

# Scalable Methods for the Construction and Analysis of Virtual Combinatorial Libraries

Victor S. Lobanov\* and Dimitris K. Agrafiotis

3-Dimensional Pharmaceuticals, Inc., 665 Stockton Drive, Exton, PA 19341, USA

## ABSTRACT

One can distinguish between two kinds of virtual combinatorial libraries: “viable” and “accessible”. Viable libraries are relatively small in size, are assembled from readily available reagents that have been filtered by the medicinal chemist, and often have a physical counterpart. Conversely, accessible libraries can encompass millions or billions of structures, typically include all possible reagents that are in principle compatible with a particular reaction scheme, and they can never be physically synthesized in their entirety. Although the analysis of viable virtual libraries is relatively straightforward, the handling of large accessible libraries requires methods that scale well with respect to library size. In this work, we present novel, efficient and scalable techniques for the construction, analysis, and *in silico* screening of massive virtual combinatorial libraries.

---

\* Corresponding author. Tel: (610) 458-5264 Ext. 6501, Fax: (610) 458-8249, E-mail: victor@3dp.com.

**INDEX TERMS**

Combinatorial library, combinatorial chemistry, high-throughput screening, compound selection, library design, molecular diversity, molecular similarity, QSAR, nonlinear mapping, multidimensional scaling.

## **I. INTRODUCTION**

Among all the tools available to the medicinal chemist, combinatorial chemistry is one of the most powerful and best suited for exploring chemical space in search of new leads. It provides access to millions of novel compounds from a limited number of building blocks using simple synthetic procedures that work reliably across a wide range of starting materials. Unfortunately, despite the ever-increasing throughput of parallel synthesis and screening technologies, in many cases the number of compounds that are accessible from commercially available reagents is too large to permit their physical synthesis. A common solution to this problem is to “virtualize” the combinatorial libraries and apply appropriate selection techniques in order to identify a smaller subset of compounds for physical synthesis and biological testing. In order to take advantage of robotic hardware, minimize the number of reagents, and simplify the logistical aspects of the experiment, physical libraries are almost invariably synthesized in the form of arrays, which represent the products derived by combining a given subset of reagents in all possible combinations as prescribed by the reaction scheme.

Depending on their use, combinatorial libraries are divided into two main categories: 1) focused or directed libraries which are biased against a specific target, structural class, or known pharmacophore, and 2) exploratory or probe libraries which are target-independent and are designed to span a wide range of physicochemical and structural characteristics. Focused libraries are typically designed to follow up on a known lead, optimize a set of properties, or validate some structure-activity hypothesis. Whatever the case, access to the chemical structures of the products is required in order to assess molecular similarity, predict biological activity, or estimate some other property of interest. In contrast, probe libraries explore chemical space in search of novel hits, and their design is based predominantly on molecular diversity. Although fairly diverse libraries can be built by selecting a diverse set of reagents, product-based designs have been shown to be substantially better [1, 2]. In addition, experience suggests that selections based exclusively on molecular diversity tend to include “extreme” reagents, which can increase cost, cause delays due to limited availability, lead to unforeseen synthetic problems, and produce unusual compounds of limited pharmaceutical interest. Besides, the hit rate achieved with such libraries has proven disappointingly low [3], and the compounds often exhibit unfavorable biological properties that could potentially result in

ADME liabilities [4, 5]. Thus, the focus in the design of probe libraries has gradually begun to shift from pure diversity to chemical feasibility, availability of monomers, and drug likeness [5, 3, 6, 7, 8].

Creating designs that combine molecular diversity or similarity with desired property profiles and drug likeness requires the use of optimization techniques such as simulated annealing [9, 10, 11, 12, 13, 14] or genetic algorithms [15, 16, 17] and, of course, access to the properties of the individual products. To that end, *in silico* enumeration or virtual library generation becomes an essential part of the design process. Despite impressive advances in the processing speed and storage capacity of modern computers, there are many combinatorial libraries that defy enumeration, let alone any form of systematic analysis. For example, it is easy to imagine a combinatorial library containing  $10^{12}$  compounds [18], which would require over three years to enumerate at a rate of 10,000 structures per second. Since most of the descriptors that are typically employed in diversity profiling, similarity searching and QSAR are calculated at a much slower rate, an exhaustive analysis of such a library would be impossible. Thus, there is a pressing need for library construction and analysis techniques that are scalable and can be applied to massive virtual libraries containing hundreds of millions of compounds.

## **II. LIBRARY CONSTRUCTION**

The construction of a virtual library involves three basic steps: reaction encoding, selection of reagents, and enumeration. Enumeration is the process of constructing the connection tables of the products from their respective building blocks as prescribed by the reaction protocol. There are generally two different approaches to this problem. The first is based on the use of a Markush structure, which represents a common scaffold with variation sites labeled as R-groups [19, 20]. In this case, the virtual library is assembled by systematic attachment of clipped reagents to the respective variation sites. Although enumeration is reduced to a simple concatenation of the corresponding connection tables, the lists of clipped reagents have to be carefully constructed from the monomers by removing the parts of the structure that are discarded during the reaction. Unfortunately, this “fragment making” approach cannot be easily applied to reactions that involve modification of the building blocks such as the Diels-Alder reaction [21].

The second approach utilizes a reaction transform to encode the chemical reaction [3, 21]. The transform specifies which parts of the reacting molecules undergo chemical transformations and what is the nature of these transformations. This approach mimics more closely the stages involved in actual synthesis, does not require the presence of a common template and the generation of clipped reagents, and can be applied to a broad spectrum of chemical reactions used in combinatorial chemistry. In order to be general, the encoding process should be able to accommodate multi-component reactions, ring cyclizations, removal of protecting groups, and modification of the core structure. In addition, there should be means to specify multiple products, designate stereochemistry, and differentiate the reactivity of different functional groups depending on their environment if more than one of these groups match the reaction requirements.

Most importantly, if it is to be scalable, the library construction process must avoid the enumeration of every product in the virtual library unless it is specifically requested, and must be able to access any desired structure at a speed that is comparable to file I/O. As mentioned above, the exhaustive enumeration of a library containing  $10^{12}$  members would require over three years of CPU time at a rate of 10,000 compounds per second, and 10 terabytes of a disk space at a modest storage requirement of a 100 bytes per structure [18]. Eliminating the need for storing an explicit record for each individual product is often referred to as “implicit” enumeration.

Taking the aforementioned considerations into account, we have developed a reaction scripting language (RSL) to facilitate the encoding of chemical transformations and the enumeration of virtual libraries that offers versatility, speed, and scalability. RSL is designed as an extension of the Tool Command Language (Tcl) [22] and has a fairly simple and human-readable syntax. Each combinatorial reaction is defined as a named Tcl procedure, thus providing a framework for creating libraries of common reaction schemes. Most importantly, RSL procedures are designed to be compilable into a sequence of parameterized function calls that need to be executed in order to assemble the product structures. The latter is essential for fast, “on-demand” enumeration of combinatorial products.

When a reaction procedure is invoked, it generates a virtual library from lists of reagents supplied in SD or SMILES format. The generated library is stored in a compact form on disk and can be used at a later time for immediate access to any of the product structures. The names of the input reagent and output

library files are passed as arguments to the Tcl reaction procedure. Fig. 1 shows an example of an RSL script based on the reductive amination reaction.

<Fig. 1>

Conceptually, an RSL procedure consists of three blocks: a definition block, an assembly instruction block and an execution trigger. The definition block defines the reagents and the product, and specifies the reactive patterns. The assembly instruction block provides explicit instructions on how to assemble the product molecule from the reagents, and what parts of the source molecules are eliminated in the process. For simplicity and speed, the assembly instructions do not explicitly construct any intermediates that might be formed during the reaction, but rather summarize the transformation of the input reagents directly into the final product. When the reaction script is executed, the assembly instructions are translated into a sequence of parameterized function calls. The assembly sequence is saved within the virtual library, and every time a product needs to be assembled, this sequence is executed with the appropriate parameters. Lastly, the execution trigger is a statement that triggers the mapping of the reactive patterns onto the supplied sets of reagents, the compilation of the assembly instructions, and the storage of the virtual library into a file. Thus, our virtual library consists of structures of input reagents in their original form, maps of every substructure pattern onto all of the input reagents, and a sequence of compiled assembly instructions that should be executed in order to build the connection table of a product.

RSL uses the SMARTS notation [23] to encode the substructural patterns that are involved in the reaction and must be present in order for the reagent to undergo the reaction. Each reagent can be defined using multiple patterns, and the order in which they are defined specifies the relative reactivity of the respective functional groups. For example, in the amination library in Fig. 1, lines 5 and 6 specify that both primary and secondary amines can react with an aldehyde. However, if both a primary and a secondary amine are present in the same molecule, the main product will be formed from the more reactive primary amine. By defining the SMARTS pattern corresponding to the primary amine before the secondary amine, we can ensure that the proper products will be assembled. Although it is possible to match both primary

and secondary amines with a single SMARTS pattern, it would not be possible to differentiate their reactivity.

Sometimes it is very difficult, if not impossible, to write a single SMARTS string that will match only the reactive substructure pattern. In this case, one can specify the substructures that are not reactive. For example, a simple amine pattern "C[NH2]" will match an amide as well, which is not susceptible to reductive amination. One can either modify the amine pattern as "[CX4][NH2]" or define the amide substructure "C(=O)[NH2]" and designate it as non-reactive. When non-reactive patterns are present, the program looks for an overlap between the matched reactive and non-reactive substructures, and if they have at least one atom in common the reactive structure will be invalidated.

After the reacting substructures of the reagents are defined, the remaining code of the reaction script encodes the instructions for product assembly. Once the product is defined (line 5 in Fig. 1), its name becomes a Tcl command that supports a series of molecular operations. These operations include addition of previously defined reagents, removal of atoms, addition and removal of bonds, changing of the bond order, etc. A list of the most frequently used operators is given in Fig. 2. Note that with the exception of the "add" command, which instructs the program to add the connection table of the reagent to the connection table of the product, the remaining assembly instructions require specification of individual atoms affected by the instructions. Individual atoms participating in the chemical transformation are referred to by the respective reagent's name and by the zero-based indices of the matching atoms in the respective SMARTS pattern. For example, the nitrogen atom from an "amine" reagent defined with the SMARTS pattern "C[NH2]" will be referred to as "amine:1". Since SMARTS strings are written in a single line, all atom specifications defined in a pattern can be unambiguously numbered from left to right as they appear in the pattern string. Note that hydrogen atoms are part of the atom specification in SMARTS and therefore cannot be individually addressed.

<Fig. 2>

Since the reagent definition can include multiple SMARTS patterns, it is important that the atoms referenced in the assembly instructions have the same indices in every pattern. For instance, the nitrogen

atom in both the primary and secondary amines should have the same index (e.g. 1) if a single assembly instruction is to apply to both of them. Fortunately, it is always possible to write SMARTS specifications in the desired order using “ring closures” (numbers), disconnections (dots) and recursive atom environments. Moreover, in most cases, SMARTS encoding lends itself naturally to this requirement since multiple patterns are typically used to define variations of the same functionality, such as primary and secondary amines.

Most of the assembly instructions are obvious: **insert bond**, **remove atom**, **remove bond**, **set atom charge**, **set bond order**, etc. Note that there is no instruction to insert a single atom since all the atoms of a product must come from the reagents in accordance with the mass preservation law. Special instructions are also provided to define the stereochemical outcome of reactions controlled by steric approach preferences. In RSL, the major product of a stereochemical reaction can be specified in two ways: 1) via the configuration of the nascent chiral center(s), and 2) via the stereochemical character of a bond during addition and elimination. Fig. 3 illustrates some simple examples of stereochemistry encoding in RSL.

<Fig. 3>

The stereochemical configuration of a formed chiral center can be identified as **unspecified**, **racemic** or **inverse**. **Unspecified** indicates that the exact configuration of the products is unknown or irrelevant (default), **racemic** indicates that both the R and S stereoisomers are formed in comparable quantities, and **inverse** exchanges one of the chiral center’s substituents during the reaction and inverts its configuration. The R/S assignment of the chiral center is automatically determined based on the original configuration and the CIP priority of the new substituent. In general, the stereochemical configuration of an atom can be specified by listing its substituents in clockwise order and designating the last substituent as an up (in front of the plane) or a down (behind the plane) wedge. In this case, the R/S assignment is based on the CIP priorities and order of the substituents. Alternatively, the R/S configuration of the chiral center could be explicitly specified, but this option is rarely used since the label depends on the CIP priorities of the individual building blocks.

Stereochemical ambiguity also emerges when the reaction mechanism involves multiple centers. For example, dehydrohalogenation leads to double bond formation via an *anti* elimination pathway, whereby two substituents are removed from opposite sides of the reduced single bond. The resulting double bond can be *cis* or *trans* depending on the original configuration of the bonded atoms. RSL defines two keywords, **syn\_product** and **anti\_product**, to specify whether an addition or elimination reaction proceeds in a *syn* or *anti* manner. Note that it is not always possible to identify a single product using these keywords. For completeness, the configuration of the double bond can also be explicitly specified as E or Z.

Finally, the **enumerate** statement triggers the creation of the virtual library. Although semantically simple, this statement is the most complicated in its implementation. For scalability, the enumeration of the products must be implicit and must circumvent the creation of a connection table or even a record for every product in the library. This objective is accomplished by dividing the enumeration process in two steps. During the first step, the reacting and interfering functionalities are identified by matching the corresponding SMARTS patterns, and any reagents that are not compatible with the reaction transform are eliminated from further processing. This step involves mostly substructure searching, and scales linearly with the number of reagents that make up the virtual library. The second step involves the generation of products and is delayed until a particular product is requested. That is, the construction of the connection table of a particular product occurs only when its structure is needed for display or evaluation, and in many cases this never happens. In the next chapter, we describe several methods to analyze a virtual library that require the enumeration of only a minor fraction of its members.

In order to accelerate the “on-demand” assembly of products, the mappings of the reactive groups matched by the SMARTS patterns are stored within the virtual library along with the compiled sequence of assembly operators. Thus, when the structure of a product is needed, no time is spent on substructure searching or parsing assembly instructions. This design and the speed of the underlying foundation classes and molecular perception algorithms upon which the software is based [24], enable the construction of products at a rate of 10,000 structures per second on a 800 MHz Pentium III processor, including full perception of valence, rings and aromaticity.

Our reaction scripting language differs from other reaction languages, such as SMIRKS [23], in that it is designed specifically for generating virtual combinatorial libraries and not for reaction database searching. Thus, RSL is less cryptic, provides more flexibility in encoding reaction transformations, and can be stored in a compiled form to allow ultra-fast product enumeration. In addition, in RSL, SMARTS patterns of the reacting functionalities are not restricted as in SMIRKS, where bond queries are not allowed and atomic expressions cannot contain queries if the bond order or connectivity change [23].

### III. LIBRARY ANALYSIS

**Structure Representation** – The main use of accessible virtual libraries is to enable the selection of smaller sub-libraries for physical synthesis and biological testing. This process is commonly referred to as virtual screening, and involves the computation of one or more molecular properties (descriptors) that are thought to be pertinent to the application at hand. These are typically classified into 1D, 2D and 3D descriptors, depending on the information required for their computation. 3D descriptors are perhaps the most informative, but also the most expensive to calculate, and their use has been limited to relatively small data sets. In contrast, 1D and 2D descriptors such as substructure keys [25], hashed fingerprints [26], topological indices [27], and information indices [28] are derived directly from the molecular graph, and are therefore much faster to compute. 1D and 2D descriptors have a proven track record in structure-activity correlation [27, 29] and have been shown to be equivalent or even outperform 3D descriptors in some applications [30, 31, 32, 33, 34]. A comparative study of several classes of descriptors in the context of library design can be found in several recent reviews [35, 36, 37, 38, 39, 40].

**Dynamic Descriptors** – Despite their simplicity, the computation of a sufficient number of 1D and 2D descriptors requires a few milliseconds on a modern PC, which is still prohibitively slow for an exhaustive characterization of large combinatorial libraries. To solve this problem, Cramer *et al.* proposed the use of “decomposable” descriptors, which can be computed in an additive or nearly additive manner from the corresponding descriptor values of the “clipped” reagents [18]. Although the products need not be assembled, the most useful descriptors, such as most connectivity and information indices, are not additive

and therefore not amenable to this approach. Recently, we proposed an alternative machine learning approach that does not require the connection tables of the products, and can be applied to a wide class of descriptors regardless of origin and complexity [41]. The method is based on probability sampling. First, a small subset of compounds from the virtual library is identified and their descriptors are calculated in a conventional manner. The resulting data is used as input to a multiplayer perceptron, which is trained to predict the descriptors of the products from the corresponding descriptors of their respective building blocks (Fig. 4). Once trained, the combinatorial neural network (CNN) is able to estimate the descriptors of the remaining members of the virtual library with remarkable accuracy, without ever generating their connection tables. The method does not require the use of clipped reagents, and the nonlinear nature of neural networks makes them ideally suited for estimating complex, non-decomposable properties.

<Fig. 4>

To demonstrate its potential, this approach was used to perform similarity searches from a 6.29 million member virtual library based on the 4-component Ugi reaction. The selections were obtained using the Euclidean distance as a measure of molecular similarity computed from a set of 30 PCs that were derived from a variety of 117 topological descriptors such as molecular connectivity indices, kappa shape indices, subgraph counts, information-theoretic indices, and topological state indices [27, 28]. The average similarity scores of the 1000 most similar compounds to 10 randomly chosen “leads” based on predicted descriptors were virtually identical to the scores of the selections obtained with directly computed properties (Fig. 5) [41]. This result is remarkable given the fact that the entire selection process required only 35 minutes on a 800 MHz Pentium III processor, including library generation, descriptor calculation, similarity searching, and network training, as opposed to 20 hours that were required for the direct, enumerative approach, which is equivalent to a 30-fold improvement in throughput.

<Fig. 5>

**Similarity Searching** – While the use of dynamic descriptors can result in dramatic computational savings, many combinatorial libraries are still exceedingly large to permit the examination of every single product in the collection. In a recently published work, we introduced a stochastic similarity search algorithm based on the principle of probability sampling [42]. The method proceeds in three stages. First, a small fraction of the products is selected at random and ranked according to their similarity against the query structure. The top-ranking compounds are then identified and deconvoluted into a list of “preferred” reagents, which are finally combined in all possible combinations to produce a smaller focused library, which is enumerated in an exhaustive manner, and systematically compared to the target to obtain the final selection.

This stochastic procedure (Fig. 6) was used to perform similarity searches from two large virtual libraries based on the reductive amination and Ugi reactions using as queries a known anti-arrhythmic agent and a product inhibitor, respectively. The stochastic selections were compared against an exhaustive approach which involved the systematic enumeration and comparison of every member of the virtual library to the query structure. It was found that the performance of the stochastic method depended on the size of the initial random sample and the number of top-ranking compounds used to extract the list of preferred reagents [42]. However, by sampling only 7% of the 6.75 million member amination library, it was possible to retrieve 96% of the most similar compounds. In the case of the 6.29 million member Ugi library, 18% of the entire library was sufficient to retrieve 98% of the true hits (Fig. 7).

<Fig. 6>

<Fig. 7>

**Compound Selection** – The similarity selections described above represent sparse arrays (also referred to as singles) that do not necessarily represent all possible combinations of the selected building blocks. While this method retrieves the best possible hits, it almost always results in experiments that are extremely difficult and expensive to execute. Thus, to contain cost and simplify synthesis, combinatorial libraries are usually synthesized in array format, even though arrays are generally inferior in terms of meeting the primary design objectives. However, although for sparse arrays similarity searching involves a simple scan

through the compounds that make up the virtual library, designing a full array involves a staggering number of combinations. This number is given by the binomial product:

$$C_a = \prod_{i=1}^R \frac{n_i!}{(n_i - k_i)!k_i!} \quad (1)$$

where  $k_i$  and  $n_i$  refer to the number of reagents requested and total number of reagent available at the  $i$ -th variation site in the  $R$ -component library. For example, for a 100×100 library of 10,000 compounds, there are  $10^{26}$  different 5×5 arrays to consider! Sorting through such a huge number of possibilities is impossible without the use of an effective optimization technique such as simulated annealing or evolutionary programming [9,10,14,16,17]. Although these techniques have been shown to deliver optimal or nearly optimal solutions for a wide variety of objective functions, they require substantial sampling of the energy landscape and can be intensive, particularly for objective functions that involve significant computation. Recently, we proposed an alternative deterministic algorithm that produces focused combinatorial arrays at a fraction of the time required by alternative Monte-Carlo techniques [43]. The method is applicable when the objective function is decomposable to individual molecular contributions, and makes use of a heuristic that allows the independent evaluation and ranking of candidate reagents in each variation site on the combinatorial library. The algorithm is extremely fast and convergent, and produces solutions that are comparable to and often better than those derived from the substantially more elaborate and computationally intensive stochastic sampling techniques (Fig. 8). Typical examples of design objectives that are amenable to this approach include maximum similarity to a known lead (or set of leads), maximum predicted activity or binding affinity according to some structure-activity or receptor binding model, containment within certain molecular property bounds, and many others.

<Fig. 8>

The algorithm capitalizes on the presence of optimal substructure when the objective function is decomposable to individual molecular contributions, and allows the selection of optimal or nearly optimal arrays within a fraction of a second on a modern personal computer. When the compounds are selected as an array, the following heuristic is employed. Given a combinatorial library of  $C$  components, a position  $i$

$\in [1, C]$ , and a particular choice of reagents for  $R_{j \neq i}$ ,  $j = 1, 2, \dots, i-1, i+1, \dots, C$ , the reagents for  $R_i$  that maximize the objective function can be determined by constructing and evaluating all possible sub-arrays derived from the combination of a single reagent from  $R_i$  with all the selected reagents for  $R_{j \neq i}$ , and selecting the ones with the highest fitness. The algorithm starts with a randomly chosen array, and optimizes each site in sequence until no further improvement is possible.

Consider, for example, the selection of a  $10 \times 10 \times 10$  array from a  $1000 \times 1000 \times 1000$  combinatorial library. The algorithm begins by selecting 10 reagents at random from each site and evaluating the fitness of the resulting array of 1000 compounds. This initial selection is refined in an iterative fashion by processing each reagent list in a strictly alternating sequence. At first, the algorithm constructs 1000  $1 \times 10 \times 10$  sub-arrays derived by combining each reagent from  $R_1$  with the selected reagents from  $R_2$  and  $R_3$ , and evaluates their fitness. Every reagent at  $R_1$  is evaluated in turn, and the 10 reagents with the highest score are selected. The process is then repeated for  $R_2$ . Each reagent in  $R_2$  is used to construct a  $10 \times 1 \times 10$  sub-array derived by combining it with the selected reagents from  $R_1$  and  $R_3$ , and the 10  $R_2$  reagents with the highest fitness are selected for that site.  $R_3$  is processed in a similar fashion, and this completes a refinement cycle. Once all the reagent lists have been processed, the selected reagents from each site are combined, and the fitness of the resulting full array is evaluated and compared to the fitness of the selection at the end of the previous cycle. If the fitness is improved, the refinement process is repeated starting at  $R_1$ . If not, the algorithm terminates. Thus, for each variation site, the algorithm evaluates the similarity of only  $10^5$  compounds. Typically, convergence is reached within a few iterations ( $< 10$ ), which means that the number of compounds that need to be evaluated during the course of optimization is less than 3% of the total size of the virtual library. When used with combinatorial networks, which have a throughput of  $\sim 10,000$  compounds per CPU second, this greedy approach can produce an optimized focused array in less than 10 minutes on a modern PC.

In a comparative study against simulated annealing, the greedy approach was shown to be extremely effective and convergent [43]. In fact, this algorithm works equally well with any objective function that can be described as a sum of individual molecular contributions, which includes virtually any measure of molecular similarity, drug-likeness as defined by the Lipinski rule of five, and many others [43].

**Diversity Profiling** – The task of designing diverse libraries for high-throughput screening involves some additional computational complexity. In addition to the computation of molecular descriptors and the combinatorial problem of selecting an optimal array, the evaluation of most diversity metrics scales to the square of the number of compounds selected [35]. Attempts to devise diversity metrics that do not exhibit quadratic complexity, such as the cosine coefficient [44], have been shown to produce unbalanced results [35]. A common solution to this problem is the use of reagent-based and cell-based selection techniques [1,45]. However, reagent-based techniques lead to designs that are substantially inferior than those selected based on the diversity of the products [1,2]. On the other hand, cell-based methods, which involve partitioning the diversity space into a fixed number of multidimensional cells and subsequent selection of compounds based on their occupancy, can be applied only to low-dimensional spaces, and depend to a large extent on the choice of the dimensions and the grid resolution [46]. Thus, for large selections containing thousands of compounds, alternative metrics must be devised to enable a more expedient estimation of molecular diversity.

Recently, we presented a novel diversity function that captures the notion of spread, is fast to compute, scales favorably with the number of compounds in the design, does not fall prey to dimensionality, and can be used to compare collections of different cardinality [47]. The method is based on the fundamental assumption that an optimally diverse sample is one that is uniformly distributed in the property space it is designed to explore. Diversity is quantified by estimating the cumulative probability distribution of inter-molecular dissimilarities in the collection of interest, and then measuring the deviation of that distribution from the respective distribution of a uniform sample using the Kolmogorov-Smirnov statistic [5, 48]. This method measures how well an experimental distribution is approximated by a particular distribution function. It is applicable to unbinned distributions that are functions of a single independent variable, and is defined as the maximum value of the absolute difference between two cumulative distribution functions:

$$K^* = \max_{-\infty < x < \infty} |P(x) - P^*(x)| \quad (2)$$

where  $P(x)$  is an estimator of the cumulative distribution function of the actual probability distribution from which it is drawn, and  $P^*(x)$  is a known cumulative distribution function. Unlike the more commonly used  $\chi^2$  test, the Kolmogorov-Smirnov statistic does not require binning of the data, which is arbitrary and leads to loss of information. More importantly, the function is very fast to compute since it involves sorting the

data in ascending order, followed by a linear scan to identify the maximum difference from the user-defined cumulative distribution function (or a simultaneous scan of two vectors in the case of two cumulative distributions) (Fig 9).

<Fig. 9>

The distinct advantage of this approach is that the cumulative distribution can be easily estimated using probability sampling and does not require exhaustive enumeration of all pairwise distances in the design, resulting in an algorithm of virtually constant time complexity. While some caution must be exercised in determining the appropriate target distribution [47], the function produces results that are consistent with our notion of spread and can do so at a fraction of the time required by alternative methodologies. The selection of an array using this algorithm is shown in Fig. 10. While the selected compounds are not as perfectly distributed in the input space, they show no preference for any particular region of the map, nor are they biased by the local density of the parent collection, i.e. they are diverse. The combination of this diversity metric with dynamic descriptors with a stochastic optimization procedure such as simulated annealing [9] enables the expedient selection of diverse arrays from very large combinatorial libraries that are intractable with conventional techniques. We have found that these optimization schemes converge very rapidly as there are multiple arrays of equal or very similar diversity, especially in a large combinatorial library.

<Fig. 10>

**Parallel Processing** – Significant improvements in screening performance can also be achieved by taking advantage of parallel and distributed hardware. Advances in computer design and networking have made PC clusters a very effective and affordable solution. Combinatorial libraries are ideally suited to parallel processing, as the virtual compounds can be easily divided into non-overlapping groups that can be enumerated, characterized and evaluated independently. Careful programming can minimize potentially

expensive inter-process communication, and can lead to dramatic reductions in the amount of time required for the analysis of virtual libraries.

#### **IV. CONCLUSIONS**

Combinatorial chemistry presents unique opportunities but also unprecedented challenges for computational drug design. Contrary to classical approaches, it is no longer sufficient to examine one synthetic candidate at a time. An effective combinatorial experiment must sort through a vast number of possibilities and take into account the collective properties of thousands of compounds. This necessitates a departure from conventional thinking and a shift of emphasis from accuracy to efficiency. As we demonstrate in this paper, combinatorial libraries have properties that make them ideally suited to probability sampling techniques, which permit the analysis of very large data sets that are otherwise computationally intractable.

#### **ACKNOWLEDGMENTS**

We wish to thank Dr. Raymond F. Salemme of 3-Dimensional Pharmaceuticals, Inc. for his insightful comments and support of this work.

## REFERENCES

- [1] Gillet, V. J.; Willett, P.; Bradshaw, J. The effectiveness of reactant pools for generating structurally-diverse combinatorial libraries, *J. Chem. Inf. Comput. Sci.*, **1997**, *37*, 731-740.
- [2] Jamois, E. A.; Hassan, M.; 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] Leach, A. R., Hann, M. M. The *in silico* world of virtual libraries, *Drug Discovery Today*, **2000**, *5*, 326-336.
- [4] Lipinski C.A. et al. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug Deliv. Rev.* **1997**, *23*, 3-25.
- [5] Rassokhin, D. N.; Agrafiotis, D. K. Kolmogorov-Smirnov statistic and its application in library design, *J. Mol. Graphics Modell.*, **2000**, *18(4-5)*, 370-384.
- [6] Sadowski, J.; Kubinyi, H. A scoring scheme for distinguishing between drugs and non-drugs. *J. Med. Chem.*, **1998**, *41*, 3325-3329.
- [7] Ajay; Walters, W. P.; Murcko, M. A. Can we learn to distinguish between “drug-like” and “nondrug-like” molecules?, *J. Med. Chem.*, **1998**, *41*, 3314-3324.
- [8] Wang, J.; Ramnarayan, K. Toward designing drug-like libraries: a novel computational approach for prediction of drug feasibility of compounds, *J. Comb. Chem.*, **1999**, *1*, 524-533.
- [9] Agrafiotis, D. K. Stochastic algorithms for maximizing molecular diversity, *J. Chem. Inf. Comput. Sci.*, **1997**, *37*, 841-851
- [10] Agrafiotis, D. K., On the use of information theory for assessing molecular diversity. *J. Chem. Inf. Comput. Sci.*, **1997**, *37(3)*, 576-580.
- [11] Agrafiotis, D. K., and Lobanov, V. S., An efficient implementatin of distance-based diversity metrics based on k-d trees. *J. Chem. Inf. Comput. Sci.*, **1999**, *39(1)*, 51-58.
- [12] Agrafiotis, D. K., Multiobjective optimization of combinatorial libraries, *IBM J. Res. Develop.*, in press.
- [13] Hassan, M.; Bielawski, J. P.; Hempel, J. C.; Waldman, M. Optimization and visualization of molecular diversity of combinatorial libraries, *J. Comput. Aided. Mol. Des.*, **1996**, *2*, 64-74.

- [14] Good, A. C.; Lewis, R. A. New methodology for profiling combinatorial libraries and screening sets: cleaning up the design process with HARPick, *J. Med. Chem.*, **1997**, *40*, 3926-3936.
- [15] Agrafiotis, D.K.; Bone, R.F.; Salemme, F.R.; Soll, R.M., United States Patents 5,463,564, **1995**; 5,574,656, **1996**; 5,684,711, **1997**; and 5,901,069, **1999**.
- [16] Brown, R. D.; Martin, Y. C. Designing combinatorial library mixtures using a genetic algorithm, *J. Med. Chem.*, **1997**, *40*, 2304-2313.
- [17] Gillet, V. J.; Willett, P.; Bradshaw, J.; Green, D. V. S. Selecting combinatorial libraries to optimize diversity and physical properties, *J. Chem. Inf. Comput. Sci.*, **1999**, *39*, 169-177.
- [18] Cramer, R.D.; Patterson, D.E.; Clark, R.D.; Soltanshahi, F.; 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.
- [19] Leland, B. A.; Christie, B. D.; Nourse, J. G.; Grier, D. L.; Carhart, R. E.; Maffett, T.; Welford, S. M.; Smith, D. H. Managing the combinatorial explosion, *J. Chem. Inf. Comput. Sci.*, **1997**, *37*, 62-70.
- [20] Downs, G. M.; Barnard, J. M. Techniques for generating descriptive fingerprints in combinatorial libraries, *J. Chem. Inf. Comput. Sci.*, **1997**, *37*, 59-61.
- [21] Leach, A. R.; Bradshaw, J.; Green, D. V. S.; Hann, M. M.; Delany, J. J., Implementation of a system for reagent selection and library enumeration, Profiling, and Design, *J. Chem. Inf. Comput. Sci.*, **1999**, *39*, 1161-1172.
- [22] Ousterhout, J. Tcl and the Tk Toolkit, Addison-Wesley, ISBN 0-201-63337-X, **1994**.
- [23] James, C. A.; Weininger, D.; Delany, J. Daylight Theory Manual Daylight 4.71. Daylight Chemical Information Systems, Inc., **2000**, <http://www.daylight.com/dayhtml/doc/theory/theory.toc.html>.
- [24] Copyright © 3-Dimensional Pharmaceuticals, Inc., 1994-2000.
- [25] Molecule Database Administration Guide, Ver. 2.0.1, MDL Information Systems, Inc., **1996**, pp 2-13, 8-57.
- [26] James, C. A.; Weininger, D.; Delany, J. 6. Fingerprints – Screening and Similarity. In Daylight Theory Manual Daylight 4.71. Daylight Chemical Information Systems, Inc., **2000**, <http://www.daylight.com/dayhtml/doc/theory/theory.finger.html>.

- [27] Hall L.H.; Kier, L.B. The molecular connectivity chi indexes and kappa shape indexes in structure-property relations. In *Reviews of Computational Chemistry*, Boyd, D.B.; Lipkowitz, K.B., Ed.; VCH Publishers, **1991**; Chapter 9, 367-422.
- [28] Bonchev, D. Information-theoretic indices for characterization of chemical structure; Research Studies Press: Letchworth, **1993**.
- [29] Devillers, J., Balaban, A. T., Eds. Topological indices and related descriptors in QSAR and QSPR, Gordon and Breach, The Netherlands, **1999**.
- [30] Brown, R. D.; Martin, Y. C.; Use of structure-activity data to compare structure-based clustering methods and descriptors for use in compound selection, *J. Chem. Inf. Comput. Sci.*, **1996**, *36*, 572-584.
- [31] Brown, R. D.; Martin, Y. C. The information content of 2D and 3D structural descriptors relevant to ligand-receptor binding, *J. Chem. Inf. Comput. Sci.*, **1997**, *37*, 1-9.
- [32] Matter, H. Selecting optimally diverse compounds from structure databases: a validation study of two-dimensional and three-dimensional molecular descriptors, *J. Med. Chem.*, **1997**, *40*, 1219-1229.
- [33] Matter, H.; Potter, T. Comparing 3D pharmacophore triplets and 2D fingerprints for selecting diverse compound subsets, *J. Chem. Inf. Comput. Sci.*, **1999**, *39*, 1211-1225.
- [34] Schuffenhauer, A.; Gillet, V. J.; Willett, P. Similarity searching in files of three-dimensional chemical structures: analysis of the BIOSTER database using two-dimensional fingerprints and molecular field descriptors, *J. Chem. Inf. Comput. Sci.*, **2000**, *40*, 295-307.
- [35] Agrafiotis, D. K. Diversity of chemical libraries, In *Encyclopedia of Computational Chemistry*; Schleyer, P.v.R.; Allinger, N. L.; Clark, T.; Gasteiger, J.; Kollman, P. A.; Schaefer III, H. F.; Schreiner, P. R., Eds., John Wiley and Sons, Chichester, **1998**, Vol. 1, pp. 742-761.
- [36] Agrafiotis, D. K., Myslik, J. C., and Salemme, F. R., Advances in diversity profiling and combinatorial series design. *Mol. Diversity*, **1999**, *4(1)*, 1-22.
- [37] Agrafiotis, D. K., Lobanov, V. S., Rassokhin, D. N., and Izrailev, S., The measurement of molecular diversity, in *Virtual screening of bioactive molecules*, Böhm, H.-J., and Schneider, G., Eds., Wiley-VCH, Weinheim, **2000**, 265-300.

- [38] Martin, Y. C.; Brown, R. D.; Bures, M. G. Quantifying diversity. In *Combinatorial Chemistry and Molecular Diversity in Drug Discovery*, Gordon, E. M.; Kerwin, J. F., Jr. Eds. Wiley, New York, **1998**, 369-385.
- [39] Gillet, V. J. Background theory of molecular diversity. In *Molecular Diversity in Drug Design*, Dean, P. M., Lewis, R. A., Eds., Kluwer, London, **1999**, 43-65.
- [40] Gorse, D.; Lahana, R. Functional diversity of compound libraries, *Cur. Opin. Chem. Bio.*, **2000**, *4*, 287-294.
- [41] Lobanov, V. S., and Agrafiotis, D. K., Combinatorial Networks, *J. Mol. Graphics Modell.*, in press.
- [42] Lobanov, V. S., and Agrafiotis, D. K. Stochastic similarity selections from large combinatorial libraries, *J. Chem. Inf. Comput. Sci.*, **2000**, *40*, 460-470.
- [43] Agrafiotis, D. K., and Lobanov, V. S., Ultrafast algorithm for designing focused combinatorial arrays, *J. Chem. Inf. Comput. Sci.*, **2000**, *40*, 1030-1038.
- [44] Turner, D. B.; Tyrrell, S. M.; Willett, P. Rapid quantification of molecular diversity for selective database acquisition, *J. Chem. Inf. Comput. Sci.*, **1997**, *37*, 18-22.
- [45] Schnur, D. Design and diversity analysis of large combinatorial libraries using cell-based methods, *J. Chem. Inf. Comput. Sci.*, **1999**, *39*, 36-45.
- [46] Cummins, D. J.; Andrews, C. W.; Bentley, J. A.; Cory, M. Molecular diversity in chemical databases: comparison of medicinal chemistry knowledge bases and databases of commercially available compounds, *J. Chem. Inf. Comput. Sci.*, **1996**, *36*, 750-763.
- [47] Agrafiotis, D. K. Constant time algorithm for estimating the diversity of large chemical libraries. *J. Chem. Info. Comput. Sci.*, in press.
- [48] von Mises, R., *Mathematical Theory of Probability and Statistics*, Academic Press, New York, **1997**.

## **CAPTIONS TO FIGURES**

1. Example of an RSL script for generating a virtual library based on the reductive amination reaction.
2. Most frequently used RSL operators (assembly instructions).
3. Examples of encoding stereochemical reactions in RSL.
4. Architecture of combinatorial neural networks.
5. Average similarity scores (Euclidean distances) of the 1000 most similar compounds to 10 randomly chosen “leads” selected from the 6.29 million-member virtual library based on the Ugi reaction. The three series represent the similarity scores of random selections, and selections based on predicted (CNN) and actual descriptors, respectively.
6. Flowchart of the stochastic similarity searching algorithm.
7. Performance of the stochastic algorithm for performing similarity searches from the (a) diamine, and (b) Ugi libraries. The quality of the selection (solid lines) is measured by the percent overlap with the corresponding ideal selections. The cost of the selection (dotted lines) is the cumulative percentage of the total number of virtual compounds evaluated.
8. Mean and standard deviation of the similarity scores of 100 compounds selected from the reductive amination library according to maximum similarity to the lead structure (10×10 array), collected over 100 optimization runs.
9. Kolmogorov-Smirnov statistic for computing the difference between two cumulative distribution functions.
10. Selection of a maximally diverse 10×10 array from the reductive amination library based on (a) the average nearest neighbor Euclidean distance, and (b) on the Kolmogorov-Smirnov diversity measure.

```
proc reductive_amination {library amines aldehydes}
{
  define reagent amine as "[C,c][NH2]" or "C[NH]C" and not "C(=O)[NH,NH2]";
  define reagent aldehyde as "C[CH]=O";
  define product p;
  p add amine from $amines;
  p add aldehyde from $aldehydes;
  p insert bond aldehyde:1 amine:1;
  p remove atom aldehyde:2;
  enumerate p as $library;
}
```

Fig. 1

<product> add <reagent> from <source>  
<product> insert bond <r1:a1> <r2:a2> [<order>] [<stereo>]  
<product> remove atom <r1:a1>  
<product> remove bond <r1:a1> <r1:a2>  
<product> remove attachment <r1:a1>  
<product> remove fragment <r1:a1> <r1:a2>  
<product> set atom <r1:a1> charge|radical <value>  
<product> set bond <r1:a1> <r1:a2> <order> [<stereo>]  
<product> set atom <r1:a1> configuration <r1:a2> <r1:a3> <r1:a4> <r1:a5> up|down

Fig. 2

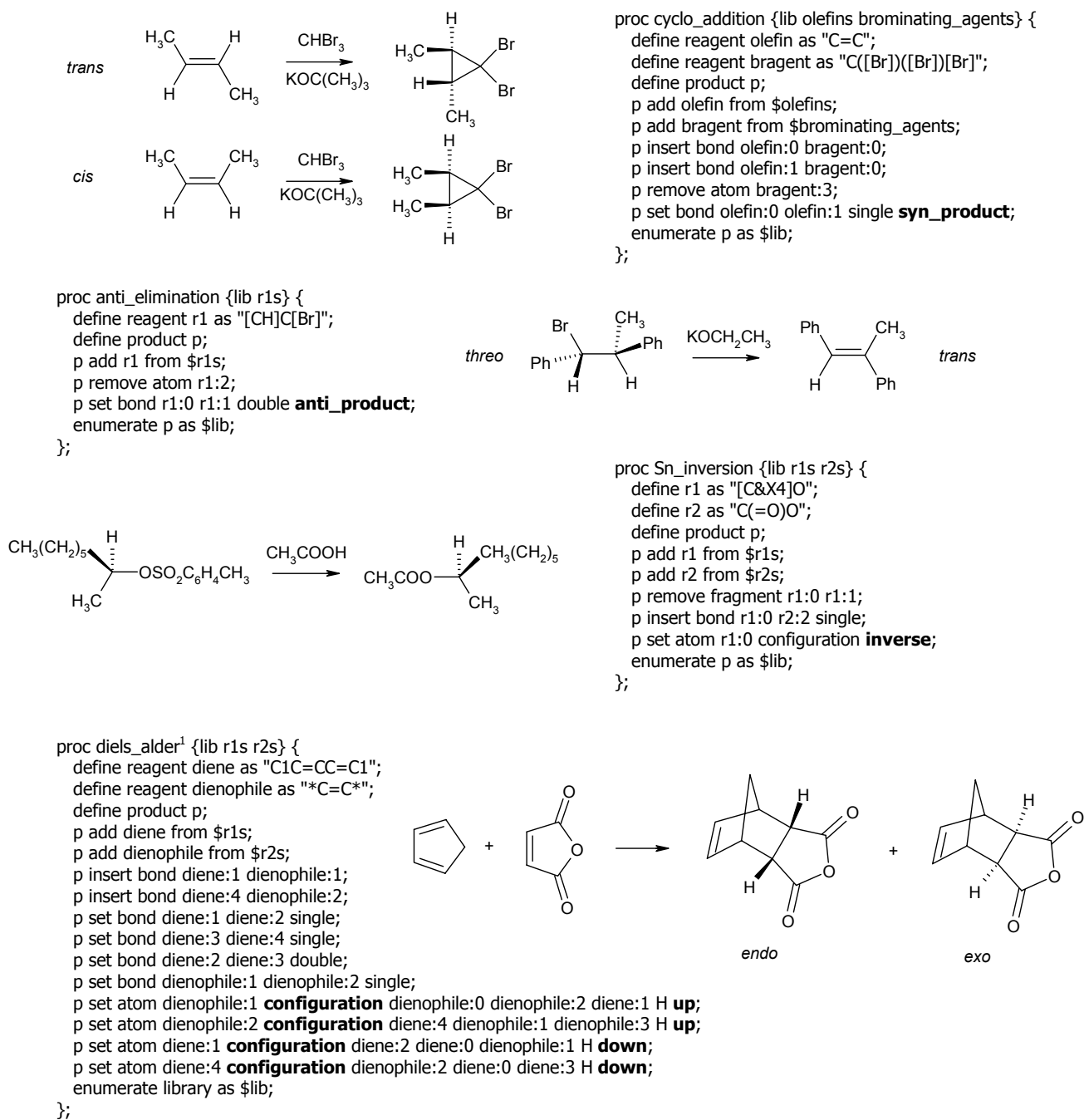


Fig. 3

<sup>1</sup> According to the empirical Alder rule, if two isomeric adducts are possible, the one that has an unsaturated substituent(s) on the alkene oriented toward the newly formed cyclohexene double bond is the preferred product. In the given example, the addition of dienophiles to cyclopentadiene usually favors the *endo* stereoisomer. Thus, an upward orientation of hydrogen atoms attached to dienophile atoms 1 and 2 should be specified explicitly along with the upward orientation of the norbornene bridge.

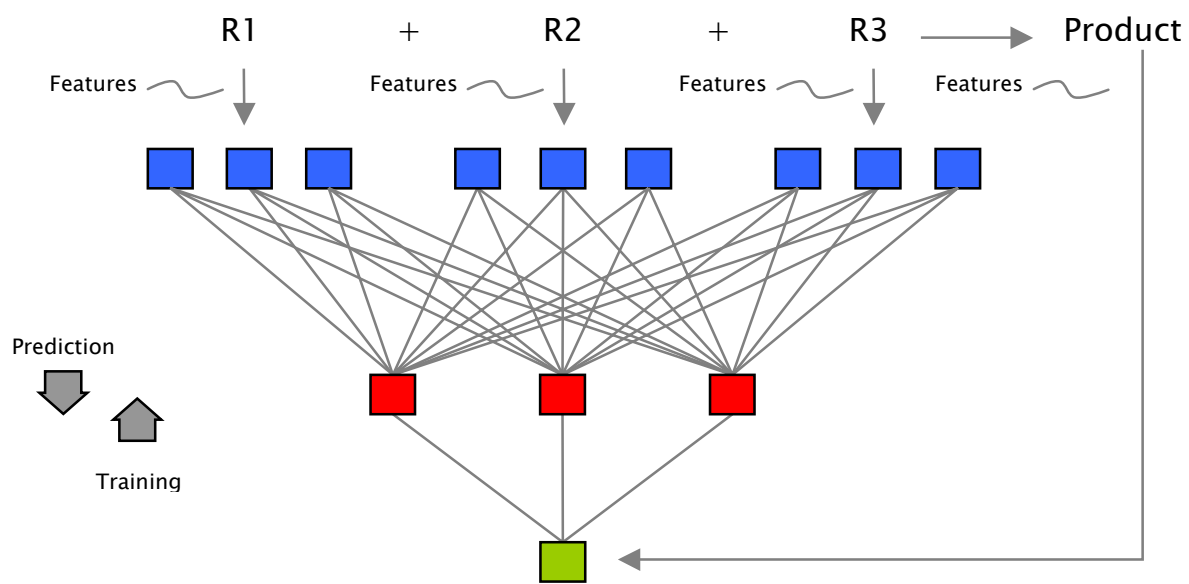


Fig. 4

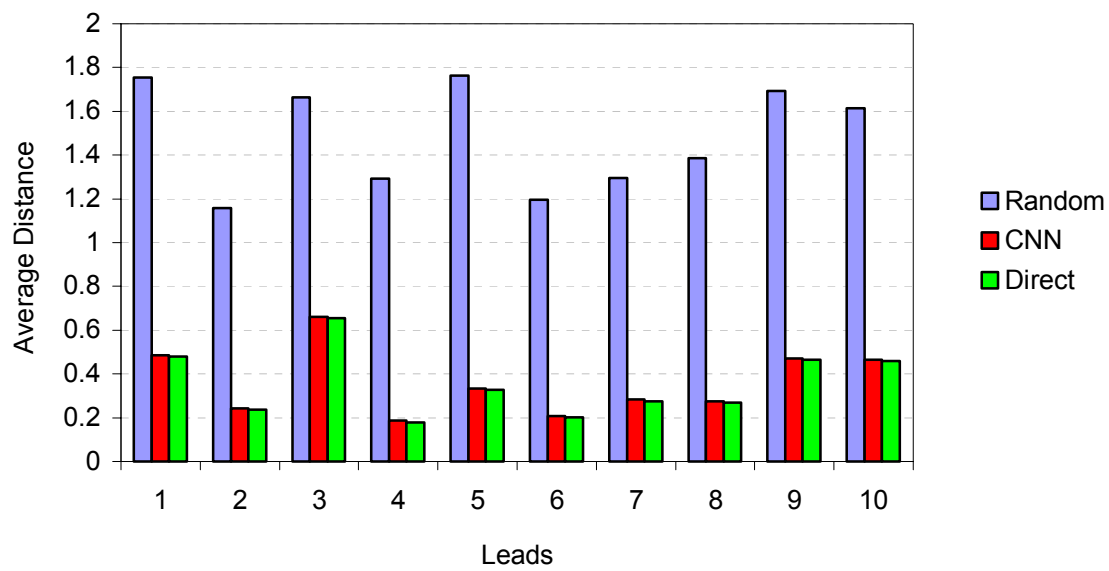


Fig. 5

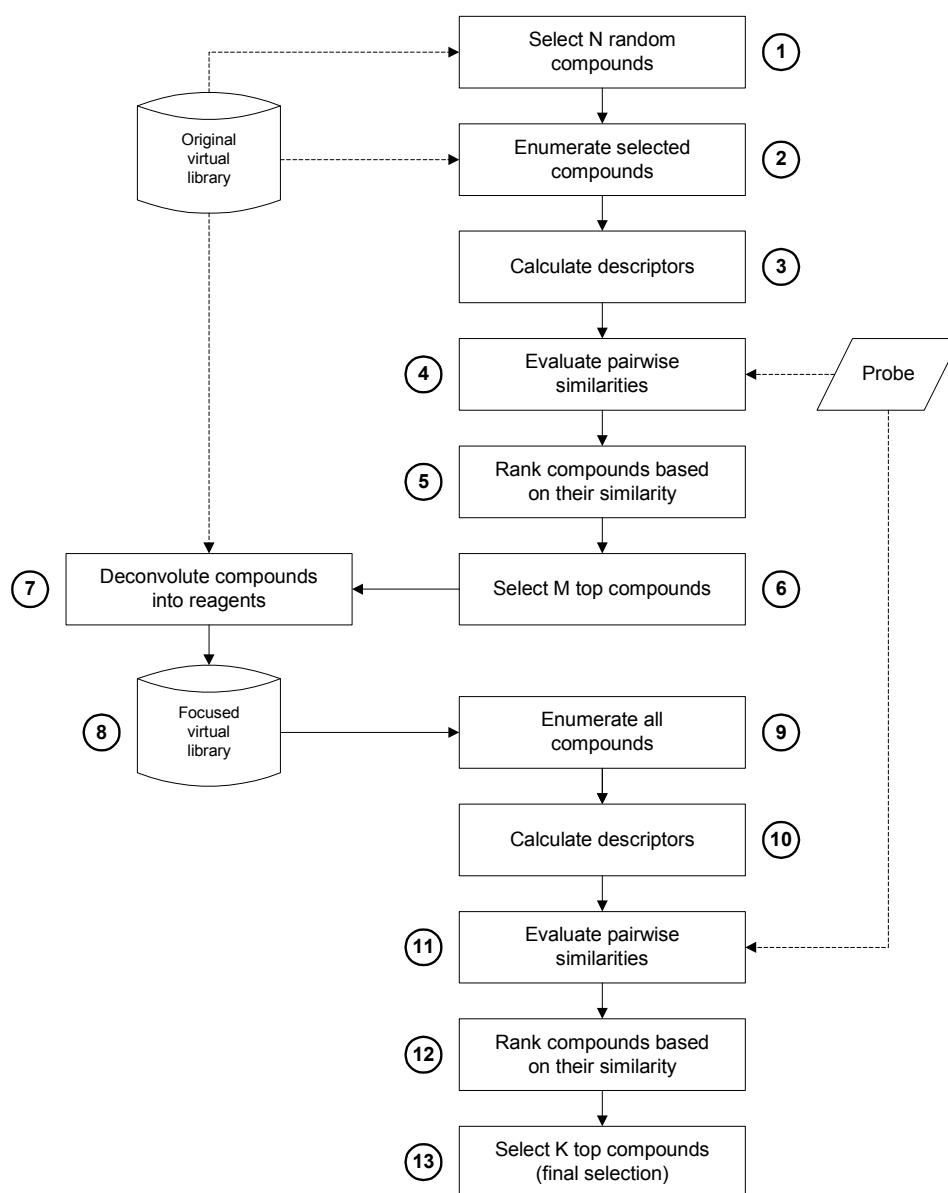


Fig. 6

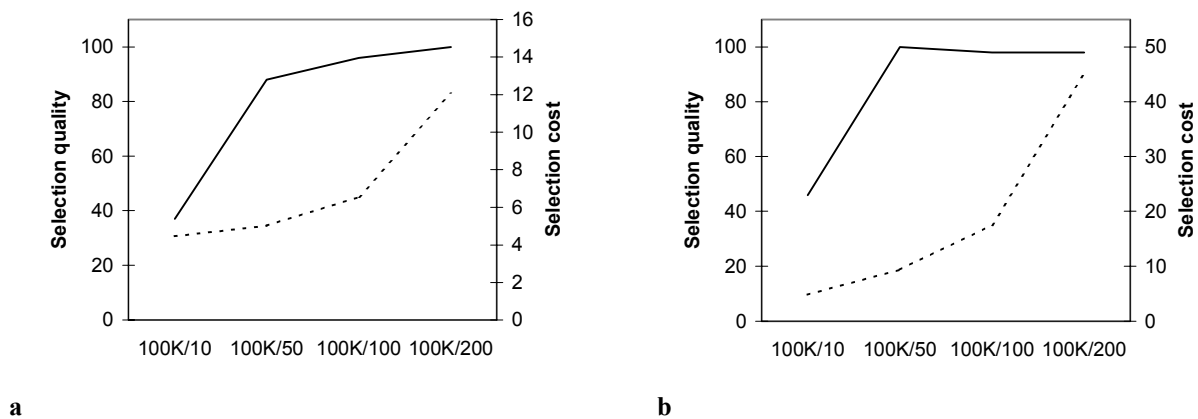


Fig. 7

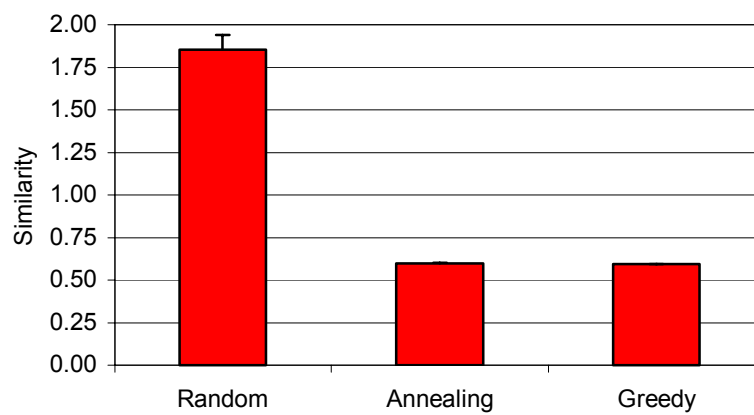


Fig. 8

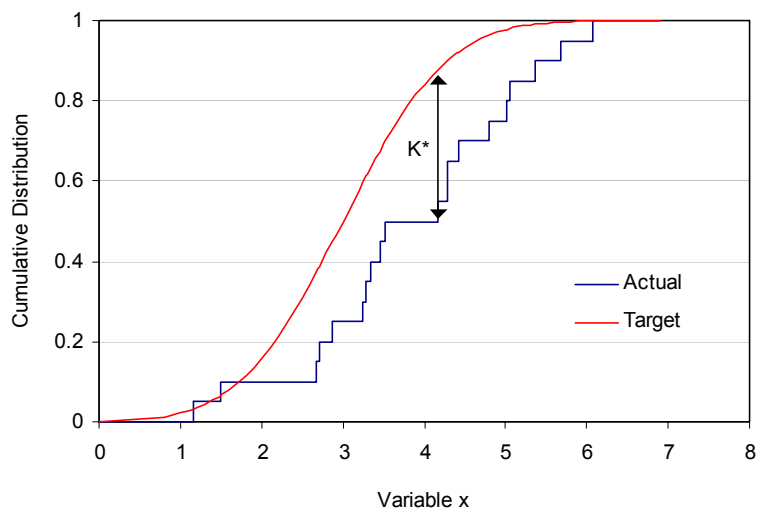


Fig. 9

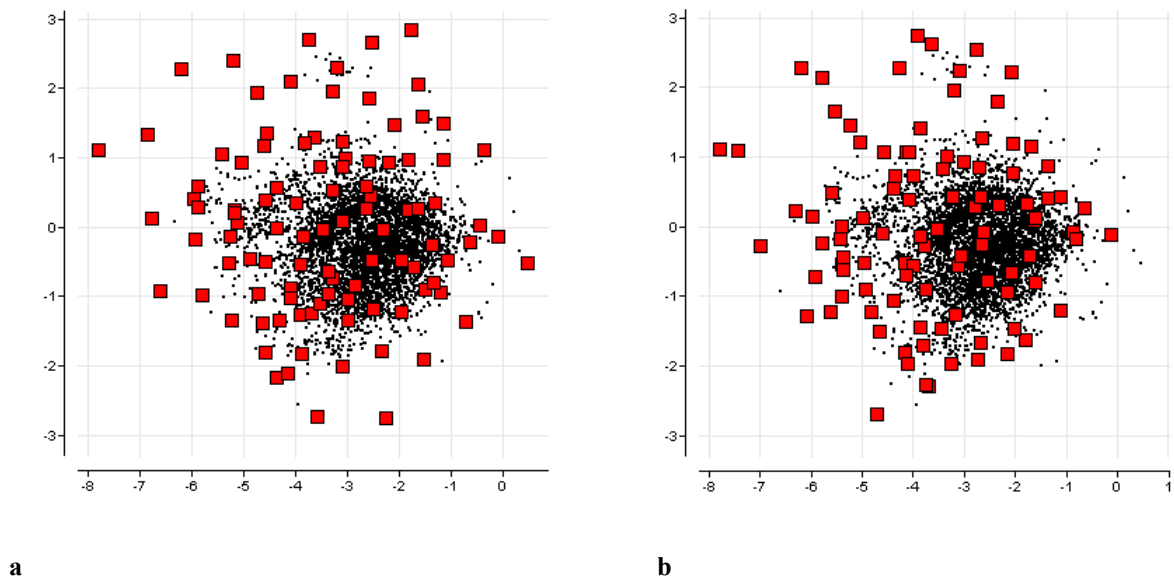


Fig. 10