Multiobjective optimization of combinatorial libraries

by D. K. Agrafiotis

Combinatorial chemistry and high-throughput screening have caused a fundamental shift in the way chemists contemplate experiments. Designing a combinatorial library is a controversial art that involves a heterogeneous mix of chemistry, mathematics, economics, experience, and intuition. Although there seems to be little agreement as to what constitutes an ideal library, one thing is certain: Only one property or measure seldom defines the quality of the design. In most real-world applications, a good experiment requires the simultaneous optimization of several, often conflicting, design objectives, some of which may be vague and uncertain. In this paper, we discuss a class of algorithms for subset selection rooted in the principles of multiobjective optimization. Our approach is to employ an objective function that encodes all of the desired selection criteria, and then use a simulated annealing or evolutionary approach to identify the optimal (or a nearly optimal) subset from among the vast number of possibilities. Many design criteria can be accommodated, including diversity, similarity to known actives, predicted activity and/or selectivity determined by quantitative structure-activity relationship (QSAR) models or receptor binding models, enforcement of certain property distributions, reagent cost and availability, and many others. The method is robust, convergent, and extensible, offers the user full control over the relative significance of the various objectives in the final design, and permits the simultaneous selection of compounds from multiple libraries in full- or sparse-array format.

1. Introduction

Historically, drug discovery has been based on a serial and systematic modification of chemical structure aimed at producing compounds which can effectively and safely alter the activity of biological targets associated with a particular disease. This process involves four major steps following the identification of a biological target: 1) hit identification, 2) lead generation, 3) lead optimization, and 4) target validation. The first part of this process is carried out by screening large compound collections, such as combinatorial libraries, natural product collections, and corporate banks, in order to identify compounds that interact with the target enzyme or receptor. Once a hit has been identified, it is chemically modified by iterative synthesis and testing of related analogs to produce leads, i.e., compounds with improved chemical characteristics that are more suitable as potential drugs. Further chemical modification optimizes the properties of these leads and converts them into drug development candidates for further preclinical and clinical development.

0018-8646/01/$5.00 © 2001 IBM

Prior to the advent of combinatorial chemistry, this process involved a simple prioritization of synthetic targets based on pre-existing structure-activity data, synthetic feasibility, experience, and intuition. This situation began to change as advances in parallel synthesis and high-throughput screening have enabled the simultaneous synthesis and biological evaluation of large chemical libraries containing hundreds to tens of thousands of compounds [1]. Although throughput has increased dramatically, the number of compounds that can be created and tested in a reliable manner represents a tiny fraction of all the molecules of potential pharmaceutical interest, and the process is still fundamentally based on trial and error. It is becoming increasingly apparent that in order to maximize the probability of identifying sustainable drug candidates, combinatorial experiments must be carefully planned and take full advantage of whatever information is available about the biological target of interest. Whether it is used for lead discovery or optimization, the design of a good combinatorial library is a complex task that requires the simultaneous optimization of several, often conflicting, design objectives. In this paper, we present an overview of a general methodology for designing combinatorial and high-throughput screening experiments rooted in the principles of multiobjective optimization.

Multiobjective (MO) optimization (also known as multicriterion, vector, or Pareto optimization) extends optimization theory by permitting several design objectives to be optimized simultaneously. Although the basic methods can be traced back to the work of Leibniz and Euler, the principle of multiobjective optimization was first formalized by Vilfredo Pareto, an Italian economist, whose theories [2] are now considered the basis of modern welfare economics exercised by socialist economic scholars. He introduced the concept of the *Pareto optimum*, a standard of judgment in which the optimum allocation of the resources of a society is not attained as long as it is possible to make at least one individual better off in his own estimation while keeping others as well off in their own estimation. In Pareto's own words: "The principal subject of our study is (economic) equilibrium. This equilibrium results from the opposition between men's tastes and the obstacles to satisfying them. Our study includes, then, three distinct parts: 1) the study of tastes; 2) the study of obstacles; and 3) the study of the way in which these two elements combine to reach equilibrium." Throughout the years, Pareto's basic theories have been extended [3] and applied to a wide spectrum of optimization problems in economics, management, engineering, and social sciences.

A multiobjective optimization problem is solved in a manner similar to the conventional single-objective (SO) problem. The goal is to find a set of values for the design variables that simultaneously optimize several objective (or cost) functions. The solutions are often referred to as Pareto optima, vector maxima, efficient points, or nondominated solutions. In general, the solutions obtained by individual optimization of each objective (i.e., SO optimization) do not represent a feasible solution to the multiobjective problem. Several methods have been devised for constructing Pareto-optimal sets, including hierarchical optimization, weighing objectives, distance functions, goal programming, tradeoff or constraint methods, min–max optimization, and many others [3]. These methods fall into two broad categories: 1) methods which attempt to optimize each criterion in turn, subject to constraints derived from the optimization of previously optimized criteria; and 2) methods which attempt to minimize a single objective function that combines the design objectives in some prescribed functional form (often referred to as a global criterion).

Unfortunately, the vast majority of multiobjective optimization theory deals with continuous or semicontinuous design variables and is not directly applicable to discrete, combinatorial problems such as the application at hand. In this paper, we review the algorithmic details of a general methodology for designing combinatorial libraries that combines the flexibility of multiobjective fitness functions with the power of simulated annealing for searching vast combinatorial state spaces. The method allows traditional design objectives such as molecular similarity or molecular diversity to be combined with other selection criteria in order to enforce certain druglike property distributions, contain the cost of the experiment, minimize potential toxicological and pharmacokinetic liabilities, etc.

Library design has evolved into a distinct subdiscipline of computational chemistry standing on the crossroads of organic synthesis, chemometrics, QSAR, and structure-based design. Although it is not our intention to provide an in-depth review of this field, the reader should be aware that the approach described here is one of many available methodologies for designing effective and well-targeted combinatorial experiments, and one that has been pursued independently by several groups. While some of the underlying principles have their origin in the field of statistical experimental design, the first reports dealing specifically with combinatorial libraries were presented in 1995 by five independent groups [4–8]. In what is considered by many the pivotal computational work on molecular diversity, Martin and coworkers reported a rational method for selecting a set of monomers for a peptoid combinatorial library based on D-optimal design [4]. To ensure that the design would capture biologically relevant information, the group employed a wide range of molecular descriptors that captured lipophilicity, shape and branching, chemical functionality, and receptor binding. Principal component analysis and multidimensional scaling were employed to reduce the dimensionality of the original data, and a D-optimal design procedure was used to select a representative subset of reagents to optimally explore the resulting diversity space.

An alternative approach that is more closely related to the one described herein was presented by groups at Merck, Hoffman–La Roche, and Sterling Winthrop Pharmaceuticals. Sheridan and Kearsley [6] employed a genetic algorithm to generate N-substituted glycine tripeptoids which maximized similarity to a known lead or predicted activity according to a trend vector developed from a series of active molecules. Weber et al. [7] presented a similar genetic scheme using a fitness function that was based on the results of an experimental enzyme assay rather than QSAR modeling, and Singh et al. [8] employed a similar methodology to optimize a set of potent and selective hexapeptide stromelysin substrates.

The group at 3-Dimensional Pharmaceuticals was also quick to recognize that biological data is the best guide for the design of combinatorial and high-throughput screening experiments, and that effective data management and process integration were key for the timely and cost-effective development of new therapeutic agents. Contemporaneously with the aforementioned publications, our group obtained the first of a series of patents describing a new drug discovery paradigm that integrates combinatorial, structural, and computational chemistry under a unifying information management system [4]. The system, known as DirectedDiversity**, is an iterative optimization process that explores combinatorial space through successive rounds of selection, synthesis, and testing. Unlike traditional combinatorial approaches, in which the entire library is made and tested in a single conceptual step, DirectedDiversity physically synthesizes, characterizes and tests only a portion of that library at a time. The selection of compounds is carried out by computational search engines that combine optimal exploration of molecular diversity with a directed search based on structure-activity relationship (SAR) information accumulated from previous iterations of the integrated machinery. The original blueprint was based on an optimization scheme that was in many respects similar to that described by the previous authors, but emphasized more strongly the need to adjust the selection criteria in the course of a discovery program and to deal with the ambiguity that is inherent in the biological response data.

In the following years, the above algorithms were elaborated, calibrated, and applied to a wide range of problems in library design. From an algorithmic perspective, the most notable examples include the independently developed simulated annealing implementations of Agrafiotis [9–12] and Hassan et al. [13, 14], and their subsequent variations by Good and Lewis [15] and Zheng et al. [16]; Brown and Martin's genetic scheme to generate libraries designed to minimize the effort required to deconvolute biological hits by mass-spectroscopic techniques [17]; the attempts by Gillet et al. [18], Rassokhin and Agrafiotis [19], and Brown et al. [20] to enforce certain property distributions on the final design; and the latest use by Sheridan et al. [21] of genetic algorithms for designing targeted libraries. These advances were complemented by important validation studies, which compared the ability of several popular sets of descriptors to differentiate between active and inactive molecules [22–27], as well as the relative merits of reagent- versus product-based designs [28, 29]. They represent only a small fraction of proposed library design methodologies, which range from conventional experimental design [4, 30, 31] to clustering [32] and cluster sampling [33], conformational sampling [34], partitioning [35, 36], Boolean logic [37–39], vector analysis [40], and some more recent "greedy" algorithms for selecting combinatorial arrays [41, 42]. For a more extensive account, the interested reader is referred to several recently published reviews [31, 43–45].

2. Methods

Architecture

In its prototypical form, the selection problem can be stated as follows: Given a collection of n compounds and a number k, find the "best" set of k compounds according to some user-defined criteria. This problem is NP-complete, and the cardinality of the search space is enormous for even the most conservative cases encountered in library design. The approach taken here is to combine all of the selection criteria into a single unifying function, and maximize (or minimize) that function using an efficient optimization algorithm such as simulated annealing or evolutionary programming in order to identify the optimal (or a nearly optimal) set among the vast number of possibilities. To simplify the description of the algorithm, we define the following entities: *collections* or *libraries*, which represent separate pools of chemical compounds (combinatorial libraries and/or regular collections) from which the selection is to be drawn; *subsets*, which represent selections of compounds from a particular compound collection; *states* or *designs*, which represent a collection of subsets from one or more chemical libraries; *selection criteria*, which encode the individual design objectives; *objective functions*, which combine one or more selection criteria in some arbitrary functional form and provide the overall quality of a particular design (state); and *optimizers*, which search through the state space associated with the problem of

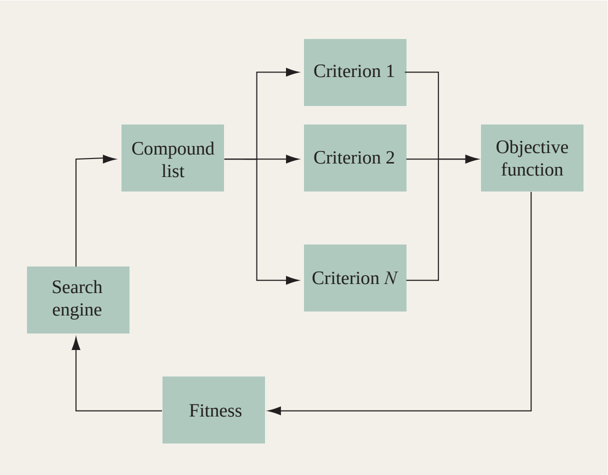


Figure 1

Overall architecture of approach used for multiobjective compound selection.

interest to identify the optimal (or a nearly optimal) solution.

The overall architecture of the approach used is shown in **Figure 1**. An optimizer (in this case a serial or parallel implementation of simulated annealing) produces a state (i.e., a collection of subsets from one or more chemical libraries), which is evaluated against all of the desired selection criteria. These are combined into a unifying objective function, which measures the overall fitness of that state—that is, its ability to collectively satisfy all of the specified selection criteria. This fitness value is used by the optimizer to produce a new set of compounds (i.e., a new state), which is in turn evaluated against the prescribed selection criteria in the manner outlined above. The process continues until a predefined termination criterion or time limit is met, and the best state identified during the course of the simulation is reported. This general scheme can be implemented in a serial or parallel fashion; in the latter case, several states are evaluated in parallel and are subsequently combined to produce a new set of states for the next iteration (see below).

The major advantage of this approach is that the search algorithm is completely independent of the performance measure, and can be applied to a wide variety of selection criteria and fitness functions. Unlike alternative algorithms such as max–min [46], cluster analysis [22], binning [36], and stepwise elimination [33], which are tailored to a particular application, this approach is completely general, programmatically simple, and easily extensible. The remaining paragraphs describe in detail the various elements of this approach, using a mixed terminology borrowed from the simulated annealing and evolutionary programming literature.

States

A state or design represents a collection of subsets from one or more chemical libraries. The system was designed to allow the simultaneous selection of multiple subsets from multiple collections, thus enabling the design of experiments that span multiple chemistries and/or corporate files. Depending on the nature of the parent collection, three types of subsets can be defined: *sparse arrays* (or singles), *full arrays* (or simply arrays), and *plates*.

Sparse arrays

A sparse array represents any conceivable subset of k compounds from an n-membered collection. This is the most general design and is not restricted by the nature or location of the compounds, or by the types of reagents required for their synthesis. The term *array* originates from the literature of combinatorial chemistry, and refers to a subset of compounds from a combinatorial library that does not necessarily represent all possible combinations of the selected building blocks. This type of selection, illustrated in **Figure 2(a)**, is formally defined as

$$S \subseteq C,\ |S| = k,\ |C| = n, \tag{1}$$

where C denotes the parent collection. The size of the state space, i.e., the number of different k-membered subsets of an n-membered set, is given by the binomial coefficient

$$\binom{n}{k} = \frac{n!}{(n-k)!k!}. \tag{2}$$

For example, the selection of 50 from a set of 1000 compounds involves 10^{85} possibilities.

Full arrays

A full array is applicable only to combinatorial libraries and refers to a subset of compounds that represent all of the products derived by combining a given subset of building blocks in all possible combinations prescribed by the reaction scheme. Note that in this context, the term *full array* does not necessarily refer to the physical layout and execution of the experiment. A full array for a hypothetical two-component library, illustrated in **Figure 2(b)**, is defined as

$$S = S_1 \times S_2 \times \ldots \times S_R,\ S_i \subseteq R_i,\ |S_i| = k_i,\ |R_i| = n_i, \tag{3}$$

where R_i represents the pool of reagents at the ith variation site of the combinatorial library, and n_i and k_i are respectively the total and selected number of reagents at that site.

The combinatorics of full and sparse arrays are vastly different. For sparse arrays, the number of states that one has to consider (the number of different k-subsets of an n-set) is given by Equation (2), whereas for full arrays the number of possibilities (i.e., the number of different $k_1 \times k_2 \times \ldots \times k_R$ arrays derived from an $n_1 \times n_2 \times \ldots \times n_R$ R-component combinatorial library) is

$$\prod_{i=1}^{R} \frac{n_i!}{(n_i - k_i)!k_i!}. \tag{4}$$

For a 10 × 10 two-component combinatorial library, there are 10^{25} different subsets of 25 compounds, and only 63504 different 5 × 5 arrays. For a 10 × 10 selection from a 100 × 100 library, those numbers increase to 10^{241} and 10^{26} for sparse and full arrays, respectively. Thus, full arrays are much simpler to design and easier to synthesize on robotic hardware. In fact, most combinatorial libraries are synthesized in (full) array format.

Plates

The final type of selection is a collection of plates. A plate represents a set of compounds grouped together according to their physical location or some other common characteristic. The primary use of this class is to enable the selection of plates from a large archive for replication and screening against one or more biological targets. A plate subset is defined as

$$S \subseteq G, |S| = k_p, |G| = n_p, \tag{5}$$

where G represents the collection of plates, and n_p and k_p are respectively the total and selected number of plates in the collection. Note that the number of compounds in the plates need not be the same; in this case the selection criteria must be defined in a manner that does not favor the selection of plates that contain either too few or too many compounds, unless it is so desired.

Criteria

A selection criterion is a function that encodes the ability of a given set of compounds to satisfy a particular design objective such as maximum intrinsic diversity, similarity to a set of known actives, enforcement of particular property distributions, and many others. These functions are simple numerical indices that can be combined into a single objective function that measures the overall quality of a candidate design, S:

$$f_o(S) = f[f_1(S), f_2(S), \cdots, f_n(S)]. \tag{6}$$

The objective function f_o can assume any desired functional form, and by convention it is maximized to produce the optimal set. The following paragraphs describe the selection criteria that are most commonly used in combinatorial library design. Note that many of these functions can be defined in a multitude of ways, and some alternative definitions are discussed in Section 4.

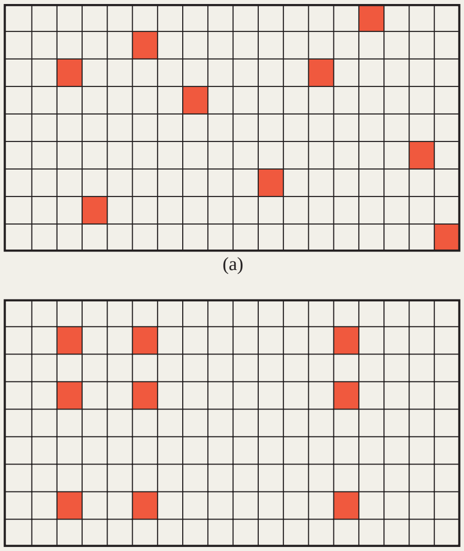


Figure 2

Combinatorial subsets: (a) sparse arrays; (b) full arrays.

Similarity

The similarity of a given set of compounds, S, to a set of leads is defined as a function of the average distance of a compound to its nearest lead:

$$S(S) = \frac{1}{k} \sum_{i=1}^{k} f[\min_{j=1}^{l} (d_{ij})], \tag{7}$$

where k is the cardinality of S, l is the number of leads, d_{ij} is the distance between the ith compound and the jth lead in some molecular descriptor space, and f is a user-defined function known as a *kernel*. The default kernel is the identity function. Since a higher similarity score typically indicates a collection of compounds that are more distant and therefore less similar to the leads, focused libraries are obtained by minimizing S.

Diversity

The intrinsic diversity of a set of compounds, S, is defined as a function of the average nearest-neighbor distance [43–45]:

$$D(S) = \frac{1}{k} \sum_{i} f[\min_{j \neq i} (d_{ij})], \tag{8}$$

where again k is the cardinality of S, d_{ij} is the Euclidean distance between the ith and jth compounds in S in some

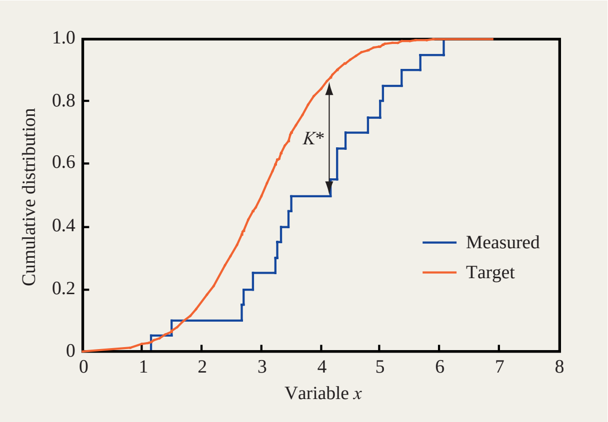


Figure 3

Kolmogorov–Smirnov statistic for computing the difference between a target cumulative distribution function (red curve) and a cumulative distribution obtained from measurements.

molecular descriptor space, and f is a user-defined kernel. The kernel is used to tailor the diversity of the design, and defaults to the identity function. Since typically the value of this function increases with spread, diverse libraries are obtained by maximizing D. We have found that this function is smoother than the more commonly used minimum dissimilarity and can discriminate more effectively among the various ensembles. Naively implemented, Equation (8) requires $k(k - 1)/2$ distance computations and scales adversely with the number of compounds selected. To reduce the quadratic complexity of the problem, D is computed using the k–d tree algorithm presented in [12]. This algorithm achieves computational efficiency by first organizing all of the points in C in a k-dimensional tree and then performing a nearest-neighbor search for each point using a branch-and-bound approach. For a relatively small number of dimensions, this algorithm exhibits $k \log k$ time complexity and scales favorably with the number of compounds selected. Other diversity functions are discussed in Section 4.

Complementarity

This criterion is closely related to diversity and represents the ability of a particular design to fill in the "diversity voids" that exist in a pre-existing collection. Its definition is similar to Equation (8):

$$D(S, S^*) = D(S \cup S^*) = \frac{1}{k} \sum_{i} f[\min_{j \neq i}(d_{ij})], \tag{9}$$

where i and j are now used to index the compounds in the combined set $S \cup S^*$, and $k = |S| + |S^*|$. Just as with Equation (8), complementary designs are obtained by maximizing D. Complementarity represents the sum of the intrinsic and extrinsic diversity of a set of compounds with respect to a reference collection.

Confinement

This criterion measures the degree to which the properties of a given set of compounds fit within prescribed limits; it is defined as

$$P(S) = \frac{1}{k} \sum_{i} \sum_{j} \max(x_j^{\min} - x_{ij}, x_{ij} - x_j^{\max}, 0), \tag{10}$$

where x_{ij} is the jth property of the ith compound, and $x_j^{\min}$ and $x_j^{\max}$ are respectively the minimum and maximum allowed limits of the jth property. Since the value of this function increases as more and more compounds fall outside the desired property range, constrained libraries are obtained by minimizing P. When multiple properties are used, they must be normalized to allow meaningful comparisons. In the special case in which the properties of interest must attain a particular target value (i.e., in the case of a degenerate range), Equation (10) can be rewritten as

$$P(S) = \frac{1}{k} \sum_{i} \sum_{j} \operatorname{abs}(x_{ij} - x_j^*), \tag{11}$$

where x_j^* represents the target value of the jth property. The confinement criterion was introduced in Reference [41].

Distribution

This criterion is used to create designs that obey certain predefined molecular property distributions. It is based on the Kolmogorov–Smirnov statistic, which 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:

$$K^* = \max_{-\infty < x < \infty} |P(x) - P^*(x)|, \tag{12}$$

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. For a set of k points x_i, $i = 1, \cdots, k$, $P(x)$ represents the fraction of data points to the left of a given value x (inclusive). The method is illustrated in **Figure 3**.

Unlike the more commonly used χ^2 test, the Kolmogorov–Smirnov statistic does not require binning of the data, which is arbitrary and leads to loss of information. More significantly, the function can be computed rapidly, 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). Speed of computation is particularly important in the application at hand, for which the fitness function must be evaluated tens of thousands of times in the course of the optimization.

The significance level of a particular value of K^* is a function of K^* and the number of data points, k. This function is relatively slow to compute, but when k is constant, it is a monotonic function of K^*. Since all we want is to determine which experimental distribution is closer to the "ideal" distribution $P^*(x)$, the significance level need not be computed.

The Kolmogorov–Smirnov criterion as defined by Equation (12) is a measure of dissimilarity and assumes values in the interval [0, 1]. Alternatively, we can define the similarity between two probability distributions, K:

$$K = 1 - K^*. \tag{13}$$

Thus, designs that obey a particular distribution function are obtained by maximizing K. This criterion was introduced by Rassokhin and Agrafiotis in Reference [19].

Activity

A common goal in combinatorial library design is to produce arrays of compounds that are predicted to be highly active against a predefined target according to some quantitative structure-activity or receptor binding model. This can easily be accomplished using the average predicted activity of the selected compounds, S:

$$Q_{\mathrm{a}}(S) = \frac{1}{k} \sum_i a_i , \tag{14}$$

where a_i is some measure of the predicted activity of the ith compound in S. Since the value of Q_{a} increases as the compounds become more active, focused libraries are obtained by maximizing Q_{a}.

Selectivity

A function similar to Equation (14) can be used to measure the selectivity against a set of biological targets:

$$Q_{\mathrm{s}}(S) = \frac{1}{k} \sum_i [a_{iq} - \max_{j \neq q}(a_{ij})], \tag{15}$$

where a_{ij} is the predicted activity of the ith compound against the jth target, and q is the target against which the molecules should be selective. Since the value of Q_{s} increases as the compounds become more selective, selective libraries are obtained by maximizing Q_{s}.

Overlap

This criterion measures the extent of overlap of a given selection, S, to another set of compounds, S^*,

$$O(S, S^*) = 1 - \frac{|S \cap S^*|}{|S|}, \tag{16}$$

where $|S|$ denotes the cardinality of S. Since the value of O decreases with the number of duplicates, nonredundant libraries are obtained by maximizing O. The most typical use of this criterion is to enable the selection of compounds in array or plate format that have not been previously synthesized or screened, and for the selection of multiple arrays from the same library (see below).

Optimization

As previously mentioned, the combinatorial nature of the problem does not permit an exhaustive enumeration of every possible combination in order to identify the optimal solution. While for simple cost functions several very effective greedy algorithms can be employed [41, 42], arbitrary objective functions have unpredictable surfaces with many local minima, and require a stochastic approach that is in principle suitable for identifying global minima. The method chosen here is based on simulated annealing. Simulated annealing is a global, multivariate optimization technique based on the Metropolis Monte Carlo search algorithm. The method starts from an initial random state and walks through the state space associated with the problem of interest by means of small, stochastic steps. In the problem at hand, a state is defined as a particular selection of compounds (i.e., a list of subsets from one or more virtual collections), and a step is defined as a small change in the composition of that set (i.e., replacement of a small fraction of the compounds comprising the set). An objective function, f_{o}, maps each state to a real value which represents its energy or fitness. While downhill transitions are always accepted, uphill transitions are accepted with a probability that is inversely proportional to the energy difference between the two states. This probability is computed using the Metropolis acceptance criterion,

$$p = e^{-\Delta E / K_{\mathrm{B}} T}, \tag{17}$$

or Felsentein's function,

$$p = \frac{1}{1 + e^{-\Delta E / K_{\mathrm{B}} T}}. \tag{18}$$

The latter ensures that the transition probability never exceeds 0.5, thus prohibiting the system from performing random walks. Boltzmann's constant, K_{B}, is used for scaling purposes, and T is an artificial temperature factor that controls the ability of the system to overcome energy barriers. The temperature is systematically adjusted during the course of the simulation in a manner that gradually reduces the probability of high-energy transitions. This protocol results in two optimization phases: one in which the system explores the state space relatively freely, and one in which it equilibrates around a low-energy minimum.

For simulated annealing to work, it is imperative that the transition probability be properly controlled. The difficulty with the problem at hand is that the cost function is not known *a priori* and can vary dramatically from problem to problem depending on the nature and weights of the criteria involved. To circumvent the problem of selecting an appropriate value for Boltzmann's constant, we use an adaptive approach in which K_B is not a true constant, but rather is continuously adjusted during the course of simulation on the basis of a running estimate of the mean transition energy. In particular, at the end of each transition, the mean transition energy is updated, and the value of K_B is adjusted so that the acceptance probability for a mean uphill transition at the final temperature is some predefined small number (usually 0.1%). The temperature is reduced using a Gaussian cooling schedule with a half-width of 5–10 deviation units. Other cooling schedules, such as linear, exponential, and Cauchy, have also been tested; in general, schedules that involve more extensive sampling at lower temperatures seem to perform best, although it is equally important that sufficient time be spent at higher temperatures in order that the algorithm not become trapped in local minima. The following sections provide a formal description of the serial and parallel implementations of this algorithm.

Serial implementation

Let S_i denote the ith subset requested, C_i the collection from which it is drawn (or any subset thereof), n_S the total number of subsets in the design, $\mathbf{T}$ the vector of temperatures in the cooling schedule, n_T the number of temperature cycles, n_C the number of sampling steps per temperature cycle, $f_o(.)$ the multiobjective fitness function, and $\overline{\Delta E}$ the average uphill transition energy (fitness). Also, let R_{ij} be the pool of available building blocks at the jth variation site if C_i is a combinatorial library, S_{ij} the selected building blocks at the jth variation site if S_i is a full array, and P_i and n_p be the set and number of available plates in C_i (if applicable). The serial annealing algorithm involves the following steps:

1. Initialize each subset, S_i, $i = 1, \cdots, n_S$, at random, and set $S = \{S_i, i = 1, \cdots, n_S\}$. If S_i is a sparse array and k is the requested number of compounds, initialize S_i with a random subset of k compounds from C_i. If S_i is a full array and k_j is the requested array size at the jth variation site, initialize each S_{ij} with a random subset of k_j reagents from R_{ij}. If S_i is a plate selection and k is the number of plates requested, initialize S_i with a random subset of k plates from P_i.
2. Set $f = f_o(S)$, $f_{\min} = f_o(S)$, and $\overline{\Delta E} = 0$.
3. Perform steps 4–9 for each $t \leq n_T$.
4. Set $T = \mathbf{T}[t]$.
5. Perform steps 6–9 for each $c \leq n_C$.
6. Select a random subset S_i, $i \in [1, n_S]$, mutate it, and denote the resulting state as S^*. If S_i is a sparse array, mutate it by replacing a small fraction of randomly chosen compounds in S_i with an equal number of randomly chosen compounds in $\overline{S_i}$, where $\overline{S_i}$ denotes the complement of S_i (i.e., the compounds in C_i that are not in S_i). If S_i is a full array, mutate it by selecting a variation site, j, at random, and replacing a randomly chosen reagent in S_{ij} with a randomly chosen reagent in $\overline{S_{ij}}$. Finally, if S_i is a plate subset, mutate it by replacing a randomly chosen plate in S_i with a randomly chosen plate from $\overline{S_i}$.
7. Set $f^* = f_o(S^*)$ and $\Delta E = |f - f^*|$.
8. Update $\overline{\Delta E}$ and set $K_B = (-\overline{\Delta E}/T \ln a)$, where a is a predefined small number in the interval [0, 1].
9. If $f^* \leq f$ or if $f^* > f$ and $r < e^{-\Delta E/K_B T}$, where r is a random number in the interval [0, 1], then
    - 9.1. Set $S = S^*$ and $f = f^*$.
    - 9.2. If $f < f_{\min}$, set $f_{\min} = f$ and $S_{\min} = S$.
10. Output $S_{\min}$ and $f_{\min}$.

To avoid repeated and potentially expensive memory allocation, S and S^* are implemented as a single object that is able to revert to its previous state after an unfavorable mutation. Moreover, the mean uphill transition energy $\overline{\Delta E}$ is computed based on the last w uphill transitions, where w is a preset window (usually a few hundred steps).

Parallel implementation

The parallel algorithm described in this section is known as synchronous annealing and is designed to keep interprocess communication and thread synchronization to a minimum. As in conventional annealing, the process begins with a random initial state and proceeds through the state space by a series of small stochastic steps. However, during each temperature cycle each execution thread is allowed to follow its own independent Monte Carlo trajectory. The threads synchronize at the end of each cycle, and the best among the last states visited by each thread is recorded and used as the starting point for the next iteration. Given sufficient simulation time, this parallel algorithm produces results that are comparable to those obtained with the traditional serial implementation (see below). The algorithm proceeds as follows:

1. Initialize each subset, S_i, $i = 1, \cdots, n_S$, at random, and set $S = \{S_i, i = 1, \cdots, n_S\}$. If S_i is a sparse array and k is the requested number of compounds, initialize S_i with a random subset of k compounds from C_i. If S_i is a full array and k_j is the requested array size at the jth variation site, initialize each S_{ij} with a random subset of k_j reagents from R_{ij}. If S_i is a plate selection and k is the number of plates requested, initialize S_i with a random subset of k plates from P_i.
2. Set $f = f_o(S)$, $f_{\min} = f_o(S)$, and $\overline{\Delta E} = 0$.
3. Perform steps 4–12 for each $t \leq n_T$.
4. Set $T = \mathbf{T}[t]$.
5. Perform steps 6–11 for each $p \leq n_p$, where n_p is the number of execution threads (processors).
6. Set $S_p = S$, $f_p = f$, $S^p_{\min} = S$, $f^p_{\min} = f$, and $\overline{\Delta E}_p = \overline{\Delta E}$.
7. Perform steps 8–11 for each $c \leq n_C$.
8. Select a random subset S_i, $i \in [1, n_S]$, mutate it, and denote the resulting state as S^*. If S_i is a sparse array, mutate it by replacing a small fraction of randomly chosen compounds in S_i with an equal number of randomly chosen compounds in $\overline{S_i}$. If S_i is a full array, mutate it by selecting a variation site, j, at random, and replacing a randomly chosen reagent in S_{ij} with a randomly chosen reagent in $\overline{S_{ij}}$. Finally, if S_i is a plate subset, mutate it by replacing a randomly chosen plate in S_i with a randomly chosen plate from $\overline{S_i}$.
9. Set $f^* = f_o(S^*)$ and $\Delta E = |f_p - f^*|$.
10. Update $\overline{\Delta E}_p$ and set $K^p_B = (-\overline{\Delta E}_p / T \ln a)$, where a is a predefined small number in the interval [0, 1].
11. If $f^* \leq f_p$ or if $f^* > f_p$ and $r < e^{-\Delta E / K^p_B T}$, where r is a random number in the interval [0, 1], then
    - 11.1. Set $S_p = S^*$ and $f_p = f^*$.
    - 11.2. If $f_p < f^p_{\min}$, set $f^p_{\min} = f_p$ and $S^p_{\min} = S_p$.
12. Set $f_{\min} = \min_p f^p_{\min}$, $S_{\min} = S^q_{\min} : S^q_{\min} \leq S^p_{\min}\ \forall\, p \neq q$, $f = \min_p f_p$ and $S = S_q : S_q \leq S_p\ \forall\, p \neq q$.
13. Output $S_{\min}$ and $f_{\min}$.

The choice of simulated annealing was based on its programmatic simplicity, the fact that the mutation function (or step in annealing terminology) can be designed in a way that guarantees the creation of valid states (something that requires extra care with genetic approaches), and in-house comparative studies which demonstrated superior convergence compared to evolutionary approaches.

Filters

Filters limit the selection to specific subsets of a particular collection. These subsets can be specified as reagent lists, product lists, or plate lists. For array selections, product lists are deconvoluted to the respective reagents (note that in this case, if the input list is a sparse array, it is possible that some of the products in the final selection may not be part of the specified list). Filters can be used for a variety of purposes, such as restricting the selection to compounds having a particular substructure, reagents that are provided by reliable vendors, plates that have a good QC score or contain compounds of a particular structural class, etc.

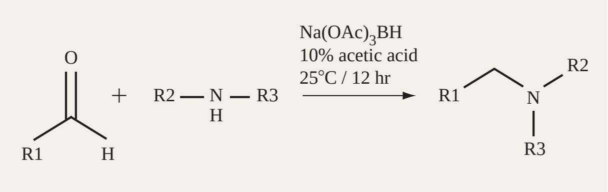


Figure 4

Synthetic sequence for the reductive amination library.

Computational details

All programs were implemented in the C++ programming language and are part of the DirectedDiversity [5] software suite. They are based on the 3-Dimensional Pharmaceuticals Mt++ class library[1] and are designed to run on all POSIX**-compliant UNIX** and Microsoft Windows** platforms. Parallel execution on systems with multiple CPUs is supported through the multithreading classes of Mt++. All calculations were carried out on a Dell workstation equipped with two 800-MHz Intel Pentium** III processors running Windows 2000 Professional. Selections were carried out in 30 temperature cycles using a Gaussian cooling schedule and 1000 sampling steps per temperature cycle. Boltzmann's constant was determined in an adaptive manner so that the acceptance probability for a mean uphill transition at the final temperature was 0.1%.

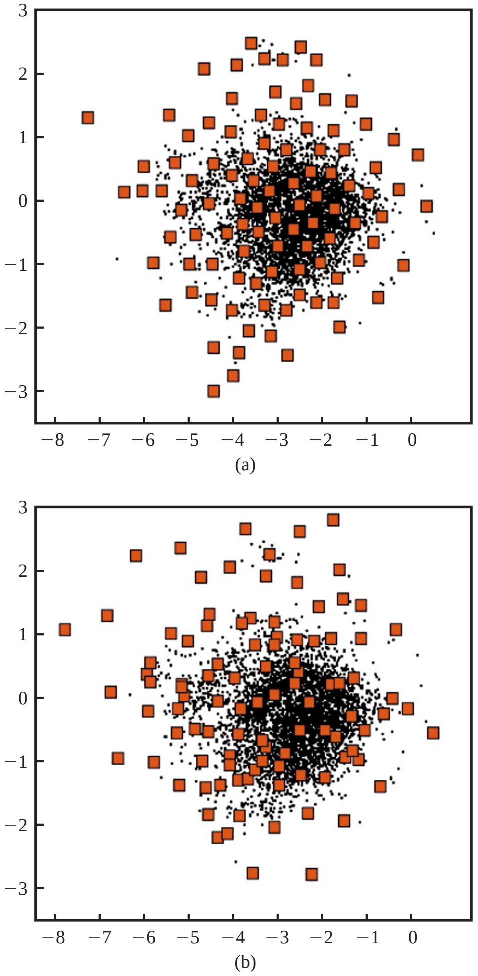
3. Datasets

Two datasets were used in this study. The first was a two-component virtual combinatorial library based on the reductive amination reaction, and the second was a subset of the 3-Dimensional Pharmaceuticals probe library, a collection of more than 250000 diverse compounds that represent more than 30 different structural classes.

The two-component reductive amination library is part of a synthetic strategy that exploits the pivotal imine intermediate and is utilized for the construction of structurally diverse druglike molecules with useful pharmacological properties, particularly in the GPCR superfamily [47]. The synthetic sequence is illustrated in **Figure 4**. This library was used in a number of previous studies and represents an internal standard for testing

---

[1] Copyright © 3-Dimensional Pharmaceuticals, Inc., 1994–2000.



(a)

(b)

Figure 5

Selection of 100 compounds from the reductive amination library based on maximum diversity on the two-dimensional nonlinear map: (a) sparse array selection; (b) full array (10 × 10) selection.

new library design methodologies. A set of 300 primary and secondary amines with 300 aldehydes were selected at random from the Available Chemicals Directory[2] and were used to generate a virtual library of 90000 products using the library enumeration classes of the DirectedDiversity toolkit.[3] These classes take as input lists of reagents supplied in SDF or Smiles format, and a reaction scheme written in a proprietary language that is based on Smarts and an extension of the scripting language Tcl. All chemically feasible transformations are supported, including multiple reactive functionalities, different stoichiometries, cleavage of protecting groups, stereospecificity, and many others. The computational and storage requirements of the algorithm are minimal (even a billion-membered library can be generated in a few seconds on a personal computer) and scale linearly with the number of reagents.

Each compound in the 90000-member library was characterized by a standard set of 117 topological descriptors [48, 49] computed with the DirectedDiversity toolkit. These descriptors include an established set of topological indices with a long, successful history in structure-activity correlation such as molecular connectivity indices, kappa shape indices, subgraph counts, information-theoretic indices, Bonchev–Trinajstic indices, and topological state indices. It has previously been shown that these descriptors exhibit proper "neighborhood behavior" [24] and are thus well suited for diversity analysis and similarity searching [22, 23, 25, 50].

The 117 molecular descriptors were subsequently normalized and decorrelated using principal component analysis. This process resulted in an orthogonal set of 23 latent variables, which accounted for 99% of the total variance in the data. To simplify the analysis and interpretation of results, this 23-dimensional dataset was further reduced to a two-dimensional set using a very fast nonlinear mapping algorithm developed by our group [51–54]. The projection was carried out in such a way that the pairwise distances between points in the 23-dimensional principal component space were preserved as much as possible on the two-dimensional map. The resulting map had a Kruskal stress of 0.187 and was used to perform and visualize the selections. The PCA preprocessing step was necessary in order to eliminate duplication and redundancy in the data, which is typical of graph-theoretic descriptors.

Finally, in addition to the 117 topological descriptors, the octanol–water partition coefficient (logP) of each compound was computed independently using the Ghose–Crippen approach [55] as implemented in the DirectedDiversity toolkit, and was used as the target variable for property-based designs (see below). This parameter was not included in the descriptor set used for similarity and diversity assessment.

The second dataset was a subset 3-Dimensional Pharmaceuticals probe library containing nearly 250000 compounds arranged in 96-well plate format containing 88 compounds each plate. This library was described using a methodology similar to that used for the reductive amination library.

4. Results and discussion

Diverse libraries

Molecular diversity represents the most common method for designing combinatorial libraries [43–45]. Although

---

[2] MDL Information Systems, Inc., 140 Catalina St., San Leandro, CA 94577.
[3] Copyright © 3-Dimensional Pharmaceuticals, Inc., 1994–2000.

there has been much controversy regarding the choice of metrics and descriptors and their ability to discover novel leads, it is generally accepted that a proper diversity function should measure spread and be algorithmically efficient so that it can be applied to the kinds of datasets that are encountered in combinatorial library design.

In general, diversity metrics fall into three main categories: *distance-based* methods, which express diversity as a function of the pairwise molecular dissimilarities defined through measurement or computation; *cell-based* methods, which define it in terms of the occupancy of a finite number of cells that represent disjoint regions of chemical space; and *variance-based* methods, which quantify diversity on the basis of the degree of correlation between the molecules' pertinent features. In their vast majority, these metrics encode the ability of a given set of compounds to sample chemical space in an even and unbiased manner, and are used to produce space-filling designs that minimize the size of unexplored regions known as "diversity voids." A thorough discussion of the relative merits of the most commonly used diversity functions can be found elsewhere [43–45].

The selection of 100 compounds from the reductive amination library based on their average nearest-neighbor Euclidean distance on the two-dimensional nonlinear map [Equation (8)] is shown in **Figure 5(a)**. The compounds, which were selected as singles, are nicely distributed in the data space occupied by the virtual library, and do not exhibit any significant clustering that reflects the density distribution of the parent collection. However, this selection does suffer from one significant drawback: It requires 69 amines and 70 aldehydes, i.e., a total of 139 reagents. Obviously, the physical synthesis of these compounds would be extremely laborious, and this is the primary reason why sparse arrays are rarely used in combinatorial chemistry. This method is usually employed in compound retrieval and acquisition, particularly when automated, efficient cherry-picking techniques are available. Thus, the remaining discussion is focused primarily on full array selections.

Figure 5(b) shows the selection of an equivalent number of compounds in the form of a 10 × 10 array using the same distance function and diversity metric that was employed in the previous selection. Although the array is somewhat less diverse in terms of spread, it requires only 20 reagents as compared to the 139 reagents required by the singles, and therefore it is much easier to synthesize in practice. A look at the selected reagents (**Figure 6**) confirms that the design is also diverse in terms of chemical structure, since it consists of building blocks containing a wide variety of atom types, connectivity patterns, ring systems, and functional groups.

Some interesting aspects of the optimization algorithm are illustrated in Figures 7 through 9. **Figures 7** and **8** show the diversity score and the percentage of accepted uphill transitions at the end of each temperature cycle for a single annealing run. In Figure 7, green points indicate the diversity of the last accepted state, and red points the cost of the best state discovered at the end of each temperature cycle. Within the first 15 cycles, the algorithm is able to extract the gross features of the minimum and recover most of the diversity that is accessible by this array size. The final cycles are spent refining that

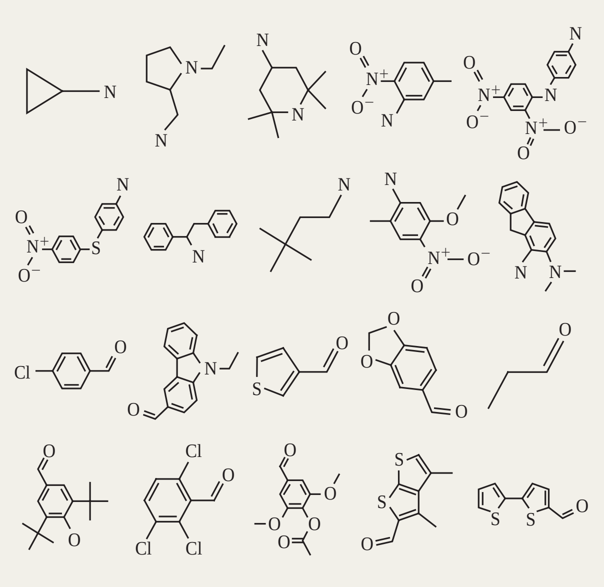


Figure 6

Amine and aldehyde reagents comprising the selection in Figure 5(b).

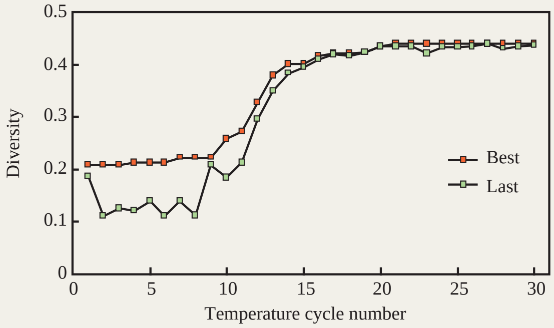


Figure 7

Diversity score as a function of time for the selection in Figure 5(b). "Best" and "Last" respectively represent the best and last state found at the end of each temperature cycle. Since simulated annealing allows uphill transitions, these two states are not always the same.

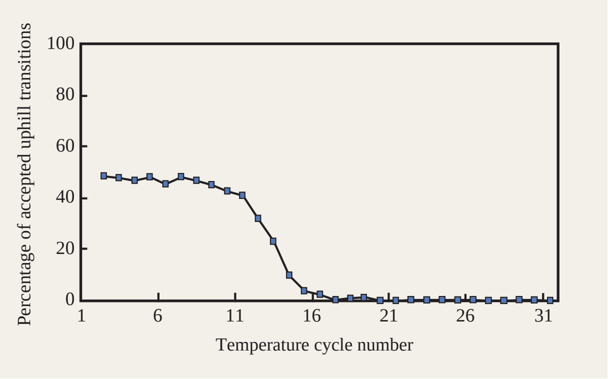


Figure 8

Percentage of accepted uphill transitions at the end of each temperature cycle for the selection in Figure 5(b).

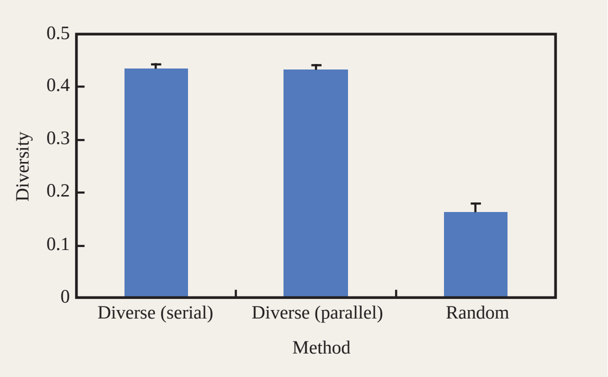


Figure 9

Mean and standard deviation of the diversity scores obtained by 50 independent selections of a maximally diverse 10 × 10 array from the reductive amination library, using the serial and parallel simulated annealing algorithms, plotted against the score of a randomly chosen array as a reference.

minimum, with relatively minimal improvements in the fitness function. The asymptotic convergence of the two curves manifests the decreasing ability of the Metropolis search algorithm to perform high-energy transitions, which is also reflected in the fraction of accepted uphill transitions in Figure 8. Indeed, at higher temperatures the algorithm is able to overcome substantial energy barriers, but this ability is diminished at lower temperatures, and the system is eventually frozen around the 20th cycle. This figure also shows that the adaptive determination of K_B appears to work reasonably well. This parameter need not be estimated by the user; rather, it is adjusted automatically by the algorithm as the simulation progresses and as the energy landscape is more thoroughly explored.

The method is also very robust. **Figure 9** shows the mean and standard deviation of the diversity scores obtained by 50 independent optimization runs carried out with the two algorithms, plotted against the diversity of a random array as a reference. The average scores obtained with the serial and parallel implementations were 0.436 and 0.433, respectively, with a standard deviation of only 0.007 in both cases. Thus, although the algorithm does converge to different local minima, depending on the starting configuration and random seed, the solutions are essentially equivalent in terms of quality, and more than sufficient for the purpose of exploring the chemical space. As for the parallel algorithm, it scales favorably with the number of CPUs (the elapsed execution time showed a nearly perfect linear relationship with respect to the number of processors used) and produces results that are virtually identical to those obtained with the serial algorithm.

A point that is worth noting relates to the computational advantages afforded by the use of the *k*–*d* tree algorithm for computing the diversity function of Equation (8) [12]. **Figure 10** shows the ratio between CPU time required by the naive and *k*–*d* tree algorithms in two dimensions as a function of the array size (the naive algorithm involves exhaustive enumeration of all pairwise distances in the data sample). It is clear that while *k*–*d* trees impose some trivial overhead for very low values of k, they scale superbly with k, resulting in enormous computational savings for large selections.

This margin, however, decreases in higher dimensions. For example, the scaling factor (i.e., the increase in the CPU time when the selection size is doubled) is 2.04 for 2D, 2.31 for 3D, 2.38 for 4D, and 2.98 for 10D, a trend shown graphically in **Figure 11**. For low dimensions (≤ 5), the scaling factor is less than 2.5, which is consistent with the reported theoretical value of $n \log n$. However, for ten dimensions the time complexity is somewhere between linear and quadratic. This performance decline is likely to continue with increasing dimensionality, and the algorithm will eventually become quadratic, with the added overhead of constructing and traversing the tree. Fortunately, our experience has shown that most high-dimensional descriptor spaces can be reduced to a relatively small number of dimensions with minimal distortion through the use of efficient nonlinear mapping techniques [52–54].

Nevertheless, there is a point at which the use of the diversity function of Equation (8) does become prohibitively expensive, no matter what algorithm is used to compute the nearest neighbors. 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 [56]. This approach 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 intermolecular 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 described in Section 2. Departure from uniformity induces sampling redundancy and formation of clusters, and thus results in diminishing diversity (see **Figure 12**). The distinct advantage of this approach is that the cumulative distribution can easily be 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 [56], the function produces results that are consistent with our notion of spread and can do so in a fraction of the time required by alternative methodologies. The selection of a 10 × 10 array using our novel diversity function is shown in **Figure 13**. 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.

A common criticism of the diversity function of Equation (8) is that it tends to favor the extremes of the feature space and produce designs containing unusual compounds of limited pharmaceutical interest. Although in theory one can exclude “peculiar” reagents from the selection through the use of appropriate filters (see Section 2) or even from the virtual library itself, a “softer” and more dynamic approach is to adjust the kernel function. A simple way to control the volume of the design without affecting its internal spread is to request that the mean nearest-neighbor distance of the selected compounds be equal to some preset value. Consider, for example, the design of Figure 5(b). The score of this array is 0.43, and represents the maximum possible nearest-neighbor intermolecular dissimilarity for any design carried out in a 10 × 10 format. Thus, by maximizing the function

$$D(S) = \frac{1}{k} \sum_{i} - \text{abs}\,[a - \min_{j \neq i}(d_{ij})], \tag{19}$$

where a is set to 0.25, one can design an array that occupies roughly half of the volume of the original selection yet is still internally diverse (**Figure 14**). The user can choose alternative values of a to produce designs with lesser or greater spread. Incidentally, Figure 14 also illustrates a known disadvantage of Equation (8) as a general measure of molecular diversity, and that is the fact that it measures only the intracluster (nearest-neighbor) distances and does not take into account the intercluster separations. As a result, designs comprising two or more internally diverse clusters cannot be easily distinguished, regardless of their relative separation.

The diversity function outlined above can also be used to design libraries that fill in the diversity voids of one or more pre-existing collections. A common problem faced by many pharmaceutical companies is the enhancement of their corporate collection through the synthesis or acquisition of chemical libraries that increase the diversity of their existing archives. Equation (9) provides a simple

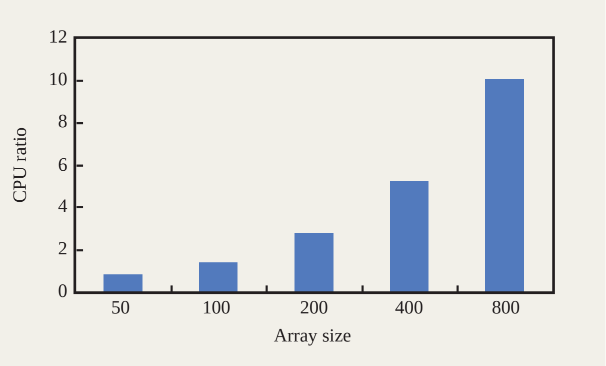


Figure 10

CPU ratio vs. array size if the *k–d* tree algorithm is used for the selection of a maximally diverse array from the reductive amination library.

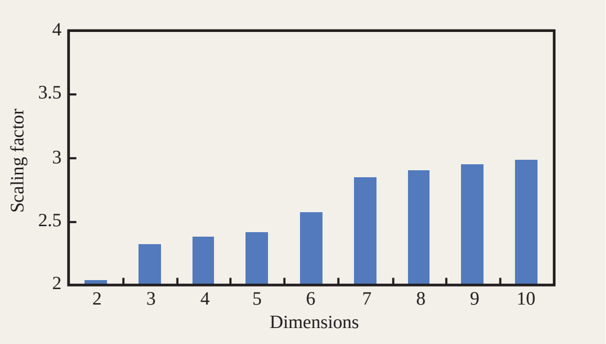


Figure 11

Scaling of the *k–d* tree algorithm: Scaling factor (the CPU ratio required for the selection of 1600 vs. 800 compounds in array format) vs. dimensionality. A value of 2 indicates $O(n)$ complexity, whereas a value of 4 indicates $O(n^2)$ complexity.

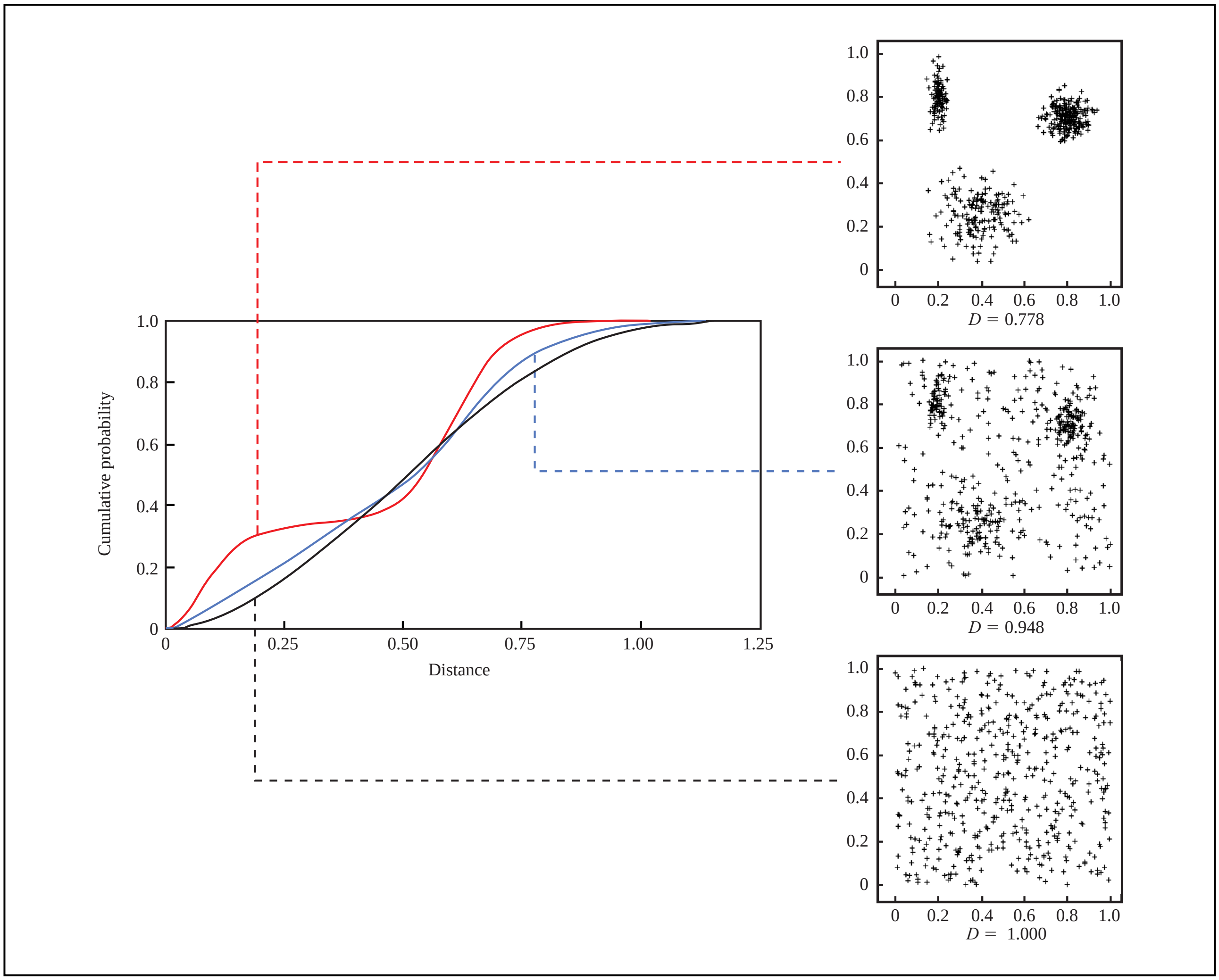


Figure 12

Effect of clustering on the shape of the cumulative probability distribution of pairwise distance for an artificial two-dimensional dataset confined in a unit square. From [56], reproduced with permission.

and straightforward solution to this problem. By maximizing the diversity of the combined set, we can ensure that the design is both internally and externally diverse. The selection of a second 10 × 10 array that complements the diversity of the original selection in Figure 5(b) is shown in **Figure 15**. The two arrays have only one amine and two aldehydes (i.e., only two products) in common. Note that in terms of scatter the design is not "perfect"; this is due to the array constraint rather than the diversity metric itself.

Druglike libraries

In all of the preceding examples, the selections were based exclusively on molecular diversity, and no attention was paid to the druglike qualities of the resulting compounds. Experience has shown that selecting compounds from a virtual library solely on that basis, no matter what diversity measure is used, often results in combinatorial libraries with poor pharmacokinetic properties or other undesirable characteristics. Recently there have been several attempts to quantify druglike qualities and incorporate them directly into the design process. Martin et al. [31] presented a reagent selection algorithm based on D-optimal design, wherein the candidate reagents were assigned to categorical bins according to their properties, and successive steps of D-optimal design were performed to generate diverse substituent sets consistent with required membership quotas from each bin. This

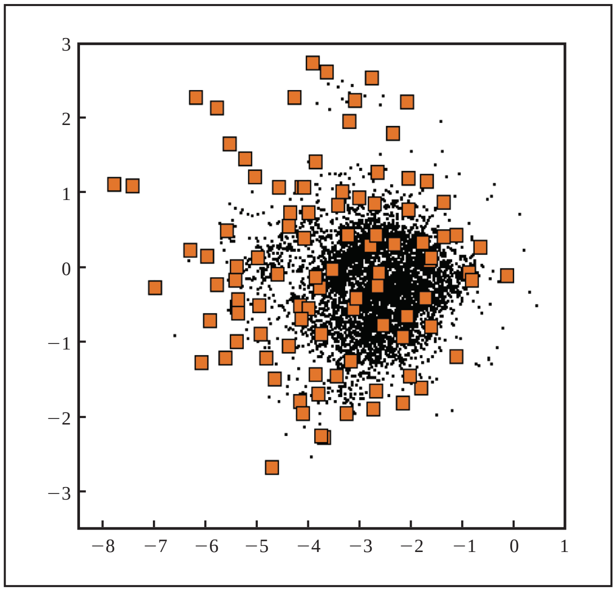


Figure 13

Selection of a maximally diverse 10 × 10 array from the reductive amination library based on the Kolmogorov–Smirnov measure of molecular diversity, by means of the diversity function depicted in Figure 12.

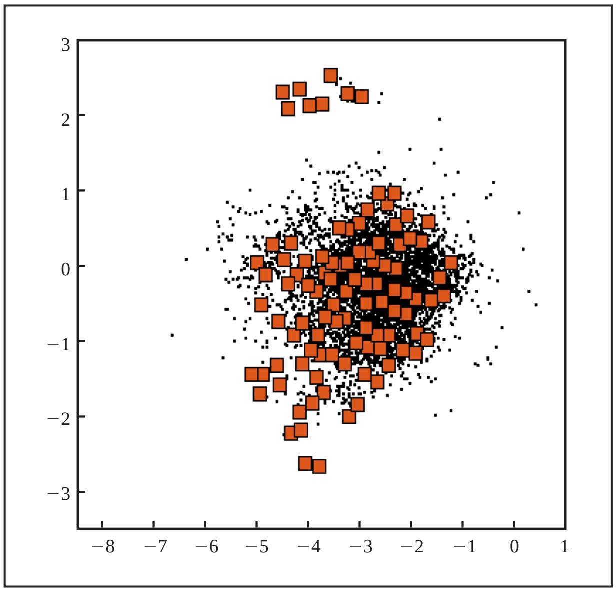


Figure 14

Selection of 100 compounds from the reductive amination library in 10 × 10 array format based on the kernel function of Equation (19). The function is used to limit the volume of the design and avoid sampling the extreme regions of the feature space.

technique was later elaborated [57, 58], and a new "parallel" sampling approach was proposed in order to eliminate the order-dependence of the original algorithm. Most recently, Koehler et al. [59] proposed a multipass algorithm designed to facilitate the addition of compounds to an existing chemical library. This method was intended to prioritize compounds that are most similar to a specified set of favorable target molecules, and, at the same time, most dissimilar to the compounds that reside in the library being augmented. The algorithm scores and ranks each compound in the external library according to the local density of similar compounds in the target and internal libraries. Density arising from the target library adds a positive contribution to the score, while density arising from the internal library adds a negative contribution. The highest-ranking compounds are then included in the internal library and the process is repeated until the specified number of compounds is selected.

Perhaps the simplest approach is to filter the candidate designs using Lipinski's "rule of 5," a simple heuristic which states that for compounds which are not substrates of biological transporters, poor absorption and permeation are more likely to occur when there are more than five H-bond donors, more than ten H-bond acceptors, the molecular weight is greater than 500, or the log *P* is greater than 5 [60]. Consider, for example, a maximally diverse 20 × 20 array from the reductive amination library shown in **Figure 16**. The molecular weight and log *P* distributions of these compounds are shown in

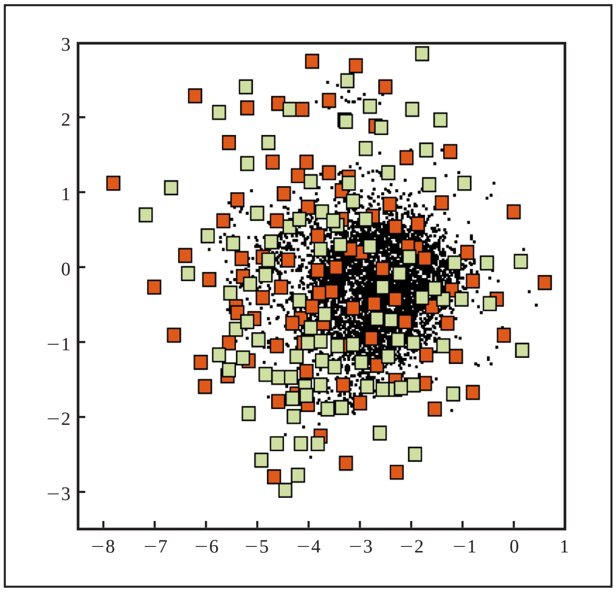


Figure 15

Selection of a 10 × 10 array that complements the diversity of the selection in Figure 5(b).

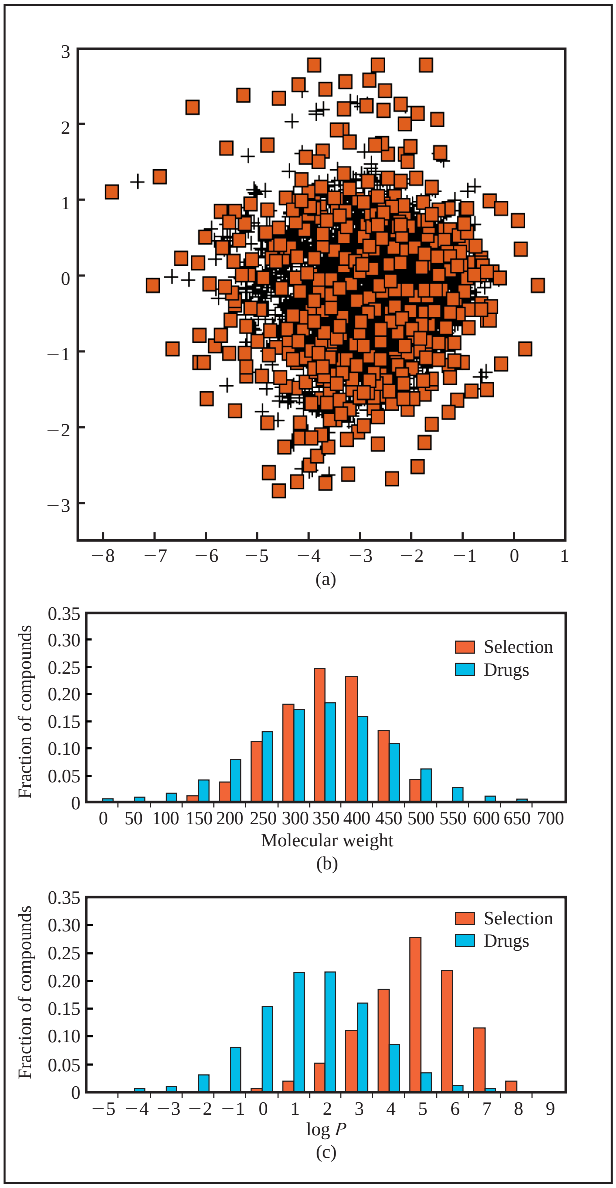


Figure 16

Selection of a maximally diverse 20 × 20 array from the reductive amination library: (a) nonlinear map; (b) molecular weight distribution; (c) log *P* distribution. In (b) and (c), the blue series shows the respective distributions of known drugs. From [19], reproduced with permission.

Figures 16(b) and 16(c), along with the respective distributions of known drugs derived by analyzing a subset of 7484 compounds from the World Drug Index, which was generously provided to us in electronic format by Dr. Christopher Lipinski of Pfizer, Inc. While the distribution of molecular weights matches closely that of the reference set, the log *P* distribution is shifted upward by nearly three log *P* units, with 125 out of 400 compounds (more than 30% of the selection) falling outside the boundaries of the Lipinski box.

In our multiobjective paradigm, correcting this adverse physicochemical profile can easily be accomplished in one of two different ways. The first is to combine the diversity criterion of Equation (8) with the confinement criterion of Equation (10) in order to penalize compounds that fall beyond the boundaries of the box. This can be achieved with an objective function of the form

$$f_o = D - 5 \cdot [0.01 \cdot P(mw) + P(\log P)], \tag{20}$$

where D represents the diversity of the ensemble, and $P(mw)$ and $P(\log P)$ are the values of the confinement criterion for molecular weight and log P, respectively. As with most problems of this type, the most difficult task is to assign a meaningful set of coefficients used to represent the weights of the individual objectives. In our case, the coefficients were determined empirically based on the maximum value that each criterion could assume independently for any given array size. In the case at hand, the coefficients were chosen on the basis of the relative scale of molecular weight and log P, and the value of the mean nearest-neighbor distance, D, of the maximally diverse 20 × 20 array. As shown in **Figure 17**, the resulting design is fully contained within the Lipinski box, while remaining sufficiently diverse.

However, given the probabilistic nature of the problem, perhaps a better approach would be to compare the actual distributions themselves, and penalize designs whose molecular weight and log P distributions depart from those of known drugs. As a numerical measure of the dissimilarity between two property distributions, we used the Kolmogorov–Smirnov criterion of Equation (13). Unlike the more commonly used χ^2 test, the Kolmogorov–Smirnov statistic does not require binning of the data [18], which is arbitrary and leads to loss of information. The distributions of known drugs are essentially normal, with a mean and sigma of 314.3 and 108.3 for molecular weight, and 1.04 and 1.78 for log P, respectively.

In order to enforce these distributions in the final design, we combined molecular diversity with the Kolmogorov–Smirnov statistic into the following objective function:

$$f_o = D + 0.2 \cdot K(\log P) + 0.2 \cdot K(mw), \tag{21}$$

where D is the diversity criterion defined in Equation (14), and $K(\log P)$ and $K(mw)$ are the Kolmogorov–Smirnov similarities between the log P and molecular weight distributions of the selected compounds and the reference World Drug Index set, respectively. Again, the coefficients were determined on the basis of the maximum values of the respective criteria, which were 0.18, 0.9, and 0.9 for diversity, and the molecular weight and log P distributions, respectively. These values suggested that for the three criteria to be placed on an equal footing, the value of K

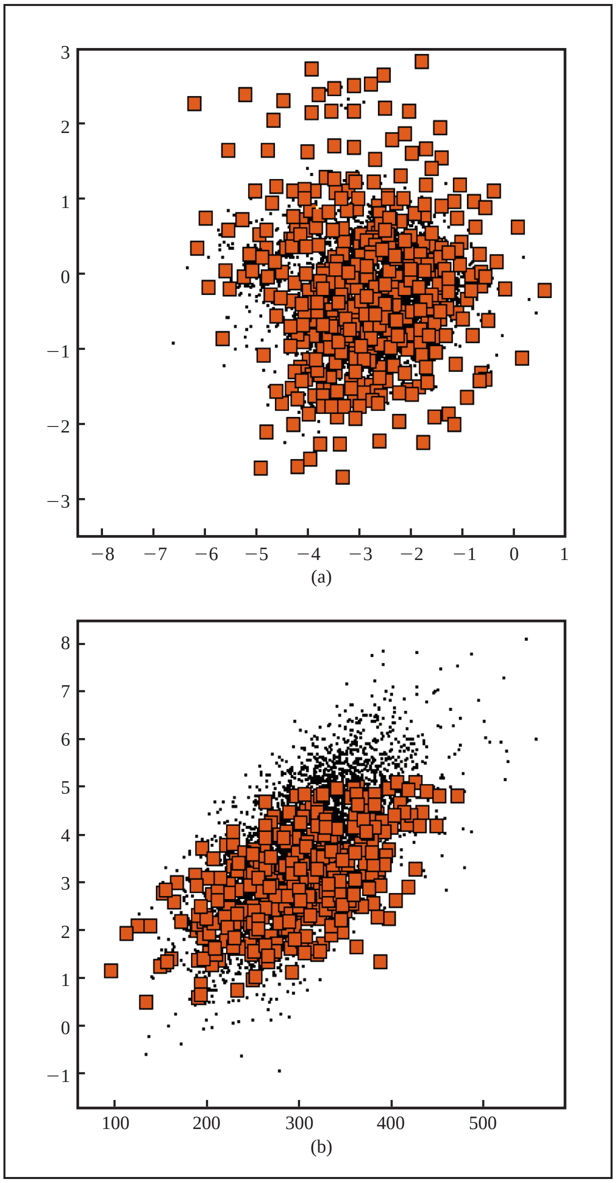


Figure 17

Selection of a maximally diverse 20 × 20 array that is confined within the boundaries of the Lipinski box: (a) nonlinear map; (b) Lipinski profile.

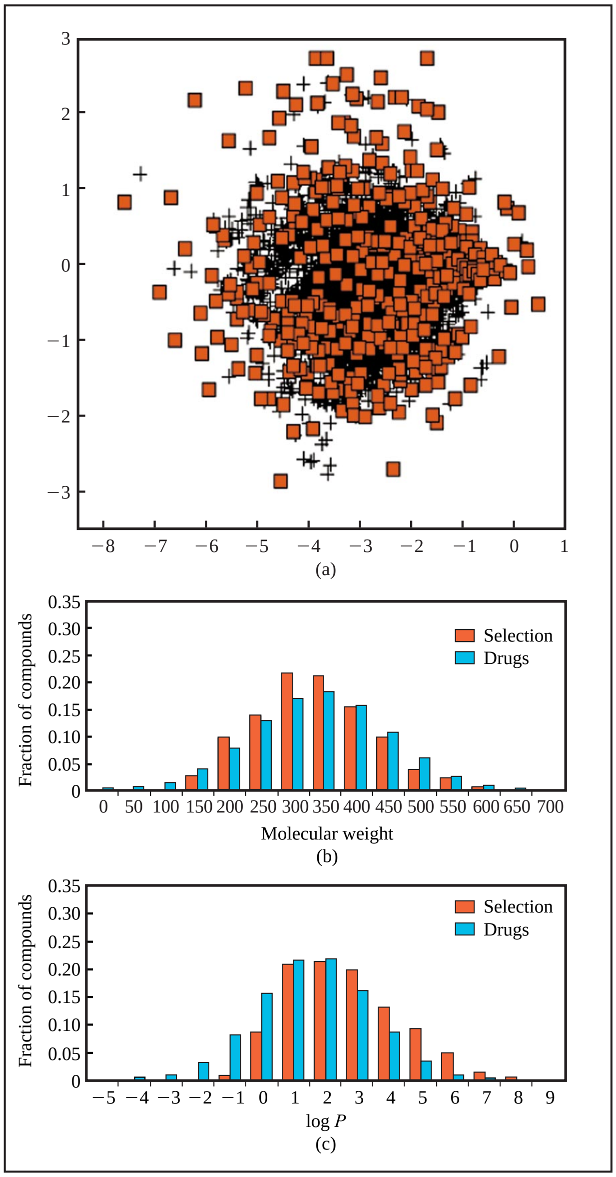


Figure 18

Selection of a maximally diverse 20 × 20 array that exhibits a druglike molecular weight and log *P* distribution: (a) nonlinear map; (b) molecular weight distribution; (c) log *P* distribution. In (b) and (c), the blue series shows the respective distributions of known drugs. From [19], reproduced with permission.

had to be scaled down by approximately a factor of 5. In pathological cases for which the energy landscapes (i.e., the distributions of scores) of the individual criteria are very different, alternative, more complex objective functions can be devised. As shown in **Figure 18**, when the selection is carried out using Equation (21), the molecular weight and log *P* distributions of the selected compounds approximate very nicely the respective distributions of known drugs, and this occurs without a significant impact on the diversity of the design [Figure 18(a)]. Note that just as in any problem of this kind, there is a point at which the various objectives begin to oppose one another: A step toward improving one of the objectives, increasing molecular diversity, is a step away from improving the other, increasing their druglike qualities.

The methodology outlined above is, of course, not limited to virtual collections. A common problem faced by

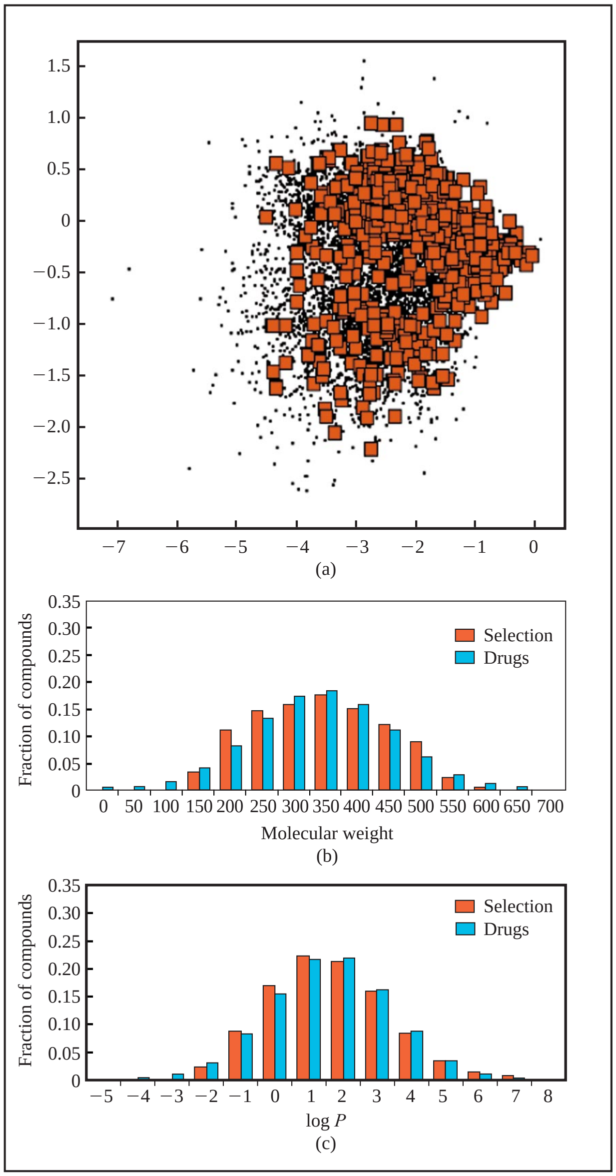


Figure 19

Selection of a maximally diverse set of 880 compounds from the 3-Dimensional Pharmaceuticals probe library. The selection was designed to exhibit druglike molecular weight and log *P* distributions, and represents the best set of ten 8 × 11 plates. (a) Nonlinear map; (b) molecular weight distribution; (c) log *P* distribution. In (b) and (c), the blue series shows the respective distributions of known drugs.

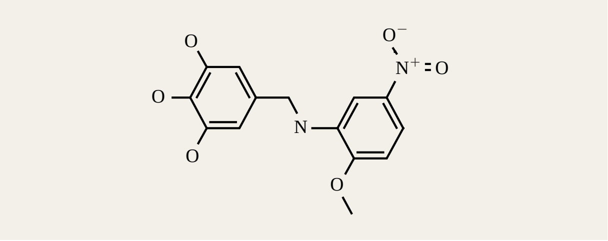


Figure 20

Lead compound used for similarity selections from the reductive amination library.

many pharmaceutical companies is the selection of a small number of compounds from their chemical banks for screening against a new biological assay. Compounds synthesized by combinatorial methods are usually stored as dimethyl sulfoxide (DMSO) solutions in 96-well plates. Extraction of individual samples from these plates can be laborious and time-consuming, and it is often much more efficient to replicate and screen the entire plate containing the compounds of interest. This, however, is a rather simplistic approach; as we demonstrated before, the quality of a particular design is determined by the collective properties of all of its constituent compounds, and the design can be vastly improved if the physical location or grouping of these compounds is taken into consideration. This is illustrated in the selection of ten plates from a subset of the 3-Dimensional Pharmaceuticals probe library, a collection of ~250000 diverse compounds representing more than 30 distinct structural classes. These compounds are stored in nearly 3000 96-well plates, containing 88 compounds each. The selection of ten plates containing 880 diverse, druglike compounds using an objective function similar to the one employed in the previous example is illustrated in **Figure 19**. Note that the requirement that these compounds exhibit a normal molecular weight distribution centered around 315 causes a noticeable shift of the selection to the right of the nonlinear map, which represents molecules of smaller molecular weight (size is a dominant factor in determining molecular similarity and therefore has a significant impact on the shape of the nonlinear map).

Focused libraries

Molecular diversity is typically employed in the design of exploratory combinatorial libraries for use in general screening. Once a promising hit has been identified and confirmed, subsequent iterations are used in an attempt to explore the structure-activity space around that compound and refine its pharmacological profile. This process involves two phases: The first is based on very sparse information and is guided primarily by molecular similarity, whereas the second involves the use of more rigorous structure-activity models which are derived from an active series with a broad dynamic range of biological activities. This task can easily be accomplished with the similarity, activity, and selectivity criteria described in Section 2. For reasons of brevity, the remaining discussion

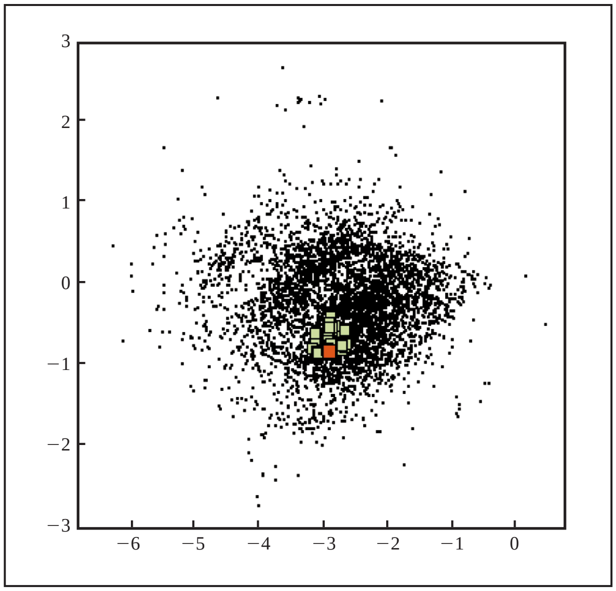


Figure 21

Selection of 100 compounds in 10 × 10 array format based on maximum similarity to the “lead” compound in Figure 20.

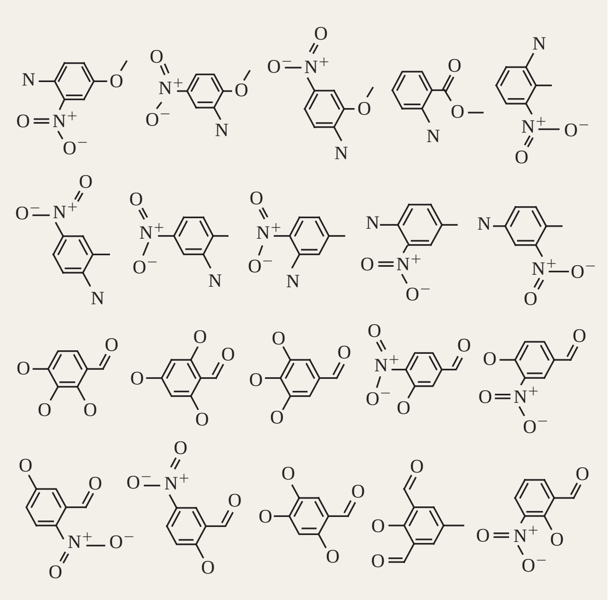


Figure 22

Amine and aldehyde reagents comprising the selection in Figure 21.

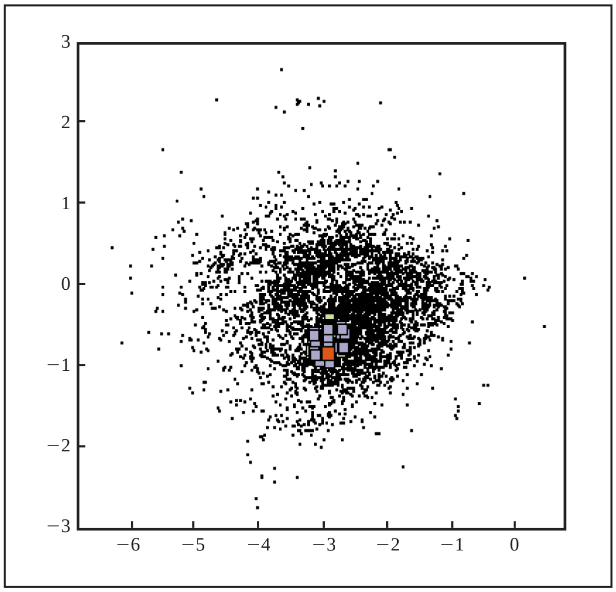


Figure 23

Selection of 100 compounds in 10 × 10 array format based on maximum similarity to the “lead” compound in Figure 20, and minimum overlap with the selection in Figure 21.

is focused on molecular similarity; the extension of these principles to QSAR should be straightforward.

Consider, again, the two-component amination library and a randomly chosen member of that library (**Figure 20**) as a lead. The distribution of the compounds in the 10 × 10 array that are maximally similar to that compound is illustrated in **Figure 21**, and their respective reagents in **Figure 22**. These reagents consist predominantly of di-substituted anilines bearing a nitro substituent, and di- and tri-substituted benzaldehydes containing three oxygen atoms in their substituents. This result demonstrates not only the ability of the algorithm to produce highly focused designs, but also that this particular set of descriptors is fully consistent with our general perception of molecular similarity.

Finally, we turn our attention to the issue of redundancy. Mining a virtual library is an iterative process that involves successive rounds of selection, synthesis, and testing. Since chemical synthesis and biological testing may be time-consuming and costly, it is imperative that the number of duplicates (i.e., compounds that have previously been screened) be kept to a minimum. While this is trivial in the case of sparse arrays, it becomes more problematic with full arrays, particularly when the requested size is relatively large. The problem stems from the fact that eliminating the possibility of duplication often results in the design of mediocre experiments that are far removed from their principal objective. Thus, a more practical approach is to use duplication as a selection criterion whose influence can be tailored by the medicinal chemist on a case-by-case basis. **Figure 23**

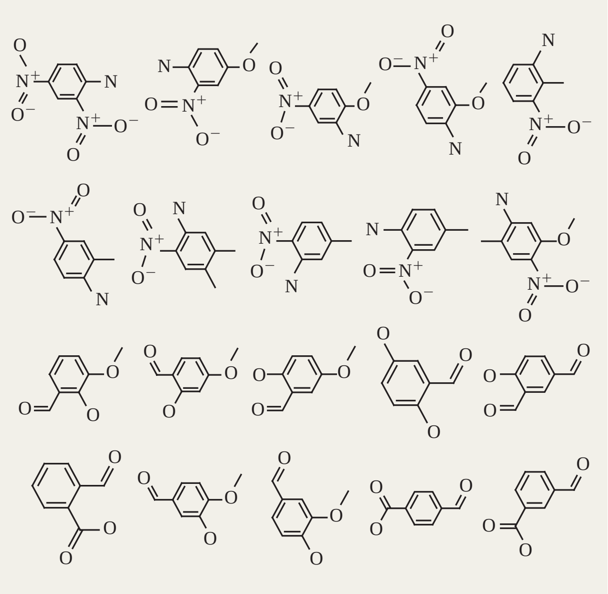


Figure 24

Amine and aldehyde reagents comprising the selection in Figure 23.

illustrates how this principle can be used to design a second focused array around the putative lead in Figure 20, in a way that minimizes the overlap with the original selection (Figure 21). The selection was carried out using the objective function

$$f_o = O - S, \tag{22}$$

where O is the overlap criterion of Equation (16). As shown in **Figure 24**, the two arrays share seven aromatic amines but contain no aldehydes (and therefore no products) in common. To achieve this goal, one oxygen atom is sacrificed from the benzaldehyde group, but except for this change, the products remain closely related to the original lead. Note that in the absence of this criterion, the second selection would have been identical to the first.

5. Concluding remarks

In this paper, we have discussed a general and flexible methodology for compound selection and combinatorial series design. The method permits the selection of multiple subsets from multiple compound collections using multiple selection criteria, in sparse array, full array, and plate format. It is rooted in the principles of multiobjective optimization, and is best viewed as a Pareto optimal process seeking a consensus in which many objectives are balanced so that the improvement of any single objective will result in a negative impact on at least one other objective. The method can accommodate a wide variety of selection criteria, and can be employed for the design of both exploratory and focused combinatorial libraries. Although the method is stochastic in nature and requires the evaluation of a relatively large number of candidate designs, typical selections are carried out in a few seconds to a few minutes on a modern personal computer.

Acknowledgments

The author is grateful to Drs. Victor S. Lobanov, Dmitrii N. Rassokhin, Sergei Izrailev, Edward Jaeger, Renee DesJarlais, Richard M. Soll, and Roger Bone of 3-Dimensional Pharmaceuticals, Inc., for many useful suggestions, and to Dr. Raymond F. Salemme for his insightful comments and support of this work.

**Trademark or registered trademark of 3-Dimensional Pharmaceuticals, Inc., The Institute of Electrical and Electronics Engineers, Inc., The Open Group, Microsoft Corporation, or Intel Corporation.

References

1. L. A. Thompson and J. A. Ellman, *Chem. Rev.* **96,** 555–600 (1996).
2. V. Pareto, *Manual of Political Economy*, 1906, p. 106.
3. H. A. Eschenauer, J. Koski, and A. Osyczka, *Multicriteria Design Optimization: Procedures and Applications*, Springer-Verlag, New York, 1986.
4. E. J. Martin, J. M. Blaney, M. A. Siani, D. C. Spellmeyer, A. K. Wong, and W. H. Moos, “Measuring Diversity: Experimental Design of Combinatorial Libraries for Drug Discovery,” *J. Med. Chem.* **38,** 1431–1436 (1995).
5. D. K. Agrafiotis, R. F. Bone, F. R. Salemme, and R. M. Soll, U.S. Patents 5,463,564, 1995; 5,574,656, 1996; 5,684,711, 1997; and 5,901,069, 1999.
6. R. P. Sheridan and S. K. Kearsley, “Using a Genetic Algorithm to Suggest Combinatorial Libraries,” *J. Chem. Inf. Comput. Sci.* **35,** 310–320 (1995).
7. L. Weber, S. Wallbaum, C. Broger, and K. Gubernator, “Optimization of the Biological Activity of Combinatorial Compound Libraries by a Genetic Algorithm,” *Angew. Chem. Int. Ed. Eng.* **34,** 2280–2282 (1995).
8. J. Singh, M. A. Ator, E. P. Jaeger, M. P. Allen, D. A. Whipple, J. E. Soloweij, S. Chowdhary, and A. M. Treasurywala, “Application of Genetic Algorithms to Combinatorial Synthesis: A Computational Approach for Lead Identification and Lead Optimization,” *J. Amer. Chem. Soc.* **118,** 1669–1676 (1996).
9. D. K. Agrafiotis, “Stochastic Algorithm for Maximizing Molecular Diversity,” presented at the *3rd Electronic Computational Chemistry Conference*, *http://hackberry.chem.niu.edu/ECCC3/paper48*, 1996.
10. D. K. Agrafiotis, “Stochastic Algorithms for Maximizing Molecular Diversity,” *J. Chem. Inf. Comput. Sci.* **37,** 841–851 (1997).
11. D. K. Agrafiotis, “On the Use of Information Theory for Assessing Molecular Diversity,” *J. Chem. Inf. Comput. Sci.* **37,** 576–580 (1997).
12. D. K. Agrafiotis and V. S. Lobanov, “An Efficient Implementation of Distance-Based Diversity Metrics Based on k-d Trees,” *J. Chem. Inf. Comput. Sci.* **39,** 51–58 (1999).
13. M. Hassan, J. P. Bielawski, J. C. Hempel, and M. Waldman, “Optimization and Visualization of Molecular Diversity of Combinatorial Libraries,” *Mol. Diversity* **2,** 64–74 (1996).

14. M. Waldman, H. Li, and M. Hassan, "Novel Algorithms for the Optimization of Molecular Diversity of Combinatorial Libraries," *J. Mol. Graphics Mod.* (special issue on combinatorial library design, D. K. Agrafiotis and E. Martin, Eds.) **18,** 412–426 (2000).
15. A. C. Good and R. A. Lewis, "New Methodology for Profiling Combinatorial Libraries and Screening Sets: Cleaning Up the Design Process with HARPcik," *J. Med. Chem.* **40,** 3926 (1997).
16. W. Zheng, S. J. Cho, and A. Tropsha, "Rational Combinatorial Library Design: 1) Focus-2D: A New Approach to the Design of Targeted Combinatorial Chemical Libraries," *J. Chem. Inf. Comput. Sci.* **38,** 251 (1998).
17. R. D. Brown and Y. C. Martin, "Designing Combinatorial Library Mixtures Using Genetic Algorithms," *J. Med. Chem.* **40,** 2304–2313 (1997).
18. V. J. Gillet, P. Willett, J. Bradshaw, and D. V. S. Green, "Selecting Combinatorial Libraries to Optimize Diversity and Physical Properties," *J. Chem. Inf. Comput. Sci.* **39,** 169–177 (1999).
19. D. N. Rassokhin and D. K. Agrafiotis, "Kolmogorov–Smirnov Statistic and Its Applications in Library Design," *J. Mol. Graphics Mod.* **18,** 370–384 (2000).
20. R. D. Brown, M. Hassan, and M. Waldman, "Combinatorial Library Design for Diversity, Cost Efficiency and Druglike Character," *J. Mol. Graphics Mod.* **18,** 427–437 (2000).
21. R. P. Sheridan, S. G. SanFeliciano, and S. K. Kearsley, "Designing Targeted Libraries with Genetic Algorithms," *J. Mol. Graphics Mod.* **18,** 320–334 (2000).
22. G. M. Downs and P. Willett, "Similarity Searching and Clustering of Chemical-Structure Databases Using Molecular Property Data," *J. Chem. Inf. Comput. Sci.* **34,** 1094–1102 (1994).
23. R. D. Brown and Y. C. Martin, *J. Chem. Inf. Comput. Sci.* **36,** 572–584 (1996).
24. D. E. Patterson, R. D. Cramer, A. M. Ferguson, R. D. Clark, and L. E. Weinberger, *J. Med. Chem.* **39,** 3049–3059 (1996).
25. R. D. Brown and Y. C. Martin, "The Information Content of 2D and 3D Structural Descriptors Relevant to Ligand-Receptor Binding," *J. Chem. Inf. Comput. Sci.* **37,** 1–9 (1997).
26. H. Matter, "Selecting Optimally Diverse Compounds from Structure Databases: A Validation Study of Two-Dimensional and Three-Dimensional Molecular Descriptors," *J. Med. Chem.* **40,** 1219 (1997).
27. Y. C. Martin, M. G. Bures, and R. D. Brown, "Validated Descriptors for Diversity Measurements and Optimization," *Pharm. Pharmacol. Commun.* **4,** 147 (1998).
28. V. J. Gillet, P. Willett, and J. Bradshaw, *J. Chem. Inf. Comput. Sci.* **37,** 731–740 (1997).
29. E. A. Jamois, M. Hassan, and M. Waldman, *J. Chem. Inf. Comput. Sci.* **40,** 63–70 (2000).
30. A. Polinsky, R. D. Feinstein, S. Shi, and A. Kuki, *Molecular Diversity and Combinatorial Chemistry*, I. M. Chaiken and K. D. Janda, Eds., American Chemical Society: Washington, DC, 1996, pp. 219–232.
31. E. J. Martin, D. C. Spellmeyer, R. E. Critchlow, and J. M. Blaney, "Does Combinatorial Chemistry Obviate Computer-Aided Drug Design?" *Reviews in Computational Chemistry*, Volume 10, K. B. Lipkowitz and D. B. Boyd, Eds., VCH Publishers, New York, 1997, pp. 75–100.
32. P. Willett, *Similarity and Clustering in Chemical Information Systems*, Research Studies Press, Letchworth, UK, 1987.
33. R. Taylor, *J. Chem. Inf. Comput. Sci.* **35,** 59–67 (1995).
34. D. Chapman, *J. Comput.-Aided Mol. Design* **10,** 501–512 (1996).
35. D. J. Cummins, C. W. Andrews, J. A. Bentley, and M. Cory, *J. Chem. Inf. Comput. Sci.* **36,** 750–763 (1996).
36. R. S. Pearlman and R. S. Smith, *Perspect. Drug Discovery Design* **9,** 339–353 (1998).
37. S. Pickett, J. S. Mason, and I. M. McLay, *J. Chem. Inf. Comput. Sci.* **36,** 1214–1223 (1996).
38. E. K. Davies and C. Briant, *Network Sci*., 1995, *http://www.awod.com/netsci/issues/*.
39. N. E. Shemetulsksis, D. Weininger, C. J. Blankley, J. J. Yang, and C. Humblet, *J. Chem. Inf. Comput. Sci.* **36,** 862–871 (1996).
40. S. M. Boyd, M. Beverly, L. Norskov, and R. E. Hubbard, *J. Comput.-Aided Mol. Design* **9,** 417–424 (1995).
41. D. K. Agrafiotis and V. S. Lobanov, "Ultrafast Algorithm for Designing Focused Combinatorial Arrays," *J. Chem. Inf. Comput. Sci.* **40,** 1030–1038 (2000).
42. R. V. Stanton, J. Mount, and J. L. Miller, "Combinatorial Library Design: Maximizing Model Fitting Compounds with Matrix Synthesis Constraints," *J. Chem. Inf. Comput. Sci.* **40,** 701–705 (2000).
43. D. K. Agrafiotis, "The Diversity of Chemical Libraries," *The Encyclopedia of Computational Chemistry*, P. v. R. Schleyer, N. L. Allinger, T. Clark, J. Gasteiger, P. A. Kollman, H. F. Schaefer III, and P. R. Schreiner, Eds., John Wiley & Sons, Chichester, UK, 1998, pp. 742–761.
44. D. K. Agrafiotis, J. C. Myslik, and F. R. Salemme, "Advances in Diversity Profiling and Combinatorial Series Design," *Mol. Diversity* **4,** 1–22 (1999).
45. D. K. Agrafiotis, V. S. Lobanov, D. N. Rassokhin, and S. Izrailev, "The Measurement of Molecular Diversity," *Virtual Screening of Bioactive Molecules*, H.-J. Böhm and G. Schneider, Eds., Wiley–VCH Verlag GmbH, Weinheim, Germany, 2000.
46. M. S. Lajiness, in *QSAR: Rational Approaches to the Design of Bioactive Compounds*, C. Silipo and A. Vittoria, Eds., Elsevier, Amsterdam, Netherlands, 1991, pp. 201–204.
47. D. S. Dhanoa, V. Gupta, A. Sapienza, and R. M. Soll, Poster 26, American Chemical Society National Meeting, Anaheim, CA, 1999.
48. L. H. Hall and L. B. Kier, "The Molecular Connectivity Chi Indexes and Kappa Shape Indexes in Structure–Property Relations," *Reviews of Computational Chemistry*, D. B. Boyd and K. B. Lipkowitz, Eds., VCH Publishers, New York, 1991, Ch. 9, pp. 367–422.
49. D. Bonchev and N. Trinajstic, *J. Chem. Phys.* **67,** 4517–4533 (1977).
50. V. S. Lobanov and D. K. Agrafiotis, "Stochastic Similarity Selections from Large Combinatorial Libraries," *J. Chem. Inf. Comput. Sci.* **40,** 460–470 (2000).
51. D. K. Agrafiotis, "A New Method for Analyzing Protein Sequence Relationships Based on Sammon Maps," *Prot. Sci.* **6,** 287–293 (1997).
52. D. K. Agrafiotis and V. S. Lobanov, "Nonlinear Mapping Networks," *J. Chem. Inf. Comput. Sci.* **40,** 1356–1362 (2000).
53. D. N. Rassokhin, V. S. Lobanov, and D. K. Agrafiotis, "Nonlinear Mapping of Massive Data Sets by Fuzzy Clustering and Neural Networks," *J. Comput. Chem.* **22,** 373–386 (2000).
54. D. K. Agrafiotis, D. N. Rassokhin, and V. S. Lobanov, "Multidimensional Scaling of Large Molecular Similarity Tables," *J. Comput. Chem.* **22,** 488–500 (2000).
55. A. K. Ghose, V. N. Viswanadhan, and J. J. Wendoloski, *J. Phys. Chem. A* **102,** 3762–3772 (1998).
56. D. K. Agrafiotis, "Constant Time Algorithm for Estimating the Diversity of Large Chemical Libraries," *J. Chem. Inf. Comput. Sci.* **41,** 159–167 (2000).
57. E. J. Martin and R. E. Critchlow, "Beyond Mere Diversity: Tailoring Combinatorial Libraries for Drug Discovery," *J. Comb. Chem.* **1,** 32–45 (1999).

58. E. J. Martin and A. Wong, "Sensitivity Analysis and Other Improvements to Tailored Combinatorial Library Design," *J. Chem. Inf. Comput. Sci.* **40,** 215–220 (2000).
59. R. T. Koehler, S. L. Dixon, and O. H. Villar, "LASSOO: A Generalized Directed Diversity Approach to the Design and Enrichment of Chemical Libraries," *J. Med. Chem.* **42,** 4695–4704 (1999).
60. C. A. Lipinski, F. Lombardo, B. W. Dominy, and P. J. Feeny, *Adv. Drug Delivery Rev.* **23,** 3–25 (1997).

Received August 22, 2000; accepted for publication November 17, 2000

Dimitris K. Agrafiotis *3-Dimensional Pharmaceuticals, Inc., 665 Stockton Drive, Suite 104, Exton, Pennsylvania 19341 (dimitris@3dp.com).* Dr. Agrafiotis is Senior Director of the DirectedDiversity group at 3-Dimensional Pharmaceuticals, Inc. He received a B.S. degree in chemistry from the University of Patras, Greece, in 1985, and a Ph.D. degree in theoretical organic chemistry from the Imperial College, University of London, in 1988. Following postdoctoral study in the laboratories of Professor A. Streitwieser at the University of California, Berkeley, and Nobel Laureate E. J. Corey at Harvard, he joined Parke–Davis Pharmaceutical Research as a Senior Scientist in their Computer-Aided Drug Design group. In 1994, he joined 3-Dimensional Pharmaceuticals, Inc., where he has focused on the development of intelligent computational tools for combinatorial chemistry, high-throughput screening, and structure-based drug design. Dr. Agrafiotis serves on the Editorial Board of the *Journal of Molecular Graphics and Modelling*; he is a co-inventor of 3-D Pharmaceuticals' proprietary DirectedDiversity® technology.