Enhancing Molecular Discovery Using Descriptor-Free Rearrangement Clustering Techniques for Sparse Data Sets

ARTICLE *in* AICHE JOURNAL · JANUARY 2009

Impact Factor: 2.75 · DOI: 10.1002/aic.11986

Genyuan Li
Princeton University
54 PUBLICATIONS **1,554** CITATIONS
SEE PROFILE

Herschel Rabitz
Princeton University
947 PUBLICATIONS **23,789** CITATIONS
SEE PROFILE

Enhancing Molecular Discovery Using Descriptor-Free Rearrangement Clustering Techniques for Sparse Data Sets

Peter A. DiMaggio Jr., Scott R. McAllister, and Christodoulos A. Floudas
Dept. of Chemical Engineering, Princeton University, Princeton, NJ 08544

Xiao-Jiang Feng, Genyuan Li, Joshua D. Rabinowitz, and Herschel A. Rabitz
Department of Chemistry, Princeton University, Princeton, NJ 08544

DOI 10.1002/aic.11986
Published online September 16, 2009 in Wiley InterScience (www.interscience.wiley.com).

This article presents a descriptor-free method for estimating library compounds with desired properties from synthesizing and assaying minimal library space. The method works by identifying the optimal substituent ordering (i.e., the optimal encoding integer assignment to each functional group on every substituent site of molecular scaffold) based on a global pairwise difference metric intended to capture smoothness of the compound library. The reordering can be accomplished via a (i) mixed-integer linear programming (MILP) model, (ii) genetic algorithm based approach, or (iii) heuristic approach. We present performance comparisons between these techniques as well as an independent analysis of characteristics of the MILP model. Two sparsely sampled data matrices provided by Pfizer are analyzed to validate the proposed approach and we show that the rearrangement of these matrices leads to regular property landscapes which enable reliable property estimation/interpolation over the full library space. An iterative strategy for compound synthesis is also introduced that utilizes the results of the reordered data to direct the synthesis toward desirable compounds. We demonstrate in a simulated experiment using held out subsets of the data that the proposed iterative technique is effective in identifying compounds with desired physical properties. © 2009 American Institute of Chemical Engineers *AIChE J,* 56: 405–418, 2010
Keywords: optimization, mathematical modeling

Introduction

Drug discovery is a tedious and expensive process, involving several phases from target identification to clinical trials.[1] One of the bottlenecks in this process is the identification of potential drug compounds, normally small organic molecules or peptides, that can achieve multiple desired biological properties.[2] Finding such lead molecules can be highly difficult even with the assistance of combinatorial chemistry and high-throughput screening.[3,4] For example, if a single molecular scaffold has N substituent sites with S distinct functional groups that may be attached at each site, then an absolute bound on the total number of compounds in this library would be S^N (one should note that this bound is typically less in practice because bound type restrictions limit the pairing of functional groups). As S and N increases, it quickly becomes impractical to synthesize and assay all the library compounds.

A common practice is to use quantitative structure-activity relationship (QSAR) methods[5–8] to computationally predict the biological properties of the library compounds (or at

Additional Supporting Information may be found in the online version of this article.
Correspondence concerning this article should be addressed to C. A. Floudas at floudas@titan.princeton.edu

least to serve as a screening and enrichment tool to eliminate chemicals that are unlikely to have drug-like properties). All existing methods for constructing predictive QSAR models involve three basic steps[7]: (1) synthesize and assay a training set of chemicals, (2) select physical/chemical/structural descriptors that can best relate to the biological properties, and (3) construct mathematical functions that quantitatively describe and predict the biological properties by these descriptors. In practice, the reliability of these methods depends critically on the choice of suitable descriptors in step (2). It should be noted here that recently developed supervised classification methods based on mixed-integer linear programming[9,10] have been shown to work well for the descriptor selection problem. In an unsupervised approach, the success of these methods is highly dependent on appropriate input from the user, and hence some level of user expertise.

We have proposed an adaptive substituent reordering and interpolation algorithm for estimating compound properties in combinatorial libraries from the synthesis and assaying of a small number of randomly sampled library compounds.[11,12] Fundamentally, both QSAR and this reordering algorithm are based on the assumption that there exists an underlying physical regularity in the compound library, and this regularity can be revealed from the structure-property relationships from sampling a subset of the library compounds. However, unlike any other QSAR methods, the reordering algorithm does not require any physical/chemical/structural descriptor to operate, which makes it a very general and easy-to-apply technique for drug discovery and other molecular discovery purposes.

For a single-scaffold compound library with N substitution sites and S_n ($n = 1,2,\ldots,N$) substituents (functional groups) at the nth site, any compound in the library can be represented by a N-dimensional vector $\mathbf{X} = \{X_1,X_2,\ldots,X_N\}$, where X_n is the index of the substituent at the n-th site and the value of X_n is an integer between 1 and S_n. As a result, a biological property y of this compound is an unknown nonlinear descriptor-free N-variable function

$$y = f(\mathbf{X}), \tag{1}$$

which can be approximated by other parameterized functions $g(\mathbf{X}) \approx f(\mathbf{X})$ (e.g., radial basis functions) using data collected on a small subset of M compounds in the library. $g(\mathbf{X})$ can then be utilized to estimate/interpolate the property of the unsampled compounds in the library. The reliability in approximating $f(\mathbf{X})$ by $g(\mathbf{X})$ and the subsequent predicative capability of $g(\mathbf{X})$ depend on the regularity of $f(\mathbf{X})$ and $g(\mathbf{X})$ as a function of $\mathbf{X}$, which in turn is determined by the integer assignment (i.e., the value of X_i) given to each substituent on the nth site. A key component of the algorithm is then to identify the optimal integer assignments (i.e., the optimal substituent ordering) for all substituents on all N sites so that $g(\mathbf{X})$ can correctly reveal the underlying regularity of the whole library space. This is not an easy task because the total number of possible orderings is $S_1! \times S_2! \times \ldots \times S_N!$ if each of the N sites is ordered independently. In addition, the relationship between $g(\mathbf{X})$ and the orderings can be highly complicated and not amenable to derivative-based optimization algorithms. In previous studies, the optimal orderings were identified by using search algorithms that either maximize the smoothness of $g(\mathbf{X})$[11] or minimize the root-mean-squared difference between interpolated and actual properties of the M sampled compounds.[12] Proof-of-principle studies were performed on the laboratory data of a copolymer library[11] and a transition metal complex library,[12] both of which demonstrated excellent predicative capability of the algorithm over the whole library space.

In this article, we introduce a new strategy for efficient substituent reordering and descriptor-free property estimation. The method views substituent reordering as a special high-dimensional rearrangement clustering problem, which eliminates the need for functional approximation and enhances computational efficiency. In comparison with functional interpolation methods, clustering techniques can be more reliable for pattern recognition in the presence of considerable data noise and therefore would be better suited for drug candidate discovery where the focus is more on identifying a subset of potential drug candidates with desired properties rather than precise quantitative property predictions. Various techniques have already been developed for the rearrangement clustering of high-dimensional data.[13–17] These techniques have important applications in clustering ensembles of structures from free energy calculations of oligopeptides[18–20] or proteins,[21,22] de novo protein design sequences,[23,24] and design and scheduling of batch processes.[25,26] An important prerequisite for these existing rearrangement clustering methods is that the data matrix is dense and contains only a few missing elements. Evidently, this limitation makes these techniques inapplicable for computational drug discovery, where the goal is to make predictions on copious amounts of missing data.

The organization of the article is as follows. In the Methods section, we present a global pairwise similarity metric to represent the smoothness of the compound property space and introduce deterministic, stochastic, or heuristic search algorithms to identify the best substituent orderings with respect to this smoothness. The proposed approach is then applied to two sparsely sampled compound libraries provided by Pfizer in the Computational Studies section. The total number of possible compounds in these libraries is 2418 and 14043, respectively. In the Computational Studies section, we show that the proposed methods provide excellent predictions of the library subspace that is densely populated by compounds with desired properties from sampling as low as 15% of the whole library space. A synthesis strategy is then presented in the Iterative Synthesis Strategy section, which iterates between synthesizing a batch of compounds and using the reordering techniques to guide the synthesis of the next batch of compounds. We demonstrate that this synthesis strategy is effective in identifying lead molecules while simultaneously minimizing the total number of sampled library compounds required. Finally, in Computational Analysis of the MILP Model section, we analyze the performance characteristics of the MILP model for solving this class of problems.

Methods

An approach for descriptor-free substituent ordering in compound libraries is presented in this section. We first

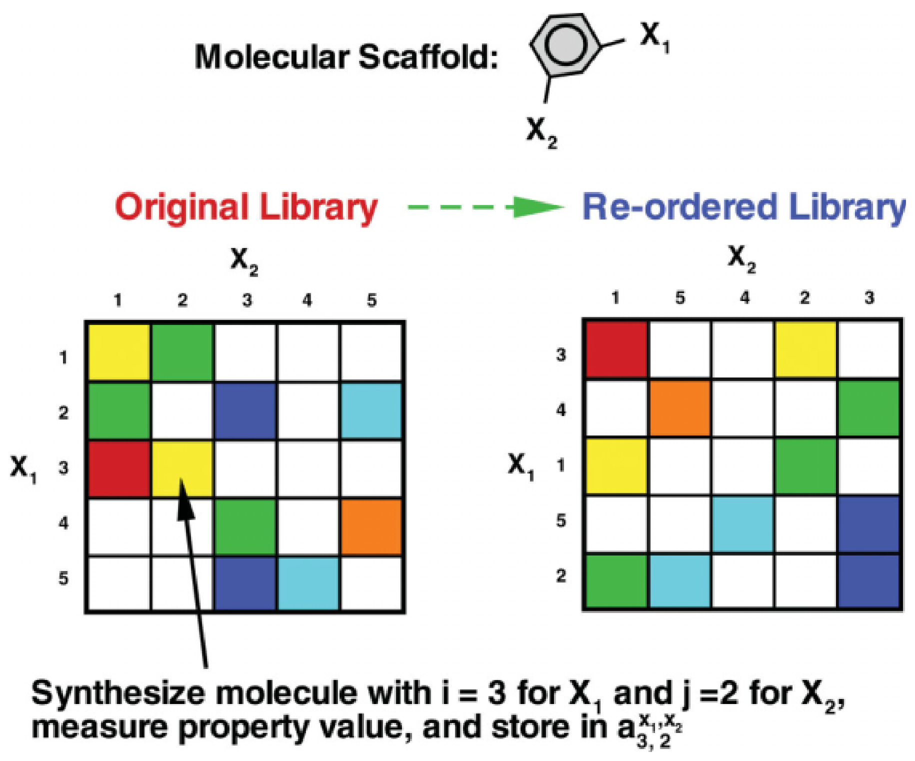
Molecular Scaffold:
X1
X2
Original Library
Re-ordered Library
X2
1 2 3 4 5
X1
1 2 3 4 5
X2
1 5 4 2 3
X1
3 4 1 5 2
Synthesize molecule with i = 3 for X1 and j =2 for X2, measure property value, and store in a^{X1,X2}_{3,2}

Figure 1. Hypothetical molecular compound library with two substituent sites and five possible functional groups.

Each value of i for X_1 and j for X_2 corresponds to a unique molecular compound, which can be synthesized and tested for some property value, such as percent inhibition, and this property value is then stored in the matrix. The values for the properties are presented on a color scale, where the colors red, organge, yellow, green, cyan, and blue span from highest to lowest. The underlying assumption is that this library can be rearranged some how to reveal an approximately regular landscape with respect to the property values, as shown by the reordered matrix on the right-hand side. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

describe an objective function that provides a measure of the smoothness of the compound property space. Three methods have been developed to identify the best substituent orderings given this measure. Deterministic optimization approaches based on mixed-integer linear programming can provide guaranteed convergence to the optimal substituent ordering according to the given objective function. A genetic algorithm is a stochastic optimization approach that provides good quality solutions which are frequently close to the optimal solution, but cannot provide theoretical guarantees for the quality of the solution. Heuristic approaches, such as random row and column swapping, have the potential to quickly identify good solutions to this problem.

Objective function

A suitable objective function first needs to be devised to quantify the smoothness of the property space in a high-dimensional sparse compound library for a given substituent ordering. For illustrative purposes, consider a library with two substituent sites and five possible functional groups, as shown in Figure 1. The functional groups (substituents) are uniquely represented by the integers $ID = \{1,2,3,4,5\}$ and a compound corresponds to any selection $i \in ID$ for X_1 and $j \in ID$ for X_2. For instance, $i = 3$ for X_1 and $j = 2$ for X_2 correponds to a molecular compound with those two substituent groups, as shown in Figure 1. Given a molecular scaffold and the specification of the functional groups 3 and 2 for the two substituent sites, we can then synthesize this molecule and measure any physical, chemical, or biological properties, and store it in element $a^{X1,X2}_{3,2}$. The numerical studies presented in this article corresponds to percent inhibition of some target protein and IC_{50} data, but the method can in principle be applied to any set of property values. Furthermore, there is no restriction that only one value is measured; the method can also address vectors of values corresponding to several properties (sometimes referred to as featue vectors for a given molecule). Note that the synthesis of all possible 25 molecules in Figure 1 is typically a cost-prohibitive procedure, so we will only have available to us a subset of the values corresponding to $a^{X1,X2}_{i,j}$, where 11 of the 25 values (44%) are known in this figure. The fundamental assumption here is that the rows i and columns j of this compound library can be rearranged in such a way that a fairly smooth surface in the property dimension is revealed. The corresponding reordered library for our example is presented in Figure 1, where we see that largest values (i.e., the red, orange, and yellow values, in decreasing order of magnitude) are grouped in the upper-left triangle of the matrix and the smallest values (i.e., the green, cyan, and blue values, in decreasing order of magnitude) in the bottom-right triangle of the matrix. Thus, we need an expression for measuring the "quality" of a particular library reordering (e.g., permutation of the rows and columns of the matrix).

Different objective functions can be used to quantify the smoothness of the compound property space. Here, we choose a pairwise similarity measure Q that depends both on the compound properties and the distance between them in the reordered dimension. This function can be generally written as

$$Q = \sum_{m=1}^{N} \sum_{n=1,n\neq m}^{N} \sum_{j=1}^{S_n} \sum_{i=1}^{S_m} \sum_{i'=1}^{S_m} \theta(d^m_{i,i'}) \cdot \phi(a^{m,n}_{i,j}, a^{m,n}_{i',j}) \quad (2)$$

where $d^m_{i,i'}$ is the distance between functional groups i and i' at the mth site with respect to their final positions in the matrix. This distance value value will be 1 if the two functional groups are adjacent to each other and $S_m - 1$ if they are on opposite ends of the data matrix. $a^{m,n}_{i,j}$ and $a^{m,n}_{i',j}$ denote the measured property values of two sampled compounds for functional group j in site n and functional groups i and i' in site m, respectively. One possible form of the component functions $\theta(d^m_{i,i'})$ and $\phi(a^{m,n}_{i,j},a^{m,n}_{i',j})$ can be written as

$$Q = \sum_{m=1}^{N} \sum_{n=1,n\neq m}^{N} \sum_{j=1}^{S_n} \sum_{i=1}^{S_m} \sum_{i'=1}^{S_m} \left(\frac{1}{w^m} \cdot \frac{S_m - d^m_{i,i'}}{S_m - 1}\right) \cdot \left(a^{m,n}_{i,j} - a^{m,n}_{i',j}\right)^2 \quad (3)$$

where $\theta(d^m_{i,i'})$ is linear with respect to $d^m_{i,i'}$, achieving a maximum value of $1/w^m$ at $d^m_{i,i'} = 1$ and a minimum value of $1/w^m \cdot 1/(S_m - 1)$ at $d^m_{i,i'} = S_m - 1$. Thus, this expression gives the largest contributions to those elements which are grouped close in the final arrangement and a lower weight to those elements that are distant from one another in the final matrix ordering. $\phi(a^{m,n}_{i,j},a^{m,n}_{i',j})$ is the squared property difference between two compounds. w^m is the number of compound pairs where both $a^{m,n}_{i,j}$ and $a^{m,n}_{i',j}$ are available from synthesis and property assaying for all i and i'.

The objective function Q quantifies the summed property difference between all pairs of rows for a given substituent ordering. Hence, the goal is to find the optimal row orderings that minimize Q. Other forms of Q could be introduced if problem-specific details suggest they would be more appropriate.

Deterministic optimization based on mixed-integer linear programming

For simplicity of model presentation, we assume that the library has two substitution sites (i.e., $N = 2$). In this case, the library becomes a two-dimensional matrix, where a row represents a substituent on site 1 being fixed and a column corresponds to a substituent on site 2 being fixed. Throughout this section, rows and columns correspond to functional groups on distinct scaffold sites. The basic model can be formulated as an assignment problem, where binary variables, $y_{i,k}$, indicate the assignment of a particular row, i, to a position in the final ordering, $1 \leq k \leq |I|$, that is:

$$y_{i,k} = \begin{cases} 1, & \text{if row } i \text{ is assigned to position} \\ & k \text{ in the final ordering} \\ 0, & \text{otherwise} \end{cases}$$

An intuitive constraint to impose on the row assignments is that a final position can contain only one row, and a row can be assigned to at most one final position. This is given by Eqs. 4 and 5.

$$\sum_k y_{i,k} = 1 \quad \forall i \quad (4)$$

$$\sum_i y_{i,k} = 1 \quad \forall k \quad (5)$$

We can represent the final position of row i in the matrix as a positive variable, p_i, and relate $y_{i,k}$ to p_i using a simple equality constraint as shown in Eq. 6.

$$p_i = \sum_k k \cdot y_{i,k} \quad \forall i \quad (6)$$

These positions can range from 1 to the total number of rows, $|I|$, which is represented by the bounds given in Eq. 7.

$$1 \leq p_i \leq |I| \quad \forall i > 1 \quad (7)$$

Issues related to problem symmetry can be alleviated by restricting the final assignment of row 1 to occupy only one half of the final matrix, as shown in Eq. 8.

$$1 \leq p_1 \leq \lfloor |I| + 1/2 \rfloor \quad (8)$$

From these final positions, we can also define the distance between any two rows i and i' in the final arrangement by the positive variable $d_{i,i'}$. The distance between two final positions i and i' is given by the nonlinear equation $d_{i,i'} = |p_i - p_{i'}|$. However, exact lower bounds on this distance can be represented by two linear inequality constraints, as shown in Eqs. 9 and 10.

$$d_{i,i'} \geq p_i - p_{i'} \quad \forall i < i' \quad (9)$$

$$d_{i,i'} \geq p_{i'} - p_i \quad \forall i < i' \quad (10)$$

Any distance must be at least 1 and cannot be greater than the total number of rows minus 1 ($|I| - 1$), which results in the bounds in Eq. 11.

$$1 \leq d_{i,i'} \leq |I| - 1 \quad \forall i, i' > i \quad (11)$$

Generating tight upper bounds on the distance variables is not straightforward and as a result the mixed-integer linear optimization problem can be extremely difficult to solve. Specifically, the resulting linear programming relaxations may not be tight and significant computational effort is required for the branch and bound tree to find and/or prove the optimal solution. However, several additional constrains can be derived to help bound and restrict the distance variables to reasonable values. One such constraint utilizes the fact that the summation of all the final distances is equal to a known constant, C, as shown in Eq. 12.

$$\sum_i \sum_{i'>i} d_{i,i'} = C \quad (12)$$

For instance, if there are only four rows, then $C = 3 \cdot 1 + 2 \cdot 2 + 1 \cdot 3 = 10$. Another constraint provides a valid upper bound on the distance by incorporating information about the final position of a row assignment. If row i is assigned to final position k, then the maximum distance between row i and any other row i' must be less than $MAX(|I| - k, k - 1)$, which is the maximum distance to either end of the matrix from position k. We represent $MAX(|I| - k, k - 1)$ as the function $F(k)$. This constraint is expressed in generalized form in Eq. 13 and 14.

$$d_{i,i'} \leq \sum_k F(k) \cdot y_{i,k} \quad \forall i < i' \tag{13}$$

$$d_{i,i'} \leq \sum_k F(k) \cdot y_{i',k} \quad \forall i < i' \tag{14}$$

Another constraint on the distance variables can be derived from the fact that once row i has been assigned to some final position k, then the sum of the distances from position k to all other positions is known, which we represent as a general function $G(k)$. For instance, if a row i is assigned to final position 2 in a four-row problem, then the sum of its distances to all other rows i' in the final arrangement is $(2 - 1) + (3 - 2) + (4 - 2) = 4 = G(2)$. The general form for this constraint is presented in Eq. 15.

$$\sum_{i'>i} d_{i,i'} + \sum_{i'<i} d_{i',i} = \sum_k G(k) \cdot y_{i,k} \quad \forall i \tag{15}$$

As the distances are Euclidean, we can impose the restriction that they satisfy the triangle inequality constraint in Eq. 16.

$$d_{i,i'} \leq d_{i,i''} + d_{i'',i} \quad \forall i, i', i'' \tag{16}$$

However, this results in $O(|I|^3)$ additional constraints and leads to memory issues for standard commercial solvers for even moderately sized problems. To circumvent this memory issue, we introduce these constraints dynamically during the program execution as cuts. In other words, we include only those triangle inequality constraints for which Eq. 16 is violated and this ensures that only the most necessary constraints are added to the problem. Another set of constraints that bounds the distances with respect to a fixed point, $1 \leq p^* \leq |I|$, are introduced as cuts into the problem, where p^* is chosen during program execution.

$$d_{i,i'} \leq \sum_k |k - p^*| \cdot \left(y_{i,k} + y_{i',k}\right) \quad \forall i, i' > i, p^* \tag{17}$$

The proposed mixed-integer linear programming formulation, which is the minimization of Eq. 3 subject to Eq. 4 through 17, when solved to optimality can guarantee that it has found the best ordering for the given objective function in Eq. 3. The largest reordering problem we have solved to optimality has contained about 90 rows. However, even though this method may not close the gap between the linear programming relaxation and the best integer solution, it is a very effective technique for establishing a lower bound on the value of the objective function. It should be mentioned that heuristic approaches (to be described in the subsequent section) can be integrated into the deterministic optimization framework for generating good integer solutions.

Stochastic optimization based on genetic algorithms

The minimization of Q can also be accomplished via a stochastic global search, such as a genetic algorithm (GA). GAs are intuitive and simple to operate and hence are generally applicable to a wide variety of problems. The underlying mechanism of the GA, known as evolution, starts from a random population of substituent orderings. The Q value of each substituent ordering in the population is evaluated and several orderings are stochastically selected from the current population based on these Q values. Operations known as mutations and crossovers between these selected substituent orderings are performed to generate a new population, which is used in the next iteration of the algorithm. This process continues until convergence is reached or satisfactory solutions are found.

The computational expense associated with the genetic algorithm is dependent on the size of the population used, where larger populations typically generate better solutions but are subject to greater computational requirements. Extensive research has been conducted to integrate GAs with local, deterministic search algorithms to enable optimal balance between global search capability and operational efficiency. The genetic algorithm requires prior selection of several parameters that are problem dependent and can produce local optima, which are close to the global optimum in certain cases. It does not offer theoretical guarantee of convergence to the global optimum in the rearrangement clustering problem.

Heuristics-based approaches

The minimization of the objective function given in Eq. 3 is amenable to a number of heuristic algorithms. The simplest heuristic is the random swapping of rows and columns (Heuristic 1: Row swap). After each swap, the objective function is evaluated and the swap is accepted if it results in a lower objective function value. The acceptance criteria of this operation could be altered to allow for objective function increases based on a probability (a basic Monte Carlo approach), but the utilization of several initial points was found to be effective. Given an initial ordering of the rows or columns, this basic random swapping operation is a reasonable local minimization approach.

The initial configuration to minimize using swapping operations can be generated using a variety of other heuristic techniques. A valid ordering can be created through a greedy assembly by repeatedly selecting rows or columns to be neighbors based on the minimum distance between unconnected elements (Heuristic 2: Greedy). A third technique randomly selects a few points from each row or column, and orders the rows or columns based on the sum of these selected points (Heuristic 3: Pick sort). A final heuristic approach looks at the highest values in each column and generates an ordering by sorting the sum of these points (Heuristic 4: High sort). A variety of orderings are generated through each of these techniques by parametrically varying the value for a missing data point or the number of points to be selected. Each of these orderings is then subject to a local minimization via the aforementioned random swapping operations.

For certain classes of problems, such as problems with roughly single-peaked property landscapes, the smoothness of the property space can be represented as an ordering-independent function R so that reordering of the rows and columns can be achieved by less expensive computational algorithms. For example, for the ith row, the objective function R_i can be devised as

$$R_i = \sum_{i'} \sum_{j} (a_{i,j} - a_{i',j}) \tag{18}$$

which quantifies the summed property difference between each compound on the ith row and its corresponding compounds on all other rows i'. As R_i does not include any distance term $\theta(d_{i,i'})$, its value is independent of the substituent ordering of the rows. As a result, the ordering of the rows can be easily obtained by a simple sorting of all the R_i ($i = 1,2,\ldots,I$) values, which is computationally very efficient (Heuristic 5: Property R sort). Other difference measures can be devised as well, which can be represented by the form of Eq. 19.

$$R_i = \sum_{i'} \sum_{j} \hat{\phi}(a_{i,j}, a_{i',j}) \tag{19}$$

where $\hat{\phi}$ is a function denoting the distance measure between two points, $a_{i,j}$ and $a_{i',j}$, similar to ϕ in Eq. 2.

The aforementioned heuristic techniques represent effective methods to obtain good solutions to the overall ordering problem. Although these methods do not provide any deterministic guarantees on the optimality of the final ordering, empirical evidence has shown that these solutions are often within less than one percent of the optimal solution. The heuristic techniques are easily scalable to much larger problems and alternative objective functions, such as those that require a dependence between the row and column problems.

As previously mentioned, these heuristics can be integrated into the MILP model to generate good integer feasible solutions. We implemented the MILP model in the C++ programming language to interface with the ILOG Concert Technology library[27] in CPLEX. This CPLEX library allows the user to manually overload many of operations and methods used in the branch-and-bound tree created by CPLEX to solve mixed-integer linear programming problems. With this library, the user can interface the CPLEX branch-and-bound algorithm with their own heuristic methods for generating integer solutions, add problem-specific cutting plane constraints (in addition to the standard CPLEX cutting planes) to improve the linear programming (LP) relaxations, specify which variables to branch on in the branch-and-bound tree, and perform various other advanced operations. This library also allows the user to retrieve problem information that is not typically accessible through other interfaces, such as the GAMS modeling language.

We utilized the heuristic callback function in the CPLEX callable library to implement the aforementioned swapping heuristic. At a given node in the branch-and-bound tree, the linear programming relaxation values of the variables are used to generate an initial ordering that can be minimized via the swapping operations of Heuristic 1: Row swap. Specifically, the initial ordering is determined by (1) identifying the two rows with the maximum distance, $d_{i,i'}$, (2) fixing these rows to be the beginning and end of the ordering, and then (3) sorting the remaining rows based on their distances to these two fixed endpoints. The resulting ordering is then subject to a local minimization via the aforementioned random swapping operations.

Computational Studies

The MILP model was applied to two sparsely sampled data sets provided by Pfizer, where the identities of these compounds are unknown to us. Compounds in each set have a common scaffold with two substitution sites (i.e., $N = 2$). Each compound is indexed by two numbers (chosen by Pfizer in an unknown way) and no descriptor information is provided to us. The goal here is to evaluate the predictive capability of the rearrangement clustering methods on these data matrices. We assess the clustering of high inhibition compounds for both data matrices in Tables 1 and 2 and the performance characteristics for the different reordering methods in Tables 3–5.

The reordering procedure starts by choosing a randomly sampled subset of the known data values. The sparse data matrices are generated by randomly removing elements from the original matrix. The indices for each entry, (i,j), are generated using a uniform random distribution over the intervals $1,\ldots,I$ and $1,\ldots,J$. Before removing a selected element $a_{i,j}$, we check the occupancy of known values along that selected row and column. That is, we count the number of known elements across the ith row and the number of known elements across the jth column. If the number of total known elements for each is above a specified threshold count then we remove the element $a_{i,j}$ from the matrix. The logic for sampling in this fashion is that because the original sparse data matrix is not sampled in a random fashion, we would like to guide the sampling so that the overall sparsity remains as uniform as possible throughout and that elements are removed from the denser regions of the data matrix.

Data matrix 1: Moderate-size compound library

The first data matrix analyzed contains 62 rows and 39 columns of percent inhibition data for an unknown set of compounds. The most desirable compounds are the strongest inhibitors of an unknown target, which correspond to the highest percentage inhibition values in this set. The original matrix has 1229 data values out of a possible 2418 (51%).

The optimal reordering identified by the deterministic algorithm for the full set of data values is shown in Figure 2f. The results for the deterministic method applied to a sampling of 30 and 50% of the available data (or 15 and 25% of the library space) are provided in Figures 2b, and d, respectively. For predictive assessment, the placement of the unsampled data values for the reorderings for 15 and 25% of the full library are also revealed in Figures 2c, and e. Contrasting Figures 2c, and e show that increasing the sampling of data values from 15 to 25% of the library space improves the quality of both the overall ordering and the selected subregion, as expected.

The reordered data matrices show a tendency to group the desired compounds into an easily identifiable subset of the matrix. Let us define rows 1–9 and columns 1–11 of Figure 2d to be Region 1 and the remaining matrix to be Region 2. When examining the reordered compounds for 25% of the whole library space (see Table 1), we see that Region 1 contains a much higher average inhibition value and a disproportionate number of data values above inhibition values of 40 and 60. Of the 12 total compounds with greater than 60% inhibition from the full data set, six sampled and three

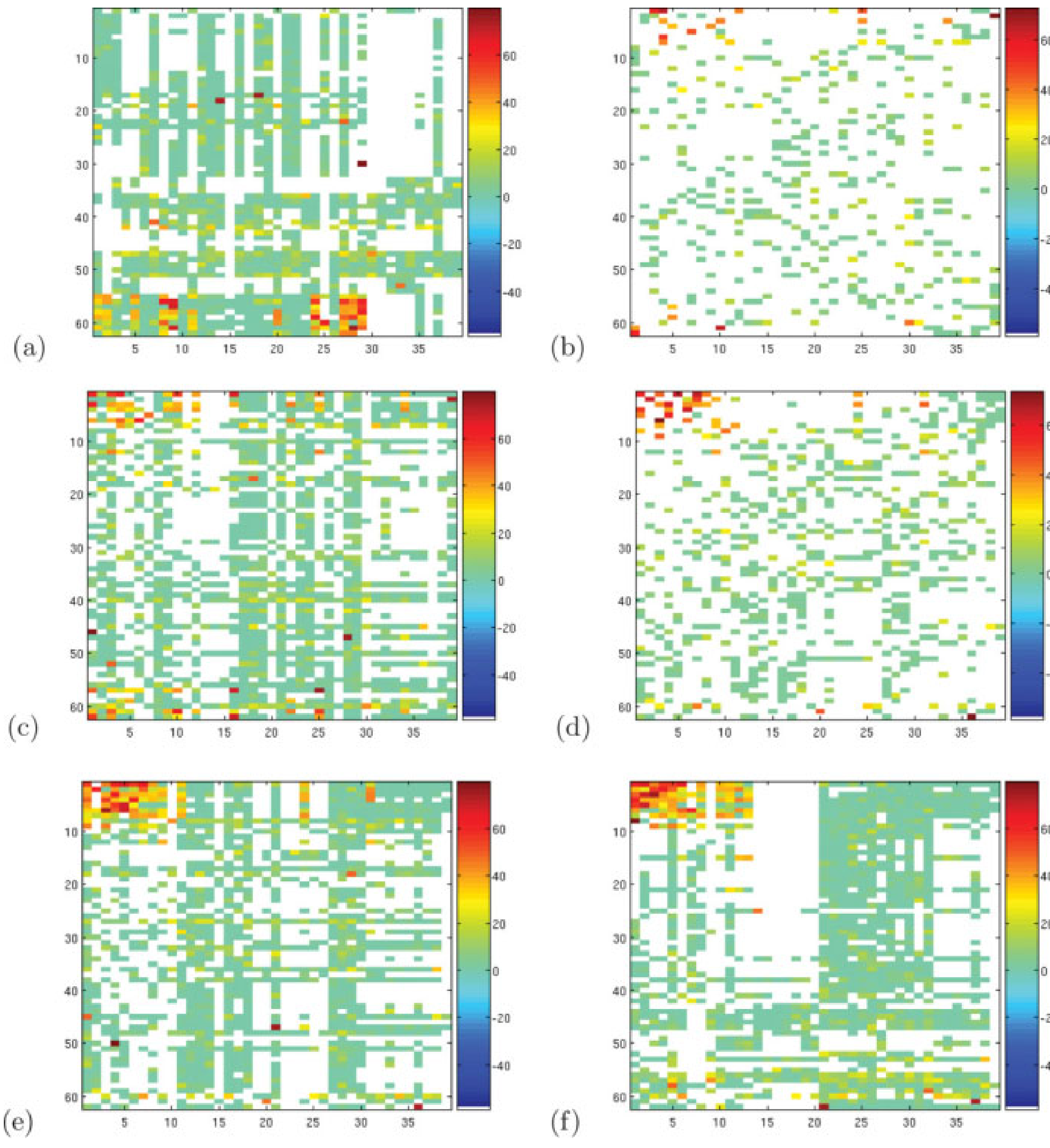

Figure 2. First data matrix reordered by deterministic optimization.

(a) Original first data matrix using all known values. (b) Optimal rearrangement of the first data matrix sampled at 15%. (c) Optimal rearrangement of the first data matrix sampled at 15%, all known values included. (d) Optimal rearrangement of the first data matrix sampled at 25%. (e) Optimal rearrangement of the first data matrix sampled at 25%, all known values included. (f) Optimal rearrangement of the first data matrix using all known values. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

unsampled compounds are found in Region 1. The consistent abundance of high inhibition values would suggest the unused compounds in Region 1 as good candidates for future synthesis.

The stochastic optimization method and the property R sorting method also result in an aggregation of the compounds with high inhibition values for the full and sampled data matrices (see Supporting Information, Figures S4 and S5, Tables S1–S4). The substituents found in Region 1 are very similar to those identified by the deterministic method but are less prominent in terms of the number of high inhibition compounds contained in Region 1.

The computational performance of the different algorithmic approaches for the full data matrix (1229 values) is compared in Table 3. In this table, the objective (Obj) column is the value from Eq. 3, the lower bound column provides a lower bound on the objective value as determined from the deterministic method, the percent gap column denotes the difference between the best lower bound and the objective value, and the CPU time (in seconds) is reported for a 3.0 GHz Intel Pentium Processor. The initial ordering of the compound library was provided by Pfizer. The solutions found by the deterministic optimization method (MILP), GA, and heuristic approaches are compared. For all the heuristics requiring random swapping operations, 100,000 swaps were performed. For all the heuristics requiring initial point selection, 40 initial points were used. The deterministic optimization approach is able to close the

Table 1. A Comparison of Two Selected Regions of the First Data Matrix Using the Optimal Ordering of the Deterministic Method from 25% Sampling to Illustrate the Clustering of High Inhibition Compounds

	Region 1	Region 1 (S)	Region 1 (U)	Region 2	Region 2 (S)	Region 2 (U)
Data values	80	34	46	1149	581	568
Average	32.9	38.8	28.5	4.5	4.2	4.8
Values ≥ 40	28 (35%)	16 (47%)	12 (26.1%)	11 (1%)	4 (0.7%)	7 (1.2%)
Values ≥ 60	9 (11.3%)	6 (17.6%)	3 (6.5%)	3 (0.3%)	1 (0.2%)	2 (0.4%)

The inhibition values correspond to percentage inhibition of an unknown target protein. Region 1 and 2 represent the full distribution of known values in the regions, whereas Region 1 (sampled) and Region 2 (sampled) correspond to the distribution of the sampled compounds in the two regions.

integrality gap in both the column and the row problems, providing a theoretical guarantee that the solution in Figure 2f has the lowest obtainable value for the objective function defined in Eq. 3. Several of the heuristic approaches were able to obtain the best possible solutions to the problem in a fraction of the time, but do not offer optimality guarantee because they only provide an upper bound. The GA was unable to identify the best solution and its required computational time is longer.

Data matrix 2: Large-size compound library

The second data matrix contains 151 rows and 93 columns of percent inhibition data for an unknown set of compounds. The initial ordering of the compounds was provided by Pfizer. This matrix is the largest data set that was analyzed using the proposed methods. Similarly as the first data matrix, the most desirable compounds for further study are the ones with strongest inhibition. The original matrix has 4110 known data values out of a possible 14043 (i.e., 29%).

This data matrix was reordered using all known values and a sampled subset of known values to test the reliability of the approach. The best identified reordering of this data matrix using all of the known values (4110 values) is presented in Figure 3d. The best reordering for a sampling of 50% of the available data values (or 15% of the whole library space) is presented in Figure 3b and the corresponding placement of the unsampled compounds based on this reordering is revealed in Figure 3c.

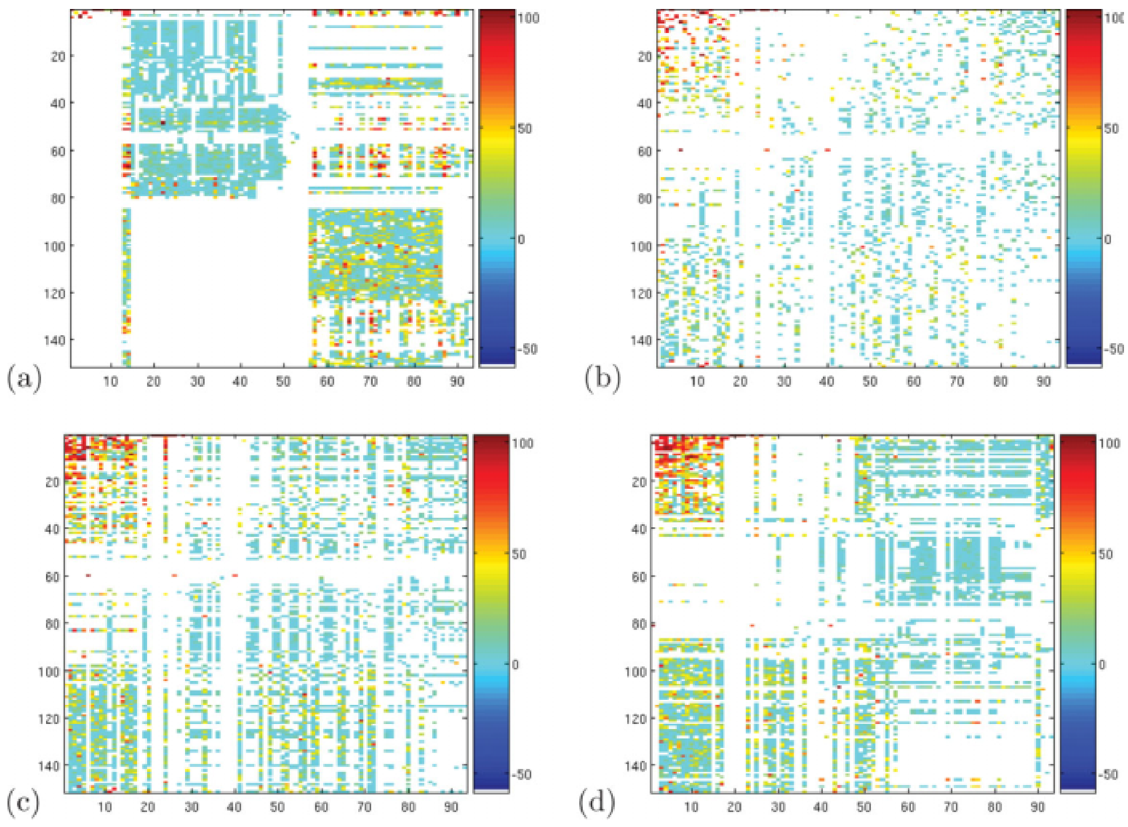

Figure 3. Second data matrix reordered by deterministic optimization.

(a) Original second data matrix using all known values. (b) Optimal rearrangement of the second data matrix sampled at 15%. (c) Optimal rearrangement of the second data matrix sampled at 15%, all known values included. (d) Optimal rearrangement of the second data matrix using all known values. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Table 2. Comparing Two Selected Regions of the Second Data Matrix Using the Deterministic Ordering from 15% Sampling to Illustrate the Clustering of High Inhibition Compounds

	Region 1	Region 1 (S)	Region 1 (U)	Region 2	Region 2 (S)	Region 2 (U)
Data values	234	99	135	3876	1951	1925
Average	49.6	54.2	46.2	12.1	12.1	12.1
Values $\geq$ 50	127 (54.3%)	62 (62.6%)	65 (48.1%)	185 (4.8%)	89 (4.6%)	96 (5%)
Values $\geq$ 70	79 (33.8%)	40 (40.4%)	39 (28.9%)	58 (1.5%)	29 (1.5%)	29 (1.5%)
Values $\geq$ 90	28 (12%)	13 (13.1%)	15 (11.1%)	12 (0.3%)	7 (0.4%)	5 (0.3%)

The inhibition values correspond to percentage inhibition of an unknown target protein (note that the laboratory conditions and target protein are different than the data presented in Table 1). Regions 1, 2 represent the full distribution of known values in the regions, whereas Region 1 (S),(U) and Region 2 (S),(U) correspond to the distribution of the sampled and unsampled compounds in the two regions, respectively.

The proposed deterministic method is able to group many of the desired compounds in an easily identifiable subset of the matrix. Let us define rows 1–21 and columns 1–20 of Figure 3d to be Region 1 and the remaining matrix to be Region 2. The statistics in Region 1 for the reordering based on using only 15% of the whole library space are provided in Table 2. From Table 2, we see that Region 1 contains a much higher average inhibition value and also a disproportionate number of data values above the inhibition cutoffs of 50, 70, and 90. Of the 40 known compounds with an inhibition greater than 90%, 28 of them are found in Region 1 and 15 of these were not used in the reordering. Thus, if one were to synthesize the 321 unknown or unsampled compounds in Region 1, then at least 15 of these compounds would have an inhibition greater than 90%. The stochastic optimization method is also able to reorder the compounds so that the high inhibition values are clustered into a small subset of the matrix (Supporting Information Figure S6). However, the orderings in other regions of the matrix are less similar to those revealed by the deterministic optimization method.

The computational performance of the algorithmic approaches for using both 15% of the whole library space and all known values is compared in Tables 4 and 5, respectively. In these tables, the objective (Obj) column is again from Eq. 3, the lower bound column is for this objective value, the percent gap column denotes the difference between the best lower bound and the objective value, and the CPU time is reported for a 3.0 GHz Intel Pentium Processor. The solutions found by deterministic optimization method (MILP), GA, and heuristic approaches are compared. For all the heuristics requiring random swapping operations, 1,000,000 swaps were performed. For all the heuristics requiring initial point selection, 40 initial points were used. The deterministic optimization approach has an integrality gap which for the row and column objective values of the sampled data set are guaranteed to be within 5.5% and 0.0% of the optimal solution, according to the objective function in Eq. 3. When using all of the known values, the objective values of the best identified row and column orderings are guaranteed to be within 1.6% and 0.0% of optimality, respectively. It is likely that the row orderings presented here are much closer to the optimal solution than this rigorous gap suggests. However, it is notable that the optimization approach achieves a better integrality gap as more data is available. Including additional data improves the ability of the algorithm to place a row or column in an appropriate position in the overall ordering. If the six rows with no available data are removed from the analysis, the row problem with 145 rows can be solved to within 1.2% of optimality, 0.4% better than when these spurious rows are included. For this data, the GA does not identify the best known solution. Both the deterministic method and the heuristic methods utilizing multiple initial points clearly outperform the stochastic optimization approach.

Iterative Synthesis Strategy

Once the optimal (or best) substituent ordering has been established, it is necessary to develop an appropriate strategy

Table 3. Method Comparison for the Arrangement of the Rows and Columns in the First Data Matrix Using All Known Data Values and the Normalized Objective Function

Problem	Method	Obj	Lower Bound	% Gap	CPU Time (s)
Row	Initial	210,908	–	13.6%	–
	Row swap	184,523	–	–	<0.1
	Greedy	182,232	–	–	1.9
	High sort	182,673	–	–	2.0
	Pick sort	182,232	–	–	1.9
	GA	183,916	–	–	8640
	MILP	182,230	182230	0.0%	3620
Column	Initial	135,003	–	29.4%	–
	Row swap	95,290	–	–	<0.1
	Greedy	95,290	–	–	1.0
	High sort	95,290	–	–	1.1
	Pick sort	95,290	–	–	1.0
	GA	96,580	–	–	1440
	MILP	95,290	95290	0.0%	123

The percent gap (i.e., deviation from the optimal ordering) of the initial orderings are provided for reference.

Table 4. Method Comparison for the Arrangement of the Rows and Columns in the Second Data Matrix Using Only Sampled Data Values and the Normalized Objective Function

Problem	Method	Obj	Lower Bound	% Gap	CPU Time (s)
Row	Initial	3.81059×10^6	–	29.6%	–
	Row swap	2.8426×10^6	–	–	0.9
	Greedy	2.82577×10^6	–	–	34
	High sort	2.82577×10^6	–	–	62
	Pick sort	2.82577×10^6	–	–	33
	GA	2.84093×10^6	–	–	100,800
	MILP	2.82577×10^6	2.67015×10^6	5.5%	569,505
Column	Initial	1,330,830	–	30.8%	–
	Row swap	950,731	–	–	0.5
	Greedy	921,153	–	–	18
	High sort	921,153	–	–	20
	Pick sort	921,153	–	–	18
	GA	926,201	–	–	403,200
	MILP	921,153	921,153	0.0%	127,344

The percent gap (i.e., deviation from the optimal ordering) of the initial orderings are provided for reference.

for molecular discovery. If the ordered compound property landscape is indeed smooth and regular, a simple local interpolation measure can be applied to adequately represent the expected property value of a compound. The most basic local interpolation measure would assume that the property value of a compound is similar to the average property values of its neighbors.

Two main issues arise if a property value is predicted by the average of neighboring compounds. If we define a set of compounds to be neighbors of specific compound, $a_{i,j}$, we may want to weight the contributions of these neighboring values by their distance from the specified compound. The distance between two points (i,j) and (i', j') in the ordered compound property landscape is defined as the Euclidian distance, as shown in Eq. 20.

$$d^{i,j}_{i',j'} = \sqrt{(i-i')^2 + (j-j')^2} \qquad (20)$$

Furthermore, we must account for the frequency of known points around a point (i,j). Let $R^{ij}_{i'j'}$ represent the set of compounds, (i',j'), that will be used to determine the estimated property value of the compound at position (i,j) as defined by Eq. 21.

$$R^{ij}_{i'j'} = \left\{(i',j') : d^{i,j}_{i',j'} \leq d^{\text{thresh}} \text{ and}\left(a_{i',j'} \text{is known or} (i = i' \text{ and } j = j')\right)\right\} \qquad (21)$$

The threshold distance, d^{thresh}, is the maximum distance a neighboring point can be from the specified compound (in this article we used the relation $d^{\text{thresh}} = (|I| \cdot 0.1 + |J| \cdot 0.1)/2$). Once we have defined $R^{ij}_{i'j'}$, Eq. 22 is used to define a normalization factor, $\Omega_{i,j}$. This normalization factor provides a higher weight to the closer neighbors and always retains the weight for the compound (i,j) to avoid long-range effects for missing values in sparsely populated regions.

$$\Omega_{i,j} = \sum_{(i',j') \in R^{ij}_{i'j'}} \frac{1}{d^{i,j}_{i',j'} + 1} \qquad (22)$$

The estimated property value at (i,j), $\rho_{i,j}$, is then defined by Eq. 23, which is a normalized average of the weighted neighboring property values.

$$\rho_{i,j} = \sum_{(i',j'):a_{i',j'} \text{is known}} \frac{a_{i',j'}}{\left(d^{i,j}_{i',j'} + 1\right) \cdot \Omega_{i,j}} \qquad (23)$$

Table 5. Method Comparison for the Arrangement of the Rows and Columns in the Second Data Matrix Using All Known Data Values and the Normalized Objective Function

Problem	Method	Obj	Lower Bound	% gap	CPU Time (s)
Row	Initial	3.90471×10^6	–	23.4%	–
	Row swap	3.03747×10^6	–	–	0.9
	Greedy	3.03741×10^6	–	–	34
	High sort	3.03741×10^6	–	–	38
	Pick sort	3.03741×10^6	–	–	33
	GA	3.05679×10^6	–	–	518,400
	MILP	3.03741×10^6	2.98092×10^6	1.9%	741,267
Column	Initial	1.84447×10^6	–	27.1%	–
	Row swap	1.34416×10^6	–	–	0.5
	Greedy	1.34416×10^6	–	–	18
	High sort	1.34416×10^6	–	–	20
	Pick sort	1.34416×10^6	–	–	17
	GA	1.35017×10^6	–	–	202400
	MILP	1.34416×10^6	1.34416×10^6	0.0%	64,516

The percent gap (i.e., deviation from the optimal ordering) of the initial orderings are provided for reference.

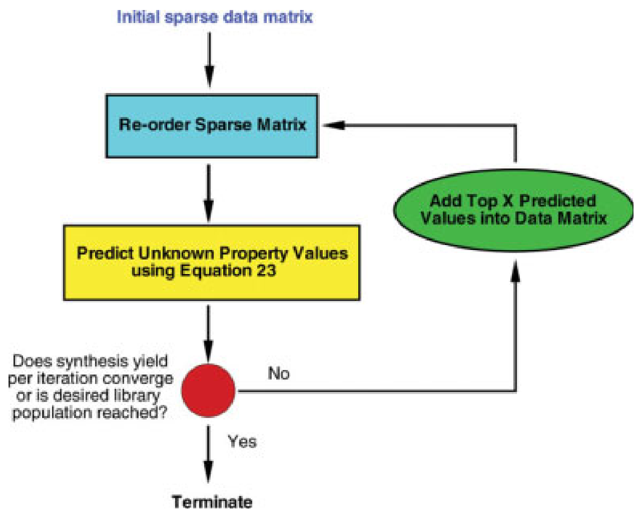
Initial sparse data matrix
Re-order Sparse Matrix
Add Top X Predicted Values into Data Matrix
Predict Unknown Property Values using Equation 23
Does synthesis yield per iteration converge or is desired library population reached?
No
Yes
Terminate

Figure 4. Flow diagram for iterative algorithm.

[Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

One strategy for synthesis, given an initial sampling of compound property values and its associated optimal substituent ordering, is to sort the estimated property values, $\rho_{i,j}$, and synthesize some number of compounds with the highest predicted property values. As more property values are determined, the process of reordering the substituents and estimating the property values can be repeated in an iterative fashion. An important feature to examine for such an iterative strategy is how much initial data is required to synthesize the important compounds while simultaneously minimizing the number of total compounds synthesized. To test the utility of such a protocol, we have developed an algorithm that begins with a small population of known compounds and uses the reordering results to perform an "in silico" synthesis. After each reordering, Eq. 23 is used to estimate the inhibition values for the unknown compounds. We then select the top 50 unknown compounds of highest predicted inhibition for in silico synthesis (i.e., reveal their actual values) and subsequently reorder this new library. A representative flow diagram for the iterative algorithm is presented in Figure 4.

This procedure was applied to four sparse data matrices: samplings of 50% (2050 compounds or 15% of the whole library space), 25% (1025 compounds or 7.3% of the whole library space), 10% (411 compounds or 3% of the whole library space), and 5% (206 compounds or 1.5% of the whole library space) at random from the original data matrix (4110 known values). The horizontal dashed line at the top of each figure denotes the total number of compounds that are above that percent inhibition in the original data. The diagonal dashed-dotted line in each figure represents the average gain per synthesis for the original synthesis procedure based on the data provided by Pfizer. To assess the overall effectiveness of the iterative strategy for finding higher inhibition compounds, we counted the total number of compounds above 40–90% inhibition in increments of 10%.

The results for compounds above 80% inhibition are shown in Figure 5 and the results for 40, 50, 60, 70, and 90% inhibition are available in the Supporting Information Figures S8–S12, respectively. In each of the figures, we see that the gain from each of the synthesis curves is fairly consistent among the initial samplings. Each of the curves exhibits a sharp slope in the beginning, indicating a good yield of higher inhibition compounds per synthesis iteration of 50 compounds. It is important to highlight that the synthesis curve starting from only 5% of the original data (206 known values) achieves a better yield of high inhibition compounds than all the other curves when 3000 known compounds are revealed (see the red curve in Figure 5 Supporting Information and Figures S8 through S12). In fact, this synthesis curve finds all of the available compounds above the inhibition thresholds of 70, 80, and 90% in less than 3000 compounds (see the red curve in Figure 5 and Supporting Information Figures S11 and S12, respectively).

Computational Analysis of the MILP Model

In this section, we examine the performance of the MILP model for solving this class of problems. From the analysis of the data sets in the previous sections and Tables 3–5, we see that the MILP model is able to generate tight lower bounds on the value of the optimal reordering for the compound libraries according to the objective function in Eq. 3. To illustrate the advantages of incorporating the aforementioned heuristics and cutting plane constraints presented in Eqs. 16 and 17 into the MILP framework, we compared the progress of the branch-and-cut tree for variants of the model with and without these additional features. The results for the MILP model (1) without the cutting planes nor the heuristics (default CPLEX), (2) without the cutting planes but with the heuristics, and (3) with both the cutting planes and heuristics are presented in Figure 6 for the rows of the second data matrix (151 elements). From this figure, one can

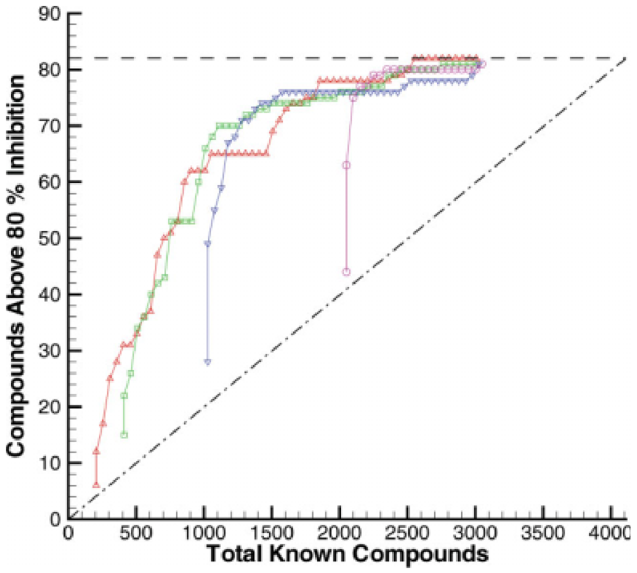

Figure 5. Iterative strategy for different initial populations of second data matrix for 80% inhibition.

The purple, blue, green, and red curves represent starting with 50, 25, 10, and 5% of the available data, respectively. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

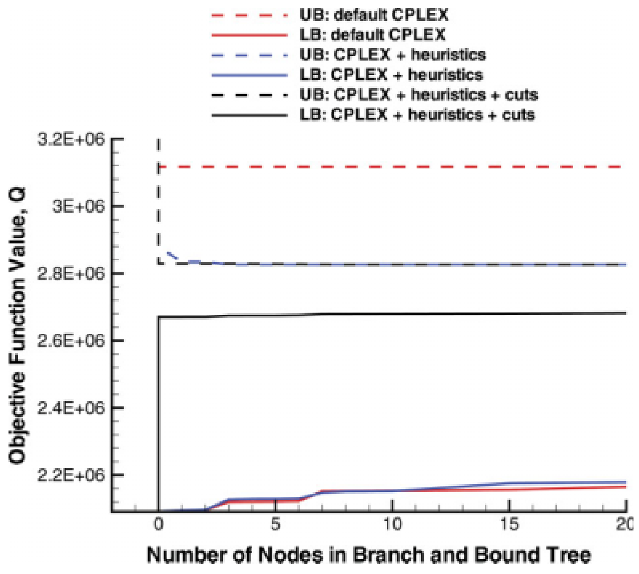

Figure 6. Performance of the branch and cut algorithm utilized in CPLEX, with and without the proposed cutting plane constraints and heuristics, on the row problem for data set 2 (151 elements).

The benefits of including both the heuristics and the cutting plane constraints into the MILP model is immediately apparent from this figure, as the MILP model that incorporates both (whose upper and lower bounds on the optimal solution are presented by the dashed and solid black lines, respectively) is able to prove that its best found integer solution is within 6% optimality at the root node. Only the first 20 nodes in the branch and bound tree are presented, because this is where the performance of the models differs the most and the lower bounds for each model subsequently increase in a parallel fashion as CPLEX utilizes similar branching operations. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

easily see that the incorporation of the cutting planes in Eqs. 16 and 17 substantially improves the quality of linear programming relaxations, thus achieving tighter lower bounds on the best possible integer solution (labeled as the lower bound on the optimal solution, LB). Also note that a significant amount of branching would be required for the two models that do not use our cutting plane constraints (e.g., “default CPLEX” and “CPLEX + heuristics”) to achieve the same lower bound at the root node as determined by the model that incorporates our cutting plane constraints, which is already within 6% of the best identified integer solution. Furthermore, the models that use our heuristics (e.g., “CPLEX + heuristics” and “CPLEX + heuristics + cuts”) are able to quickly find integer solutions (labeled as the upper bound on the optimal solution, UB) that have significantly lower values than those found by CPLEX alone for the objective function presented in Eq. 3.

A practical issue associated with some large-scale mixed integer linear programming problems is the existence of multiple integer solutions that have objective function values close to the optimal value. This issue is commonly referred to as solution degeneracy, and can often inhibit a branch-and-cut solver from proving the global optimality of an integer solution. Some of the problems studied in this article, such as the one analyzed in Figure 6, do exhibit solution degeneracy, but we point out that the existence of multiple solutions does not limit the applicability of our method. To understand this, consider the sparse compound library presented in Figure 3b. If we were to swap two rows that have primarily elements with low inhibition values, such an exchange would correspond to a new integer feasible solution, but would not substantially change the value of the objective function. Thus, the degeneracy is with respect to the existence of multiple relative orderings for rows containing primarily lower inhibition compounds, which in turn has no effect on the interpolation procedure presented in the Iterative Synthesis Strategy section and hence on the compounds that are selected for additional synthesis (as the selected compounds are close to the known, high inhibition compounds).

To prove this claim, we generated a large number of feasible solutions for the row problem of data matrix 2 (151 elements) by conducting excessive random sampling runs and stored the unique final orderings that had an objective function value less than 2.826×10^6. Note that in Table 4, this cutoff value for the objective function is less than the best solution reported by the GA. A total of 588 unique final orderings were identified with objective function values less than 2.826×10^6 and a heat map of these final orderings is presented in Figure 7 for visual analysis. In Figure 7, each cross-section of the *y*-axis corresponds to a unique final

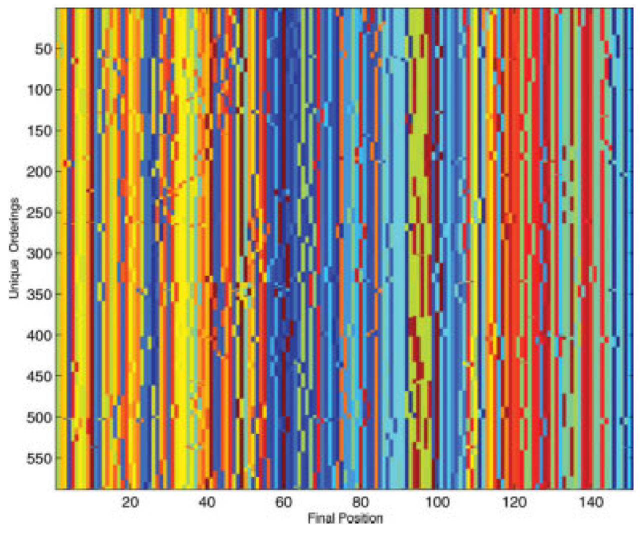

Figure 7. Unique row orderings for data set 2 (151 elements) such that the objective function value is less than 2.826×10^6.

A total of 588 unique row orderings with an objective function less than this threshold were found via several runs of random sampling. Each cross-section of the *y*-axis corresponds to a unique final ordering of the 151 functional groups, where the *x*-axis is the final position of the functional group and the *z*-axis is a color map, ranging from 1 (blue) to 151 (red), indicative of the functional group in the library (their identities were assigned as 1 through 151 based on their initial ordering in the compound library). From this figure, it is apparent that all of these solutions are very similar in terms of the final orderings of the functional groups. In fact, the difference between the orderings is primarily due to local shuffling of neighboring elements. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

ordering, the x-axis denotes the position of the functional group in the final ordering, and the color map (ranging from 1 (blue) to 151 (red)) denotes the distinct functional groups in the library (which were labeled as 1–151 based on their initial ordering). We can see from Figure 7 that the global placements of the functional groups are extremely similar and only vary with respect to local exchanges of elements in the final ordering. Hence, the overall placement of the functional groups is very consistent across these 588 unique orderings. Furthermore, the similarity of functional groups between the unique orderings is most consistent within the last 20 rows of the final ordering (i.e., along the x-axis). This is consistent with our claim, as the highest inhibition functional groups were moved to this region of the ordering, and thus are essentially unaffected by the solution degeneracy.

Discussion and Conclusion

This article introduces three classes of descriptor-free computational algorithms for property prediction within large combinatorial libraries of drug molecules or other molecules. Similar to our previous work,[11,12] these algorithms serve to identify (from the property data collected on a small subset of the library compounds) the optimal ordering of the functional groups on all substitution sites that yields the most regular/smooth compound property landscape. The reordered data matrices reveal small submatrices in which compounds with desired properties are abundant. From these observations, it appears that a significant amount of the compound synthesis and assaying was misdirected. This indicates that molecular discovery in large combinatorial libraries should be performed in an adaptive fashion, where each iteration is initiated by the synthesis and assaying of a small set of compounds and then followed by a suitable reordering algorithm or other QSAR method to estimate the library subspace with the highest possibility for finding desired compounds. The iterative synthesis strategy we presented and applied to sparse samplings of the second data matrix was able to uncover a significant percentage of the lead molecules while using only a fraction of total compound library, even when starting from a mere 3% of the total library space.

A main difference between the original reordering algorithm and the rearrangement clustering techniques presented in this article is that the former method represents a compound property by an inseparable high-dimensional function of all N substituent variables $X_1,X_2,\ldots,X_N$ (hence all substitution sites need to be reordered simultaneously), whereas the latter method separates these variables and reorders each substitution site individually, thereby leading to greatly enhanced computational efficiency. The applicability of the latter approach depends on the assumption that the compound property space is largely regular (i.e., containing no/few local optima) when the substituents are properly ordered. In all considered applications, we have been able to reorder the substitution sites to form regular, roughly single-peaked property landscapes, which validates the underlying assumption of landscape regularity and enables reliable property predictions from these reorderings. The underlying physical conditions for the existence of such smooth landscapes would be interesting to study.

One should note that a smooth landscape does not necessarily correspond to linearly additive contributions from individual functional groups on the substitution sites. Nonadditivity originates from cooperative interactions of multiple substitution sites and could affect the reliability of the rearrangement clustering algorithms because the objection function in Eq. 3 treats each substitution sites separately. However, this issue could be resolved by utilizing an objective function that specifically takes into account cooperative effects.

We need to emphasize here that the reliability of all the descriptor-free reordering methods, including those involving function interpolation,[11,12] depends critically on the resolution of sampling. For example, if a substitution site is unsampled then it is not possible to reorder this site. Our experience indicates that a random sampling over all the sites simultaneously (where each site is sampled at least a few times) yields a starting point that provides the best balance between the algorithms predictive reliability and the cost associated with compound synthesis and assaying. With this sampling strategy, the number of required samples quickly converges as N and S increase, leading to a significant reduction in the sampling percentage for large libraries.

Acknowledgments

The authors are grateful to Dr. T.-C. Hua of Pfizer for providing the data analyzed in this work. The authors acknowledge support from the US EPA (GAD R 832721-010). Although the research described in the article has been funded in part by the US Environmental Protection Agency's STAR program through grant (R 832721-010), it has not been subjected to any EPA review and therefore does not necessarily reflect the views of the Agency, and no official endorsement should be inferred.

Literature Cited

1. Ng R. *Drugs—From Discovery to Approval*. New Jersey: WileyLiss, 2006.
2. Lutz M, Kenakin T. *Quantitative molecular pharmacology and informatics in drug discovery*. New Jersey: Wiley, 2001.
3. Bannwarth W, Hinzen B, Mannhold R, Kubinyi H, Folkers G. *Combinatorial Chemistry: From Theory to Application (Methods and Principles in Medicinal Chemistry)*. New Jersey: Wiley-VCH, 2006.
4. Huser J, Mannhold R, Kubinyi H, Folkers G. *High-Throughput Screening in Drug Discovery (Methods and Principles in Medicinal Chemistry)*. New Jersey: Wiley-VCH, 2006.
5. Hansch C, Leo A. *Exploring QSAR—Fundamentals and Applications in Chemistry and Biology*. Washington, DC: American Chemical Society, 1995.
6. Hansch C, Telzer BR, Zhang L. Comparative qsar in toxicology: examples from teratology and cancer chemotherapy of aniline mustards. *Crit Rev Toxicol,* 1995;25:67–89.
7. Perkins R, Fang H, Tong W, Welsh W. Quantitative structure-activity relationship methods: perspectives on drug discovery and toxicology. *Environ Toxicol Chem*. 2003;22:1666–1679.
8. Tong W, Welsh W, Shi L, Fang H, Perkins R. Structure-activity relationship approaches and applications. *Environ Toxicol Chem,* 2003;22:1680–1695.
9. Kahraman P, Turkay M. Classification of 1,4-dihydropyridine calcium channel antagonists using the hyperbox approach. *Ind Eng Chem Res*. 2007;46:4921–4929.
10. Armutlu P, Ozdemir ME, Uney-Yuksektepe F, Kavakli IH, Turkay M. Classification of drug molecules considering their ic50 values

using mixed-integer linear programming based hyper-boxes method. *BMC Bioinform*. 2008;9:411–424.
11. Shenvi N, Geremia JM, Rabitz H. Substituent ordering and interpolation in molecular library optimization. *J Phys Chem A*. 2003;107:2066–2074.
12. Liang F, Feng X, Lowry M, Rabitz H. Maximal use of minimal libraries through the adaptive substituent reordering algorithm. *J. Phys Chem B*. 2005;109:5842–5854.
13. McCormick WT, Schweitzer PJ, White TW. Problem decomposition and data reorganization by a clustering technique. *Oper Res*. 1972;20:993–1009.
14. Lenstra JK. Clustering a data array and the traveling salesman problem. *Oper Res*. 1974;22:413–414.
15. Lenstra JK, Rinnooy Kan AHG. Some simple applications of the traveling salesman problem. *Oper Res Q*. 1975;26:717–733.
16. DiMaggio P, McAllister S, Floudas CA, Feng XJ, Rabinowitz J, Rabitz H. Biclustering via optimal reordering of data matrices in systems biology: rigorous methods and comparative studies. *BMC Bioinform*. 2008;9:458–473.
17. DiMaggio P, McAllister S, Floudas CA, Feng XJ, Rabinowitz J, Rabitz H. A network flow model for biclustering via optimal reordering of data matrices. *J Glo Opt*. In press.
18. Androulakis IP, Maranas CD, Floudas CA. Prediction of oligopeptide conformations via deterministic global optimization. *J Glo Opt*. 1997;11:1–34.
19. Klepeis JL, Floudas CA. Free energy calculations for peptides via deterministic global optimization. *J Chem Phys*. 1999;110:7491–7512.
20. Klepeis JL, Floudas CA, Morikis D, Lambris JD. Predicting peptide structures using NMR data and deterministic global optimization. *J Comp Chem*. 1999;20:1354–1370.
21. Klepeis JL, Floudas CA. Ab initio tertiary structure prediction of proteins. *J Glo Opt*. 2003;25:113–140.
22. Klepeis JL, Floudas CA. ASTRO-FOLD: a combinatorial and global optimization framework for ab initio prediction of three-dimensional structures of proteins from the amino acid sequence. *Biophys J*. 2003;85:2119–2146.
23. Klepeis JL, Floudas CA, Morikis D, Tsokos CG, Argyropoulos E, Spruce L, Lambris JD. Integrated computational and experimenal approach for lead optimization and design of compstatin variants with improved activity. *J Am Chem Soc*. 2003;125:8422–8423.
24. Fung HK, Floudas CA, Taylor MS, Zhang L, Morikis D. Towards full sequence de novo protein design with flexible templates for human beta-defensin-2. *Biophys J*. 2008;94:584–599.
25. Lin X, Floudas CA. Design, synthesis and scheduling of multipurpose batch plants via an effective continuous-time formulation. *Comput Chem Eng*. 2001;25:665–674.
26. Janak SL, Lin X, Floudas CA. Enhanced continuous-time unit-specific event based formulation for short-term scheduling of multipurpose batch processes: resource constraints and mixed storage policies. *Ind Eng Chem Res*. 2004;43:2516–2533.
27. CPLEX. *ILOG CPLEX C++ API 11.1 Reference Manual*. Gentilly, France: ILOG SA, 2008.

Manuscript received Sept. 16, 2008, revision received Mar. 15, 2009, and final revision received May 28, 2009.