OptDesign: Extending Optimizable *k*-Dissimilarity Selection to Combinatorial Library Design

Robert D. Clark,* Julia Kar, Lakshmi Akella, and Farhad Soltanshahi

Tripos, Inc., 1699 South Hanley Road, St. Louis, Missouri 63144

Received December 26, 2002

Optimizable *k*-dissimilarity (OptiSim) selection entails drawing a series of subsamples of size *k* from a population and choosing the “best” candidate from each such subsample for inclusion in the selection set. By varying the size of the subsample, one can control the balance between representativeness and diversity in the selection set obtained. In the original formulation, a uniform random sampling from among valid candidates was used to draw the subsamples from a single target population. Here we describe in detail two key modifications that serve to extend the OptiSim methodology to vector selection for interdependent variables, specifically as applied to the design of combinatorial sublibraries. The first modification involves *pivoting* between variables: subsamples are drawn from each reagent pool in turn, with the viability of each candidate being evaluated in isolation as well as in terms of the products it will produce from complementary reagents already selected. The filters applied may be static or dynamic in nature, with molecular weight and hydrophobicity being examples of the former and structural diversity with respect to reagents already selected being an example of the latter. The second key modification is adding the ability to *bias* the selection of candidate reagents for inclusion in the subsamples. Taken together, these modifications support the efficient generation of multiblock and other sparse matrix designs that are both representative and diverse, and for which “backfilling” of designs edited to remove undesirable reagents or products is straightforward. The method is intrinsically fast and efficient, since enumeration of the full combinatorial is not required— only those candidates actually considered for inclusion need be evaluated. Moreover, because the subsample selection step is separate from the diversity-based selection of the “best” candidate, incorporating such bias in favor of a competing criterion such as low price provides a “natural,” nonparametric mechanism for generating designs that are likely to be “good” in a double-objective, Pareto sense.

INTRODUCTION

Optimizable *k*-dissimilarity (OptiSim[1,2]) selection was originally developed as a fast and general way to extract selection sets that are both representative and diverse from large populations. The key parameter controlling the balance between those two contrasting properties is the size *k* of the subsample chosen from the target population for consideration at each iteration. When the “best” candidate in each such subsample is identified on the basis of its dissimilarity to those selected in previous iterations, selection sets are produced that are similar to those obtained by agglomerative hierarchical clustering using complete linkage; higher values of *k* mimic other linkage methods.[3]

One benefit of OptiSim in general is that it scales as kM^2, where *M* is the size of the selection set required. Another is that properties of interest need only be calculated for the relatively small fraction of the total population considered for inclusion in the subsamples. These attributes make the methodology well suited for sampling combinatorial libraries as combinatorials per se, i.e., without enumeration or the need to characterize every potential product. There is, however, no provision in the original OptiSim method for drawing subsamples from multiple populations, nor for taking into account multiple interactions between new candidates and complementary subsets of the existing selection set (for library design, products arising from reaction with selected reagents from other reactant classes). Hence, though the method is useful for generating “cherry picked” sublibraries, it cannot in general be used to identify “good” combinatorial sublibraries.

This limitation has been alleviated by *pivoting* between iterations, so that choosing candidates for subsamples alternates between reagent pools. This allows the efficient selection of multiblock sublibraries from full combinatorials.[4−6] Here, that logic has been further extended to allow generation of sparse blocks and by incorporating bias into the system so as to shift reagent selection in favor of one or another subclass of reagents.

METHODS

Basic OptiSim Methodology. The fundamentals of optimizable *k*-dissimilarity selection (*aka* OptiSim[1]) are relatively simple. Three parameters are involved: *k*, which specifies a subsample size; *r*, which specifies the minimum proximity allowed between a valid candidate and any individual already selected; and *M*, the total number of representatives to be selected. A simple iterative loop is then applied:

1. Seed the selection list **S**
   a. one or more seeds may suggest themselves, or one can be selected at random

* Corresponding author phone: (314)647-1099; fax: (314)647-9241; e-mail: bclark@tripos.com.

10.1021/ci025662h CCC: $25.00 © 2003 American Chemical Society
Published on Web 04/18/2003

2. Choose k valid candidates from the candidate pool at random
   a. only candidates at least r away from elements already in **S** are valid
3. Select the best candidate from among the K in the subsample and add it to **S**
   a. "best" by default meaning maximally dissimilar to **S**
4. If $|\mathbf{S}| < M$ and valid candidates remain to consider, go to step 1; else quit.

A full implementation of the method is somewhat more involved,[1] especially when, as is usually the case, sampling the target population is carried out without replacement until all valid candidates have been considered.

Varying k adjusts the balance between diversity (or any other criterion for "best") among the individuals selected and their aggregate representativeness with respect to the set from which they are drawn, with larger subsamples yielding more diverse sets. The method is inherently fast, since it scales with order kM^2, yet the selection sets created by applying it have very similar properties to those drawn from clusters obtained by agglomerative hierarchical clustering.[1] Moreover, partitions obtained by assigning each individual *not* selected from the population to the nearest exemplar that *was* selected provides a fast alternative to full hierarchical clustering.[3] Small values of k (3−5) are analogous to complete linkage agglomerative clustering, whereas somewhat higher values correspond, in turn, to group average and single linkage clustering.

Reagent Bias. Not all reagents are created equal. Some are cheaper than others—indeed, those already in hand may be nearly free—and some bear privileged substructures likely to confer a more or less specific activity upon the corresponding products. In that event, the simple approach of sampling at random from a uniform distribution to choose candidates for each successive subsample can lead to quite expensive or otherwise suboptimal sublibrary designs, because complexity-conferring reagents are favored in selection from the subsample. *Bias* has been incorporated into the choice of candidates to get around this problem. This involves assigning each reagent to a class based on some attribute (e.g., price or availability), and each class is assigned an integer weight specifying the number of "slots" it gets on a virtual roulette wheel. Candidates are then chosen on the basis of *two* numbers drawn from a uniform distribution—the first number determines the class from which a candidate will be drawn, and the other determines which member of that class will be chosen. The first random number is constrained to fall between 1 and the total of all weights (inclusive), whereas the second falls between 1 and the number of reagents in the chosen class.

Suppose, for illustrative purposes, that there are three classes **A**, **B**, and **C** with weights 6, 3, and 1 corresponding to class populations of 3, 21, and 126. Then the random number series 5, 2; 4, 1;10, 100; 7, 12 for $k = 4$ will lead to a subsample comprised of reagents 2, 1, 124, and 15 if the virtual roulette wheel is arranged as shown in Figure 1.

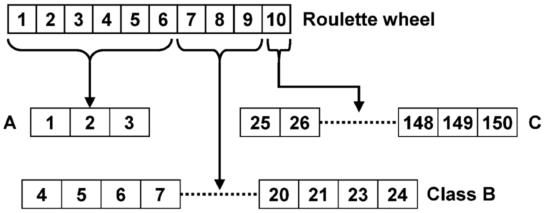


Figure 1. Illustration of how bias is applied in selection of candidate reagents. Class A, with three members, is assigned a weight of 6; Class B, with 21 members, is assigned a weight of 3; and Class C, with 126 members, is assigned a weight of 1. Selection involves drawing two numbers: first from a uniform random distribution between 1 and 10 and then from a uniform random distribution between 1 and 3, 21 or 126, depending upon the class indicated by the first random number.

Reagent Selection. Extension of OptiSim to multiple interacting factors X and Y (and Z, etc., if desired, though the current commercial implementation[6] is limited to working with two reagents at a time) entails iterative alternation between the factors under consideration.[4] In doing so one must allow for the fact that more reagents may be desired from population **X** than from population **Y**. In combinatorial library designs involving a disubstituted scaffold, for example, intermediates generated in the first reaction step are often synthesized and purified in bulk (i.e., on a multigram scale), and then parallel synthesis and high-throughput microscale chromatography are used to synthesize and purify products obtained in secondary reactions. Hence cost considerations lead to unsymmetrical designs in which M_X, the number of primary reagents required, is considerably smaller than M_Y, the number of secondary reagents. Pivoting to a reagent ceases once the specified quota for that reagent has been filled or none of the candidate reagents are viable candidates—i.e., none pass all pertinent static and dynamic reagent and product filters. Once X_3 has been selected in the scheme shown in Figure 2A, for example, pivoting to that reagent ceases and only reagents drawn from **Y** get added to the design. The block is complete once the number of products in the design exceeds the number requested, or all reagent quotas have been met, or no viable candidate reagents remain, whichever comes first.

Some library design tools are limited to generating full block designs wherein every X_i is "reacted" with every Y_j (and every Z_k etc.) included in the design.[7,8] Such a "combinatorial constraint" sharply narrows the piecewise properties of individual reagents and forces the designer to rather arbitrarily restrict the range of properties allowed among the reagent classes. Suppose, for example, that only products with molecular weights below 500 are acceptable and the scaffold of interest contributes 150 Daltons to each product. Then, taken together, the reagents used can contribute no more than 350 Daltons. The "usual" way to deal with this would be to consider no x contributing more than, say, 200 daltons to the product and no y contributing more than 150. But doing so excludes a wide range of very desirable candidates for which many candidate products would be perfectly acceptable. It also tends to increase the structural redundancy of the design obtained.

One way to broaden the reagent base utilized is to create a series of fully combinatorial blocks, i.e., a multiblock design. In OptDesign, this is accomplished by creating a first block as described above and then starting each new block by choosing k products (k each from **X** and **Y**) at random and selecting from these the one that is "best" *with respect to the products already selected for all previous blocks*. The second and subsequent blocks are then filled just as for the original one (Figure 2B). All dynamic product filters (e.g.,

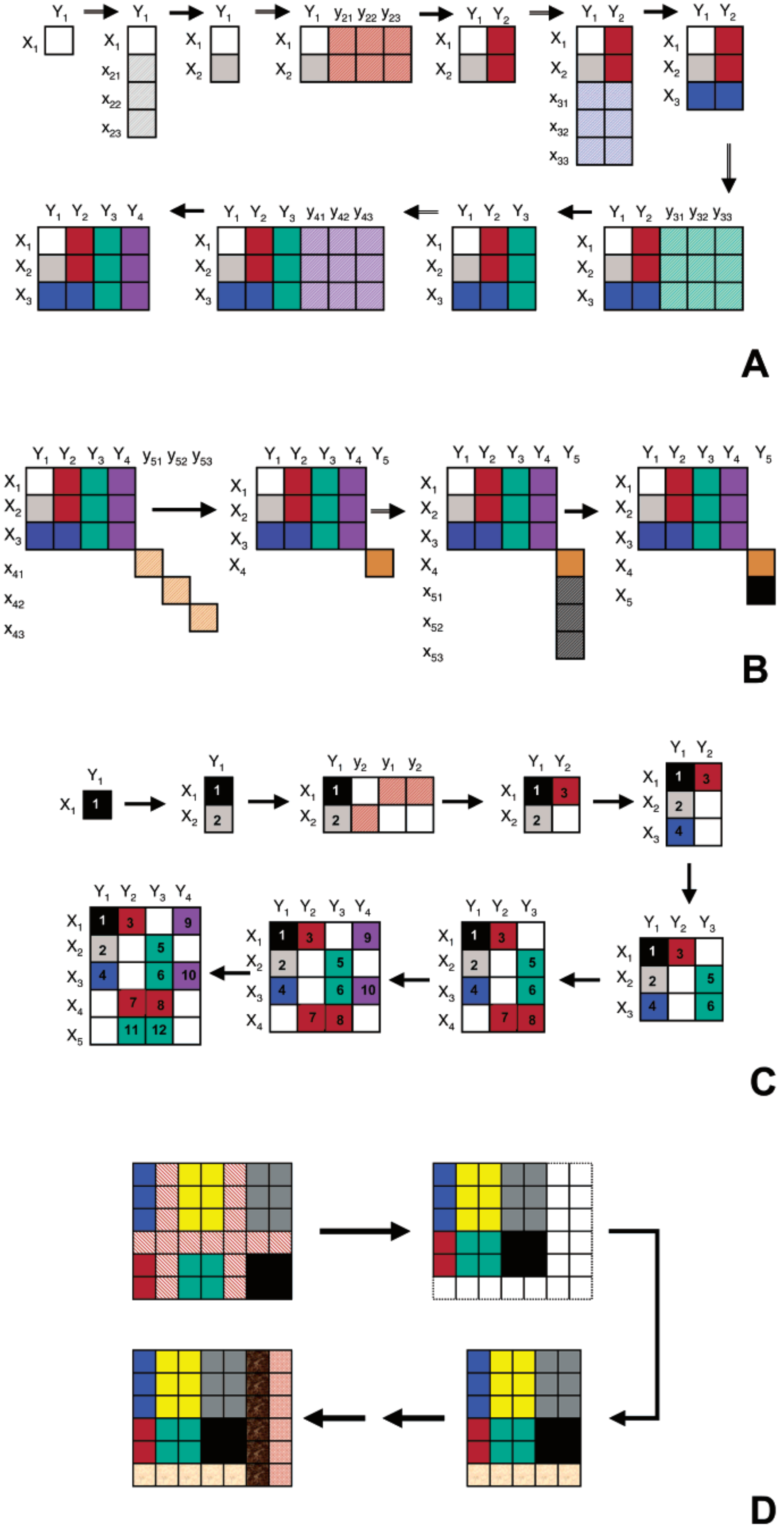


Figure 2. Schematic illustrations of OptDesign methodology. (**A**) Full block design illustrating pivoting between reagents, with $m_X = 3$, $m_Y = 4$, $k = 3$, and $M = 12$. (**B**) Starting a second block with $k = 3$. (**C**) Creating a sparse block design with density $q = 0.5$. (**D**) Filling in a design after removing undesirable products.

r) and selection criteria are evaluated with respect to all products selected so far, not just those in the current block.

Multiblock designs can be efficient if a relatively low total coverage of the fully combinatorial sublibrary is desired. A sublibrary comprised of 10 blocks, for example, will provide up to 10% coverage of the combinatorial space defined by the full complement of its constituent reagents (less if some reagents are reused). For cases in which a more synthetically efficient use of reagents is desired (say, 25−50% density *q*), it is often better to create a single block design in which only the *best* $q \times m_Y$ products are included for each selected X_i, where m_Y is the number of Ys selected for the current block at the present iteration. Integer arithmetic applies, so in the example shown in Figure 2C with $q = 0.50$, $q \times m_Y = 2$ when $m_Y = 3$; white unnumbered boxes indicate potential products not included in the design. As a result of such rounding up, the final design includes a fraction of the fully combinatorial sublibrary slightly greater than *q*. Sparse multiblock designs are also possible, of course.

In the absence of additional constraints, sparse designs tend to become very unbalanced, with some reagents in the design used much more or much less than are others. Provision has been made for specifying a maximum allowed *excess usage* to even out the actual usage of each reagent in the final design.

Filling-in Designs. Some of the reagents or products selected by a design program may ultimately be rejected by the synthetic chemist charged with realizing the submitted design. When this happens (as it does with remarkable frequency), the fact that a design is not the result of direct optimization of some objective function can be a significant practical advantage. Here, the undesirable reagents (indicated by red hashing in the Figure 2D) can simply be dropped, and the remaining reagents and products are used to seed another design run. The subsequent filling out of the sublibrary to the desired size requires far fewer computational steps than would restarting the selection from scratch, because there is no need to compare the seed products or reagents with each other. This facility makes it easy to recover from unusually "bad" starts, though these are—by definition—uncommon.

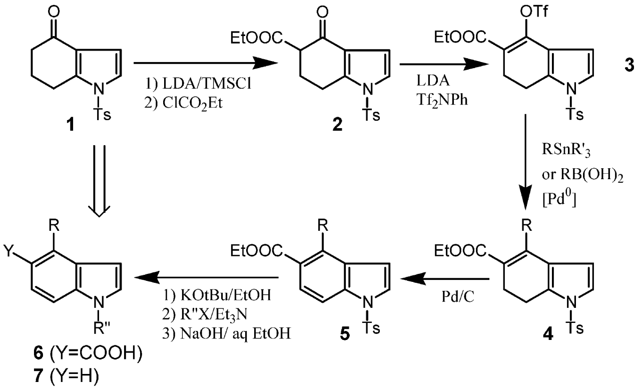
In OptDesign, the only requirement for inclusion is that the new reagents and their products *complement* the reagents and products retained from the original design, not that they have any *specific* properties. This makes it less likely that the replacements found will share the properties or substructures that made the rejected reagents or products undesirable than is the case for designs obtained by direct optimization.

All product seeds are treated as immediately included in the design, i.e., at step 0. Reagent seeds are considered in the order presented. If no product seeds are present when design generation is initiated, reagents and products are selected as described above. Pivoting between seed queues, however, replaces random selection of subsample candidates and a "temporary" *k* of 1 is used. The process reverts to the original *k* and random sampling once a seed queue is exhausted. When product seeds are provided along with reagent seeds, it is assumed that all desired products from those reagents are included among the product seeds. Hence no new products arising from combinations of seed reagents are added to the design; only products of seed reagents with newly selected reagents are generated and added to the design.

RESULTS

Virtual Library Composition. Some reasonably realistic design problem was required to illustrate the effects of the various OptDesign input parameters on the properties of the sublibraries produced. To this end, a synthesis scheme starting from *N*-tosyl-4-oxotetrahydroindole (**1**) was selected as an example of how the method can be applied in practice. Suzuki and Stille coupling chemistry were envisioned for the first combinatorial step, followed by aromatization, deprotection, combinatorial substitution at the indole nitrogen, and hydrolysis to the corresponding library of indole carboxylic acids (**6**; Scheme 1).

Scheme 1



N-Tosyl-4-oxotetrahydroindole can be treated with lithium diisopropylamide and trimethylsilyl chloride (LDA/TMSCl) to form the trimethylsilyl enolate, which can then be trapped with ethyl chloroformate to produce *N*-tosyl-5-carboethoxy-4-oxotetrahydroindole (**2**). Subsequent reaction with LDA and *N*-phenyltriflimide should yield the enol triflate **3** for use in subsequent combinatorial cross-coupling reactions. Palladium catalyzed reaction of **3** with organo-tin reagents or boronic acids (Stille or Suzuki coupling) is expected to give **4** in good yield, with subsequent aromatization using Pd on carbon yielding **5**. *N*-Derivatization of **5** will then afford the desired products represented by generic structure **6** (Scheme 1). A list of 193 primary substituents (R in Scheme 1) derived from alkyl tins and boronic acids suitable for the proposed palladium^0-catalyzed enol triflate chemistry[9,10] was compiled along with their normalized prices. A separate list of 238 potential secondary substituents was created for broadly compatible, commercially available alkylating, acylating, and sulfonylating reagents (R" in Scheme 1). The corresponding full virtual library of 45 934 products was created in combinatorial SYBYL line notation (CSLN) using the Legion module in SYBYL.[6] Omitting the initial chloroformate (virtual) reaction and terminal hydrolysis step yielded the analogous indole virtual library **7**.

Static filters applied included fragment molecular weights, product molecular weights (≤ 450), and product ClogP values (≤ 5 for indole carboxylates, but ≤ 6 for the more hydrophobic indoles). Price constraints were applied as indicated. Dynamic filters included a maximum UNITY fingerprint similarity of 0.90 with respect to any *reagent fragment* already selected for the *current* block as well as a maximum similarity of 0.95 with respect to any *product* already selected for the current *or any preceding* block.

Figure 3 shows molecular weight and hydrophobicity distributions for designs derived from **7**. Such critical product property profiles, which are often used to evaluate the "drug-likeness" of a library,[11,12] were calculated for two different designs. Each sublibrary had maximum block dimensions

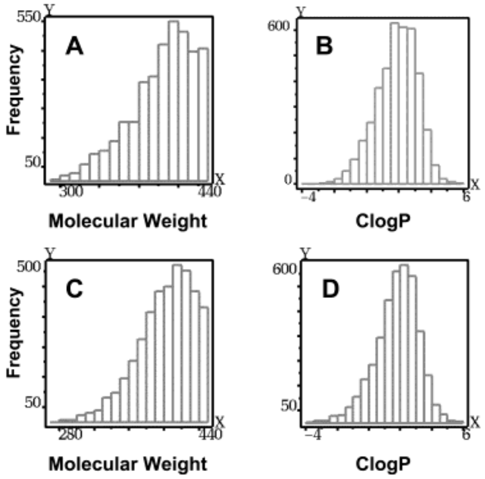


Figure 3. Molecular weight (**A** and **C**) and ClogP (**B** and **D**) distributions for 3000-product indole sublibraries generated with maxuimum block dimensions of 20×400 at a density $q = 0.5$. (**A** and **B**) No limit on excess usage. (**C** and **D**) The maximum excess usage allowed was set to 0.15, i.e., no reagent was allowed to contribute more than 13 or 260 ($m_i \times (q + 0.15)$) products to any one block.

of 20 and 400 reagents for R (X) and R”(Y), respectively, and a target density q of 0.5 and was comprised of 3000 products. They differed in the excess usage allowed: reagent usage was unconstrained for the first library (Figure 3A,B) but was restricted to an overrun of 0.15 or less for each reagent going into the second library (Figure 3C,D). The target density q is strictly enforced for each reagent *as it is added* to the design but not for products derived from complementary reagents added subsequently, so failing to limit excess usage can (and generally does) result in quite unbalanced designs—e.g., reagent Y_3 contributes to four products in the last OptiSim stage shown in Figure 2C, whereas Y_4 is only “used” twice.

Limiting excess usage had a salutary effect on both profiles, in that there was an appreciable shift toward lower molecular weight and less hydrophobic products. Both shifts reflect the fact these are more “average” (representative) products, whereas large lipophilic reagents are necessarily favored by any selection method seeking to increase library diversity. Without an excess usage constraint, the design will push up against the filter limits as much as it can.

Reagent Bias. Somewhat larger designs (4000 products) were used to characterize the effect of introducing bias into the selection of reagent candidates for inclusion in each subsample. Here, the maximum block dimensions were set to 20×100. At a density of 0.50, this gives about 1000 products per block, with four or more blocks in each design.[13] In the first example, boronic acids as a class were given a bias weight of 30, whereas the less numerous (28 vs 165) but more structurally diverse tin reagents received a bias weight of 10. Figure 4A shows the reagent usage for each structural class as a function of library size. Not surprisingly, boronic acids are utilized more frequently than are tin reagents under this selection regime, especially in the first and last blocks (1−1000 and 3000−4000). Swapping the bias weights (as one might wish to do should the wider applicability of the Stille coupling reaction outweigh the greater unit cost of the reagents) gives the usage curve shown in Figure 4B. Here, tin reagents overwhelmingly predominate for the first and second blocks; boronic acids are selected only after the supply of tin candidates has been nearly exhausted.

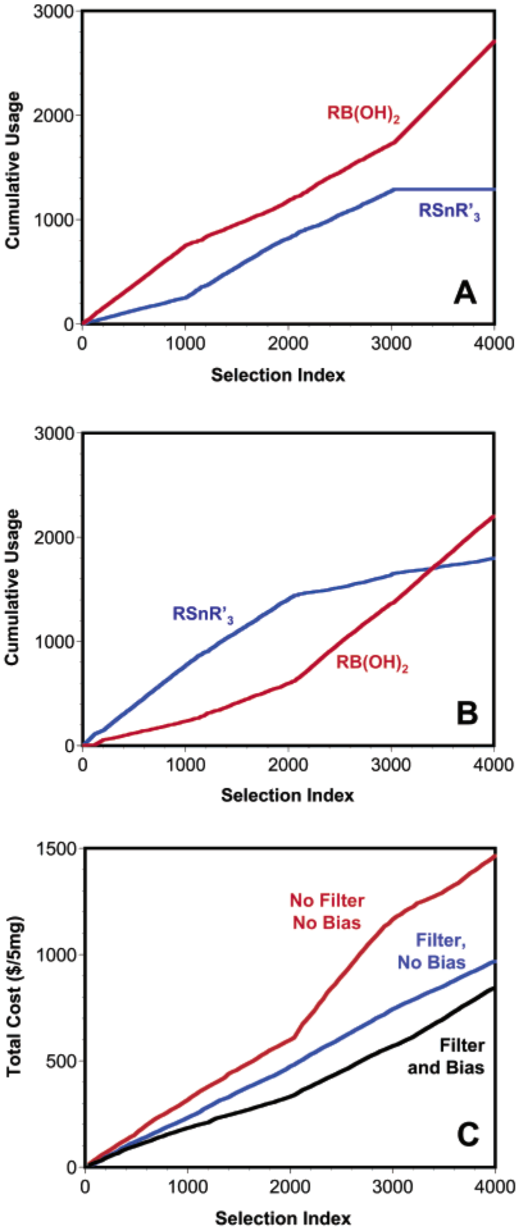


Figure 4. Effects of bias on usage curves for 4000-product designs (20×100 blocks at 50% density, 15% excess usage allowed) based on indole carboxylic acids (**6**). “Selection index” indicates the number of products selected at each step of sublibrary generation. (**A**) 3:1 bias in favor of boronic acids. (**B**) 3:1 bias in favor of organo-tin reagents. (**C**) Total cost progress curves for minimal constraints (No Filter, No Bias); with disallowance of any product costing more than $800/5 g (Filter, No Bias); and product cost filtering plus reagent bias by price involving three classes of coupling reagents (<$50, $50−300, and >$300 for 5 g, with weights of 4, 2, and 1, respectively) and five classes of acylation/alkylation reagents (<$10, $10−$30, $30−$100, $100−$300, and >$300, with weights of 5, 4, 3, 2, and 1, respectively) (Filter and Bias).

Figure 4C shows the synergistic effects of incorporating reagent cost directly, as a product filter, and indirectly, as a bias. The “product filtering” applied here entailed disallowing any *product* whose component reagents together cost more than $800/5 g. “Bias” involved three classes of coupling reagents and five classes of alkylation/acylation/sulfonylation

Table 1. Effect of Product Filters and Bias on Reagent Costs for 3000-Compound Indole Carboxylic Acid Sublibrary Designs Generated Using OptDesign Using a Density of 0.50

		R[a]			R"[b]			sum		
product filter	bias	min	max	mean	min	max	mean	min	max	mean
−	−	18[c]	1125	$274.03	1	675	$63.68	20	1800	$337.71
+[d]	−	12	695	173.46	1	675	57.29	14	799	230.75
+	+[e]	11	285	83.21	1	285	49.98	12	570	133.19

[a] Selected from among 193 alkyl tin or boronic acid reagents, up to 20 per block. [b] Selected from among 238 alkylating and acylating reagents, up to 100 per block. [c] Cost in dollars per 5 g of reagent based on 2002 catalog listings. [d] Any product costing more than $800/5 g was considered invalid. [e] Applied bias classes (and associated weights) for R were <$50/5 g (4), $50−300 (2), and >$300 (1). Bias classes (and weights) for R" were <$10/5 g (5), $10−$30 (4), $30−$100 (3), $100−$300 (2), and >$300 (1).

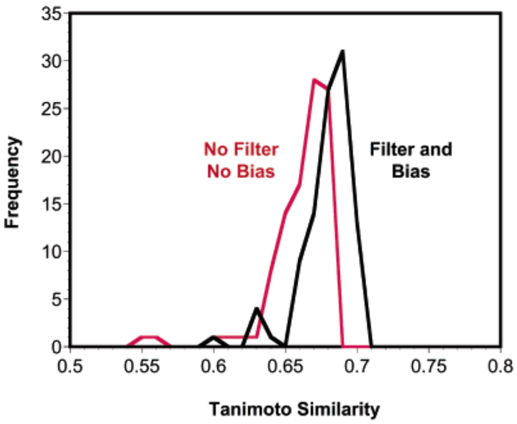


Figure 5. Self-similarity nearest-neighbor profiles for maximally dissimilar 100-compound subsets drawn from the minimally constrained sublibrary described in Figure 4C (red) and from the corresponding one obtained by combining a product price filter and a reagent price bias (black). Subsets were selected using *dbdiss* and analyzed using *dbcmpr*.

reagents. Including a product filter cut the overall cost of the design substantially, yielding a very linear cost curve. Adding a bias in favor of cheaper reagents reduced (virtual) costs even further, especially for the first two blocks generated. The reagent costs begin to edge back up toward those seen without bias later in the progress curve, which reflects exhaustion of the supply of good, cheap reagents (recall that 4000 products represents a substantial fraction (9%) of the full combinatorial chemistry space available).

Table 1 presents the corresponding results in a more detailed numerical format. Even the relatively mild bias described here was enough to drop the reagent costs associated with a 4000-compound production library at 100 mg maximum yield or each compound from $20K to $8K.

What concomitant "costs" in diversity were incurred by using less expensive reagents? This question can be addressed by considering the *redundancy* of each design indicated by the average nearest-neighbor Tanimoto (Soergel) distances found among the products included in each design[14]—0.0677, 0.0615, and 0.0578 for the unfiltered, filtered, and filtered with bias (Table 1). A complementary measure of the coverage afforded by each design is given by the nearest-neighbor self-similarities for maximally dissimilar subsets of 100 products;[15] the corresponding average values were 0.331, 0.328, and 0.318. The self-similarity profiles for the two extreme cases are shown in Figure 5.

A more revealing qualitative indicator of the cost in structural diversity of applying reagent price bias and product filtering can be obtained by visualizing the corresponding sublibraries based on their UNITY fingerprints.[4] The plot shown in Figure 6 is based on subsets of 250 products selected from each sublibrary by applying OptiSim (subsample size $k = 5$ and exclusion radius $r = 0.15$) using the Tanimoto dissimilarity of the corresponding UNITY substructural fingerprints as a measure of diversity.[1,4] The subsets from the two libraries were combined and subjected to a modified nonlinear mapping procedure in which pairwise stresses falling beyond a specified horizon are ignored (NLM-H); here, a "Tanimoto distance" (Soergel coefficient)[16] horizon of 0.3 was used.[4] This corresponds to slicing through empty regions of the high-dimensional fingerprint space and unfolding it into two dimensions so as to minimize *local* distortions in distances between the compounds under consideration. Dark symbols are used in Figure 6 to represent the filtered and biased design, whereas red symbols represent the unconstrained sublibrary; symbol size corresponds to the cost anticipated for each product. Illustrative products are appended to give an indication of the structural class represented in the various parts of the projection. This projection reinforces the notion that the reduction in cost evidenced by the small size of the dark symbols comes at little or no real cost in terms of coverage or diversity—nearly every red symbol has a (smaller) dark one falling near it in the projection.

DISCUSSION

The primary goal of this work was to create an extension of OptiSim that could be used to generate good sparse and multiblock combinatorial sublibrary designs yet still be flexible enough to accommodate a range of static and dynamic reagent and product filters as well as reagent bias. Not coincidentally, the method described is quite computationally efficient, in that the designs described here were all obtained in less than 75 min on a 250 MHz SGI workstation, despite the fact that some of the filters (e.g., ClogP calculation) and the fitness function used (minimal nearest-neighbor similarity as a measure of diversity) are very demanding in terms of CPU time. This reflects, at least in part, the fact that the method works directly from the combinatorial; such properties need only be evaluated for products actually considered for inclusion in the design.

OptDesign balances representativeness against diversity just as the parent OptiSim methodology does, and striking such a balance in combinatorial (sub)library design is a desirable end in itself. It is also, however, a way to address the concern that designing combinatorial sublibraries with an eye to maximizing diversity alone will tend to produce expensive libraries exhibiting grossly undruglike property profiles because of an intrinsic bias toward large, complex,

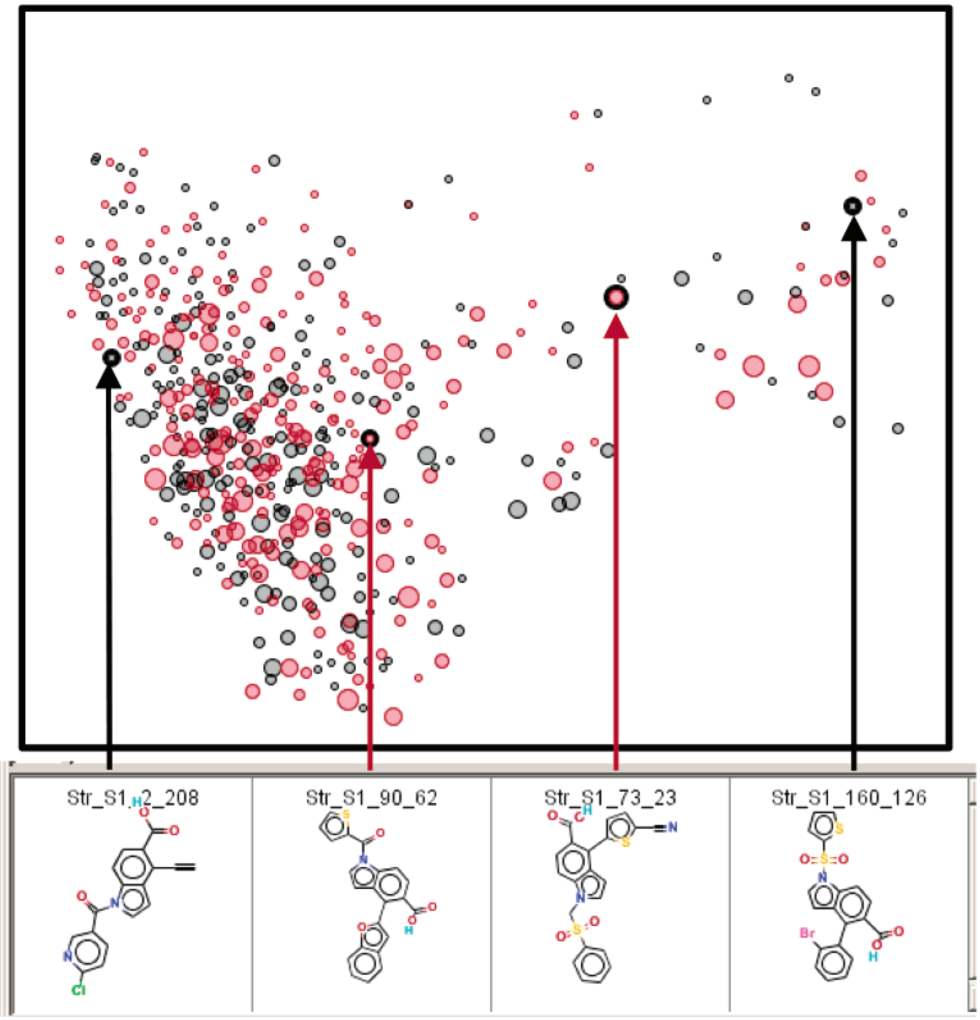


Figure 6. Nonlinear map created from UNITY substructural fingerprints using Tanimoto dissimilarities with an horizon of 0.3.[4] Red symbols indicate a subset of 250 products drawn from the library constructed without regard to price, whereas dark symbols correspond to 250 products similarly drawn from the library obtained using reagent price bias as well as product price filters. Subsets were obtained by applying OptiSim to each sublibrary with subsample size $k = 5$ and exclusion radius $r = 0.15$. The size of each symbol indicates the relative price of the corresponding product.

and hydrophobic reagents.[8,17,18] The stochastic nature of the subsampling used in OptDesign serves as a counterweight to this tendency of a diversity-based fitness function to maximize diversity at the expense of being representative of the library as a whole.

Note that the lists of candidate reagents used here were "natural" in the sense that they were obtained from relatively simple substructure searches[4] for commercially available alkylating agents, acylating agents, boronic acids, and organotins compatible with the chemistry involved. Use of these lists results in sublibrary designs exhibiting gratifyingly drug-like property profiles, especially when the blocks are constrained to be reasonably evenly populated by products (Figure 3). This will not be the case, of course, if the reagent pools used are skewed toward "bad" reagents. Perhaps economic forces resulting from "ordinary" medicinal and organic chemistry favor commercialization of "reasonably good" reagents. It may also be that the aggregate properties of the pool of available reagents shapes the property profile for drugs to some degree.

When the representativeness of OptiSim selection sets is plotted as a function of their diversity, points corresponding to $k > 2$ fall in a relatively narrow band along a line defining the best value of the one attribute attainable given an attainable good value of the other.[1] Though not recognized at the time, this is a hallmark of "Pareto-good" (versus strictly Pareto optimal) solutions for double-objective functional optimization.[18] This behavior results directly from the separation of the selection process into two parts—the stochastic subsample selection favoring representativeness and the objective ("Which candidate is best?") part favoring diversity. Hence OptDesign by construction represents a good approximation to direct Pareto optimization for two factors that is readily extensible to multiblock and sparse designs.

The bias applied to selection of candidates for subsamples operates similarly independently of the fitness function used to select the "best" compound from each subsample, so it, too, will strongly promote generation of sublibraries that are "Pareto good" whenever the bias involves a property that is more or less at odds with the fitness function. The cost and reagent biases considered here illustrate that effect. In particular, biasing in favor of lower-cost reagents while selecting for diversity among products leads to relatively less expensive but still diverse sublibraries without having to determine a single, fixed "cost" per unit of diversity a priori, as is necessary when designing libraries using multilinear objective functions.[8,11,19−21] Such multiobjective selection has been built directly into a single fitness function for design of *full* combinatorial sublibraries in the MoSELECT program, which makes use of a genetic algorithm to produce an ensemble of such designs.[18] Varying *k* as well as random number seed will be required if one wishes to explore a comparable range of Pareto-good solutions using OptDesign, but the approach as described here is otherwise minimally parametric in nature.

Additional objective functions can be incorporated by including additional bias classes, since there is no requirement that a reagent appear in only one class. Doing so will allow more than two objective functions to be accommodated, though the bias components will be nonlinearly related to one another rather than fully independent. In addition, although alternative fitness functions such as similarity to a set of known actives are readily incorporated into the program, the only way to accommodate multiple product-based objectives (e.g., similarity to known actives *and* molecular diversity) simultaneously at present is by way of a multilinear fitness function.[8,17] Work is underway to incorporate Pareto scoring directly into OptDesign as a way to get around this limitation. Note, however, that the *specific* designs obtained will generally be quite different—not least in terms of representativeness—from those obtained by direct optimization of this or any other objective function, just as OptiSim selection sets differ qualitatively and quantitatively from maximally diverse subsets.[1,3]

ACKNOWLEDGMENT

The authors would like to thank Drs. Mark Warne and Michael Lawless of Tripos, Inc., for their cooperation, helpful suggestions, and patience as OptDesign was being developed.

REFERENCES AND NOTES

(1) Clark, R. D. OptiSim: An Extended Dissimilarity Selection Method for Finding Diverse Representative Subsets. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 1181−1188.

(2) OptiSim, which is a registered trademark of Tripos, Inc., is patent pending.

(3) Clark, R. D.; Langton, W. J. Balancing Represenativeness Against Diversity using Optimizable *K*-Dissimilarity and Hierarchical Clustering. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 1079−1086.

(4) Clark, R. D.; Patterson, D. E.; Soltanshahi, F.; Blake, J. F.; Matthew, J. B. Visualizing substructural fingerprints. *J. Mol. Graph. Model.* **2000**, *18*, 404−411.

(5) Clark, R. D. Getting Past Diversity in Assessing Virtual Library Designs. *J. Braz. Chem. Soc.* **2002**, *13*, 788−794.

(6) OptDesign and Legion are available as part of the 6.9 release of SYBYL from Tripos, Inc., 1699 S. Hanley Road, St. Louis MO 63144 (http://www.tripos.com).

(7) Good, A. C.; Lewis, R. A. New Methodology for Profiling Combinatorial Libraries and Screening Sets: Cleaning Up the Design Process with HARPick. *J. Med. Chem.* **1997**, *40*, 3926−3936.

(8) Gillet, V. J.; Willett, P.; Bradshaw, J.; Green, D. V. S. Selecting Combinatorial Libraries to Optimize Diversity and Physical Properties. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 169−177.

(9) Doi, K.; Mori, M. Short Step Synthesis of 4-Substituted Indoles using Palladium-Catalyzed C−C Bond Forming Reaction. *Heterocycles* **1996**, *42*, 113−116.

(10) Ritter, K. Synthetic Transformations of Vinyl and Aryl Triflates. *Synthesis* **1993**, 735−762.

(11) Brown, R. D.; Hassan, M.; Waldman, M. Combinatorial library design for diversity, cost efficiency and drug-like chararcter. *J. Mol. Graphics Model.* **2000**, *18*, 427−437.

(12) Barn, D.; Caulfield, W.; Cowley, P.; Dickins, R.; Bakker, W. I.; McGuire, R.; Morphy, J. R.; Rankovic, Z.; Thorn, M. Design and Synthesis of a Maximally Diverse and Druglike Screening Library Using REM Resin Methodology. *J. Comb. Chem.* **2001**, *3*, 534−541.

(13) Blocks can terminate prematurely if reagents are selected early on that are especially restrictive in terms of complementary reagents it can "legally" react with—e.g., that give substituents that are too large or too hydrophobic. In that case, creation of a new block is initiated automatically.

(14) Clark, R. D. Relative and Absolute Diversity Analysis of Combinatorial Libraries. In *Combinatorial Library Design and Evaluation for Drug Discovery: Principles, Software Tools and Applications in Drug Discovery;* Ghose, A. K., Viswanadhan, V. N., Eds.; Marcel Dekker: New York, 2001; pp 337−362.

(15) Cramer, R. D.; Patterson, D. E.; Clark, R. D.; Soltanshahi, F.; Lawless, M. S. Virtual Compound Libraries: A New Approach to Decision Making in Molecular Discovery Research. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 1010−1023.

(16) Cheng, C.; Maggiora, G.; Lajiness, M.; Johnso, M. Four Association Coefficients for Relating Molecular Similarity Measures. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 909−915.

(17) Agrafiotis, D. K. Multiobjective optimization of combinatorial libraries. *J. Comput.-Aided Mol. Des.* **2002**, *16*, 335−356.

(18) Gillet, V. J.; Willett, P.; Fleming, P. J.; Green, D. V. S. Designing focused libraries using MoSELECT. *J. Mol. Graph. Model.* **2002**, *20*, 491−498.

(19) Pickett, S. D.; Luttmann, C.; Guerin, V.; Laoui, A.; James, E. DIVSEL and COMPLIB − Strategies for the Design and Comparison of Combinatorial Libraries using Pharmacophoric Descriptors. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 144−150.

(20) Cf. the reagent-biased product selection facility in DiverseSolutions (DVS), developed by R. S. Pearlman and K. M. Smith at the University of Texas at Austin and distributed by Tripos, Inc., 1699 S. Hanley Rd., St. Louis, MO 63144 USA.

(21) Brown, R. D.; Hassan, M.; Waldman, M. Tools for Designing Diverse, Druglike, Cost-Effective Combinatorial Libraries. In *Combinatorial Library Design and Evaluation for Drug Discovery: Principles, Software Tools and Applications in Drug Discovery*; Ghose, A. K., Viswanadhan, V. N., Eds.; Marcel Dekker: New York, 2001; pp 301−335.

CI025662H