PLUMS: a Program for the Rapid Optimization of Focused Libraries

Gianpaolo Bravi,* Darren V. S. Green, Michael M. Hann, and Andrew R. Leach

Computational Chemistry and Informatics, GlaxoWellcome R&D, Medicines Research Centre, Gunnels Wood Road, Stevenage, Hertfordshire, SG1 2NY U.K.

Received May 8, 2000

PLUMS is a new method to perform rational monomer selection for combinatorial chemistry libraries. The algorithm has been developed to optimize focused libraries with specific two-dimensional and/or three-dimensional properties. A preliminary step is the identification of those molecules in the initial virtual library which satisfy the imposed property constraints; we define these molecules as the *virtual hits*. From the virtual hits, PLUMS generates a starting library, which is the true combinatorial library that includes all the virtual hits. Monomers are then removed in an iterative fashion, thus reducing the size of the library. At each iteration, the worst monomer is removed. Each sublibrary is selected using a global scoring function, which balances *effectiveness* and *efficiency*. The iterative process continues until one is left with a library that consists entirely of virtual hits. The optimal library, which is the best compromise between effectiveness and efficiency, can then be selected according to the score. During the iterative process, equivalent solutions may well occur and are taken into account by the algorithm, according to a user-defined parameter. The number of monomers for each substitution site and the size of the library are parameters that can be either optimized or used to constrain the selection. The results obtained on two test libraries are presented. PLUMS was compared with genetic algorithms (GA) and monomer frequency analysis (MFA), which are widely used for monomer selection. For the two test libraries, PLUMS and GA gave equivalent results. MFA is the fastest method, but it can give misleading solutions. Possible advantages and disadvantages of the different methods are discussed.

INTRODUCTION

There are many reported methods for the design of combinatorial libraries.[1] However, most of these techniques are aimed at general screening libraries, with emphasis on the design of diverse sets of molecules. Whereas *diverse libraries* are designed to span as much as possible the chemical and/or property space, *focused libraries* are designed to satisfy specific property boundaries. The purposes of diverse and focused libraries are diametrically opposed, and the demands in the design methods are accordingly different. We herein describe a recently developed program, named PLUMS, aimed at optimizing libraries which satisfy a predefined set of constraints.

Chemical properties, such as molecular weight, log *P*, and number of hydrogen bond donor/acceptor atoms, have been shown to play a crucial role in absorption and transportation phenomena.[2] These properties are easily computable and can be used as constraints in an attempt to incorporate the concept of drug-likeness in the design of any virtual libraries, no matter what the target is. In this regard, the polar surface area[3,4] has been applied as well.[5]

When more information is available on the biological target, more complex constraints can be introduced in the design efforts. For instance, a pharmacophore derived either through a comparison of known inhibitors or from the analysis of a receptor−ligand complex can be used as a constraint to find those members of a combinatorial library that match it. Alternatively, a QSAR model can be used to predict the activity of the library products, or the similarity of the library products to a target molecule can be evaluated.

Whatever the nature of the applied constraints, they provide a partition of the library products into two categories. Those molecules able to satisfy the constraints are defined as favorable and are *virtual hits*. Those molecules unable to satisfy the constraints are unfavorable. This is exemplified in Figure 1 for a simplified library with two points of diversity.

In principle only those molecules belonging to the favorable set should be synthesized. Unfortunately these rarely represent a true combinatorial set such that when each monomer reacts with every other monomer the resulting product is a virtual hit. The task is then to select a subset of the monomers for each substitution site which maximizes the number of virtual hits in the library while minimizing the efforts in doing this (i.e., keeping the size of the library as small as possible).

PLUMS: DEFINITION OF THE SCORING FUNCTION

In practice, there are usually external factors that constrain the library size or even the number of monomers for each substitution site. In the simplest case the numbers of monomers might be predetermined, in which case we would wish to find, for example, the best 10×10 library in the case of a library with two points of diversity. Here the aim is simply to maximize the number of favorable molecules in the 100-molecule library. The monomers, which lead to

* To whom correspondence should be addressed. Phone: +44 (0)1438 763389. FAX: +44 (0)1438 764918. E-mail: gb94807@GlaxoWellcome.co.uk.

10.1021/ci000389+ CCC: $19.00
Published on Web 10/21/2000

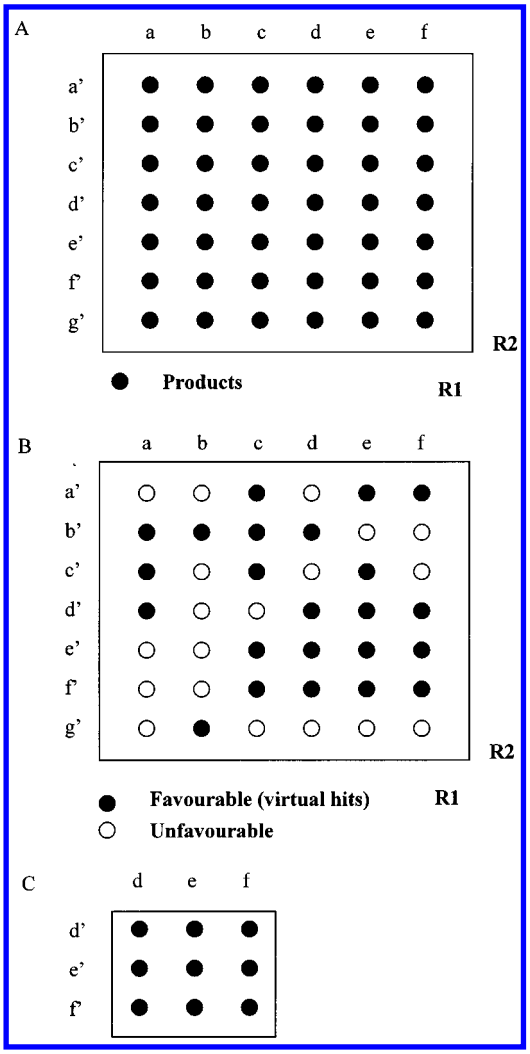

Figure 1. (a) Grid representation of a 6 × 7 virtual library. Each node represents a single compound. (b) Compounds in the library have been assigned to two categories, named favorable and unfavorable, as a result of applying a set of constraints. (c) The largest sublibrary whose members are all favorable.

the 100-member library that contains the highest number of favorable compounds, would be those we wish to select.

However, it is generally more useful for the chemist to know whether a 10 × 10 library is indeed the optimal size or whether other libraries, be they smaller or larger, might be better. Hence we need a different scoring mechanism which allows us to compare libraries with different numbers of product molecules.

Figure 1b shows the library obtained after removing those monomers that do not give rise to any favorable compounds. In this particular case no monomers were excluded, since every monomer in both R1 and R2 occurs in at least one favorable compound. This library is our starting point. We define it as the most *effective* we can make. This is because it contains all the virtual hits. However, it does also contain many unfavorable compounds and so is rather inefficient. We consider the most *efficient* library to be the largest sublibrary, which only contains favorable molecules. This is represented in Figure 1c. Note however that this library is not very effective, as we have lost many favorable molecules with respect to the starting virtual library.

Accordingly, for the *i*th library we define *effectiveness* as the ratio between the number of virtual hits in the library and the total number of virtual hits available (eq 1) and *efficiency* as the ratio between the number of virtual hits in the library and the size of the library (eq 2).

$$\text{effectiveness}_i = \frac{N_{i,\text{favorable}}}{N_{\text{tot,favorable}}} \tag{1}$$

$$\text{efficiency}_i = \frac{N_{i,\text{favorable}}}{N_{i,\text{product}}} \tag{2}$$

Figure 1 shows two extremes: optimal effectiveness (Figure 1b) and optimal efficiency (Figure 1c). However, what we might need is a compromise between the two. There may exist an intermediate library, which represents a better balance between efficiency and effectiveness. We can combine these two definitions to derive a score.

$$\text{score}_i = w_1\,\text{effectiveness}_i + w_2\,\text{efficiency}_i \tag{3}$$

This scoring function is used by PLUMS to compare and rank different libraries.

PLUMS: DESCRIPTION OF THE ALGORITHM

PLUMS uses the list of favorable molecules as input. Each molecule identifier is given by the concatenation of its corresponding monomer names (this is the standard output of the program ADEPT[6] that we use to perform the library enumeration). PLUMS identifies all the monomers occurring in the favorable list from the molecule names. The starting virtual library is encoded in a bit string where each bit corresponds to a different monomer. If *n* monomers are found for the first substitution site, the first *n* bits in the string are assigned to the first substitution site. The length of the bit string thus equals the total number of monomers, and initially all its elements are set to 1. PLUMS works in an iterative fashion. At each iteration, the *worst monomer* in the library is identified and removed; i.e., the corresponding bit is set to zero. The worst monomer is the monomer that, once removed, produces the library with the best score for that iteration. Constraints on the library size or the number of monomers can be easily applied at this stage. The process continues until all the unfavorable molecules have been removed from the library. Any subsets of the resulting library would have a lower (or equal if $w_1 = 0$ in eq 3) score since more favorable compounds would be lost without further improving the efficiency.

For the example in Figure 1b, the whole PLUMS process is illustrated in Figure 2 (equal weights were assigned to effectiveness and efficiency). The starting library has a score of 0.774. From this library monomer *g′* is removed at the first iteration leading to an improved score library, 0.784. The process continues until the efficiency reaches its maximum, after seven iterations. The best solution corresponds to a maximum in the score. In this example the score reaches a maximum of 0.806 (effectiveness = 0.913, efficiency = 0.700) after two iterations upon removal of monomers *g′* and *b* and the corresponding library contains 11 monomers.

During the iterative process equivalent situations may well occur. For instance, if at the *i*th iteration the removal of monomer *j* or monomer *k* produces libraries with identical

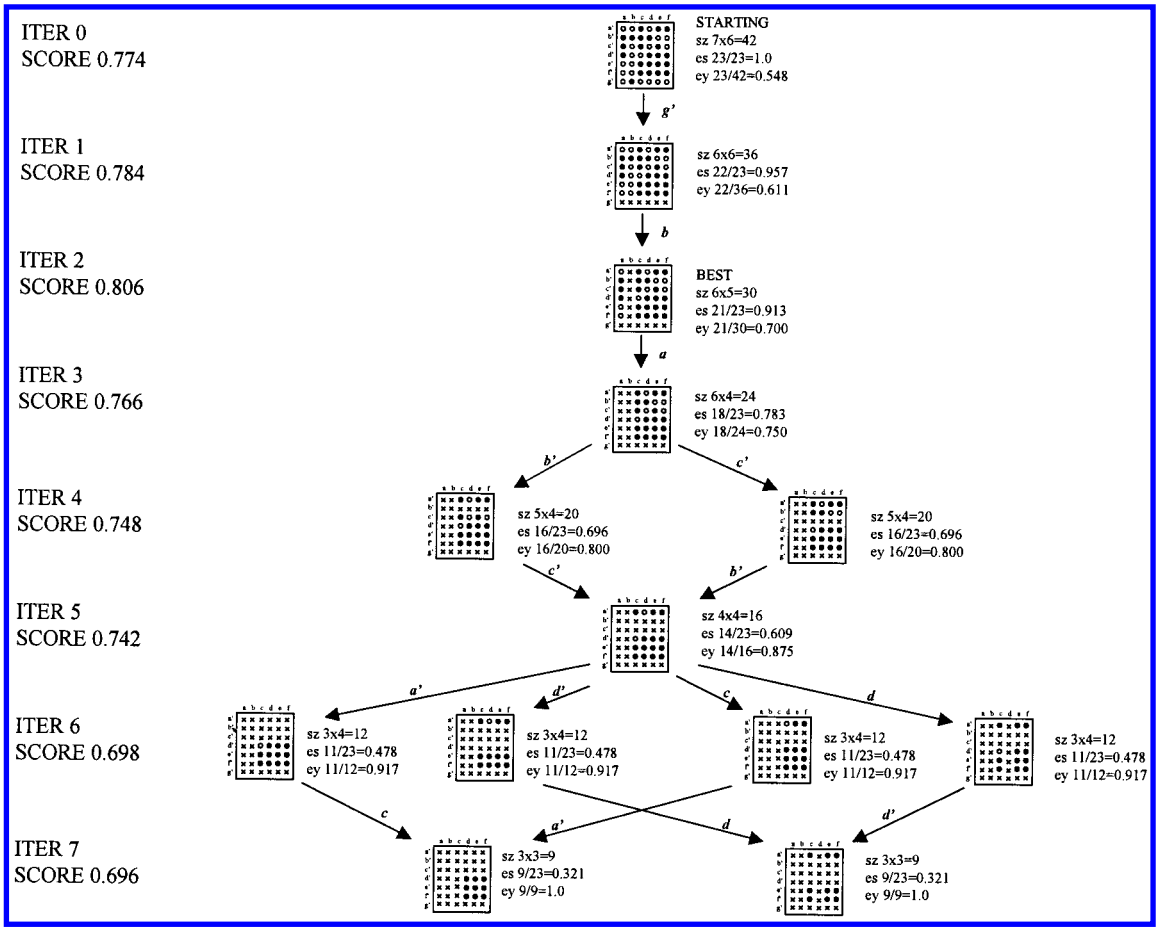

Figure 2. Detailed description of the PLUMS iterative process for the text case of Figure 1b. Filled circles represent favorable compounds, while unfilled circles indicate unfavorable ones. sz is size, ey efficiency, and es effectiveness. Score is computed as $(0.5 \times \text{ey}) + (0.5 \times \text{es})$. Monomers, which are removed upon each iteration, are highlighted in italics and indicated by a cross. Equivalent solutions are found at iterations 4, 6, and 7.

scores, the program removes both monomers in turn and generates two bit strings for analysis at the next iteration. The algorithm keeps a record of any equivalencies found (and therefore removes any potential duplicate solutions). As a general rule, fewer equivalent situations are found at the beginning of the run and more at the end, when the program is approaching the most efficient library. However, the number of equivalent bit strings may or may not grow iteration after iteration. The behavior is difficult to predict and is very set dependent. Since this number might be rather considerable (depending on the set) and could affect the speed of the program, the user can decide whether to keep only one solution at each iteration, to keep them all, or to keep a certain number (e.g., no more than 10). This third option, in the test cases reported below, has proved to be a rather good compromise between quality of results and computational speed.

In the above example (see Figure 2), two equivalencies are found at the fourth iteration (either monomers *b′* or *c′* removed) which converge to only one solution at the fifth iteration (both *b′* and *c′* removed). At iteration 6 four equivalent solutions are detected and they converge to two final libraries of nine molecules whose components are all favorable. The first one includes monomers *d′*, *e′*, *f′* and *d*, *e*, *f*. The second one includes monomers *a′*, *e′*, *f′* and *c*, *e*, *f*. This result was obtained using PLUMS and the all_equivalent option. If the 1_equivalent option is turned on, only the first solution (represented in Figure 1c) is found, corresponding to the pattern on the left of Figure 2.

It is worth noting that, if equivalent situations are considered, monomers excluded at the *i*th iteration can actually still be present. This is the case, for example, of monomers *a′*, *d′*, *c*, and *d*, which were removed at iteration 6. However, they are all present in the two final solutions because they were kept in alternative solutions.

APPLICATIONS

Library-1. The first example involves the design of a focused library with three points of diversity. Sets of 13 aldehydes (R1), 41 azole aldehydes (R2) and 59 amines (R3) were selected based on availability and two-dimensional diversity. The library was fully enumerated and profiled using the ADEPT software.[6] The "rule of five"[2] was considered in an attempt to enhance the druglike properties of the products. Because the numbers of hydrogen-bond donors and acceptors were already within acceptable limits, filters were applied only on the molecular weight and *C*log *P*.[7] Two more properties were considered, the number of rotatable bonds and the "maximal binding energy",[8] and appropriate filters were defined on the basis of known active molecules. As a result, 10 532 out of 31 477 molecules passed the filters and were used to build a multiconformational Catalyst database.[9] A previously derived three-dimensional pharmacophore retrieved 4907 compounds as virtual hits, which

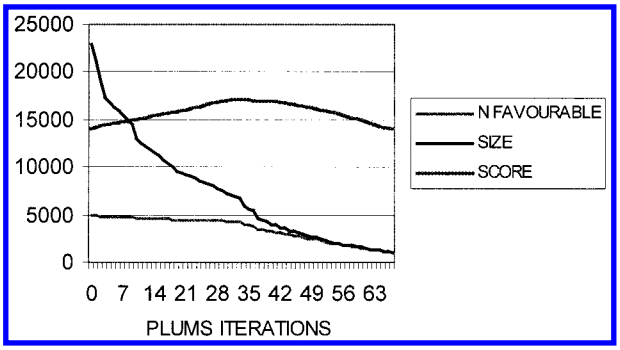

Figure 3. PLUMS results for Library-1. The best library is identified by a maximum in the score (which was appropriately rescaled).

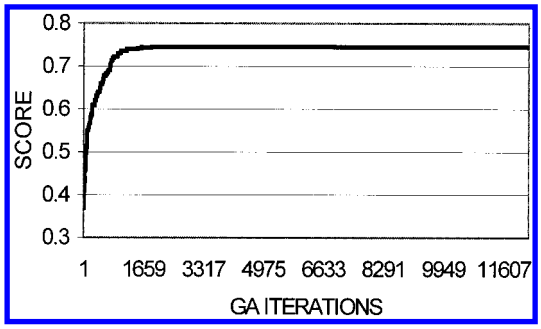

Figure 4. Results from applying the genetic algorithm (GA) to Library-1. The GA was required to optimize the same score formula used by PLUMS (eq 3 in the text). The best score value was found after 1874 iterations. GA- and PLUMS-optimal libraries are identical.

corresponds to 46% of the entire database and 15% of the whole virtual library. These molecules, according to the PLUMS definitions, represent the favorable set; the rest of the molecules are considered unfavorable. However, the 4907 compounds do not represent a combinatorial set. A total of 12 aldehydes, 39 azole aldehydes, and 49 amines are present in the favorable set. Hence a library of 22 932 compounds would have to be synthesized to include all of the favorable molecules in a true combinatorial library. This library would be the most effective we can make (effectiveness = 1.000) but not very efficient (efficiency = 0.214). We therefore used PLUMS to find libraries with a better balance between effectiveness and efficiency.

Figure 3 summarizes the PLUMS results obtained by assigning equal weights to the effectiveness and efficiency. For each iteration, three parameters are reported: size, number of favorable molecules, and score (appropriately scaled). The number of favorable compounds decreases along with the size of the library, while removing monomers, but at a typically much lower rate. The algorithm terminated when a library of $6 \times 11 \times 18 = 1188$ elements was found. This library contains only favorable molecules (effectiveness = 0.242, efficiency = 1.000) and was found after the removal of 65 monomers (6 aldehydes, 28 azole aldehydes, and 31 amines).

The optimal point, representing the best balance between size and number of favorable compounds, is identified by a maximum in the score. In this particular case, all the libraries whose size is between 7500 and 4500 compounds are well-balanced. Smaller libraries are not worth making since, although the efficiency is higher, too many favorable molecules have been lost. The best score is 0.746 (effectiveness = 0.863, efficiency = 0.630) and corresponds to a library of $8 \times 24 \times 35 = 6720$ compounds, which includes 4235 favorable compounds. This library was found at the 33th iteration. Comparing this library with the starting one, it is clear that PLUMS has improved the efficiency while keeping the effectiveness as close to the maximum as possible.

This result was obtained using the all_equivalent option. For comparison purposes, two more runs were executed considering (a) only 1 and (b) up to 10 equivalent solutions at each iteration. Decreasing the number of equivalent solutions taken into account can speed up the program but, in general, decreases the chances of finding the best solution. Both runs, in this case, were able to find the 0.746 best score library. However, differences were noticed at the end of the runs in that the final generated libraries were slightly smaller than the 1188-molecule library obtained using all the equivalent solutions. Run a stopped after 67 iterations, yielding a library of 1056 favorable compounds, while run b produced a library of 1122 elements after 66 iterations.

Because of the way in which PLUMS sequentially removes monomers, it is not guaranteed to find the globally optimal solution, even when using the all_equivalent option. We therefore wanted to compare the results from PLUMS with other popular methods for selecting combinatorial subsets. We have concentrated here on comparisons with genetic algorithms (GA) and monomer frequency analysis (MFA).

The GA herein applied resembles the program GALOPED described by Brown et al.,[10] in the sense that each chromosome represents a different potential library. The chromosome length equals the total sum of monomers, and each gene in the chromosome corresponds to a particular monomer. Genes are binary: a value of 1 means that the corresponding monomer is present while a null value indicates that the monomer is absent. The score for each potential library is calculated in a way identical to PLUMS (eq 3). The main operations in the GA are the following:

1. Constraints on either the min/max size of the library or the min/max number of monomers for each substitution site are specified.

2. An initial population of chromosomes is generated and a score is calculated for each chromosome according to eq 3.

3. Parent chromosomes are randomly selected, children are created by crossover and mutation operations and scored.

4. High-scoring children are inserted in the chromosome population and low-scoring chromosomes are displaced.

5. Return to step 3 if the number of iterations does not exceed the desired limit.

Figure 4 illustrates the result from a GA run (equal weights were assigned to the effectiveness and efficiency). No constraints, as defined in step 1, were applied to this experiment. The best library, corresponding to the highest value in the score, was found after 1874 iterations. Ten thousand more iterations did not further improve this result. The best library from the GA run was identical to that one found by PLUMS.

In a recent paper[11] Zheng et al. suggested monomer frequency analysis (MFA) as a tool to select plausible monomers for focused library optimization. Frequencies of the monomers are computed and compared with the random

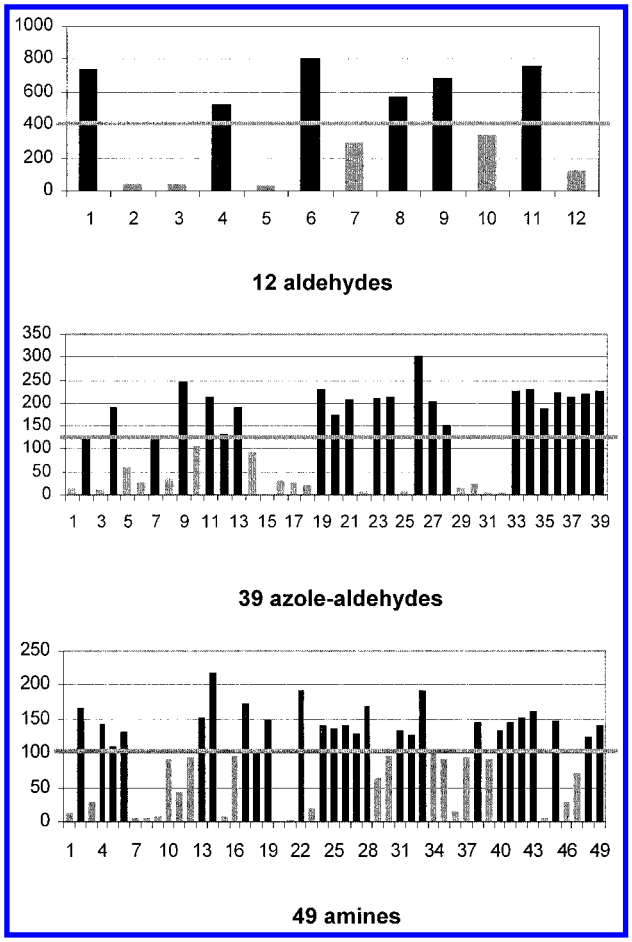

Figure 5. Frequencies for each monomer of each substitution site of Library-1 represented as histogram plots. Straight lines indicate the expected random frequency values (computed according to eq 4 in the text). Monomers with frequencies above these lines (highlighted in plain black) are likely to be statistically significant.

frequency values. Only monomers whose frequencies are above this threshold are considered for the experiment. For each monomer *m* we calculated the frequency as the number of its occurrences in the favorable set (N_m). The frequency of each monomer is compared with the random frequency value which is calculated for each substitution site as the average frequency:

$$\text{random frequency}_s = \frac{1}{N_s}\sum_{m=1}^{N_s} N_m \qquad (4)$$

where N_s is the number of monomers for substitution site *s* which occur in the favorable set. Note that the sum in eq 4 equals the total number of virtual hits, $N_{\text{tot,favorable}}$ (4907 in the case of Library-1).

Figure 5 illustrates the results from MFA. The frequencies of each monomer of each substitution site are reported as a histogram plot. Those monomers having a frequency higher than the random value were selected. In this way, 6 aldehydes, 22 azole aldehydes, and 26 amines were chosen yielding a library of 3432 compounds with 2763 favorable ones. Comparing the monomers chosen by the three techniques, we found that the MFA library represents a subset of the best library found by GA and PLUMS. This means that PLUMS and GA selected monomers whose frequency is below the expected random value. However, no monomers above this threshold were excluded.

The score, as defined in eq 3, for the MFA result is 0.684 (effectiveness = 0.563 and efficiency = 0.805). This value is lower than the one found by PLUMS and GA. Nevertheless, it does represent a rather good result considering the simplicity and the speed of the method which does not attempt to optimize the score in any way. In this regard, one referee pointed out that we should not compare the total scores, but limit the comparison to the raw effectiveness and efficiency. If this is the case, MFA would lead to a less effective (0.563 vs 0.863) yet more efficient (0.805 vs 0.630) library with respect to the best PLUMS result. However, it is worth mentioning that PLUMS, during its iterative process (10_equivalent run), found seven consecutive libraries (iterations 40−46) with scores in the range 0.737−0.720, which are both more effective and more efficient than the MFA result. In particular, the library retrieved at iteration 45, which contains 6 aldehydes, 18 azole aldehydes, and 31 amines, is lower in size (3348 vs 3432) than the MFA solution, yet it contains more virtual hits (2879 vs 2763).

Library-2. The second example was taken from the literature. Leach et al.[6] enumerated a 10 000 molecule library using 100 diverse carboxylic acids (R1) and amines (R2) extracted from the MedChem database. Filters on molecular weight, CMR,[12] number of rotatable bonds, "maximal binding energy",[8] and complexity[13] (the number of bits in the Daylight fingerprint[14]) were then applied (the mean values from the World Drug Index[15] were used as cutoffs). Only 409 molecules passed the imposed bounds (see Table 1 in ref 6).

The 409 molecules represent the favorable set, and PLUMS could be used to maximize the number of favorable molecules and minimize the size of the library. However, in this case, we asked PLUMS to generate the best 10 × 10 library to have a direct comparison with the results reported in ref 6.

Analysis of the favorable set revealed that 67 carboxylic acids and 71 amines were still present. Hence the library, which includes all 409 favorable compounds, contains 4757 molecules. We again applied equal weights for the effectiveness and efficiency. The score of the starting library is 0.543 (effectiveness = 1.000, efficiency = 0.086). The PLUMS process stopped after 118 iterations when 20 monomers, 10 for each substitution site, remained. The number of favorable compounds in the final library was 69, and the score was 0.429 (effectiveness = 0.169, efficiency = 0.690). The starting score was not improved by this library. This result was achieved using the all_equivalent option. The 10_equivalent option found the same result, whereas the 1_equivalent option produced a library containing fewer favorable compounds, 67.

Leach et al.[6] applied the program VOLGA[16] and ended up with an identical solution, 69 favorable molecules. VOLGA is another GA-based program for optimization of combinatorial libraries. As with the GA program in the previous example, VOLGA encodes the whole library in a chromosome. However, it differs because it was developed to optimize libraries of specific size. With this particular example, VOLGA used the efficiency, as defined in eq 2, to score each chromosome.

Leach et al.[6] also applied an approach similar to MFA. The 10 most frequent monomers for each substitution site were chosen. The library obtained contains only 39 favorable

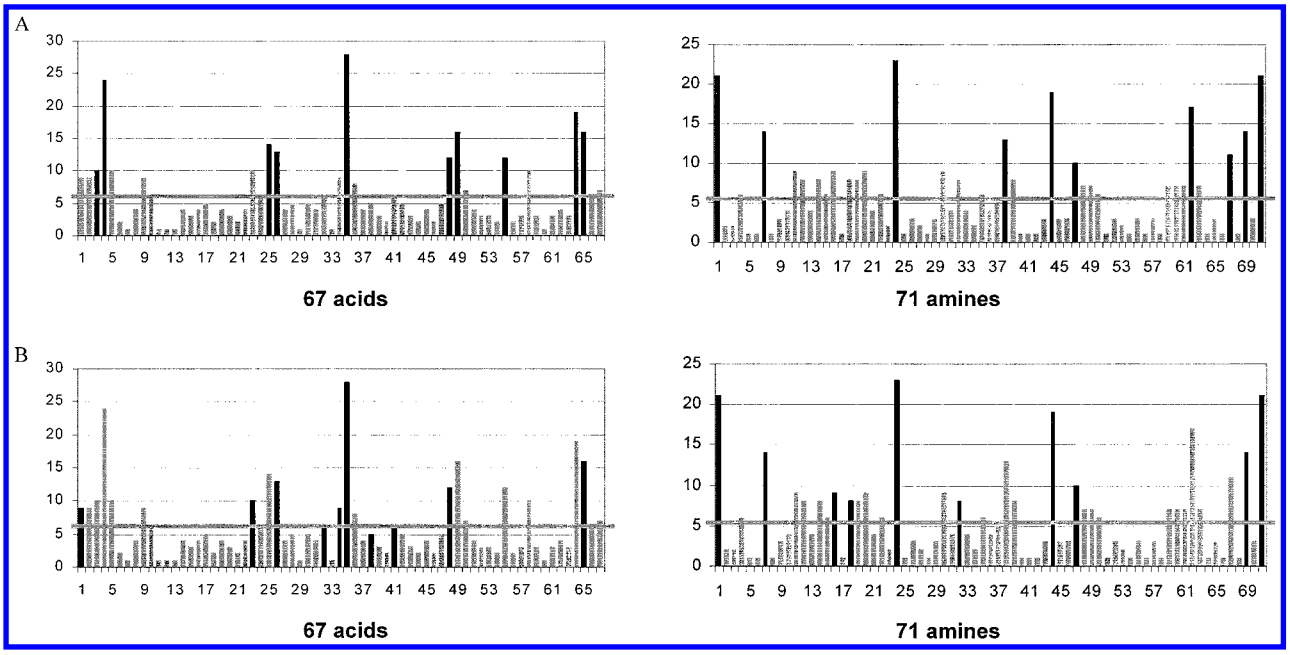

Figure 6. Frequencies for each monomer of each substitution site of Library-2 represented as histogram plots. Monomers highlighted in black are those selected by (a) monomer frequency analysis (MFA) and (b) PLUMS. MFA selected for each reagent the 10 most frequent monomers. Their frequencies are well above the random expected values, indicated by the straight lines (computed according to eq 4 in the text). PLUMS, instead, selected a few monomers with low frequency (below the straight lines). Also some of the most frequent monomers were not picked up. However, PLUMS yielded a better result than MFA (see text for more details).

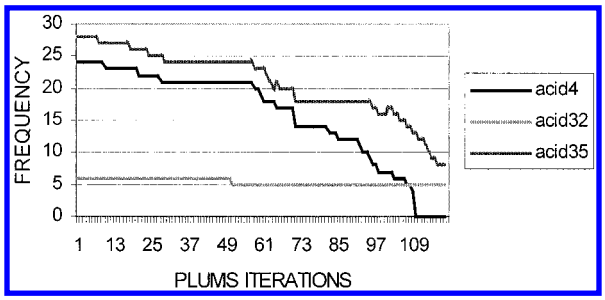

Figure 7. Frequencies of acid4, acid32, and acid35 (first substitution site of Library-2) versus number of PLUMS iterations. At each iteration a monomer is removed from the set so that the frequency scenario changes. In the example acid35 and acid32 are selected by PLUMS for the final experiment. Acid4 is removed at iteration 111 (its frequency drops to zero).

compounds instead of 69 found by PLUMS and VOLGA, despite the fact that the size is the same.

Figure 6 illustrates the acids and amines chosen by the three techniques. The 10 monomers chosen by MFA have frequencies all above the random value. Noteworthy is that PLUMS and VOLGA selected some monomers with frequencies below this threshold, e.g., acid32, acid38, and acid41. Even more surprisingly, monomers with very high frequencies, e.g., acid4, acid64, and amine62, were excluded by the genetic algorithm and PLUMS. Figure 7 shows how the frequency of a few representative monomers changes during the PLUMS iterative process. Acid35 represents a monomer that maintains a high importance during the entire iterative process. Acid4 has a high frequency at the beginning, but then loses importance and is finally removed. By contrast, acid32, which is not a crucial monomer according to its initial frequency (below random expected value), is finally kept for the experiment.

When analyzing in more detail the occurrences of the above-mentioned three acids, we noted that acid32, though not very frequent (thus occurring only six times in the favorable set), does occur up to four times in combination with amines belonging to the top 10 frequency list as defined by MFA. In contrast acid4, despite occurring as many as 24 times, couples successfully (i.e., to give a favorable product) with only two of the top amines. Finally acid35, the most frequent monomer among the acids, occurs in combination with top amines as many as eight times. This is illustrated in Figure 8a, which shows the number of occurrences of each acid in the favorable set when combined with any of the top 10 frequent amines and vice versa. It is clear that some of the top 10 frequent monomers, as defined by MFA (shown in black), are no longer the most frequent ones (cf. Figure 6a). These observations led us to the conclusion that some highly frequent acids (e.g., acid3, acid4, and acid64) are "poor monomers" because they occur with only few of the top 10 frequent amines. On the other hand, some low frequency acids (e.g., acid32, acid38, and acid41) are "good monomers" because they mainly occur in combination with the top 10 amines. A similar analysis of amines versus acids led us to similar conclusions. For instance, low frequency amine32 is "good" because it couples with five of the top 10 acids. Despite that amine62 and amine67 are highly frequent monomers and occur with five and four of the top 10 acids, respectively, they can be considered "poor". Thus a further investigation revealed that these two amines couple with "poor" acid4 and acid64.

These observations prompted us to implement a modified MFA approach, which we called *dynamic monomer frequency analysis* (DMFA). For the specific example of Library-2, DMFA performs the following operations:

1. Compute the number of occurrences of each acid and each amine in the favorable set (N_m).

2. Sort and select the top 10 acids and the top 10 amines.

3. Compute the number of occurrences of each acid in the favorable set when combined with any of the top 10 amines (N_t).

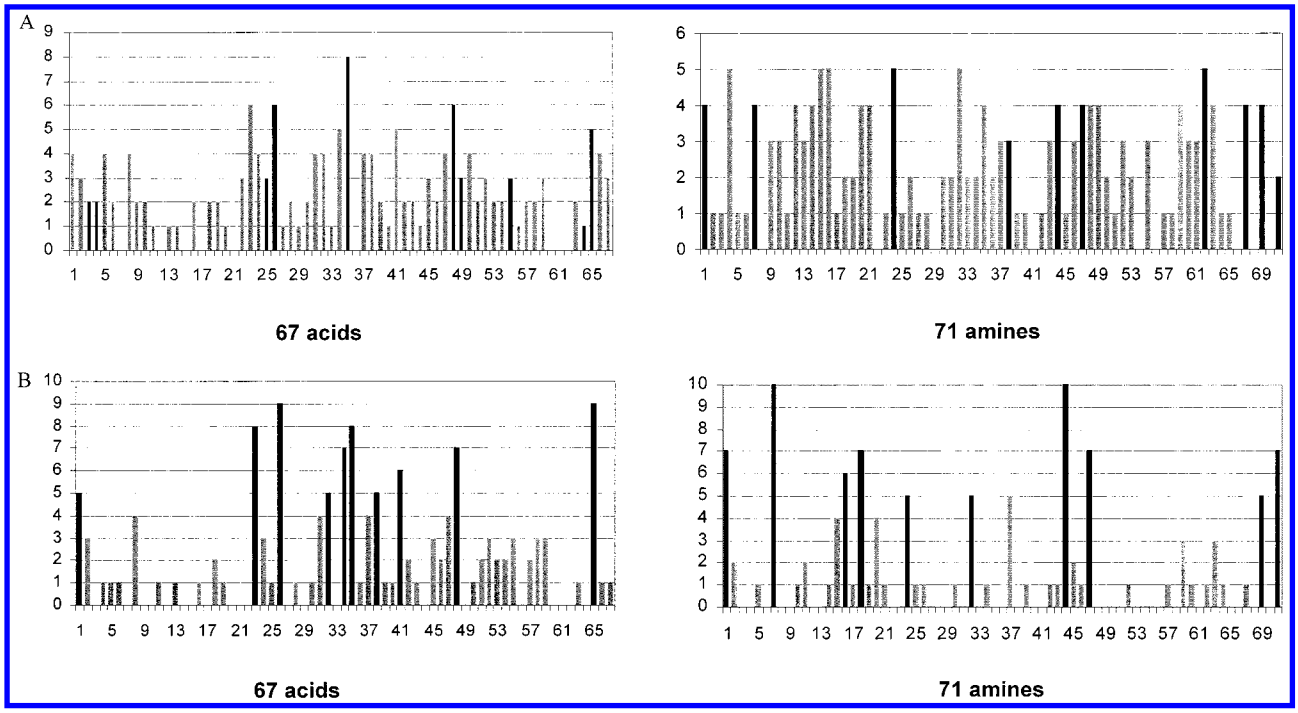

Figure 8. Library-2: number of occurrences of each acid in the favorable set when combined with any of the top 10 amines (left) and number of occurrences of each amine in the favorable set when combined with any of the top 10 acids (right). Top 10 acids and amines are highlighted in black. (a) Top 10 acids and amines as defined by MFA. (b) Top 10 acids and amines as found after the DMFA iterative process.

4. Sort and update the top 10 acids.

5. Compute the number of occurrences of each amine in the favorable set when combined with any of the top 10 acids (N_t).

6. Sort and update the top 10 amines.

7. Return to step 3 until convergence is reached or DMFA runs out of iterations.

DMFA consists of updating the top list of monomers in an iterative fashion. The list of top 10 acids is updated in light of the list of top 10 amines and vice versa. Convergence is reached when there is no change between the lists of top monomers of two consecutive iterations.

For Library-2, DMFA reached convergence after only two iterations and yielded the same monomers selected by PLUMS and VOLGA, thus producing a library with 69 favorable molecules (Figure 8b shows the number of occurrences of each acid in the favorable set when combined with any of the top 10 frequent amines and vice versa after the DMFA iterative process). This considerable improvement over MFA was achieved at a very low cost in terms of computational speed. Further experiments, carried out by selecting libraries with different dimensions, confirmed the robustness and the validity of the DMFA approach.

DISCUSSION

We have defined a focused library to be one that has a bias toward molecules able to satisfy specific two- and/or three-dimensional properties. We define the molecules able to satisfy these constraints either as favorable or as virtual hits. Accordingly, the remaining molecules are defined as unfavorable. PLUMS was developed to maximize the number of favorable compounds in a focused library while minimizing the effort (i.e., size of the library) for doing this. Through an iterative process, PLUMS identifies and removes the less important monomers according to a scoring function. Presently this function is a linear combination of the effectiveness and efficiency. Effectiveness refers to the virtual hits present in the sublibrary relative to the number of favorable compounds in the starting virtual library. Efficiency is the ratio of the number of virtual hits in the sublibrary to the total number of molecules in the sublibrary. The iterative process stops when no clear improvements can be achieved by removing further monomers. PLUMS is not a global optimization algorithm, especially if the user chooses not to consider equivalent solutions during the iterative process. This is because a monomer, once removed, is lost forever. However, if equivalent solutions are taken into account, monomers removed in previous iterations do have the chance to return. Even so, not all possible combinations are explored, and thus PLUMS does not guarantee to find the globally optimal solution.

We have presented the results for two test libraries, and compared the performance of PLUMS with those from genetic algorithms. The two approaches gave equal results in both cases; however PLUMS was notably faster. This comparison is particularly important because, even though both methods are not totally rigorous, the GA does have, in principle, a superior ability to cover the optimization space. Therefore, it is encouraging that the GA did not find better solutions in any case. Also, we believe that the results from PLUMS are generally more informative. Due to the iterative process, PLUMS ranks monomers according to their importance. This is particularly helpful when for reasons of availability, cost, synthetic ease, or some other unforeseen reasons one or more monomers cannot be used. In this situation one simply looks for a monomer further down the list. We also find that the PLUMS output can be useful to help the user decide which libraries are worth making and

which libraries are not by visual analysis of the results. In this regard, when no practical synthetic constraints are limiting the selection procedure, PLUMS can offer valuable indications on the optimal number of monomers to be used for each substitution site. On the other hand, the GA is a very flexible optimization engine. For instance, the program VOLGA,[16] which we used on Library-2, is able to use any form of user-defined fitness function. Another example of the use of a GA in library subset selection is the program SELECT.[17] This program is able to optimize a function which combines chemical diversity (based on Daylight fingerprints[14]), similarity to a certain property profile (e.g., molecular weight profile of the World Drug Index database[15]), and fit to a specific property (e.g., match to a pharmacophore). The possibility to control the tradeoff between different objectives is an appealing feature. For instance, it is generally accepted that any focused library would benefit from a diversity component (which should reflect the degree of information available on the target examined[18]). With SELECT each component (i.e., fit, similarity, and diversity) can be separately weighted and properly scaled to obtain the desired balance. However, the task is not trivial and may require several trial-and-error experiments. PLUMS, instead, in its current implementation, adopts a scoring function which reflects only a library's ability to fit a certain property. Therefore any diversity analysis needs to be run as a preliminary step. This might result in a library that could not represent the optimal balance between focus and diversity. In principle, PLUMS could be modified to make it SELECT-like, i.e., able to simultaneously manage multiple objectives. However, the search space might become too complex for a direct algorithm like PLUMS.

We compared PLUMS with monomer frequency analysis. The analysis of frequencies, i.e., the number of occurrences of each monomer in the favorable set, is a very fast and straightforward tool. Zheng et al.[11] have recently described an application where monomers are selected if their frequencies are higher than the randomly expected values, i.e., the average number of monomer occurrences. This strategy provided a good result in the case of Library-1, although not as good as PLUMS and GA. However, it was less effective in the case of Library-2. A detailed analysis of monomer occurrences highlighted the limitations of such a method and prompted us to implement a modified MFA approach, which we called dynamic MFA (DMFA). This new approach showed considerably improved performances over MFA while still retaining high computational speed. DMFA yielded equivalent results to more rigorous methods on Library-2, and the approach could be swiftly extended to libraries including more than two points of diversity. However, it is worth noting that DMFA can be effectively used only when the number of monomers for each substitution site has been predetermined. If this is not the case, we strongly recommend the use of PLUMS or GA.

CONCLUSIONS

PLUMS is a new program for rapid optimization of focused libraries. It is simple, fast, and highly informative. The number of monomers for each substitution site and the size of the library are parameters that can be either optimized or used to constrain the selection. Presently PLUMS is aimed at finding the best-balanced library in terms of effectiveness and efficiency. Alternative scoring functions or alternative ways to combine effectiveness and efficiency can be easily implemented and explored. Also, PLUMS, in its current implementation, adopts a simple yes/no metric to decide whether each member of the virtual library is desirable or not. However, the method can be easily modified to consider continuous scoring of the library products. PLUMS was applied on two test cases, giving results as good as those from GA and better than MFA. The algorithm is written in PERL and has been recently implemented in ADEPT.[6]

ACKNOWLEDGMENT

We thank Nick Bailey and Duncan Judd for their contribution to the design of Library-1, and Brian Evans and Tony Parkhouse for the synthesis. Xiao Qing Lewell and Emanuela Gancia made useful suggestions to this work, and their contribution is very much appreciated. Finally we thank the unknown referees for their valuable comments.

REFERENCES AND NOTES

(1) *Computational Methods for the Analysis of Molecular Diversity*; Willet P., Ed.; KLUWER/ESCOM: The Netherlands, 1997.

(2) Lipinsky, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and Computational Approaches to Estimate Solubility and Permeability in Drug Discovery and Development Settings. *Adv. Drug Delivery Rev.* **1997**, *23*, 3−25.

(3) Clark, D. E. Rapid Calculation of Polar Molecular Surface Area and Its Application to the Prediction of Transport Phenomena. 1. Prediction of Intestinal Absorption *J. Pharm. Sci.* **1999**, *88*, 807−814.

(4) Clark, D. E. Rapid Calculation of Polar Molecular Surface Area and Its Application to the Prediction of Transport Phenomena. 2. Prediction of blood-Brain Barrier Penetration. *J. Pharm. Sci.* **1999**, *88*, 815−821.

(5) Pickett, S. D.; McLay, I. M.; Clark, D. E. Enhancing The Hit-to-Lead Properties of Lead Optimization Libraries. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 263−272.

(6) Leach, A. R.; Bradshaw, J.; Green, D. V. S.; Hann, M. M.; Delany, J. J., III. Implementation of a System for Reagent Selection, Library Enumeration, Profiling and Design. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 1161−1172.

(7) Leo, A. *CLOGP*; Daylight Chemical Information Systems: Irvine, CA.

(8) Andrews, P. R.; Craik, D. J.; Martin, J. L. Functional Group Contributions to Drug-Receptor Interactions. *J. Med. Chem.* **1994**, *27*, 1648−1657.

(9) *Catalyst*, v. 4.0; Molecular Simulations Inc.: San Diego, CA.

(10) Brown, R. D.; Martin, Y. C. Designing Combinatorial Library Mixtures Using a Genetic Algorithm. *J. Med. Chem.* **1997**, *40*, 2304−2313.

(11) Zheng, W.; Cho, S. J.; Tropsha, A. Rational Combinatorial Library Design. 1. Focus-2D: A New Approach to the Design of Targeted Combinatorial Chemical Libraries. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 251−258.

(12) Leo, A. *CMR*; Daylight Chemical Information Systems: Irvine, CA.

(13) Lewell, X. C.; Congreve, M. S. Complexity Descriptor—A New Structural Descriptor and Its Applications to High-Throughput-Screening and Compound Set Selection and Profiling. *J. Chem. Inf. Comput. Sci.* Submitted for publication.

(14) James, C. A.; Weininger, D. *Daylight Theory Manual*; Daylight Chemical Information Systems Inc.: Irvine, CA, 1995.

(15) The World Drug Index is available from Derwent Information, 14 Great Queen St., London WC2B 5DF, U.K.

(16) Pozzan, A.; Leach, A. R.; Feriani, A.; Hann, M**.** Virtual Optimization of Chemical Libraries Using Genetic Algorithms. Book of Abstracts, 218th ACS National Meeting, New Orleans, Aug 22−26, 1999.

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

(18) Hann, M.; Green, R. Chemoinformatics—a new name for an old problem? *Curr. Opin. Chem. Biol.* **1999**, *3*, 379−383.

CI000389+