessible surface and polar van der Waals surface, have been used to predict drug oral bioavailability and/or cell permeability.[24] Figure 13 shows that, unlike the exact PSA values that are always positive, the estimated ones from the "direct reactants" Equation 6 and the "basis products" Equation 11 can be negative, an artifact of the methods. Even though the estimated PSA from Equation 6 or 11 correlates well with the exact PSA (R = 0.969

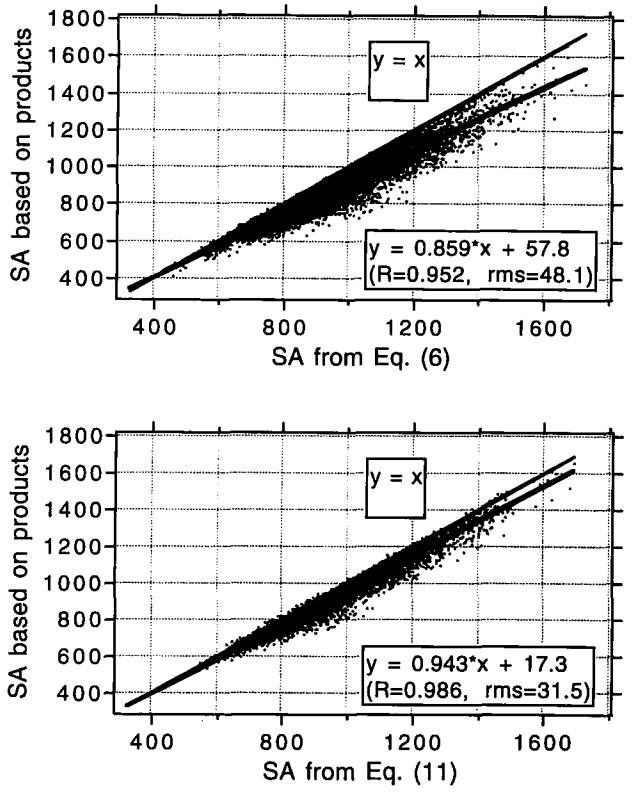
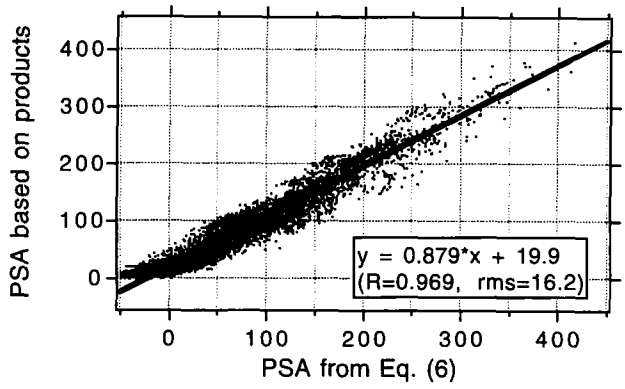


Figure 12. Scatter plots of exact solvent accessible surface area (SA) values vs the fast approximations from the "direct reactants" Equation 6 and the "basis products" Equation 11 for all 9,702 product molecules generated by the 21 × 21 × 22 test library. See Figure 8 caption for details. Equations 6 and 11 overestimate SA systematically when compared to exact values. Even though both estimations are nicely correlated with the exact SA, errors from Equations 6 and 11 are significant for applications requiring precision and accuracy.



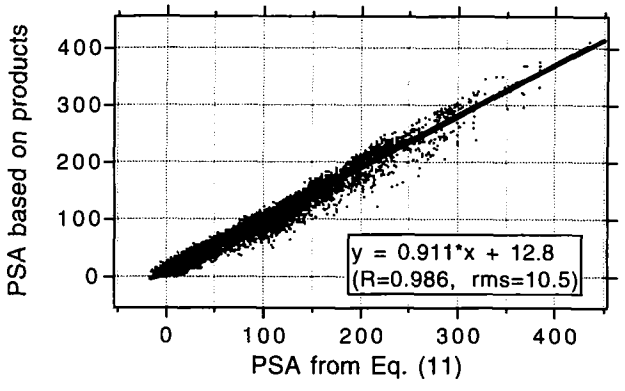
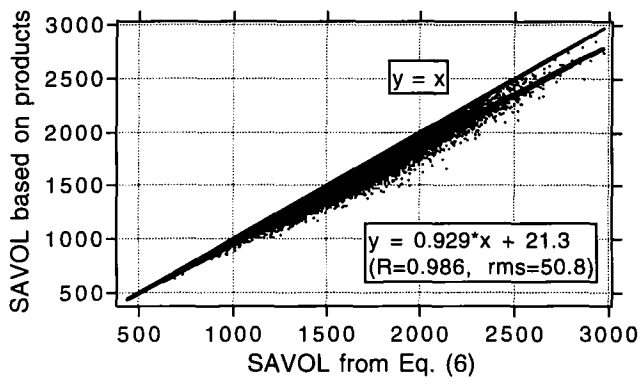


Figure 13. Scatter plots of exact polar solvent accessible surface (PSA) values vs the fast approximation from the "direct reactants" Equation 6 and the "basis products" Equation 11 for all 9,702 product molecules generated by the 21 × 21 × 22 test library. See Figure 8 caption for details. Like SA, PSA estimations based on Equations 6 and 11 are nicely correlated with the exact values but with larger relative errors. They can still be used for applications that do not demand high accuracy and precision.



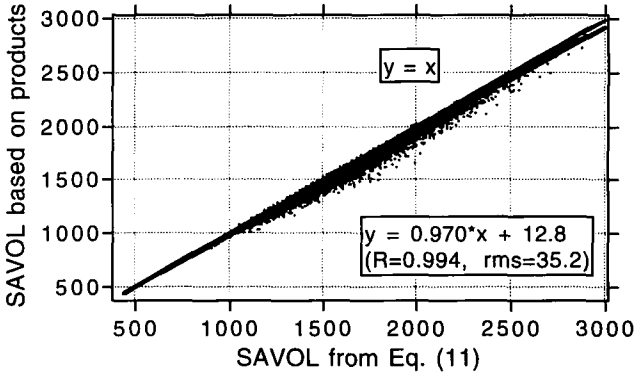


Figure 14. Scatter plots of exact solvent accessible volume (SAVOL) values vs the fast approximations from the "direct reactants" Equation 6 and the "basis products" Equation 11 for all 9,702 product molecules generated by the 21 × 21 × 22 test library. See Figure 8 caption for details. The "basis products" Equation 11 provides a good estimation for product SAVOL values.

and 0.986, respectively), the spread of the data points are quite large with respect to the exact PSA values (RMS = 16.2 and 10.5 respectively). Nevertheless, Equations 6 and 11 are able to identify compounds with low, medium, or high PSA values. For example, let us assume that one wants to classify compounds with PSA < 100 Å^2 as Low and other compounds as High. If one uses the estimated PSA from Equation 11 to do the classification and compares the result with the one generated by using exact PSA, 9,269 products are correctly classified and only 4% of 9,702 compounds are misclassified.

Solvent Accessible Volume: Figure 14 shows that both the "direct reactants" Equation 6 and the "basis products" Equation 11 systematically overestimate SAVOL. Like SA, SAVOL depends on molecular conformation. The good fitting results for the "basis products" Equation 11 suggest that Equation 11 may be used to approximate product SAVOL with a reasonably small error.

van der Waals Volume: Both the "direct reactants" Equation 6 and the "basis products" Equation 11 overestimate VWVOL only slightly (Figure 15). This means that VWVOL is nearly additive in nature and one can use either approximation method to estimate product VWVOL. This observed additivity could be attributed to the fact that van der Waals spheres infrequently penetrate each other when they are in a 1-4 relationship or farther apart in reasonable low-energy molecular conformations. The exceptions are at a few sites of intramolecular hydrogen bonding where atomic van der Waals spheres do penetrate each other. The contribution from 1-2 (bond) and 1-3 (angle) related van der Waals sphere overlaps vary little because bond lengths and angles are nearly constant for most molecular conformations.

Our test results of a limited scope have shown that there are molecular properties (such as MLOGP) that could not be approximated with either the "direct reactants" Equation 6 or the "basis products" Equation 11. However, other molecular properties (such as VWVOL, SLOGP, CLOGP, SAVOL, SA, and PSA) of products can be suitably approximated by a carefully designed additivity without fitting. For these molecular properties, as indicated in the Methodology section, the "basis products" Equation 11 represents a much better approximation than the "direct reactants" Equation 6 (Table 2). Those "closely additive" properties can be used by our efficient filtering method over a relatively broad application domain.

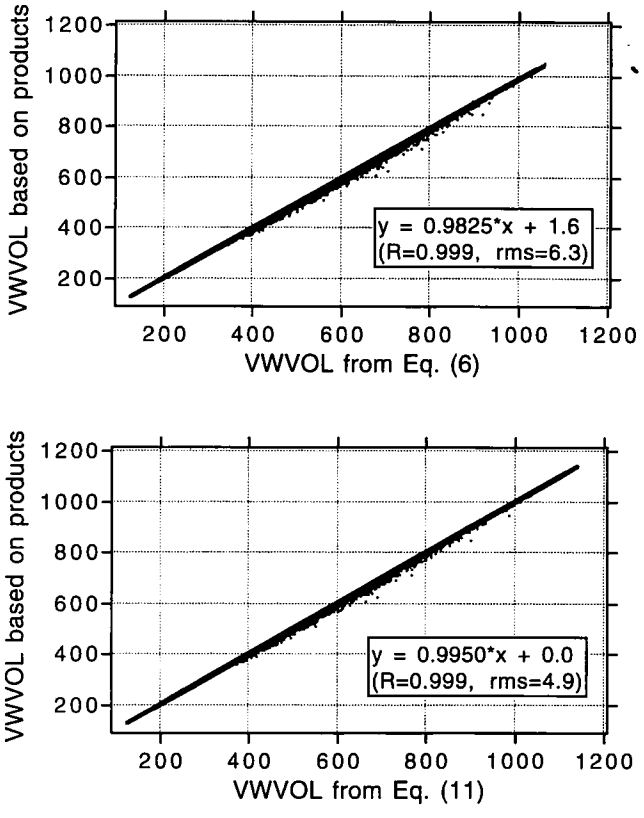


Figure 15. Scatter plots of exact van der waals volume (VWVOL) values vs the fast approximation froms the "direct reactants" Equation 6 and the "basis products" Equation 11 for all 9,702 product molecules generated by the 21 × 21 × 22 test library. See Figure 8 caption for details. Both approximations work well for VWVOL. This means that VWVOL is an excellent closely additive molecular property.

Virtual Product Mining for Desired Molecular Properties

For the first set of calculations, we again use the three-component, two-step reaction depicted in Figure 2 as an example. However, in this study, the search domain spans the whole virtual product space (~1.5 billion virtual products) of this reaction. The properties considered are molecular weight, number of hydrogen bond acceptors, number of hydrogen bond donors, and SLOGP. The compounds MFCD00006991, MFCD00008104, and MFCD00000719 (shown in Figure 5) are selected as the particular set of the reaction components $\mathbf{A}^*$. The molecular properties of all the (3,023 + 1,132 + 430 = 4,585) reactants and the corresponding 4,585 "basis products" with the particular set $\mathbf{A}^*$ are calculated first. The reaction-dependent constants $\Delta\mathbf{Q}$(rxn) and the properties $\mathbf{Q}(P^*)$ for the "capped core," the particular product formed from the particular set, MFCD00006991, MFCD00008104, and MFCD00000719, are listed in Table 3.

Five filtering calculations are carried out for this reaction. Test calculation 1 uses the "direct reactants" Equations 18 and 19. The "basis products" Equations 22 and 23 are used for test calculations 2, 3, 4, and 5. As shown in Table 4, a relatively broad range of molecular weight and SLOGP values are used for test calculations 1 and 2, and a narrow range of property values are selected for the filtering criteria in test calculation 3 and 4. For test calculation 5, enlarged filter windows are used to account for the binning error.

In the first three calculations, the bin sizes for molecular weight and SLOGP are 5 amu and 0.2, respectively. Therefore, the error bars due to the binning are 20 amu for molecular weight and 0.8 for SLOGP in test calculation 1; and 15 amu for molecular weight and 0.6 for SLOGP in test calculations 2, 3, and 5. In test calculation 4, smaller bin sizes are applied: 1 amu for molecular weight and 0.05 for SLOGP. As a result, the binning errors are reduced: 3 amu for molecular weight and 0.15 for SLOGP.

All five calculations are done on an SGI O2 machine with a MIPS R10000 processor. The user CPU time is <1 second (for the first three calculations) and 4.3 seconds (for test calculation 4) for the task of selecting ~5,000 reaction products from a pool of ~1,500,000,000. For comparison, the user CPU time for calculating the molecular properties of the ~5,000 reaction products alone is ~1,800 seconds. (Note that this 1,800 seconds was only necessary to perform the validation.)

From filter test calculations 1 and 2, 4,153 and 6,755 products are selected, respectively. The third and fourth test calculations give rise to 6,658 and 2,578 products, respectively. The properties of the products from these four filter calculations were then computed directly. The product property calculations show that all the products selected from these four filter calculations have four hydrogen bond acceptors and four hydrogen bond donors as specified by the filtering requirements. The distributions of the molecular weights for the four sets of products are shown in Figure 16. It is seen that all the compounds have molecular weights within the bounds specified in the filtering criteria if the respective errors due to binning have been taken into account.

The SLOGP values of the four sets of products are presented in Figure 17 as distribution plots. There are no errors in the filtered products from calculation 2, and only 8 out of 6,658 from calculation 3 have SLOGP values outside the bounds given in the filtering criteria (considering the binning error bar 0.6). There are many SLOGP values of the products from test calculation 1 that fall outside of the filtering window. This is another indication that the "direct reactants" Equation 6 is not a good approximation for SLOGP, while Equation 11 is performing well.

Test calculation 4, with smaller bin sizes, reveals that all the 324 false positives for molecular weight are within the error bar due to the binning. Although for SLOGP there are 185 false positives, 26 of these virtual products are outside the range of the binning error bar and thus are due to the error of the "basis products" Equation 11. Given that test calculations 3 and 4 have the same filtering criteria, it is interesting that 1,223 out

Table 3. Property values for $\Delta\mathbf{Q}$(rxn) and $\mathbf{Q}(P^*)$ for the three-component reaction depicted in Figure 2

	MW	N_HBA	N_HBD	SLOGP
$\Delta\mathbf{Q}$	−52.47	−1	−1	1.31
$\mathbf{Q}(P^*)$	101.15	1	0	0.01

Table 4. Filtering criteria used for various test calculations for the three-component reaction depicted in Figure 2

Test case	MW			N_HBA		N_HBD		SLOGP			Approx. model used
	w_1	w_2	Bin size	w_1	w_2	w_1	w_2	w_1	w_2	Bin size	
1	0	350	5	4	4	4	4	2.5	4.5	0.2	Equation 6
2	0	385	5	4	4	4	4	2.5	4.5	0.2	Equation 11
3	495	500	5	4	4	4	4	3.5	4.0	0.2	Equation 11
4	495	500	1	4	4	4	4	3.5	4.0	0.05	Equation 11
5	480	505	5	4	4	4	4	2.9	4.2	0.2	Equation 11

of 2,578 virtual products selected by filter calculation 4 are not in the set of 6,658 virtual products picked by calculation 3. To see how one may eliminate these false negatives in calculation 3, broader filter windows (Table 4) are used in test calculation 5. Calculation 5 renders 37,311 virtual products, which include all the 2,578 virtual products obtained in calculation 4. As pointed out in the section on An Efficient Tree Algorithm for Filtering Discrete or Continuous Properties of Products, the price paid for eliminating these false negatives is that a significantly larger set of virtual products (37,311) is selected in calculation 5.

To explore further the false-negative issue, a one-step, two-

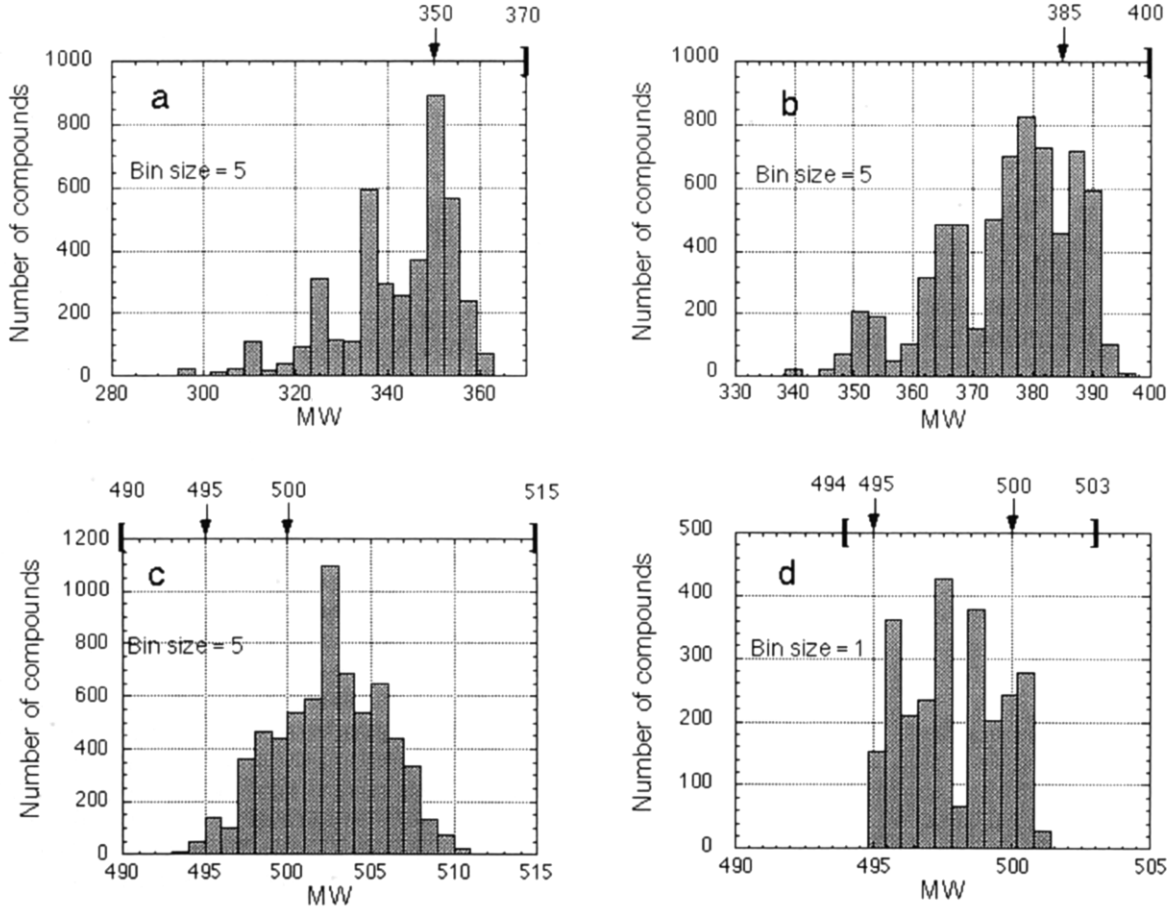


Figure 16. Distributions of molecular weight (MW) for filtered product molecules out of a 1.5 billion virtual product pool. The upper and lower bounds (except for the MW=O bound) used in the filtering calculations are indicated by the arrows. The error bars due to binning are marked by the square brackets. (a) MW distribution for 4,153 filtered products from test calculation 1 using the "direct reactants" Equations 18 and 19. (b) MW distribution for 6,755 filtered products from test calculation 2 using the "basis products" Equations 22 and 23. (c) MW distribution for 6,658 filtered products from test calculation 3 using the "basis products" Equations 22 and 23. (d) MW distribution for 2,578 filtered products from test calculation 4 using the "basis products" Equations 22 and 23. Comparing plots c and d, we see that by reducing the bin size (from 5 to 1 amu), the number of false positives is significantly reduced.

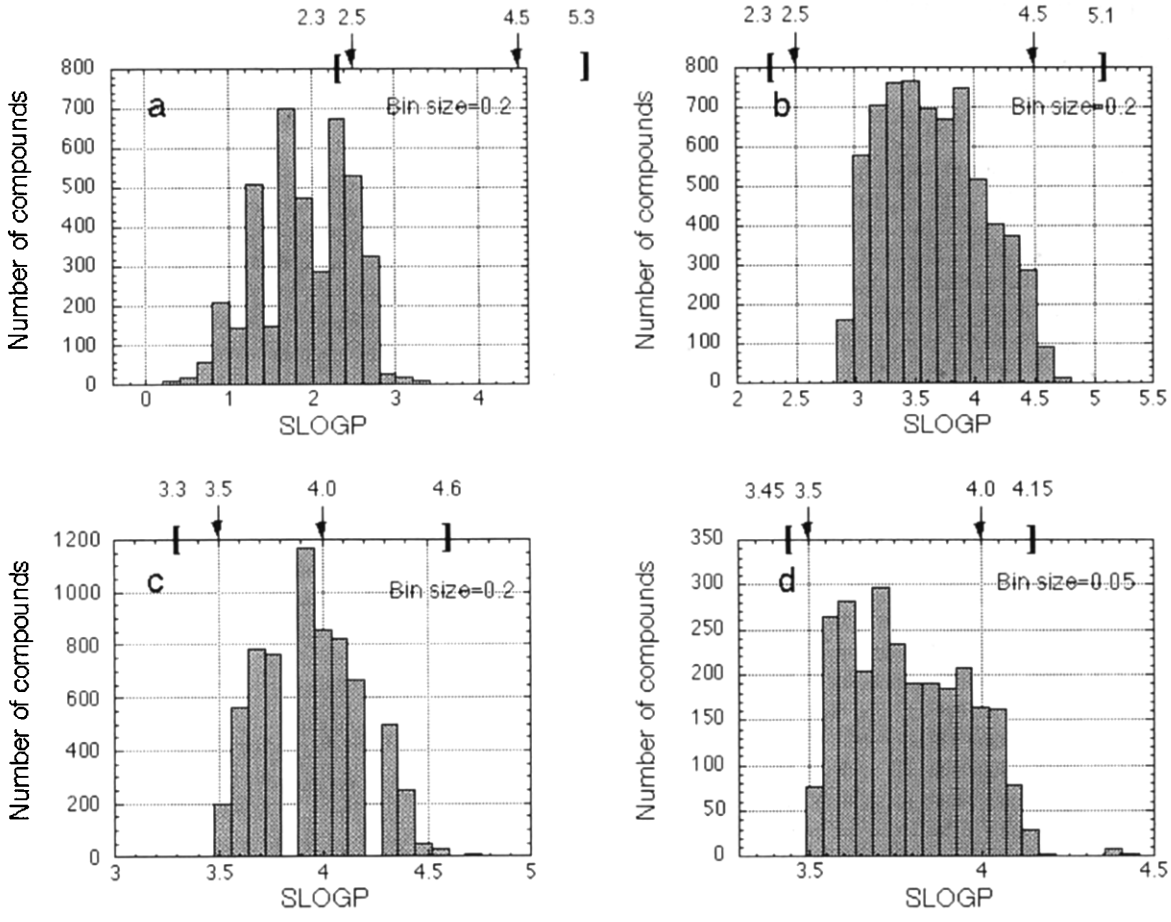


Figure 17. Distributions of SLOGP for filtered product molecules out of a 1.5 billion virtual product pool. The upper and lower bounds used in the filtering calculations are indicated by the arrows. The error bars due to binning are marked by the square brackets. (a) Distribution of SLOGP for 4,153 filtered products from test calculation 1 using the "direct reactants" Equations 18 and 19 with a filter window [2.5, 4.5] and bin size of 0.2. Thus, the error bar due to binning is 4 × 0.2 = 0.8. There are many filtered products with SLOGP outside the window [2.5 − 0.2, 4.5 + 0.8] or [2.3, 5.3]. This indicates that Equation 6 is not a good approximation for product SLOGP. (b) Distribution of SLOGP for 6,755 filtered products from test calculation 2 using the "basis products" Equations 22 and 23 with a filter window [2.5, 4.5] and bin size of 0.2. Thus, the error bar due to binning is 3 × 0.2 = 0.6. There are no filtered products outside of the window [2.5 − 0.2, 4.5 + 0.6] or [2.3, 5.1]. (c) Distribution of SLOGP for 6,658 filtered products from test calculation 3 using the "basis products" Equations 22 and 23 with a filtering window [3.5, 4.0] and bin size of 0.2. The error bar due to binning is 3 × 0.2 = 0.6. There are as few as 8 (out 6,658) filtered products with SLOGP outside the window [3.5 − 0.2, 4.0 + 0.6] or [3.3, 4.6], which is due to the error of Equation 11. (d) Distribution of SLOGP for 2,578 filtered products from test calculation 4 using the "basis products" Equations 22 and 23 with bin size of 0.05. The error bar due to the binning is 3 × 0.05 = 0.15. There is only 1% (26 out of 2,578) of filtered product with SLOGP outside the window [3.5 − 0.05, 4.0 + 0.15] or [3.45, 4.15], which is due to the error of Equation 11.

component reaction, reductive amination, schematically shown in the upper part of Figure 2, is used. Five hundred reactants for each of the two-reaction components, A and B, are randomly selected. One thousand "basis products" and 250,000 virtual products are formed. Their molecular properties are calculated. As shown in Table 5, the same filtering criteria are used in both the direct filtering of virtual products (calculation 6) and the filtering calculation by the "basis products" Equations 22 and 23 (calculation 7). For the latter, bin sizes of 1 amu for molecular weight and 0.05 for SLOGP are chosen. The results are presented in Figure 18. The direct filtering of virtual products gives the first set of 1,125 products. The filtering by the "basis products" Equations 22 and 23 selects the second set of 1,295 products. The overlap of these two sets of virtual products contains 1,116 products, that is, the filtering by the "basis products" Equations 23 and 24 picked 99% of the virtual products selected by direct filtering. There are 179 false positives generated by the "basis products" Equations 22 and 23. The nine false negatives that exhibit the desired properties are missing from the second set selected by using the "basis products" Equations 22 and 23. This is 0.8% of the desired target products. By broadening the filter windows, as shown in Table 5, filter calculation 8 gives rise to the third set of 1,717 virtual products. The virtual products that are in the first set,

Table 5. Property values of Q(*P) and filtering criteria for a two-component reaction (the reductive amination step in Figure 2)**

	MW				N_HBA			N_HBD			SLOGP				
Test case	Q(*P**)	w_1	w_2	Bin size	Q(*P**)	w_1	w_2	Q(*P**)	w_1	w_2	Q(*P**)	w_1	w_2	Bin size	Method of filtering
6		310	350			2	2		2	2		3.5	4.0		Directly on product properties
7	60.12	310	350	1	0	2	2	1	2	2	0.31	3.5	4.0	0.05	Equation 11
8	60.12	308	351	1	0	2	2	1	2	2	0.31	3.4	4.05	0.05	Equation 11

selected by exact direct product filtering, all are in the third set, that is, there are no false negatives. However, the false positives, a total of 592, are significantly increased in comparison with 179 included in the second set.

The filtering calculations show that both filtering methods, the "direct reactants" Equations 18 and 19 and the "basis products" Equations 22 and 23, are extremely efficient and are reliable for molecular weight, number of hydrogen bond donors, or hydrogen bond acceptors. For SLOGP, although the "direct reactants" Equations 18 and 19 are no longer reliable, the "basis products" Equations 22 and 23 still give rise to excellent results. For real-valued properties, false negatives as well as false positives may result from the binning. Broadening the filter windows can eliminate false negatives but increase false positives. Decreasing the bin sizes will reduce both the false negatives and false positives.

REMARKS

In the present work, two methods are presented for efficient combinatorial calculation of, and filtering by, molecular properties of virtual products from known combinatorial reactions without explicitly forming the product structures. As indicated in Table 1, the first method, based on the "direct reactants" Equations 6 and 7 (or 8), is accurate for strictly additive properties such as molecular weight, number of hydrogen bond donors, etc. (for details see Table 1). It also performs with good accuracy for those molecular properties that are closely additive (e.g., VDWVOL) in the sense that the fragments contribute additively to the property and have acceptably small influence on each other's contributions. This method uses only the molecular properties of reactants that are *reaction independent.* For each molecular property, there is a reaction-dependent parameter ΔQ(rxn) that needs to be calculated for each of the known reactions.

Equation 11, the second method, utilizes the "basis products," and is not only accurate for the strictly additive and closely additive molecular properties, but is also applicable to certain molecular properties that are calculated empirically with additive fragment contributions, such as SLOGP and CLOGP. This method requires the determination of the structures of the "basis products," which have a one-to-one correspondence with the reactants in the reaction components, and the calculation of their molecular properties for each of the given reactions. Because all the computed inputs are calculated on complete molecules that are all within the *P* product chemical family, where the nearest neighbor influences occurring across the bonds formed in the reaction are already fully incorporated, the "basis products" Equation 11 represents a much better approximation.

The results presented in section on Combinatorial Property Calculation of Reaction Products suggest that the accuracy of combinatorial filtering for desired molecular properties of reaction products may be improved by using regression equations such as those shown in Figures 8–10

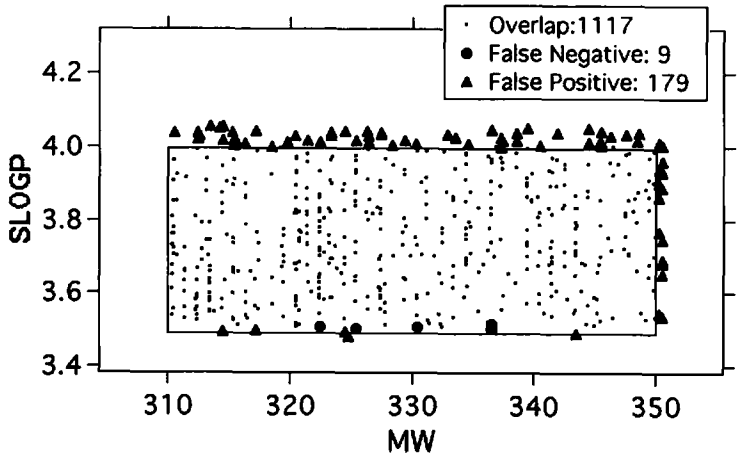


Figure 18. Comparison of the combinatorial filtering method (calculation 7) and exact product filtering by direct calculation of product properties (calculation 6) on a domain of 250,000 virtual products from a two-component reaction. The same filtering criteria listed in Table 5 are used for the two calculations. The direct product filtering (calculation 6) yields 1,125 products, whereas the combinatorial filtering renders 1,295 products. There are 1,116 overlaps. The combinatorial filtering fails to identify nine false negatives that are selected by direct product filtering. There are 179 false positives. All the false negatives and false positives are within the error bars due to the binning.

and 12–15 in addition to the "direct reactants" Equation 6 or the "basis products" Equation 11. Because the primary concerns of the present work are the efficiency and adequate accuracy for the desired filtering, we emphasized the performance with the scaling coefficient k = 1 (no fitting) and have not pursued the issue of improving the accuracy by fitting.

The combinatorial hierarchical filtering procedure and the multiproperty tree-sorting algorithm described in the present work serve to filter efficiently for the reaction products with desired molecular properties. Therefore, it can be useful in the acceleration of product-based targeted library design in structure-based or pharmacophore-directed libraries. For example, by filtering with specified requirements on molecular weight, number of hydrogen bond donors, and number of hydrogen bond acceptors, SLOGP (or CLOGP), and SA or PSA, the number of virtual products can be substantially reduced (e.g., from 1.5×10^9 to 5×10^3) so that computationally more intensive modeling tools such as docking can be applied to further enrich the selection.

ACKNOWLEDGMENTS

The authors are indebted to Dr. Zhongxiang Zhou for generating the "basis products" and calculating their molecular properties; to Drs. Peter Rose and Djamal Bouzida for providing programs used in the calculation of molecular properties; and to Phil Deak for helping in manuscript preparation.

REFERENCES

1 Van Drie, J.H., and Lajiness, M.S. Approaches to virtual Library design. *Drug Discovery Today* 1998, **3,** 274–283 (see also the references therein)
2 Jamois, E.A., Hassan, M., and Waldman, M. Evaluation of reagent-based and product-based strategies in the design of combinatorial library subsets. *J. Chem. Inf. Comput. Sci.* 2000, **40,** 63–70
3 Cramer, R.D., Patterson, D.E., Clark, R.D., Soltanshahi, F., and Lawless, M.S. Virtual compound libraries: A new approach to decision making in molecular discovery research. *J. Chem. Inf. Comput. Sci.* 1998, **38,** 1010–1023
4 Martin, E., and Critchlow, R. Beyond mere diversity: Tailoring combinatorial libraries for drug discovery. *J. Comb. Chem.* 1999, **1,** 32–45
5 Lipinski, C.A., Lombardo, F., Dominy, B.W., and Feeney, P.J. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug Delivery Rev.* 1997, **23,** 3–25
6 Verkhivker, G.M., Rejto, P.A., Gelhaar, D.K., and Freer, S.T. Exploring the energy landscapes of molecular recognition by a genetic algorithm: Analysis of the requirements for robust docking of HIV-1 protease and FKBP-12 complexes. *Proteins,* 1996, **25,** 342–353; Walters, P.W., Stahl T.M., and Murcko M.A. Virtual screening: An overview. *Drug Discovery Today* 1998, **3,**160–178
7 Brown, R., and Martin, Y.C. Designing combinatorial library mixtures using a genetic algorithm. *J. Med. Chem.* 1997, **40,** 2304–2313
8 Gillet, V.J., Willett, P., and Bradshaw, J. The effectiveness of reactant pools for generating structurally-diverse combinatorial libraries. *J. Chem. Inf. Comput. Sci.* 1997, **37,** 731–740
9 Sheridan, R.P., and Kearsley, S.K. Using a genetic algorithm to suggest combinatorial libraries. *J. Chem. Inf. Comput. Sci.* 1995, **35,** 310–320
10 Zheng, W., Cho, S.J., and Tropsha, A. Rational combinatorial library design. 1. Focus-2D: A new approach to the design of targeted combinatorial libraries. *J. Chem. Inf. Comput. Sci.* 1998, **38,** 251–258
11 Pickett, S.D., McLay, I.M., and Clark, D.E. Enhancing the hit-to-lead properties of lead optimization libraries. *J. Chem. Inf. Comput. Sci.* 2000, **40,** 263–272
12 Leland, B.A., Christie, B.D., Nourse, J.G., Grier, D.L., Carhart, R.E., Maffett, T., Welford, S.M., and Smith, D.H. Managing the combinatorial explosion. *J. Chem. Inf. Comput. Sci.* 1997, **37,** 62–70
13 Available Chemicals Directory, version 99. 1, Molecular Design Limited, San Leandro, CA 94577
14 Proprietary programs, LCALC, of Agouron Pharmaceuticals, Inc.
15 Meylan, W.M., and Howard, P.H. Atom/fragment contribution method for estimating octanol-water partition coefficients. *J. Pharm. Sci.* 1995, **84,** 83–92
16 The Basis Products approach and its application in the design of combinatorial libraries, to be published
17 Pan-reaction Combinatorial Filtering for efficient library design across broad virtual libraries, to be published
18 Jensen, K.J., Alsina, J., Songster, M.F., Vágner, J., Albericio, F., and Barany, G. Backbone amide linler (BAL) strategy for solid-phase synthesis of c-terminal-nodified abd cyclic peptides. *J. Am. Chem. Soc.* 1998, **120,** 5441–5452
19 Gasteiger, J., Sadowski, J., Schuur, J., Selzer, P., Steinhauer, S., and Steinhauer, V. Chemical information in 3D. *J. Chem. Inf. Comput. Sci.* 1996, **36,** 1030–1037
20 Moriguchi, I., Hirono, S., Liu, Q., Izumi, N., and Matsushita, Y. Simple method of calculating octanol/water partition coefficient. *Chem. Pharm. Bull.* 1992, **40,** 127–130; Moriguchi, I., Hirono, S., Izumi N., and Hirano, H. Comparison of reliability of LogP values for drugs calculated by several methods. *Chem. Pharm. Bull.* 1994, **42,** 976–978
21 CLOGP v 4.0 for Unix, BioByte Corp., 201 W. Fourth Street, Suite 204, Claremont, CA 91711; see Leo, A.J. Calculating Log P_{oct} from structures. *Chem. Rev.* 1993, **93,** 1281–1306 and references cited within for details
22 Leo, A.J. Critique of recent comparison of log P calculation methods. *Chem. Pharm. Bull.* 1995, **43,** 512–513
23 William Meylan, Personal Communication
24 For application of various polar molecular surfaces to drug absorption, one may consult the following references and references cited by them. Palm, P., Stenberg, P., Luthman, K., and Artursson, P. Polar molecular surface properties predict the intestinal absorption of drugs in humans. *Pharm. Res.* 1997, **14,** 568–571; Krarup, L.H., Christensen, I.T., Hovgaard, L., and Frokjaer, S. Predicting drug absorption from molecular surface properties based on molecular dynamics simulations. *Pharm. Res.* 1998, **15,** 972–978; Clark, D.E. Rapid calculation of polar molecular surface area and its application to the prediction of transport phenomena. 1. Prediction of intestinal absorption. *J. Pharm. Sci.* 1999, **88,** 807–814

APPENDIX A

The following is a pseudo C code for implementing Equations 20 and 21 for the molecular property Q with non-negative integer values i_j for the reaction component A_j sorted initially, with j = 1,...,K:

```
For (i1 = 0; i1 < W2 + (K-1)Q(P*) + 1; i1 ++ ){
  For (i2 = 0; i1 < W2 + (K-1)Q(P*) + 1-i1; i2 ++ ){
    For (i3 = 0; i3 < W2 + (K-1)Q(P*) + 1-i1-i2; i3
++ ){
    . . . . . .
Io = W1 + (K-1)Q(P*) − (io + i1 + . . . + iK-1);
if (Io < 0)
  Io = 0;
For (iK = Io; iK < W2 + (K-1)Q(P*) + 1 − (io + i1 + . . .
                                    + iK-1); iK ++ ){
  Retrieve desired products as Product[i1, i2, i3, . . .iK];
      }
    }
  }
}
```

APPENDIX B: THE BINNING ERRORS

Suppose that the lower and upper bounds of a filter for property Q are w_1 and w_2, respectively, and the bin size is δ. If one takes the integer value of (Q/δ) as the bin value for property Q, then the virtual products with property Q,

$$w_2 < Q < (w_2 + E_u^p) \text{ or } w_1 > Q > (w_1 - E_l^p), \tag{B.1}$$

could be included as false positives. Whereas the virtual products with property Q,

$$w_2 > Q > (w_2 - E_u^n) \text{ or } w_1 < Q < (w_1 + E_l^n), \tag{B.2}$$

could, as false negatives, be excluded. Here, for a K-component reaction, the errors, E_u^p, E_l^p, E_u^n, and E_u^n, due to a bin size of δ are:

$$E_u^p = E_l^n = (K + 1)\,\delta \quad \text{and} \quad E_l^p = E_u^n = \delta, \tag{B.3}$$

for Equations 18 and 19, and

$$E_u^p = E_l^n = K\,\delta \quad \text{and} \quad E_l^p = E_u^n = \delta, \tag{B.4}$$

for Equations 22 and 23.