# Chapter 4

# A Scalable Approach to Combinatorial Library Design

**Puneet Sharma, Srinivasa Salapaka, and Carolyn Beck**

## Abstract

In this chapter, we describe an algorithm for the design of lead-generation libraries required in combinatorial drug discovery. This algorithm addresses simultaneously the two key criteria of diversity and representativeness of compounds in the resulting library and is computationally efficient when applied to a large class of lead-generation design problems. At the same time, additional constraints on experimental resources are also incorporated in the framework presented in this chapter. A computationally efficient scalable algorithm is developed, where the ability of the deterministic annealing algorithm to identify clusters is exploited to truncate computations over the entire dataset to computations over individual clusters. An analysis of this algorithm quantifies the trade-off between the error due to truncation and computational effort. Results applied on test datasets corroborate the analysis and show improvement by factors as large as ten or more depending on the datasets.

**Key words:** Library design, combinatorial optimization, deterministic annealing.

## 1. Introduction

In recent years, combinatorial chemistry techniques have provided important tools for the discovery of new pharmaceutical agents. Lead-generation library design, the process of screening and then selecting a subset of potential drug candidates from a vast library of similar or distinct compounds, forms a fundamental step in combinatorial drug discovery (1). Recent advances in high-throughput screening such as using micro/nanoarrays have given further impetus to large-scale investigation of compounds. However, combinatorial libraries often consist of extremely large collections of chemical compounds, typically several million. The time and cost of associated experiments makes it practically

J.Z. Zhou (ed.), *Chemical Library Design*, Methods in Molecular Biology 685,
DOI 10.1007/978-1-60761-931-4_4, 

impossible to synthesize each and every combination from such a library of compounds. To overcome this problem, chemists often work with virtual combinatorial libraries (VCLs), which are combinatorial databases containing enumeration of all possible structures of a given pharmacophore with all available reactants. A subset of lead compounds from this VCL is selected which is used for physical synthesis and biological target testing. The selection of this subset is based on a complex interplay between various objectives, which is cast as a combinatorial optimization problem. The main goal of this optimization problem is to identify a subset of compounds that is representative of the underlying vast library as well as manageable, where these lead compounds can be synthesized and subsequently tested for relevant properties, such as activity and bioaffinity. The combinatorial nature of the selection problem makes it impractical to exhaustively enumerate each and every possible subset of obtaining the optimal solution. For example, to select 30 lead compounds from a set of 1,000, there are approximately $3 \times 10^{25}$ different possible combinations. Selection based on enumeration is thus impractical and requires numerically efficient algorithms to solve the constrained combinatorial optimization problem.

## 2. Issues in Lead-Generation Library Design

In addition to the computational complexity that arises due to the combinatorial nature of the problem, any algorithm that aims to address the lead-generation library design problem must address the following key issues:

*Diversity versus representativeness*: The most widely used method to obtain a lead-generation library involves maximizing the diversity of the overall selection (2, 3), based on the premise that the more diverse the set of compounds, the better the chance to obtain a lead compound with desired characteristics. Such a design strategy suffers from an inherent problem that using diversity as the sole criterion may result in a set where a large number of lead compounds disproportionately represent outliers or singletons (4, 5). However, from a drug discovery point of view, it is desirable for the lead-generation library to more proportionally represent all the compounds, or at least to quantify how representative each lead compound is in order to allot experimental resources. A maximally diverse subset is of little practical significance because of its limited pharmaceutical applications. Therefore, representativeness should be considered as a lead-generation library design criterion along with diversity (6, 7).

*Design constraints*: In addition to diversity and representativeness, other design criteria include confinement, which quantifies the degree to which the properties of a set of compounds lie in a prescribed range (8), and maximizing the activity of the set of compounds against some predefined targets. Activity is usually measured in terms of the quantitative structure of the given set. Additionally, the cost of chemical compounds and experimental resources is significant and presents one of the main impediments in combinatorial diagnostics and drug synthesis. Different compounds require different experimental supplies which are typically available in limited quantities. The presence of these multiple (and often conflicting) design objectives makes the library design a multiobjective optimization problem with constraints.

## 3. Basic Problem Formulation and Modifications

*Basic formulation*: The problem of selecting lead compounds for lead-generation library design can be stated in general as follows:

*Given a distribution of N compounds, $x_i$, in a descriptor space* $\Omega$, *find the set of M lead compounds, $r_j$, that solves the following minimization problem:*

$$\min_{r_j, 1 \le j \le M} \sum_{i=1}^{N} p(x_i) \left\{ \min_{1 \le j \le M} d(x_i, r_j) \right\} \qquad [1]$$

Here, $\Omega$ represents the chemical property space corresponding to the VCL, $d(x_i, r_j)$ represents an appropriate distance metric between the lead compound $r_j$ and the compound $x_i$, $p(x_i)$ is the relative weight that can be attached to compound $x_i$ (if all compounds are of equal importance, then the weights $p(x_i) = \frac{1}{N}$ for each $i$), and $M$ is typically much smaller than $N$. That is, this problem seeks a subset of $M$ lead compounds $r_j$ in a descriptor space such that the average distance of a compound $x_i$ from its nearest lead compound is minimized. Alternatively, this problem can also be formulated as finding an optimal partition of the descriptor space $\omega$ into $M$ clusters $\{R_j\}$ and assigning to each cluster a lead compound $r_j$ such that the following cost function is minimized:

$$\sum_{j=1}^{M} \sum_{x_i \in R_j} d(x_i, r_j) p(x_i)$$

*Incorporating diversity and representativeness*: One drawback of the basic formulation is that all the lead compounds are

weighted equally. However, design constraints often require distinguishing them from one another to reflect different aspects of the clusters. For example, when addressing the issue of representativeness in the lead-generation library, the lead compounds that represent larger clusters need to be distinguished from those that represent outliers.

We incorporate representativeness into the problem formulation by specifying an additional *relative weight* parameter $\lambda_j, 1 \leq j \leq M$ for each lead compound. This parameter $\lambda_j$ quantifies the size of the cluster represented by the compound $r_j$, and it is proportional to the number of the compounds in that cluster. Thus, the resulting library design will associate lead compounds that represent outliers with low values of $\lambda$ and the lead compounds that represent the majority members with corresponding high values. In this way, the algorithm can be used to identify distinct compounds through property vectors $r_j$ in the descriptor space $\Omega$ that denote the *jth* lead compound and at the same time determine how representative each lead compound is. For instance, $\lambda_j = 0.2$ implies that lead compound $r_j$ represents 20% of all compounds in the VCL. The following modified optimization problem adequately describes the diversity goals in the basic formulation as well as the representativeness through the relative weights $\lambda_j$:

$$\min_{r_j,\lambda_j,1\leq j\leq M} \sum_i p(x_i) \left\{ \min_{1\leq j\leq M} d(x_i, r_j) \right\} \qquad [2]$$

such that $\sum_{j=1}^{M} \lambda_j = 1$

where $\lambda_j$ is the fraction of compounds in VCL that are nearest to (represented by) the lead compound $r_j$.

*Incorporating constraints on experimental resources*: Experiments associated with compounds with different properties often require different experimental resources. The constraints on availability of these resources can vary depending on their respective handling costs and time. These constraints can be incorporated in the selection problem by associating appropriate weights to lead compounds. For instance, consider a VCL that is classified into $q$ types of compounds corresponding to $q$ types of experimental supplies required for testing. More specifically, the *jth* lead compound can avail only $W_{jn}$ amount of the *nth* experimental resource ($1 \leq n \leq q$). The modified optimization problem is then given by (9, 10)

$$\min_{r_j} D = \sum_n \sum_i p_n(x_i^n) \left\{ \min_j d(x_i^n, r_j) \right\} \qquad [3]$$

such that $\lambda_{jn} = W_{jn} 1 \leq j \leq M, 1 \leq n \leq q$

where $p_n(x_i^n)$ is the weight of the compound location $x_i^n$, which requires the *nth* type of supply.

## 4. Computational Issues

Problem formulations [1–3] for designing lead-generation library under different constraints belong to a class of combinatorial resource allocation problems, which have been widely studied. They arise in many different applications such as minimum distortion problems in data compression (11), facility location problems (12), optimal quadrature rules and discretization of partial differential equations (13), locational optimization problems in control theory (9), pattern recognition (14), and neural networks (15). Combinatorial resource allocation problems are nonconvex and computationally complex and it is well documented (16) that most of them have many local minima that riddle the cost surface. Therefore, the main computational issue is developing an efficient algorithm that avoids local minima. Due to the large size of VCLs, and the combinatorial nature of the problem, the issue of algorithm scalability takes central importance. Since the number of computations to be performed by the lead-generation library design algorithm scales up exponentially with an increase in the amount of data, most algorithms become prohibitively slow and expensive (computationally) for large datasets.

### 4.1. Deterministic Annealing Algorithm

The main drawback of most popular algorithms that address the basic combinatorial resource allocation problem [1], such as Lloyd's or K-means algorithms (11, 17), is that they are extremely sensitive to initialization step in their procedures and typically get trapped in local minima. Other algorithms such as simulated annealing that actively try to avoid local minima are often computationally inefficient. Other drawbacks of these algorithms mainly stem from the lack of flexibility to incorporate various constraints on the resource locations discussed in **Section 3**. The deterministic annealing (DA) algorithm (18) overcomes these drawbacks; this algorithm is heuristically based on law of minimum free energy in statistical chemistry that models similar combinatorial problems occurring in nature. The DA algorithm is versatile in terms of accommodating constraints on resource locations while simultaneously it is designed to be insensitive to the initialization step and to avoid local minima.

The central concept of the DA algorithm is based on developing a homotopy from an appropriate convex function to the nonconvex cost function; the local minima of cost function at every

step of homotopy serves as the initialization for the subsequent step. Since minimization of the initial convex function yields a global minimum, this procedure is independent of initialization. The heuristic is that the global minimum is tracked as the initial convex function deforms into the actual nonconvex cost function via the homotopy. Accordingly, the DA algorithm solves the following multiobjective optimization problem:

$$\min_{r_j} \min_{p(r_j|x_i)} \underbrace{D - T_k H}_{:=F}$$

over iterations indexed by $k$, where $T_k$ is a parameter called temperature which tends to zero as $k$ tends to infinity. The cost function $F$ is called free energy, where this terminology is motivated by statistical chemistry (18). Here the distortion

$$D = \sum_{i=1}^{N} p(x_i) \sum_{j=1}^{M} d(x_i, r_j) p(r_j|x_i)$$

which is similar to the cost function in equation [1] is the "weighted average distance" of a lead compound $r_j$ from a compound $x_i$ in the VCL. This formulation associates each $x_i$ to every $r_j$ through the weighting parameter $p(r_j|x_i)$ and thus diminishes the sensitivity of algorithm to initialization of locations $r_j$. The more uniformly (or randomly) these weights are distributed, the more insensitive is the algorithm with respect to the initialization. The term $H = -\sum_{i,j} p(y_j|x_i) \log p(y_j|x_i)$ is the entropy of the weights $\{p(y_j|x_i)\}$ that quantifies their uniformity (or randomness). The annealing parameter $T_k$ defines the homotopy from the convex function $-H$ to the nonconvex function $D$. Clearly, for large values of $T_k$, we mainly attempt to maximize the entropy. As $T_k$ is lowered, we trade entropy for the reduction in distortion, and as $T_k$ approaches zero, we minimize $D$ directly to obtain a hard (nonrandom) solution, where $p(y_j|x_i)$ is either 0 or 1 for each pair *(i,j)*. Minimizing the free energy term $F$ with respect to the weighting parameter $p(r_j|x_i)$ is straightforward and gives the Gibbs distribution

$$p(r_j|x_i) = \frac{e^{-d(x_i,r_j)/T_k}}{Z_i}, \text{ where } Z_i := \sum_j e^{-d(x_i,r_j)/T_k} \qquad [4]$$

Note that the weighting parameters $p(r_j|x_i)$ are simply radial basis functions, which clearly decrease in value exponentially as $r_j$ and $x_i$ move farther apart. The corresponding minimum of $F$ is obtained by substituting for $p(r_j|x_i)$ from equation [4]:

$$\widehat{F} = -T_k \sum_i p(x_i) \log Z_i \qquad [5]$$

To minimize $\widehat{F}$ with respect to the lead compounds $\{r_j\}$, we set the corresponding gradients equal to zero, i.e., $\frac{\partial \widehat{F}}{\partial r_j} = 0$; this yields the corresponding implicit equations for the locations of lead compounds:

$$r_j = \sum_i p(x_i|r_j)x_i, 1 \le j \le M, \text{where}$$
$$p(x_i|r_j) = \frac{p(x_i)p(r_j|x_i)}{\sum_k p(x_k)p(r_j|x_k)} \quad [6]$$

Note that $p(x_i|r_j)$ denotes the posterior probability calculated using Bayes' rule and the above equations clearly convey the *centroid* aspect of the solution.

The DA algorithm consists of minimizing $\widehat{F}$ with respect to $\{r_j\}$ starting at high values of $T_k$ and then tracking the minimum of $\widehat{F}$ while lowering $T_k$. At each $k$,

1. Fix $\{r_j\}$ and use equation [4] to compute the new weights $\{p(r_j|x_i)\}$.
2. Fix $\{p(r_j|x_i)\}$ and use equation [6] to compute the lead compound locations $\{r_j\}$.

### 4.2. A Scalable Algorithm

As noted earlier, one of the major problems with combinatorial optimization algorithms is that of scalability, i.e., the number of computations scales up exponentially with an increase in the amount of data. In the DA algorithm, the computational complexity can be addressed in two steps – first by reducing the number of iterations and second by reducing the number of computations at every iteration. The DA algorithm, as described earlier, exploits the phase transition feature (18) in its process to decrease the number of iterations (in fact in the DA algorithm, typically the temperature variable is decreased exponentially which results in few iterations). The number of computations per iteration in the DA algorithm is $O(M^2\ N)$, where $M$ is the number of lead compounds and $N$ is the total number of compounds in the underlying VCL. In this section, we present an algorithm that requires fewer computations per iteration. This amendment becomes necessary in the context of the selection problem in combinatorial chemistry as the sizes of the dataset are so large that DA is typically too slow and often fails to handle the computational complexity. We exploit the features inherent in the DA algorithm that, for a given temperature, the farther an individual compound is from a cluster, the lower is its influence on the cluster (as is evident from equation [4]). That is, if two clusters are far apart, then they have very small interaction between them. Thus, if we ignore the effect of a separated cluster on the remaining compound locations, the resulting error will not be significant (*see* **Fig. 4.1**). Ignoring the effects of separated regions (i.e., groups

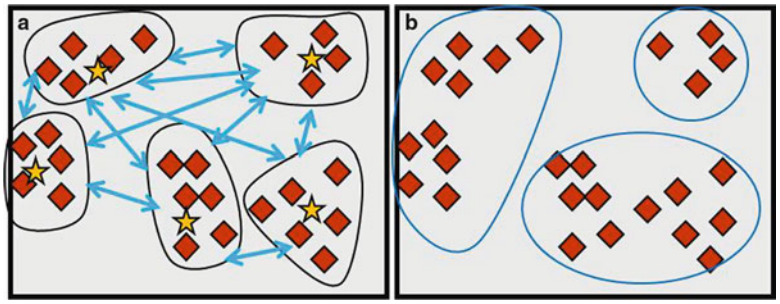


Fig. 4.1. (**a**) Illustration depicting the different clusters in the dataset, together with the interaction between each pair of points (and *clusters*). (**b**) Separated regions determined after characterizing intercluster interaction and separation.

of clusters) on one another will result in a considerable reduction in the number of computations since the points that constitute a separated region will not contribute to the distortion and entropy computations for the rest. This computational saving increases as the temperature decreases since the number of separated regions, which are now smaller, increases as the temperature decreases.

*4.2.1. Cluster Interaction and Separation*

In order to characterize the interaction between different clusters, it is necessary to consider the mechanism of cluster identification during the process of the DA algorithm. As the temperature ($T_k$) is reduced after every iteration, the system undergoes a series of phase transitions (*see* (18) for details). In this annealing process, at high temperatures that are above a precomputable critical value, all the lead compounds are located at the centroid of the entire descriptor space, thereby there is only one distinct location for the lead compounds. As the temperature is decreased, a critical temperature value is reached where a phase transition occurs, which results in a greater number of distinct locations for lead compounds and consequently finer clusters are formed. This provides us with a tool to control the number of clusters we want in our final selection. It is shown (18) for a square Euclidean distance $d(x_i, r_j) = \|x_i - r_j\|^2$ that a cluster $R_i$ splits at a critical temperature $T_c$ when twice the maximum eigenvalue of the posterior covariance matrix, defined by $C_{x|r_j} = \sum_i p(x_i)p(x_i|r_j)(x_i - r_j)(x_i - r_j)^T$, becomes greater than the temperature value, i.e., when $T_c \leq 2\lambda_{\max}\left[C_{x|r_i}\right]$. This is exploited in the DA algorithm to reduce the number of iterations by jumping from one critical temperature to the next without significant loss in performance. In the DA algorithm, the lead location $r_j$ is primarily determined by the compounds near it since far-away points exert small influence, especially at low temperatures. The association probabilities $p(r_j|x_i)$ determine the level of interaction between the cluster $R_j$ and the data-point $x_i$. This interaction decays exponentially with the increase in the distance between $r_j$ and $x_i$. The total interaction exerted by all the data-points in a

given space determines the relative weight of each cluster, $p(r)_j := \sum_i^N p(x_i, r_j) = \sum_i^N p(r_j|x_i)p(x_i)$, where $p(r_j)$ denotes the weight of cluster $R_j$. We define the level of interaction that data-points in cluster $R_i$ exert on cluster $R_j$ by $\varepsilon_{ji} = \sum_{x \in R_i} p(r_j|x)p(x)$. The higher this value is, the more interaction exists between clusters $R_i$ and $R_j$. This gives us an effective way to characterize the interaction between various clusters in a dataset. In a probabilistic framework, this interaction can also be interpreted as the probability of transition from $R_i$ to $R_j$. Consider the $m \times n$ matrix $m \geq n$

$$A = \begin{pmatrix} \sum_{x \in R_1} p(r_1|x)p(x) & \cdots & \sum_{x \in R_m} p(r_1|x)p(x) \\ \sum_{x \in R_1} p(r_2|x)p(x) & \cdots & \sum_{x \in R_m} p(r_1|x)p(x) \\ \vdots & \ddots & \vdots \\ \sum_{x \in R_1} p(r_m|x)p(x) & \cdots & \sum_{x \in R_m} p(r_1|x)p(x) \end{pmatrix}$$

In a probabilistic framework, this matrix is a finite-dimensional Markov operator, with the term $A_{j,i}$ denoting the transition probability from region $R_i$ to $R_j$. The higher the transition probability, the greater is the amount of interaction between the two regions. Once the transition matrix is formed, the next step is to identify regions, that is, groups of clusters, which are separate from the rest of the data. The separation is characterized by a quantity which we denote by ε. We say a cluster ($R_j$) is ε-separate if the level of its interaction with each of the other clusters ($A_{j,i}, i = 1, 2, \ldots, n, i \neq j$) is less than ε. The value ε is used to partition the descriptor space into separate regions for reduced and scalable computational effort, and it quantifies the increase in the distortion cost function of the proposed scalable algorithm with respect to the DA algorithm.

*4.2.2. Trade-Off Between Error in Lead Compound Location and Computation Time*

As was discussed in **Section 4.2**, the greater the number of separate regions we use, the smaller the computation time for the scalable algorithm. At the same time, a greater number of separate regions results in a higher deviation in the distortion term of the proposed algorithm from the original DA algorithm. This trade-off between reduction in computation time and increase in distortion error is systematically addressed in the following. For any pair $(r_j, V)$, where $r_j$ is a lead compound and $V$ is a subset of the descriptor space Ω, we define

$$G_j(V) := \sum_{x_i \in V} x_i p(x_i) p(r_j|x_i),$$
$$H_j(V) := \sum_{x_i \in V} p(x_i) p(r_j|x_i) \qquad [7]$$

Then, from the DA algorithm, the location of the lead compound ($r_j$) is determined by $r_j = \frac{G_j(\Omega)}{H_j(\Omega)}$. Since the cluster $\Omega_j$ is separated from all the other clusters, the lead compound location $r'_j$ will be determined in the scalable algorithm by

$$r'_j = \frac{\sum_{x_i \in \Omega_j} x_i p(x_i) p(r_j|x_i)}{\sum_{x_i \in \Omega_j} p(x_i) p(r_j|x_i)} = \frac{G_j(\Omega_j)}{H_j(\Omega_j)} \qquad [8]$$

We obtain the component-wise difference between $r_j$ and $r'_j$ by subtracting terms. Note that we use the symbols $\prec$ and $\succ$ for component-wise operations. On simplifying, we have

$$\left| r'_j - r_j \right| \prec= \frac{\max\left( G_j(\Omega_j^c) H_j(\Omega_j), G_j(\Omega_j) H_j(\Omega_j^c) \right)}{H_j(\Omega_j) H_j(\Omega)}, \qquad [9]$$

where $\Omega_j^c = \Omega \backslash \Omega_j$

Denoting the cardinality of $\Omega$ by $N$ and $M_j^c = \frac{1}{N} \sum_{x_i \in \Omega_j^c} x_i$, we note that

$$G_j(\Omega_j^c) \leq \left( \sum_{x_i \in \Omega_j^c} x_i \right) H_j(\Omega_j^c) = N M_j^c H_j(\Omega_j^c) \qquad [10]$$

We have assumed $x \succ= 0$ without any loss of generality since the problem definition is independent of translation or scaling factors. Thus,

$$\begin{aligned} \left| r'_j - r_j \right| \prec= & \frac{\max\left( N M_j^c H_j(\Omega_j), G_j(\Omega_j) \right) H_j(\Omega_j^c)}{H_j(\Omega_j) H_j(\Omega)} \\ = & \max\left( N M_j^c, \frac{G_j(\Omega_j)}{H_j(\Omega_j)} \right) \left( \frac{H_j(\Omega_j^c)}{H_j(\Omega)} \right) \end{aligned} \qquad [11]$$

then dividing through by $N$ and using $M = \frac{1}{N} \sum_{x_i \in \Omega} x_i$ gives

$$\frac{\left| r'_j - r_j \right|}{MN} \prec= \max\left( \frac{M_j^c}{M}, \frac{M_j}{M} \right) \eta_j, \text{where } \eta_j = \frac{\sum_{k \neq j} \varepsilon_{kj}}{\sum_k \varepsilon_{kj}} \qquad [12]$$

and $\varepsilon_{kj}$ is the level of interaction between cluster $\Omega_j$ and $\Omega_k$. For a given dataset, the quantities $M, M_j$,and $M_j^c$ are known a priori. For the error in lead compound location $\left| r'_j - r_j \right| / M$ to be less than a given value $\delta_j$ (where $\delta_j > 0$), we must choose $\eta_j$ such that

$$\eta_j \leq \frac{\delta_j}{N \max\left(\frac{M_j^e}{M}, \frac{M_j}{M}\right)} \quad [13]$$

*4.2.3. Scalable Algorithm*

1. Initiate the DA algorithm and determine lead compound locations together with the weighting parameters.
2. When a split occurs (phase transition), identify individual clusters and use the weights $\left(p(r_j|x)\right)$ to construct the transition matrix.
3. Use the transition matrix to identify separated clusters and group them to form separated regions. $\Omega_k$ will be separated from $\Omega_j$ if the entries $A_{j,k}$ and $A_{k,j}$ are less than a chosen $\varepsilon_{jk}$.
4. Apply the DA to each region, neglecting the effect of separate regions on one another.
5. Stop if the terminating criterion (such as maximum number of lead compounds ($M$) or maximum computation time) is met, otherwise go to 2.

Identification of separate regions in the underlying data provides us with a tool to efficiently scale the DA algorithm. In the DA algorithm, at any iteration, the number of computations is $M^2\ N$. In the proposed scalable algorithm, the number of computations at a given iteration is proportional to $\sum_{k=1}^{s} M_k^2 N_k$, where $N_k \left(N = \sum_{k=1}^{s} N_k\right)$ is the number of compounds and $M_k$ is the number of clusters in the *kth* region. Thus, the scalable algorithm saves computations at each iteration. This savings increases as temperature decreases since corresponding values of $N_k$ decrease. Moreover, since the scalable algorithm can run these $s$ DA algorithms in *parallel*, it will result in additional potential savings in computational time.

## 5. Simulation Results

### *5.1. Design for Diversity and Representativeness*

As a first step, a fictitious dataset (VCL) was created to present the "proof of concept" for the proposed optimization algorithm. The VCL was specifically designed to simultaneously address the issue of diversity and representativeness in the lead-generation library design. This dataset consists of few points that are outliers while most of the points are in a single cluster. Simulations were carried out in MATLAB. The results for dataset 1 are shown in **Fig**. **4.2**. The pie chart in **Fig**. **4.2** shows the relative weight of each lead compound. As was required, the algorithm gave larger weights at locations which had larger numbers of similar compounds. At the same time, it should be noted that the key issue of diversity is not

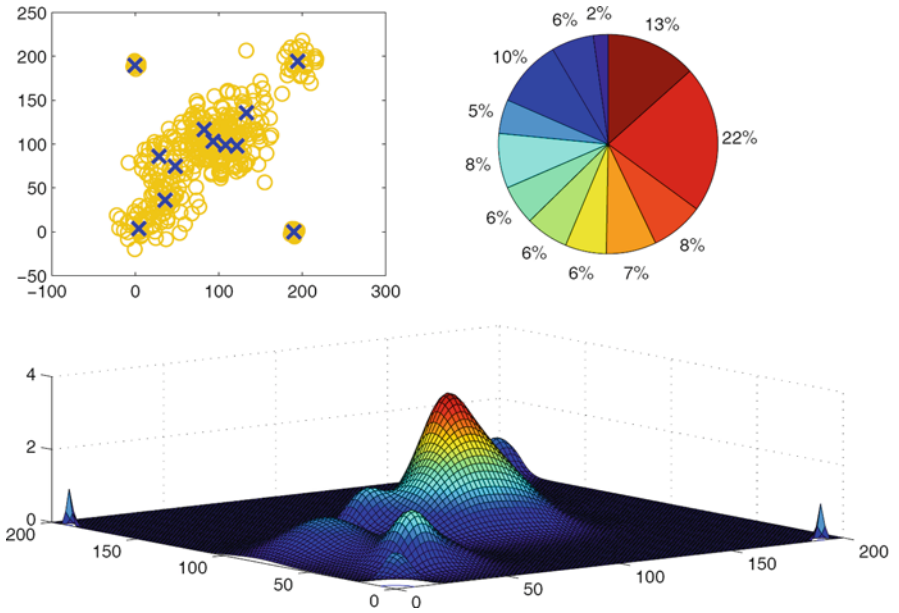


Fig. 4.2. Simulation results for dataset 1. (**a**) The locations $x_i, 1 \leq i \leq 200$, of compounds (*circles*) and $r_j, 1 \leq j \leq 10$, of lead compounds (*crosses*) in the 2-d descriptor space. (**b**) The weights $\lambda_j$ associated with different locations of lead compounds. (**c**) The given weight distribution $p(x_i)$ of the different compounds in the dataset. Reprinted ("adapted" or "in part") with permission from *Journal of Chemical Information and Modeling*. Copyright 2008 American Chemical Society.

compromised. This is due to the fact that the algorithm inherently recognizes the natural clusters in the VCL. As is seen from the figure, the algorithm identifies all clusters. The two clusters which were quite distinct from the rest of the compounds are also identified albeit with a smaller weight. As can be seen from the pie chart, the outlier cluster was assigned a weight of 2%, while the central cluster was assigned a significant weight of 22%.

### *5.2. Scalability and Computation Time*

In order to demonstrate the computational savings, the algorithm was tested on a suite of synthesized datasets. The first set was obtained by identifying ten random locations in a square region of size $400 \times 400$. These locations were then chosen as the cluster centers. Next, the size of each of these clusters was chosen and all points in the cluster were generated by a normal distribution of randomly chosen variance. A total of 5,000 points comprised this dataset. All the points were assigned equal weights (i.e., $p(x_i) = \frac{1}{N}$ for all $x_i \in \Omega$). **Figure 4.3** shows the dataset and the lead compound locations obtained by the original DA algorithm. The crosses denote the lead compound locations $(r_j)$ and the pie chart gives the relative weight of each lead compound $(\lambda_j)$.

The algorithm starts with one lead compound at the centroid of the dataset. As the temperature is reduced, the cluster is split and separate regions are determined at each such split.

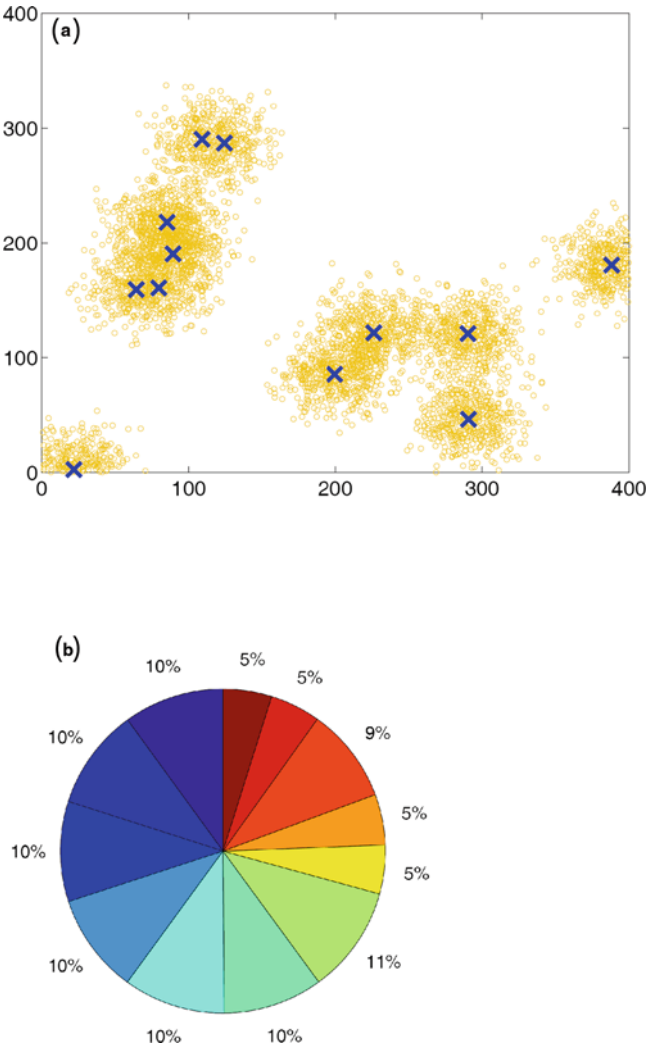


Fig. 4.3. (**a**) Locations $x_i, 1 \leq i \leq 5,000$, of compounds (*circles*) and $r_j, 1 \leq j \leq 12$, of lead compounds (*crosses*) in the 2-d descriptor space determined from the original algorithm. (**b**) Relative weights $\lambda_j$ associated with different locations of lead compounds. Reprinted ("adapted" or "in part") with permission from *Journal of Chemical Information and Modeling*. Copyright 2008 American Chemical Society.

**Figure 4.4a** shows the four separate regions identified by the algorithm (as described in **Section 4.2.1**) at the instant when 12 lead compound locations have been identified. **Figure 4.4b** shows a comparison between the two algorithms. Here the crosses represent the lead compound locations ($r_j$) determined by the original DA algorithm and the circles represent the locations ($r'_j$) determined by the proposed scalable algorithm. As can be seen from the figure, there is little difference between the locations obtained by the two algorithms. The main advantage of the scalable algorithm is in terms of computation time and its ability to

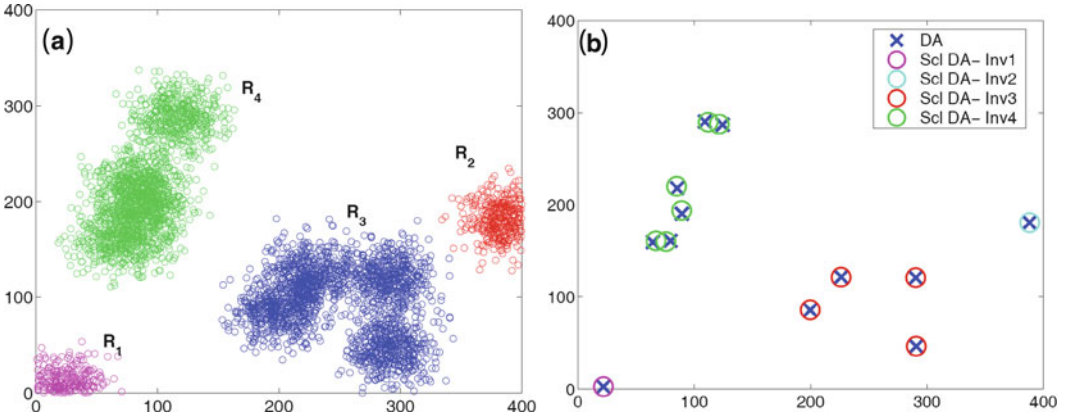


Fig. 4.4. (**a**) Separated regions $R_1, R_2, R_3$, and $R_4$ as determined by the proposed algorithm. (**b**) Comparison of lead compound locations $r_j$ and $r'_j$. Reprinted ("adapted" or "in part") with permission from *Journal of Chemical Information and Modeling*. Copyright 2008 American Chemical Society.

**Table 4.1**
**Comparison between the original and proposed algorithm**

| Algorithm | Distortion | Computation time (s) |
|---|---|---|
| The original DA | 300.80 | 129.41 |
| Proposed algorithm | 316.51 | 21.53 |

Reprinted ("adapted" or "in part") with permission from *Journal of Chemical Information and Modeling*. Copyright 2008 American Chemical Society

handle larger datasets. The results from the two algorithms are presented in **Table 4.1**. As can be seen, the proposed scalable algorithm takes just about 17% of the time used by the original (nonscalable) algorithm and results in only a 5.2% increase in distortion; this was obtained for $\varepsilon = 0.005$. Both the algorithms were terminated when the number of lead compounds reached 12. The computation time for the scalable algorithm can be further reduced (by changing ε), but at the expense of increased distortion.

*5.2.1. Further Examples*

The scalable algorithm was applied to a number of different datasets. Results for three such cases have been presented in **Fig. 4.5**. The dataset in Case 2 is comprised of six randomly chosen cluster centers with 1,000 points each. All the points were assigned equal weights (i.e., $p(x_i) = \frac{1}{N}$ for all $x_i \in \Omega$). **Figure 4.5a** shows the dataset and the eight lead compound locations obtained by the proposed scalable algorithm. The dataset in Case 3 is also comprised of eight randomly chosen cluster locations with 1,000 points each. Both the algorithms were executed till they identified eight lead compound locations in the underlying dataset. Case 4 is comprised of two cluster centers with 2,000

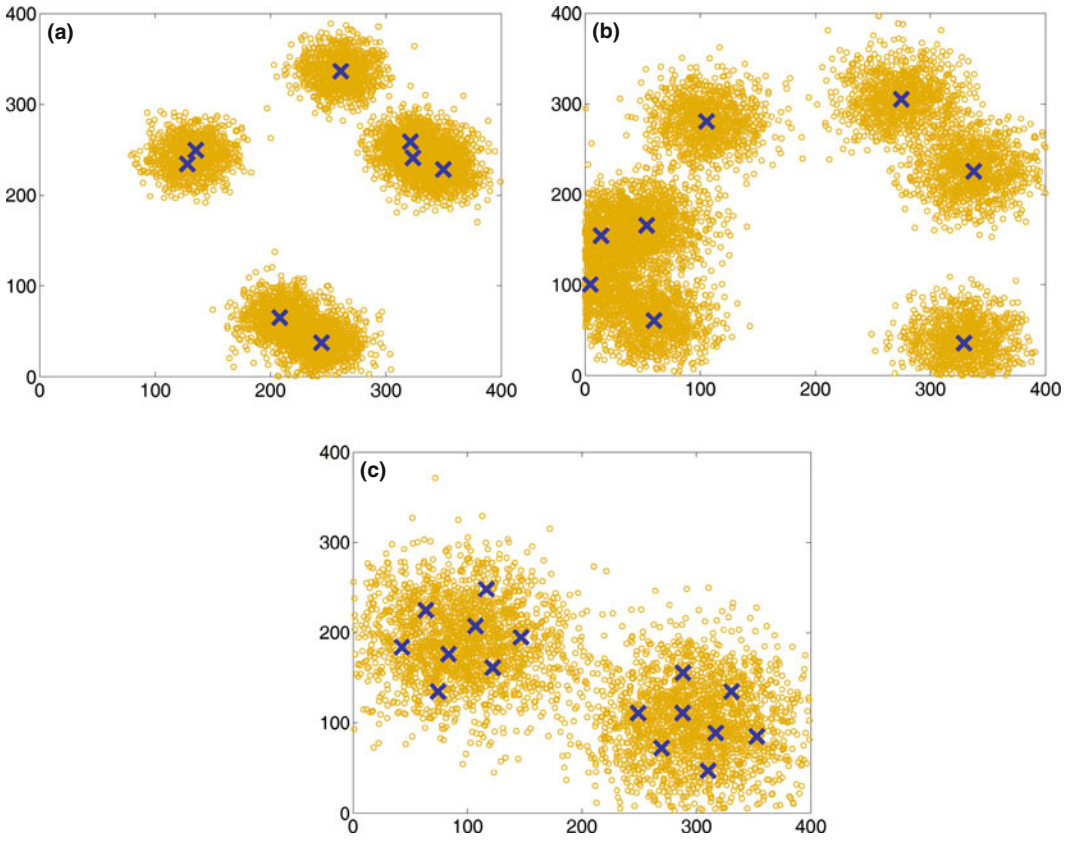


Fig. 4.5. (**a**, **b**, **c**) Simulated dataset with locations $x_i$ of compounds (*circles*) and lead compound locations $r_j$ (*crosses*) determined by the algorithm. Reprinted ("adapted" or "in part") with permission from *Journal of Chemical Information and Modeling*. Copyright 2008 American Chemical Society.

**Table 4.2**
**Distortion and computation times for different datasets**

| Case | Algorithm | Distortion | Computation time (s) |
|---|---|---|---|
| Case 2 | The original DA | 290.06 | 44.19 |
| | Proposed algorithm | 302.98 | 11.98 |
| Case 3 | The original DA | 672.31 | 60.43 |
| | Proposed algorithm | 717.52 | 39.77 |
| Case 4 | The original DA | 808.83 | 127.05 |
| | Proposed algorithm | 848.79 | 41.85 |

Reprinted ("adapted" or "in part") with permission from *Journal of Chemical Information and Modeling*. Copyright 2008 American Chemical Society

points each. Both the algorithms were executed till they identified 16 lead compound locations. Results for the three cases have been presented in **Table 4.2**.

It should be noted that both the algorithms were terminated after a specific number of lead compound locations had been identified. The proposed algorithm took far less computation time when compared to the original algorithm while maintaining less than 5% error in distortion.

### 5.3. Drug Discovery Dataset

This dataset is a modified version of the test library set (19). Each of the 50,000 members in this set is represented by 47 descriptors which include topological, geometric, hybrid, constitutional, and electronic descriptors. These molecular descriptors are computed using the Chemistry Development Kit (CDK) Descriptor Calculator (20, 21). These 47-dimensional data were then normalized and projected onto a two-dimensional space. The projection was carried out using Principal Component Analysis. Simulations were completed on this two-dimensional dataset. The proposed scalable algorithm was used to identify 25 lead compound locations from this dataset (*see* **Fig. 4.6**). The algorithm gave higher weights at locations which had larger numbers of similar compounds. Maximally diverse compounds are identified with a very small weight. The original version of the algorithm could not complete the computations for this dataset (on a 512 MB RAM 1.5 GHz Intel Centrino processor).

### 5.4. Additional Constraints on Lead Compounds

As was discussed in **Section 3**, the multiobjective framework of the proposed algorithm allows us to incorporate additional constraints in the selection problem. In this section, we have addressed two such constraints, namely the experimental resources constraint and the exclusion/inclusion constraint.

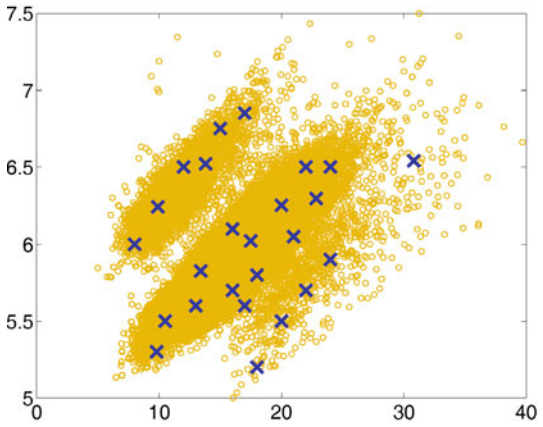


Fig. 4.6. Choosing 25 lead compound locations from the drug discovery dataset. Reprinted ("adapted" or "in part") with permission from *Journal of Chemical Information and Modeling*. Copyright 2008 American Chemical Society.

*5.4.1. Constraints on Experimental Resources*

In this dataset, the VCL is divided into three classes based on the experimental supplies required by the compounds for testing, as shown in **Fig. 4.7a** by different symbols. It contains a total of 280 compounds with 120 of the first class (denoted by circles), 40 of the second class (denoted by squares), and 120 of the third class (denoted by triangles). We incorporate experimental supply constraints into the algorithm by translating them into direct constraints on each of the lead compounds. With these experimental supply constraints, the algorithm was used to select 15 lead compound locations ($r_j$) in this dataset with capacities ($W_{jn}$) fixed for

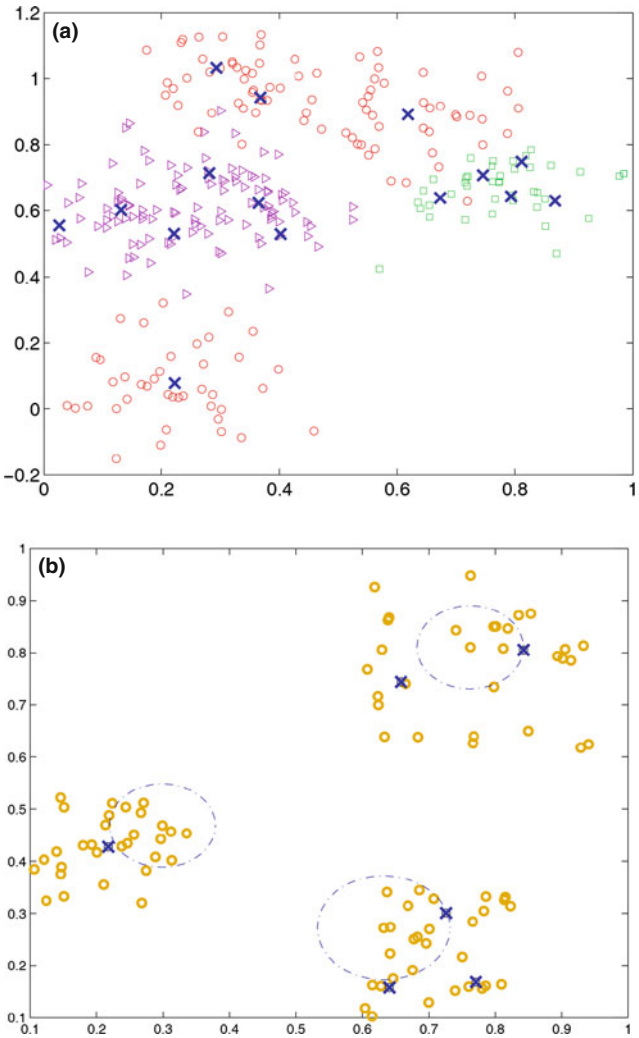


Fig. 4.7. (**a**) Simulation results with constraints on experimental resources. (**b**) Simulation results with exclusion constraint. The locations $x_i, 1 \leq i \leq 90$, of compounds (*circles*) and $r_j, 1 \leq j \leq 6$, of lead compounds (*crosses*). *Dotted circles* represent undesirable properties. Reprinted ("adapted" or "in part") with permission from *Journal of Chemical Information and Modeling*. Copyright 2008 American Chemical Society.

each class of resource. The crosses in **Fig**. **4.7a** represent the selection from the algorithm in the wake of the capacity constraints for different types of compounds. As can be seen from the selection, the algorithm successfully addressed the key issues of diversity and representativeness together with the constraints that were placed due to experimental resources.

*5.4.2. Constraints on Exclusion and Inclusion of Certain Properties*

There may arise scenarios where we would like to inhibit selection of compounds exhibiting properties within certain prespecified ranges. This constraint can be easily incorporated in the cost function by modifying the distance metric used in the problem formulation. Consider a case in a 2-d dataset where each point $x_i$ has an associated radius (denoted by $\chi_{ij}$). The selection problem is the same, but with the added constraint that all the selected lead compounds $(r_j)$ must be at least $\chi_{ij}$ distance removed from $x_i$. The proposed algorithm can be modified to solve this problem by defining the distance function, given by $d(x_i, r_j) = \left(\|x_i - r_j\| - \chi_{ij}\right)^2$, which penalizes any selection $(r_j)$ which is in close proximity to the compounds in the VCL. For the purpose of simulation, a dataset was created with 90 compounds $(x_i, i = 1, \ldots, 90)$. The dotted circle around the locations $x_i$ denotes the region in the property space that is to be avoided by the selection algorithm. The objective was to select six lead compounds from this dataset such that the criterion of diversity and representativeness is optimally addressed in the selected subset. The selected locations are represented by crosses. From **Fig**. **4.7b**, note that the algorithm identifies the six clusters under the constraint that none of the cluster centers are located in the undesirable property space (denoted by dotted circles).

## 6. Conclusions

In this chapter, we proposed an algorithm for the design of lead-generation libraries. The problem was formulated in a constrained multiobjective optimization setting and posed as a resource allocation problem with multiple constraints. As a result, we successfully tackled the key issues of diversity and representativeness of compounds in the resulting library. Another distinguishing feature of the algorithm is its scalability, thus making it computationally efficient as compared to other such optimization techniques. We characterized the level of interaction between various clusters and used it to divide the clustering problem with huge data size into manageable subproblems with small size. This resulted in significant improvements in the computation time and enabled the algorithm to be used on larger sized datasets. The trade-off between computation effort and error due to truncation is also characterized, thereby giving an option to the end user.

## References


1. Gordon, E. M., Barrett, R. W., Dower, W. J., Fodor, S. P. A., Gallop, M. A. (1994) Applications of combinatorial technologies to drug discovery. 2. Combinatorial organic synthesis, library screening strategies, and future directions. *J Med Chem* **37**(10), 1385–1401.
2. Blaney, J., Martin, E. (1997) Computational approaches for combinatorial library design and molecular diversity analysis. *Curr Opin Chem Biol* **1**, 54–59.
3. Willett, P. (1997) Computational tools for the analysis of molecular diversity. *Perspect Drug Discov Design*, **7/8**, 1–11.
4. Rassokhin, D. N., Agrafiotis, D. K. (2000) Kolmogorov-Smirnov statistic and its applications in library design. *J Mol Graph Model* **18**(4–5), 370–384.
5. Lipinski, C. A., Lomabardo, F., Dominy, B. W., Feeny, P. J. (1997) Experimental and computational approaches to estimate solubility and permeability in drug discovery and development setting. *Adv Drug Del Review* **23**, 2–25.
6. Higgs, R. E., Bemis, K. G., Watson, I. A., Wikel, J. H. (1997) Experimental designs for selecting molecules from large chemical databases. *J Chem Inf Comput Sci* **37**, 861–870.
7. Clark, R. D. (1997) Optisim: an extended dissimilarity selection method for finding diverse representative subsets. *J Chem Inf Comput Sci* **37**(6), 1181–1188.
8. Agrafiotis, D. K., Lobanov, V. S. (2000) Ultrafast algorithm for designing focussed combinatorial arrays. *J Chem Inf Comput Sci* **40**, 1030–1038.
9. Salapaka, S., Khalak, A. (2003) Constraints on locational optimization problems. *Proceedings of the IEEE Control and Decisions Conference*. Maui, HI, 9–12 December 2003, pp. 1741–1746.
10. Sharma, P., Salapaka, S., Beck, C. (2008) A scalable approach to combinatorial library design for drug discovery. *J Chem Inf Model* **48**(1), 27–41.
11. Gersho, A., Gray, R. (1991) *Vector Quantization and Signal Compression*. Kluwer, Boston, Massachusetts.
12. Drezner, Z. (1995) Facility location: a survey of applications and methods. *Springer Series in Operations Research*, Springer, New York.
13. Du, Q., Faber, V., Gunzburger, M. (1999) Centroidal Voronoi tessellations: applications and algorithms. *SIAM Rev* **41**(4), 637–676.
14. Therrien, C. W. (1989) *Decision, Estimation and Classification: An Introduction to Pattern Recognition and Related Topics*, 1st ed. Wiley, New York.
15. Haykin, S. (1998) *Neural Networks: A Comprehensive Foundation*, Prentice Hall, Englewoods Cliffs, NJ.
16. Gray, R., Karnin, E. D. (1982) Multiple local minima in vector quantizers. *IEEE Trans Inform Theor* **28**, 256–361.
17. Lloyd, S. P. (1982) Least squares quantization in PCM. *IEEE Trans Inform Theory* **28(2)**, 129–137.
18. Rose, K. (1998) Deterministic annealing for clustering, compression, classification, regression and related optimization problems. *Proc IEEE* **86**(11), 2210–2239.
19. Mcmaster hts lab competition. HTS data mining and docking competition. http://hts.mcmaster.ca/downloads/82bfbeb4-f2a4-4934-b6a8-804cad8e25a0.html (accessed June 2006).
20. Guha, R. (2006) Chemistry Development Kit (CDK) descriptor calculator GUI (v 0.46). http://cheminfo.informatics.indiana.edu/rguha/code/java/cdkdesc.html (accessed October 2006).
21. Steinbeck, C., Hoppe, C., Kuhn, S., Floris, M., Guha, R., Willighagen, E. L. (2006) Recent developments of the Chemistry Development Kit (CDK) – an open-source JAVA library for chemo and bioinformatics. *Curr Pharm Des* **12**(17), 2110–2120.