ELSEVIER

Designing targeted libraries with genetic algorithms

Robert P. Sheridan, Sonia G. SanFeliciano,[1] and Simon K. Kearsley

Department of Molecular Systems, Merck Research Laboratories, Rahway, New Jersey, USA

In combinatorial synthesis, molecules are assembled by linking chemically similar fragments. Because the number of available chemical fragments often greatly exceeds the number that can be used in one synthetic experiment, one needs a rational method for choosing a subset of desirable fragments. If a combinatorial library is to be targeted against a particular biological activity, virtual screening methods can be used to predict which molecules in a virtual library are most likely to be active. When the number of possible molecules in a virtual library is very large, genetic algorithms (GAs) or simulated annealing can be used to quickly find high-scoring molecules by sampling a small subset of the total combinatorial space. We previously demonstrated how a GA can be used to select a subset of fragments for a combinatorial library, and we used topology-based methods of scoring. Here we extend that earlier work in three ways. (1) We demonstrate use of the GA with 3D scoring methods developed in our laboratory. (2) We show that the approach of assembling libraries from fragments in high-scoring molecules is a reasonable one. (3) We compare results from a library-based GA to those from a molecule-based GA.

Keywords: SQ, FLOG, library design, stromelysin, angiotensin II

INTRODUCTION

The idea behind combinatorial synthesis is that large libraries of chemical compounds can be created by joining a *basis set* of reagents into short oligomers. One of the problems in combinatorial synthesis is that the number of available reagents is much larger than the number of reagents that actually can be used in the basis set for a given experiment. For instance, there are thousands of chemically suitable basic amines, but an experiment typically will use a few dozen at most. The field of combinatorial library design[1-18] addresses the problem of rationally selecting subsets of n reagents from a large universe of N reagents, where n << N. Early on, most work concentrated on designing very diverse libraries, with the idea that these libraries would be tested against a variety of biological assays. Nowadays it is equally common to design targeted libraries to have a maximal activity in a particular assay. Usually the molecular modeler is given the reaction scheme for the library and must do the design within that framework.

The field of "virtual screening" (reviewed in Walters et al.[5]) has grown up around designing targeted libraries by scoring molecules from a virtual library by some computational structure-activity method. Virtual libraries are generated by constructing connection tables for molecules by joining reagents *in silico* with the appropriate simulated chemistry. For our purposes, we will refer to a "fragment" as that part of a reagent that remains after it is incorporated into a molecule. For instance, a carboxylate reagent has two carboxylate oxygens. If incorporated into an amide, it becomes a carboxylate fragment with only one oxygen. Scoring methods can be 2D (e.g., the topological similarity to a target molecule or the predicted activity by some topological QSAR) or 3D (interaction energy with a receptor, possession of a pharmacophore, score on a CoMFA field, etc.) The assumption is that the highest scoring molecules will be most likely to have the desired activity and either the molecules, or the fragments contained in them, should be given priority in experimental design. One particularly fruitful subfield focuses on finding oligomers that will bind to a particular receptor. These methods[6-11] start with a common scaffold, say a peptide, already positioned in a receptor. Side chains are substituted at particular positions, perhaps flexed, and then scored. This approach has the advantage of addressing combinatorial chemistry at the level of fragments rather than molecules, so exhaustive investigation of all fragments is possible. In contrast, there are other approaches where entire molecules are scored, but the fragment universe is small enough that all molecules can be enumerated and scored in a reasonable time.[12-14]

Many practical problems in library design, however, involve large fragment universes and situations where scoring entire molecules is desirable. An exhaustive exploration of fragment

Color Plates for this article are on page 525.

Corresponding author: Robert P. Sheridan, Department of Molecular Systems, Merck Research Laboratories, P.O.B. 2000, Rahway, NJ 07065, USA. Tel.: 732-594-3859; fax: 732-592-4224. *E-mail address:* sheridan@merck.com (R.P. Sheridan).

[1]Present address: Dept. Quimica Organica, Universidad de Salamanca. Pz. de los Caidos 1-5, 37008 Salamanca, Spain.

655 Avenue of the Americas, New York, NY 10010

1093-3263/00/$–see front matter
PII S1093-3263(00)00060-7

space is not practical for such problems, and stochastic methods for exploring the space, such as genetic algorithms (GA) or simulated annealing (SA), must be applied. Both GA and SA are efficient ways of searching large combinatorial spaces. The results of both depend on particular series of pseudorandom numbers, so multiple runs usually need to be done, and there is no guarantee that the global best solution will be found. However, good results usually are found much more quickly than in a purely random search or a systematic search.

There are two levels at which one can apply GAs or SAs. One can make the fundamental unit the molecule, or the library. For example, say we decide to make a library of A-B-C trimers. In a molecule-based GA, for instance, we would start with a random population of individual molecules. Higher-scoring molecules would be selected for survival, and "offspring" from these molecules would be generated by mutation and crossover. The mean score of the population would rise and eventually level off. (SA would similarly produce high-scoring molecules by mutation and Boltzmann-weighted selection.) The score can be the predicted activity by some structure activity method, but other molecule-based properties also may be included (molecular weight, predicted permeability,etc.). Given the final population of high-scoring molecules, we could see what As, Bs, and Cs they contain. Our basis sets probably should contain these fragments. The frequency of fragments in the final population has been suggested[15-17] as one figure of merit for selecting them from the final population. Molecule-based GAs or SAs are most suited to designing targeted libraries.

In contrast, one can start with a population of libraries, each implicitly containing a large number of individual molecules, and evolve toward high-scoring libraries. Each library is already by definition a basis set, so the highest-scoring libraries at the end of a GA or SA run can be used directly. One potential limitation is that the desired number of As, Bs, and Cs (n_A, n_B, n_C) must be decided beforehand. Library-based GAs or SAs have been used mostly for generating diverse libraries (for example, Brown and Martin[18]), because diversity is easily defined for libraries, but is undefined for individual molecules. The library-based approach also can be used for generating targeted libraries, although few examples have appeared in the literature to date.

In a previous article,[15] we demonstrated the use of a molecule-based GA to find high-scoring molecules based on topological scoring methods. The GA proved to be remarkably efficient at this. In one of our test examples, it was able to select a specific target molecule out of a combinatorial space of 10^{10} after scoring only a few thousand molecules. It was suggested in that article that selection of fragments for a library should be done based on the frequency of fragments in the set of high-scoring molecules. A number of issues were not addressed in that article, and most have not been addressed to date in the literature. Is it necessary to score entire molecules? Does picking frequent fragments from high-scoring molecules lead to high-scoring libraries? What are the relative merits of molecule-based GAs *vs* library-based GAs? In this article, we will extend our previous GA scheme to include 3D scoring methods. We will show, using a 2D and a 3D example, how libraries created from fragments taken from molecule-based GA runs score relative to molecules in the GA. For the 2D example, we compare the results of a library-based GA to those of a molecule-based GA.

METHODS

Overview

We encoded the GA as a Unix shell script that ties together several FORTRAN, C, or PERL programs. The scheme is represented in Figure 1. An input control file contains parameters for the run (population size, number of generations, convergence criterion), names of files containing the fragment universes, parameters for conformation generation and scoring, etc. For molecule-based GA, the program **random_molecules** generates a population of molecules by randomly assigning for each residue a number from 1 to N, where N is the number of fragments in the universe for that residue. For example, part of the population of A-B-C trimers could look like:

	A	B	C
Molecule 1	1615	435	2540
Molecule 2	5	1234	601
Molecule 3	2134	456	301

For a library-based GA, part of the population might look like:

	A	B	C
Library 1	5,7,1615	4,435,456	3,301,601
Library 2	15,73,185	14,358,489	38,1011,2022
Library 3	40,58,115	28,64,486	54,381,641

for a 3 × 3 × 3 library. For further processing, each library here would be expanded into 27 molecules. It is clear that a library-based GA is generalization of the molecule-based GA in the sense that there can be any number of instances (here three) of a fragment per member of the population. The program **fuserk** takes each list of molecules and generates a connection table by following the instructions on how to join the original fragments. This is discussed in detail in our previous article.[15] Molecules are scored by either a 2D or 3D scoring function that predicts the desired activity. Supplementary terms can be added to the molecule score. In either case, the score of a library is the mean score over all its constituent molecules. The program **get_stats** monitors the average, minimum, and maximum score for the population and writes the population and scores to a log file. The utility **converg** monitors the record of scores to decide whether the GA run has converged. The program **mutate** prepares the next population from the previous one. These steps are discussed in detail in the following.

2D Scoring Function

In our previous article,[15] one way to score a virtual molecule was to calculate its descriptor similarity to one or more probe molecules with interesting activity. Another way was to score against a trend vector. In our original work, the similarity metric used only the regular atom pair descriptor (AP). However, subsequent work[19] has shown us that using the AP in combination with other descriptors often gives better prediction of activity. The binding point (BP) descriptor, for instance, directly captures information on physiochemical types (cations, H-bond donors, etc.). Here we will set the score as the AP+BP

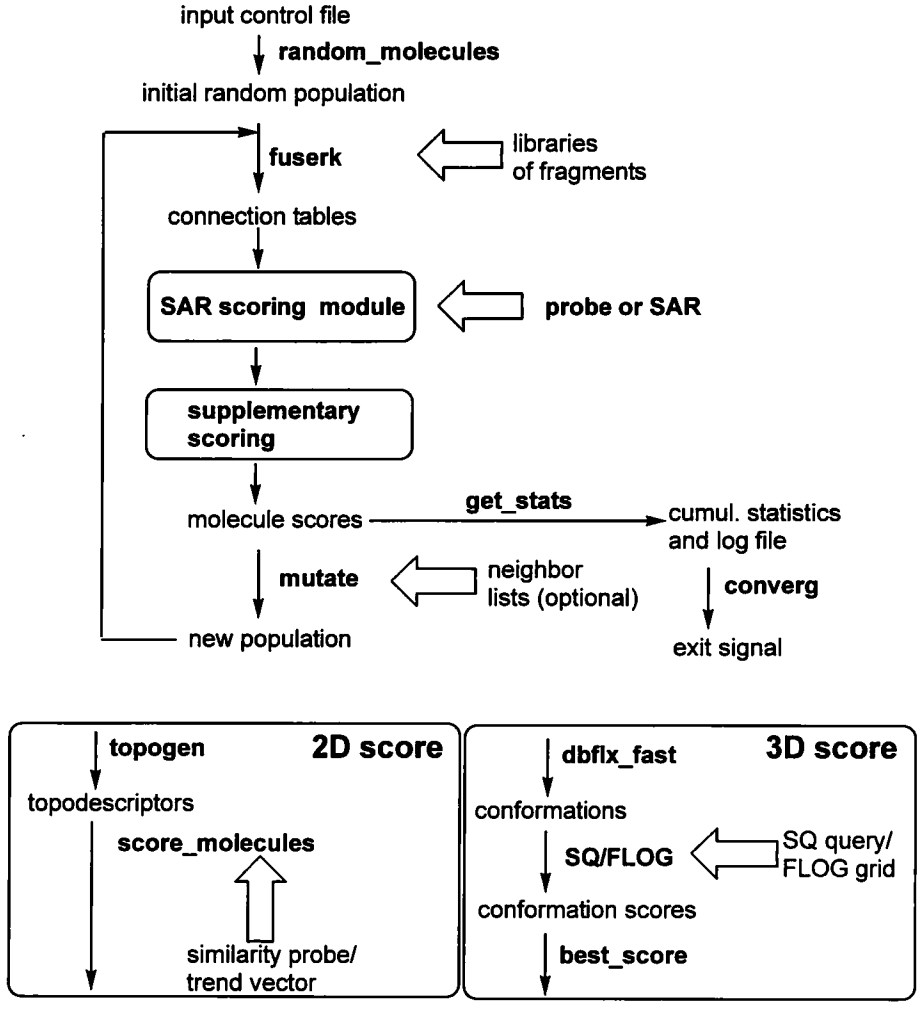


Figure 1. Schematic diagram of the genetic algorithm script. FORTRAN, C, or PERL programs are indicated in bold.

similarity, which is the mean of the similarities using AP and BP descriptors.

3D Scoring Function

The 3D scoring methods are more complicated in that the conformations of the virtual molecules become an issue. The script **dbflx_fast** generates a set of 3D conformers from the connection tables. We use a variation of the flexibase-building methodology given previously.[20] In that work, explicit diverse conformations are generated from a connection table through the following steps: generation of an initial 3D structure from a 2D drawing using a rapid in-house molecular mechanics method called **idealize**, generation of many conformations by distance geometry, cleanup of the distance geometry structures by **idealize**, and selection of a subset of diverse conformations by a sphere-exclusion algorithm based on the root mean square (rms) deviation between conformations. For **dbflx_fast**, we make four changes. First, the 2D→3D conversion is made with the builder CORINA[21] (Version 1.7) rather than **idealize**. This is necessary because the connection tables from **fuserk** have arbitrary 2D coordinates that usually cannot be minimized into reasonable 3D structures. Second, conformations are generated by an in-house rule-based method called **et**[22] instead of by distance geometry. This is faster and tends to produce more energetically realistic conformations. Stereochemistry is inconsistently treated in many fragment collections and requires special consideration. If the absolute chirality of a stereo center is indicated, **et** will preserve it. If it is not indicated, **et** will allow that center to invert. Third, the selection of diverse conformations is made before the structures are cleaned up. This greatly speeds up the generation of conformations by reducing the use of molecular mechanics. For the work here, we requested **dbflx_fast** to produce up to 50 diverse conformations with at least 1.2Å rms between them.

The conformers can be scored using one of two related methods, SQ[23] or FLOG.[24] SQ is a method of superimposing a candidate conformation onto a target molecule; FLOG is a method of docking a candidate conformation into a known receptor site. Both SQ and FLOG use a set of "match centers" as a target. A distance-based clique-finding algorithm is used to find sets of candidate atom-match center pairs wherein the types are similar and the distances are the same within a tolerance. One can specify that certain match centers are "essential," i.e., must be matched with some candidate atom.

Typically there are many sets of pairs per candidate. Each match generates an initial orientation, which is then SIMPLEX-optimized according to a scoring function. The ori-

entation with the highest final score is taken as the orientation of the candidate. In SQ, the scoring function is based on atom-centered Gaussians. It measures the overlap of similar atoms of the candidate and the target. One can add supplementary terms to the scoring function, such as the "cavity term," which penalizes the candidate molecule when any of its atoms move beyond a certain distance from the match centers in the target. In FLOG, the scoring function is a grid that stores the pseudointeraction energy of an atom of a given type at various positions around the receptor.

In both SQ and FLOG, more positive scores mean better similarity or fit to the receptor. Very negative scores occur if there is a severe violation of the cavity term (SQ), a steric clash with the receptor (FLOG), or if an essential point cannot be matched (either method). For the application of GA, we set all negative scores to zero. A score for a molecule is taken as the score of its highest-scoring conformation. This is done by the utility **best_score**.

Selection/Generation Protocols

Several protocols for generating a new population of molecules (or libraries) from the old one with the program **mutate** were discussed in our previous work.[15] We found that *best third* selection with *neighbor* mutation worked well for 2D scoring, and we will follow that scheme here. In *best third* selection, an elitist algorithm, the highest-scoring third of a population is saved and three copies are made. One copy survives unchanged to the next generation, one copy is mutated, and one copy is crossed over. In *neighbor* mutation, an allele can mutate only to 1 of the 10 fragments most similar to it in topological similarity. To use this option, one must have a list of neighbors for each fragment appropriate for a given fragment database.

Protocols for mutation and crossover for libraries are analogous to those for molecules, except that there is one more layer for making choices. For a molecule, a mutation would be:

```
A  B   C      A  B   C
11 45  135 →  11 89  135
```

where residue B was randomly chosen for mutation. (Here the fragment B-89 is assumed to be a neighbor of B-45). For a library, a mutation would look like:

```
   A          B          C
11,33,38  33,45,78  108,135,602

                        A          B          C
                  → 11,33,38  33,78,89  108,135,602
```

where residue B was randomly chosen, and then "instance" 2 of residue B was chosen. Note that the instances in residue B are rearranged in ascending order. A similar situation applies to crossovers. For molecules, a residue is randomly chosen at which the crossover is to be made. For libraries, a residue is randomly chosen, and then an instance of that residue. If more than one instance of the same fragment exists in one residue after crossover, say two copies of fragment B-45, mutations are made until there are no duplicates.

Convergence Criterion

In our previous work, we ended each GA run after a fixed number of generations. It was clear, though, that much computer time was spent unnecessarily on populations that were not improving in score. Wasted time can be significant for 3D scoring and library-based GAs. Therefore, in this work we introduce a test to stop a run. The routine **converg** monitors two ratios at the end of Generation i:

$$\frac{[\text{meanscore}(i) - \text{meanscore}(i-3)]}{[\text{meanscore}(i) - \text{meanscore}(0)]}$$

$$\frac{[\text{maxscore}(i) - \text{maxscore}(i-3)]}{[\text{maxscore}(i) - \text{maxscore}(0)]}$$

where meanscore is the mean score of the population, maxscore is the maximum score, and generation 0 is the initial random population. When both of these ratios go below a cut-off, we consider the score curves sufficiently "leveled off" to end the GA run. Experimentation with molecule-based GAs has shown that once a cut-off of 0.02 is met, extending the run usually is unproductive.

Fragment Universe

In this article, we will construct A-B-C trimers wherein A is a carboxylate, B is an amino acid, and C is a primary or secondary basic amine. These are taken from the ACD Version 97.1[25] with the following constraints:

1. Number of nonhydrogen atoms $\leq$20.
2. There are no interfering chemical groups: oximes, sulfhydryls, azides, nitros, and nitrosos.
3. There should be no more than one additional reactive group of the same type as required by the fragment. For the amines, there can be up to one additional primary or secondary amine in the molecule. For the carboxylates, up to one additional carboxylate is allowed. For amino acids, either one additional amine or one additional carboxylate is allowed.

For fragments with more than one reactive group, duplicates were made and modified. For instance, there would be two copies of glutamate, one with the alpha-carboxylate as the reactive group and one with the gamma-carboxylate as the reactive group.

In the ACD, there are many fragments that are topologically identical but differ in salt form or stereochemistry. We ignored counterions and eliminated molecules that were topological duplicates with two exceptions. If the fragments differed in stereochemistry, we kept all (e.g., L- *vs* D-alanine *vs* racemic alanine). If the position of the reactive group were different, we kept both (e.g., the two copies of glutamate discussed earler).

The final fragment universe had 5,321 carboxylates, 1,030 amino acids, and 2,851 basic amines.

Examples

We will look at two examples, one using topological scoring and one using FLOG scoring. The first example simulates a situation, usually early in a project, where not much information is available to the modeler other than the 2D structure of

some actives. The goal is to construct a library that best resembles the actives. The second example simulates the situation where one has an active site and wishes to design a library of compounds likely to bind to it.

For the topological scoring we will use the similarity of the candidates to the joint probe (descriptor average) of selected angiotensin II (A-II) antagonists from the MDL Drug Data Report (MDDR).[26] These are shown in Figure 2.

For the FLOG example, we will score A-B-C trimers as potential inhibitors of the metalloprotease stromelysin-1. The structure, solved at Merck,[27] of stromelysin-1 plus a bound inhibitor L-702842 is available as the Protein Data Bank[28] dataset 1SLN. L-702842 is a carboxyalkylpeptide. The carboxylate acts as a zinc ligand. A homophenylalanine side chain fits into the S1′ site, which is a predominantly hydrophobic tunnel. An arginine side chain extends into solvent, and the N-phenyl cap binds in a hydrophobic pocket. An active site was extracted within 8Å of L-702842, L-702842 was removed, and a FLOG grid was generated around the active site using the FLOG protocols.[24] Match centers were created at interaction maxima on the grid. The match center nearest the catalytic zinc was defined as "essential by type" and could be matched only by an atom that is an anion. The active site with the interaction field contoured is shown in Color Plate 1.

RESULTS

Three molecule-based GA runs each were performed against the two queries using a population of 300. Runs were terminated when the convergence criteria were met. A typical 2D run takes <1 minute per generation on a Silicon Graphics Indigo2 with R10000 processor. A typical 3D run takes roughly 2 to 2.5 hours per generation. Three library-based GA runs were made against the A-II query assuming a library of 5 × 5 × 5 and a population of 300. These take 2 hours per generation. A library-based GA run on the FLOG query would take an estimated 150 days and was not done.

Molecule-Based GA for A-II

The mean and maximum score of the three GA runs as a function of generation is shown in Figure 3 (top). As expected, each run converges at a different number of generations and, generally, the final scores can be different. Making the convergence criterion more stringent did not change the result. The top scoring molecule in the final generation is shown at the top of Figure 4. This is the most similar A-B-C trimer to the probe we can make out of the fragments, and one can see the resemblance of these molecules to the target molecules, in particular the aromatic acid group and the short aliphatic chain. To confirm that these structures, assembled on the basis on a topological scoring system, are reasonable in 3D, we attempted docking some molecules from the last generation of Run 1 on the crystal structure of a macrocyclic A-II antagonist[29] using SQ.[23] Several can make plausible superpositions onto the macrocycle. A selected set is shown at the bottom of Figure 4. One superposition is shown in Color Plate 2. The five most frequent fragments from the highest-scoring 100 molecules in the GA are shown in Figure 5. Although the fragments and their frequencies vary from run to run, one can clearly see commonality between the runs.

One type of experiment to compare against the molecule-based GA is to score each universe of individual fragments (with an appropriate capping group(s)) against the joint probe

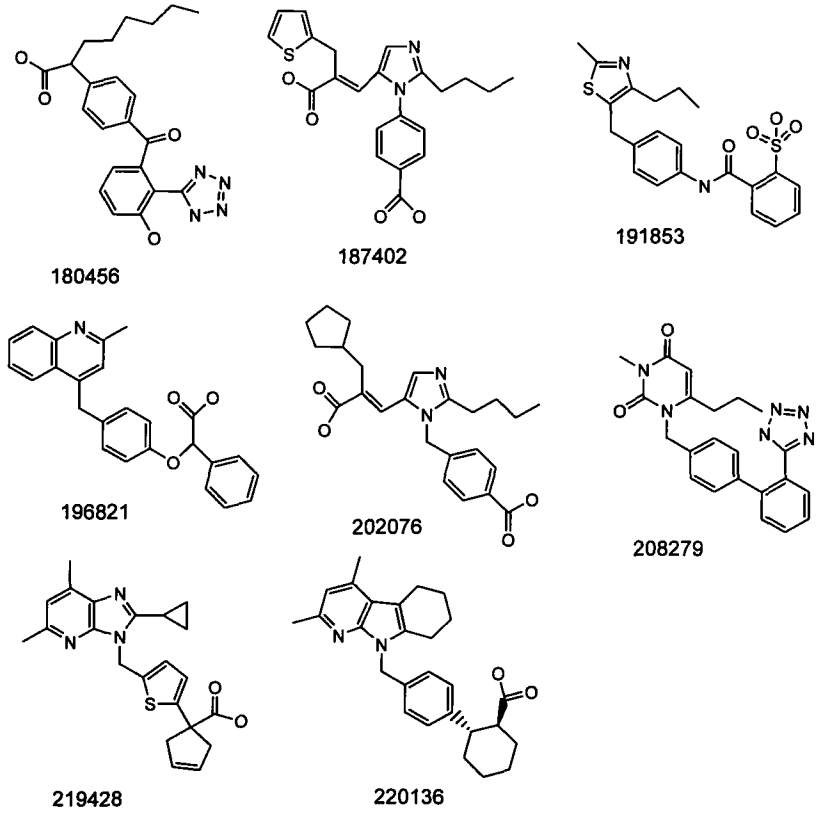


Figure 2. Diverse angiotensin-II antagonists selected from the MDDR database that constitute a joint probe for the A-II example.

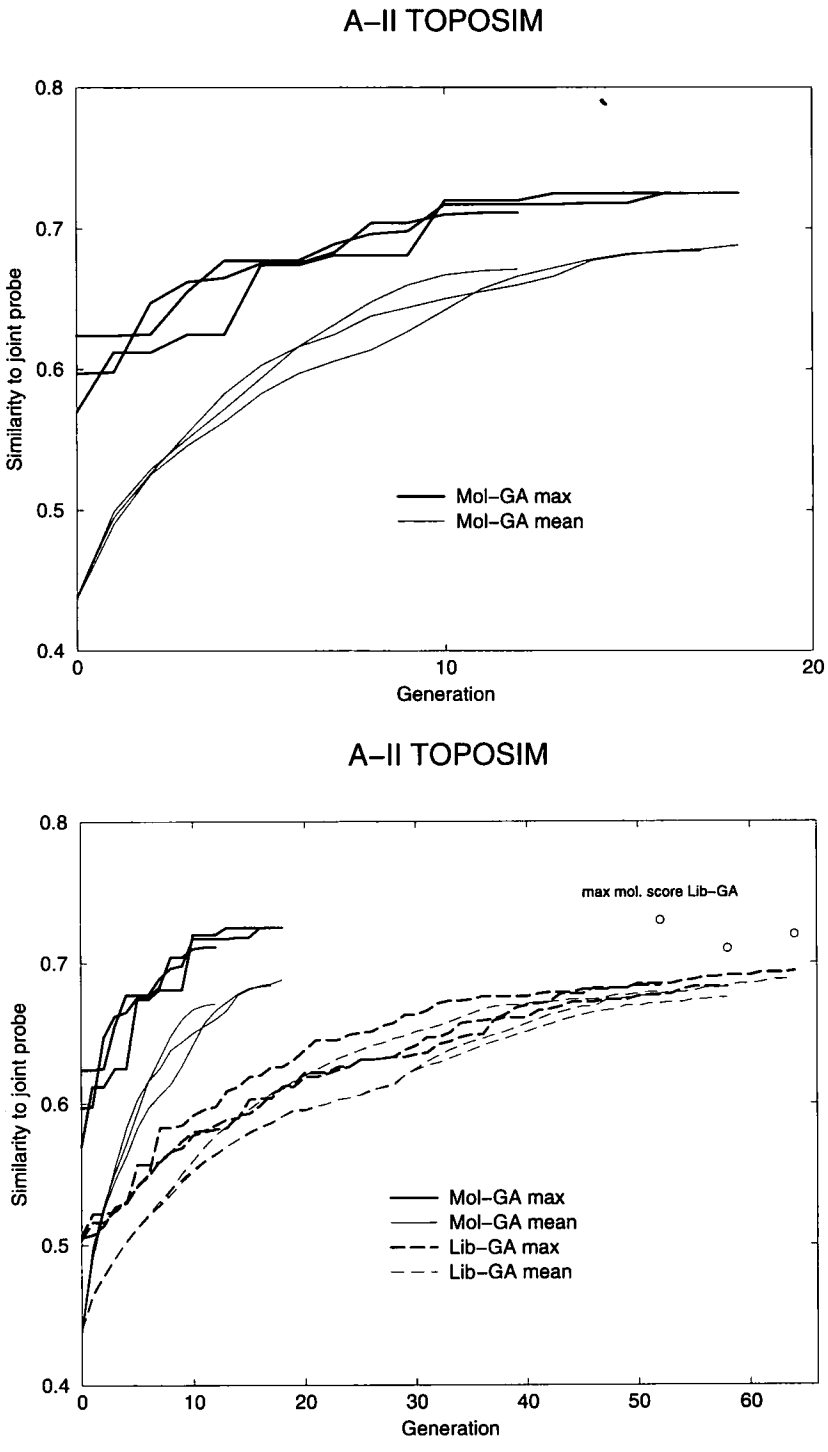


Figure 3. (Top) Mean and maximum score of a population of 300 molecules as a function of generation for molecule-based GA runs on the A-II example. Three runs are shown, each with a different random number seed. The number of unique molecules scored during the runs are 3433, 2575, and 3614. The score is the similarity of a molecule to the joint probe in Figure 2 using a combined AP+BP descriptor (see text for reference). (Bottom) Mean and maximum score of a population of 300 libraries as a function of generation for library-based GA runs on the A-II example. The score of each library in the population is taken as the mean score of all individual molecules in that library. A circle indicates the score of the highest-scoring individual molecule produced by each run. The curves for the molecule-based runs are shown for reference. The number of unique molecules scored in the runs are 210803, 187165, and 189631.

and list the five highest scoring As, Bs, and Cs. When this is done, the fragments look nothing like the fragments in Figure 5. When they are assembled into molecules, the molecules have scores similar to those expected from a random selection of fragments (the highest score is 0.58). The obvious reason for this is that there is nothing forcing the individual fragments to map to distinct parts of the probe molecules. Thus, the premise of the GA approach that it is necessary to score entire molecules, rather than just individual fragments, is justified.

Library-Based GA for A-II

The mean and maximum score of the library-based runs are shown in Figure 3 (bottom). Because the scores rise so slowly, the convergence cut-off of 0.02 ends the runs before the scores fully level off. A cut-off of 0.005 is more appropriate and that is what we use here. Note that the "maximum score" for a library-based run is really the average score over 125 molecules and should be compared to the "mean score" for the molecule-based runs. The score of the highest-scoring individ-

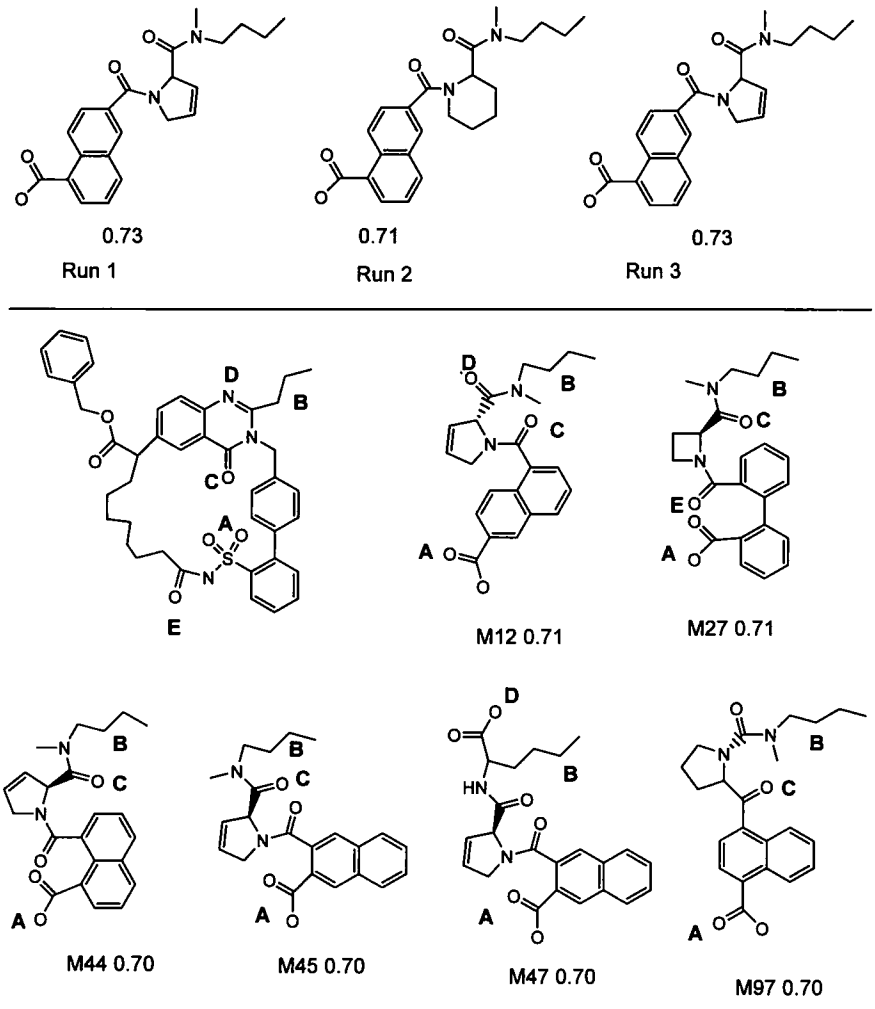


Figure 4. (Top) Top-scoring molecules from each molecule-based GA run and their scores. (Bottom) Diagram of a macrocyclic A-II antagonist is shown, with key groups labeled A–E. Structures of selected molecules from the top-scoring 100 molecules from Run 1 that can superimpose with the antagonist in 3D are listed. Spatially equivalent groups are labeled with the same convention.

ual molecule in the highest-scoring 100 libraries is shown for the last generation. They are in the range of the highest-scoring molecules from the molecule-based GA (0.71–0.73). The fragments in the highest-scoring library for the last generation are shown in Figure 6. Although an exact match to all the fragments from the molecule-based GA runs in Figure 5 was not obtained, many are in common.

Comparison of Score Distribution for A-II

In this section, we will compare the distribution of scores of molecules from the molecule-based GA and molecules from various libraries generated in a variety of ways. One way to do this is to show the distribution of scores. The scores of molecules are sorted from high to low and presented as a plot of score *vs* rank. In Figure 7, the scores of the top-scoring 100 molecules (i.e., the elite third) from the molecule-based GA are shown as bold solid lines. All have high scores. For each run, a 5 × 5 × 5 library is constructed from the most frequent fragments in those same molecules (the fragments in Figure 5). The curve for each set of 125 molecules is displayed as a thin solid line. In this example, the two sets of lines are close to each other, implying that libraries constructed from the most frequent fragments generally have as high scores as individual molecules from the GA. This is a robust result. When the population is changed to 1,000 or the size of the library is changed to 20 × 20 × 20, similar results are obtained. The highest-scoring library from each library-based GA run is shown as a dotted line. These curves are generally indistinguishable from the curves for the library constructed from the most frequent fragments. Finally, three 5 × 5 × 5 libraries created by fragments randomly selected from the universe are plotted as dashed lines. As expected, these have the lowest scores.

1SLN Molecule-Based GA

Mean and max score as a function of generation is shown in Figure 8. Color Plate 3 shows the highest-scoring molecule from each run docked in the site. Figure 9 shows schematic diagrams of the highest-scoring molecule from each run. Generally, other than being peptide-like, the molecules produced by the GA do not closely resemble L-702842. However, the dockings appear reasonable. The scores tend to be higher than for L-702842 because more contacts are made with the receptor site. In particular, some penetrate further into the S1′ pocket. The docked molecules form some, but not all, of the H-bonds formed by L-702842 with the receptor, and sometimes form new ones, e.g., to the backbone oxygen of Leu-222. Detailed inspection of docked molecules shows that the anionic group

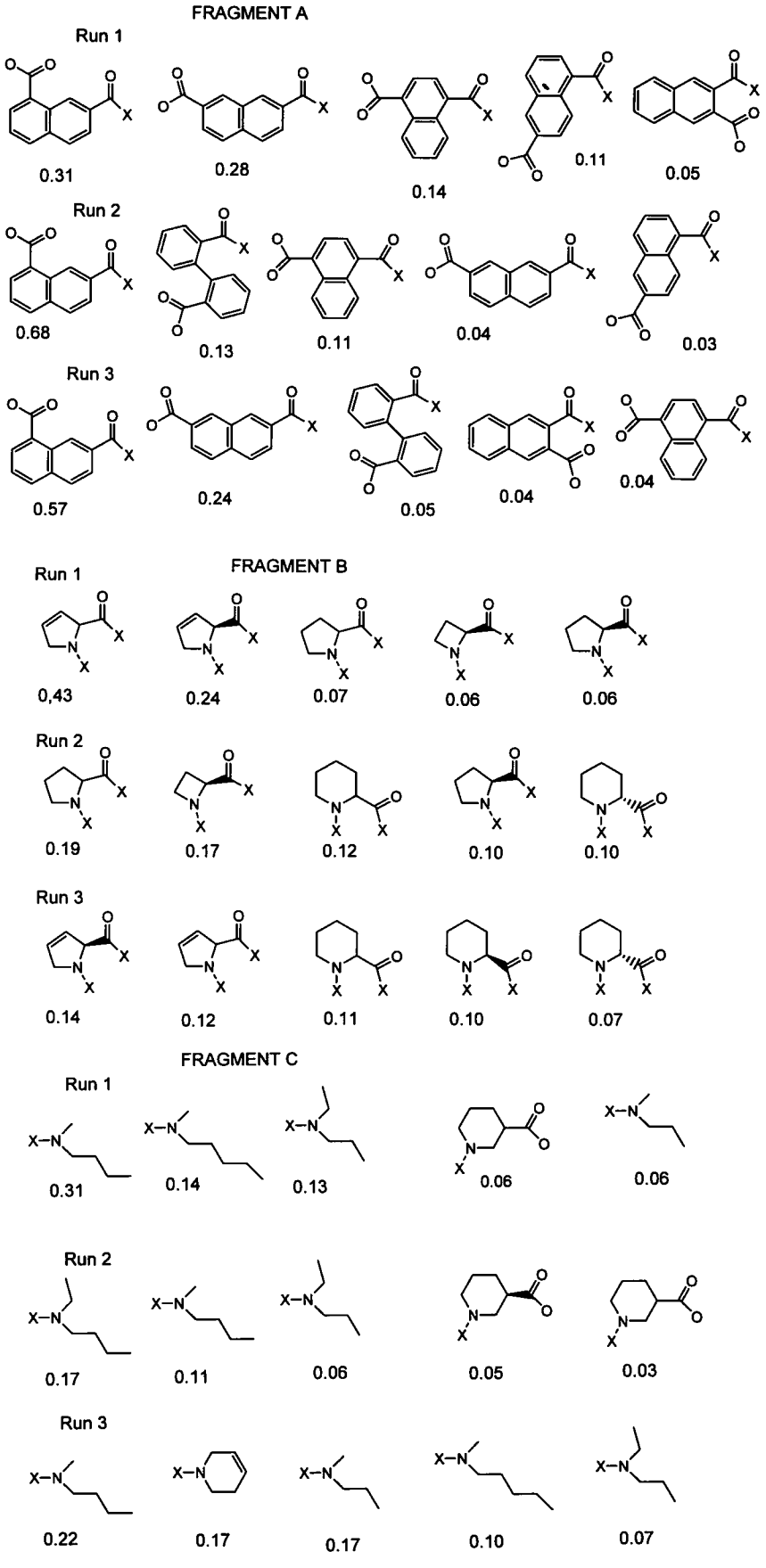


Figure 5. The five most frequent fragments in the top-scoring 100 molecules from the molecule-based GA runs for the A-II example. The frequency of the fragments is indicated. Two or more topologically equivalent molecules may appear because stereoisomers are treated as distinct fragments in our fragment universe.

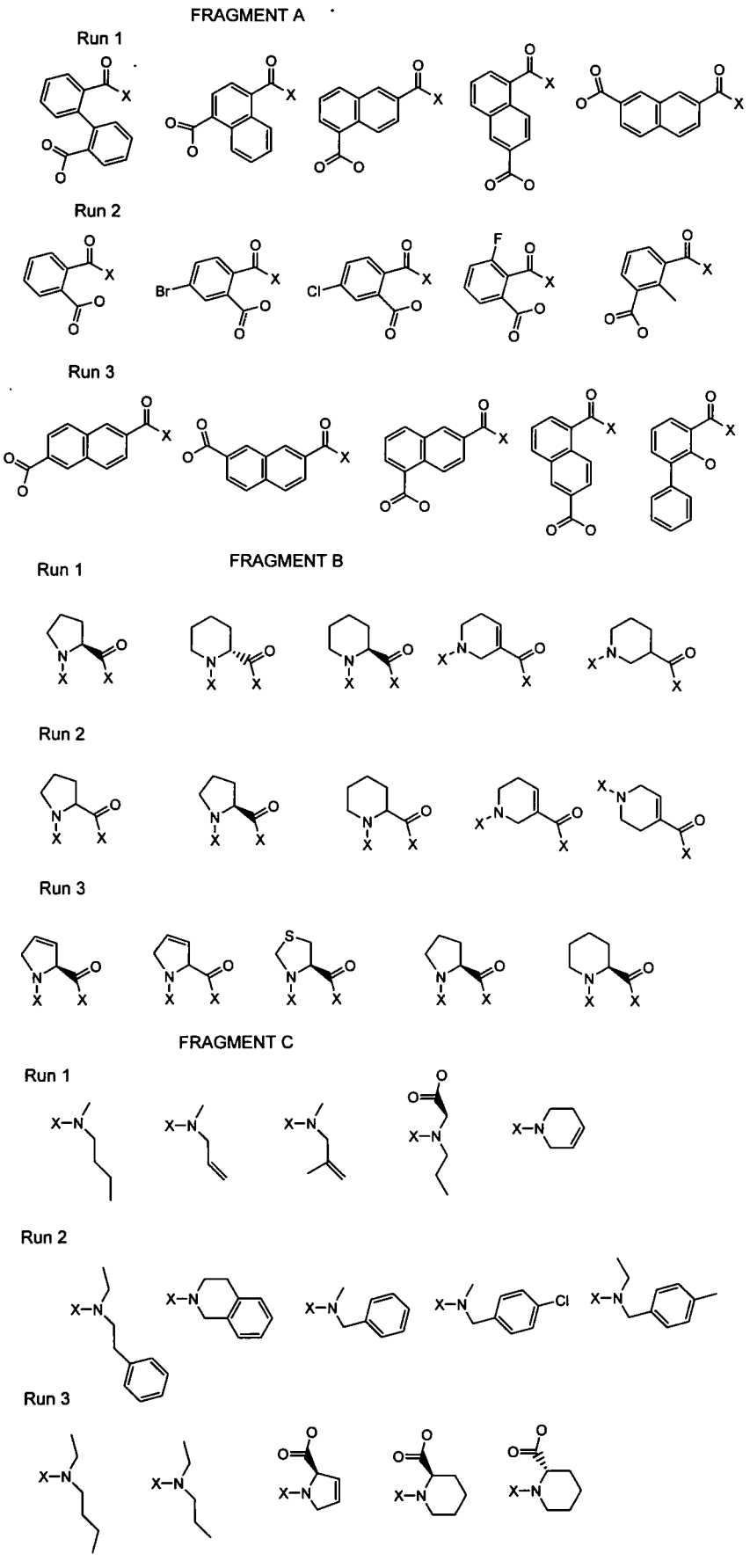


Figure 6. The fragments in the highest-scoring library from the library-based GA on A-II.

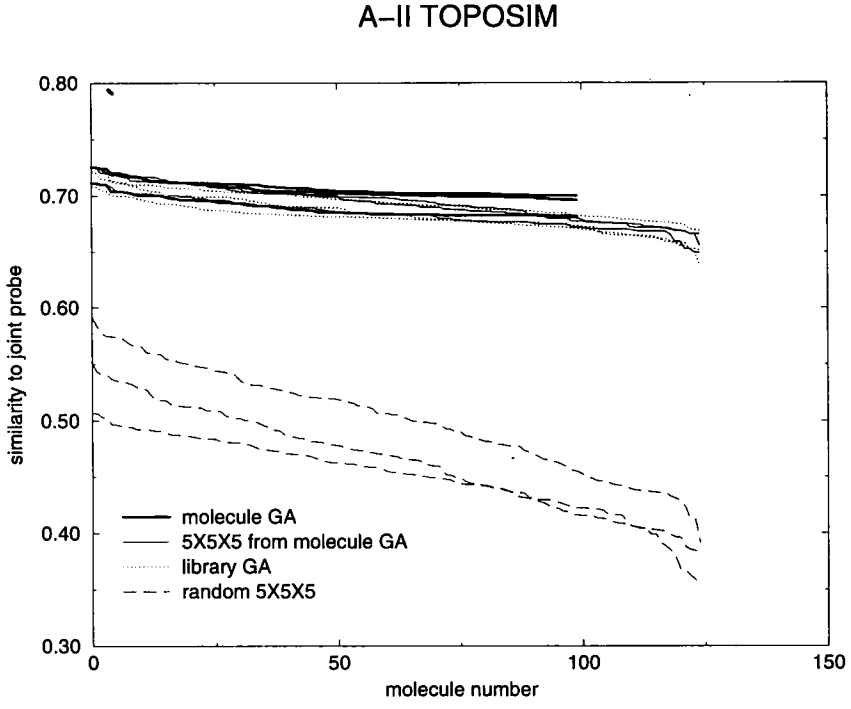


Figure 7. Scores of molecules in decreasing order plotted as a function of rank for the A-II example. Shown are the scores of molecules from the molecule-based GA run, molecules in a 5 × 5 × 5 library built from the fragments in Figure 5, molecules in the highest-scoring 5 × 5 × 5 library in the library-based GA run, and molecules in the 5 × 5 × 5 library constructed from fragments randomly selected from the fragment universe.

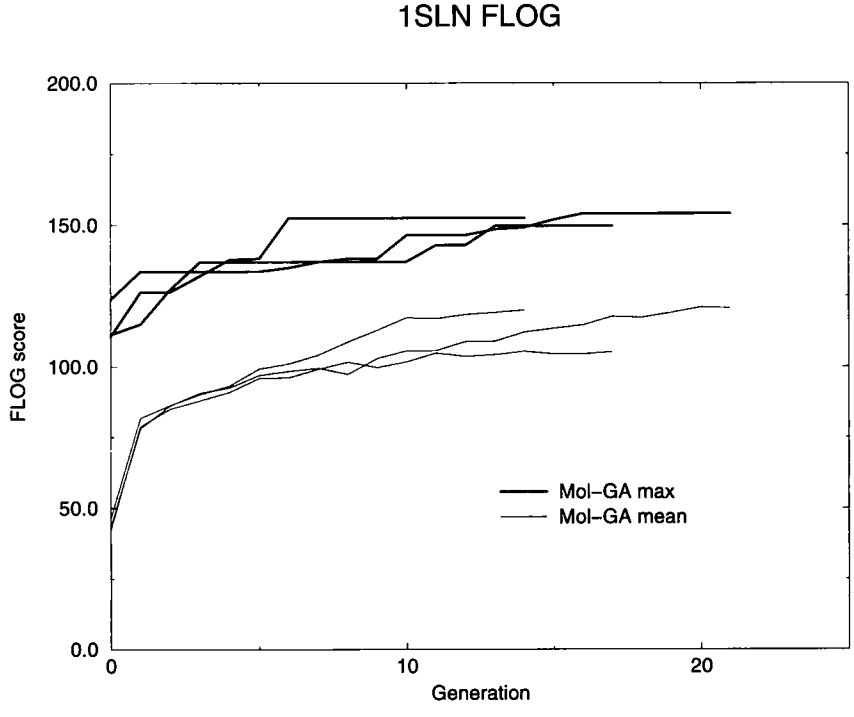


Figure 8. Graph of the mean and maximum score as a function of generation for a population of 300 against the 1SLN active site using FLOG scoring. Three runs are shown. The number of individual molecules scored in the runs are 2937, 4093, and 3442.

can be provided by A, B, or C even within the same run. That is, several docking modes are found for an A-B-C trimer.

Figure 10 shows the five most frequent fragments from the three runs. One can still see the resemblance of the fragments from one run to another, but the resemblance is not nearly as close as that in the A-II example.

In Figure 11, the scores of the top-scoring 100 molecules from the molecule-based GA are shown as bold solid lines. All have high scores. In this example, the curves for the libraries constructed from the most frequent fragments in those same molecules start with comparable scores, but tend to fall quickly. In one run, the highest-scoring molecule in the library has a score lower than the highest-scoring molecule from the GA, indicating that the highest scoring molecule in that run is not built from the most frequent fragments. As expected, three 5 × 5 × 5 libraries created by fragments randomly selected from the universe have the lowest scores.

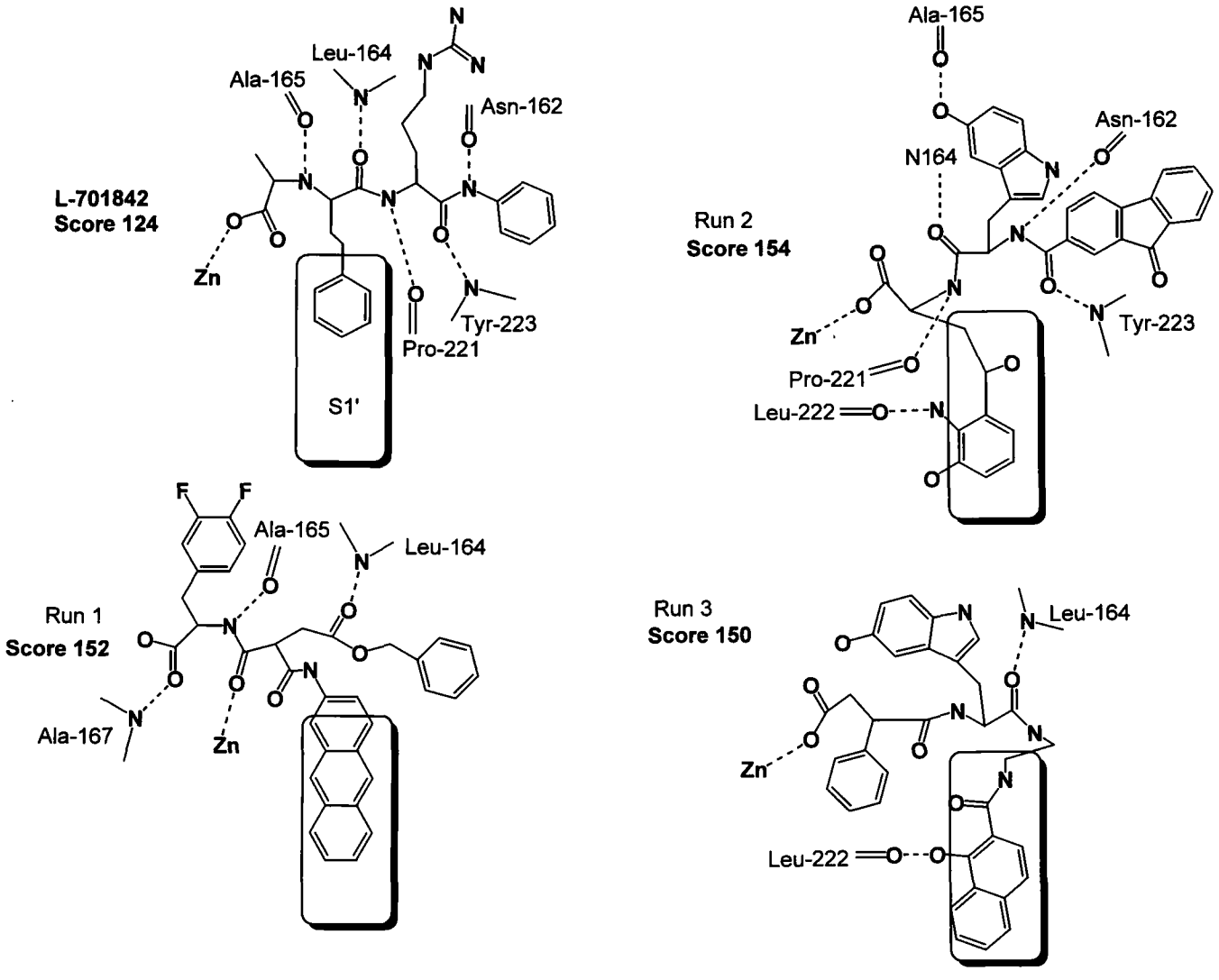


Figure 9. Schematic diagram of highest-scoring molecules from each 1SLN run and their interactions with the receptor. The score of L-701842 is the score of this molecule in its x-ray observed binding mode on the FLOG grid.

DISCUSSION

In this article, we have tried to extended our previous work in three directions. First, we extended our molecule-based GA to use 3D scoring functions developed at Merck. Second, we investigated whether constructing libraries from fragments taken from the highest-scoring molecules result in high-scoring libraries. Third, we compared the results of a molecule-based GA with a library-based GA.

Many methods of library design use 3D methods of scoring,[6-14] including ones that involve GAs.[17] As far as we know, ours is the first published demonstration of a GA using a 3D scoring against a receptor site that does not require the candidate molecules to be oriented by a common scaffold or a starting seed. Thus, we believe our methodology, despite its limitations (see later), is one way to fill a niche in 3D virtual screening that has not been fully addressed to date.

The 3D scoring methods have a number of important advantages over 2D methods. One can fit topologically discontinuous targets. Stereochemistry is taken into account. The final molecules tend to be more topologically diverse. In the case of docking to a known receptor, one does not need to rely on having a known active molecule to start with. There are equally important drawbacks. One must be able to specify the 3D structure of the target, either as a molecule in its receptor-bound form or an atomic model of a receptor site. Finding the correct conformation of candidate molecules introduces a serious complication. In our implementation, we let GA handle the search for the molecule topology and leave conformational searching as part of the scoring function; this is consistent with our "flexibase" approach wherein we generate and store explicit conformations that are docked as rigid bodies to the target. The flexibase approach has two important limitations. First, not all the conformational space is covered by 50 conformations, so some molecules are falsely rejected as low scoring if a potentially good conformation is missed. Second, generating conformations on the fly in the GA is expensive in terms of CPU time. It would be desirable in future work to avoid the limitations of precomputed conformations, either by having a docking function that flexed the molecule onto the target or by having the GA modify the conformation as well as

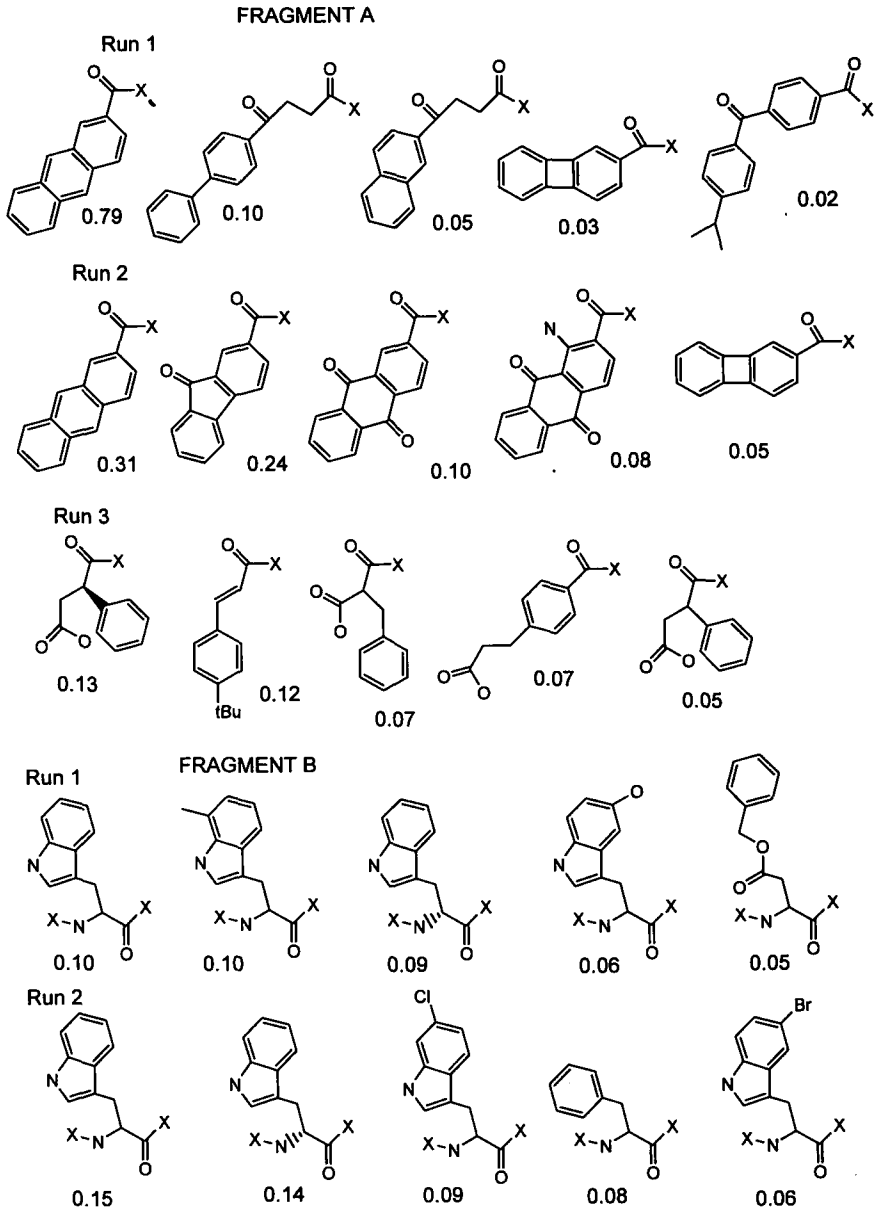
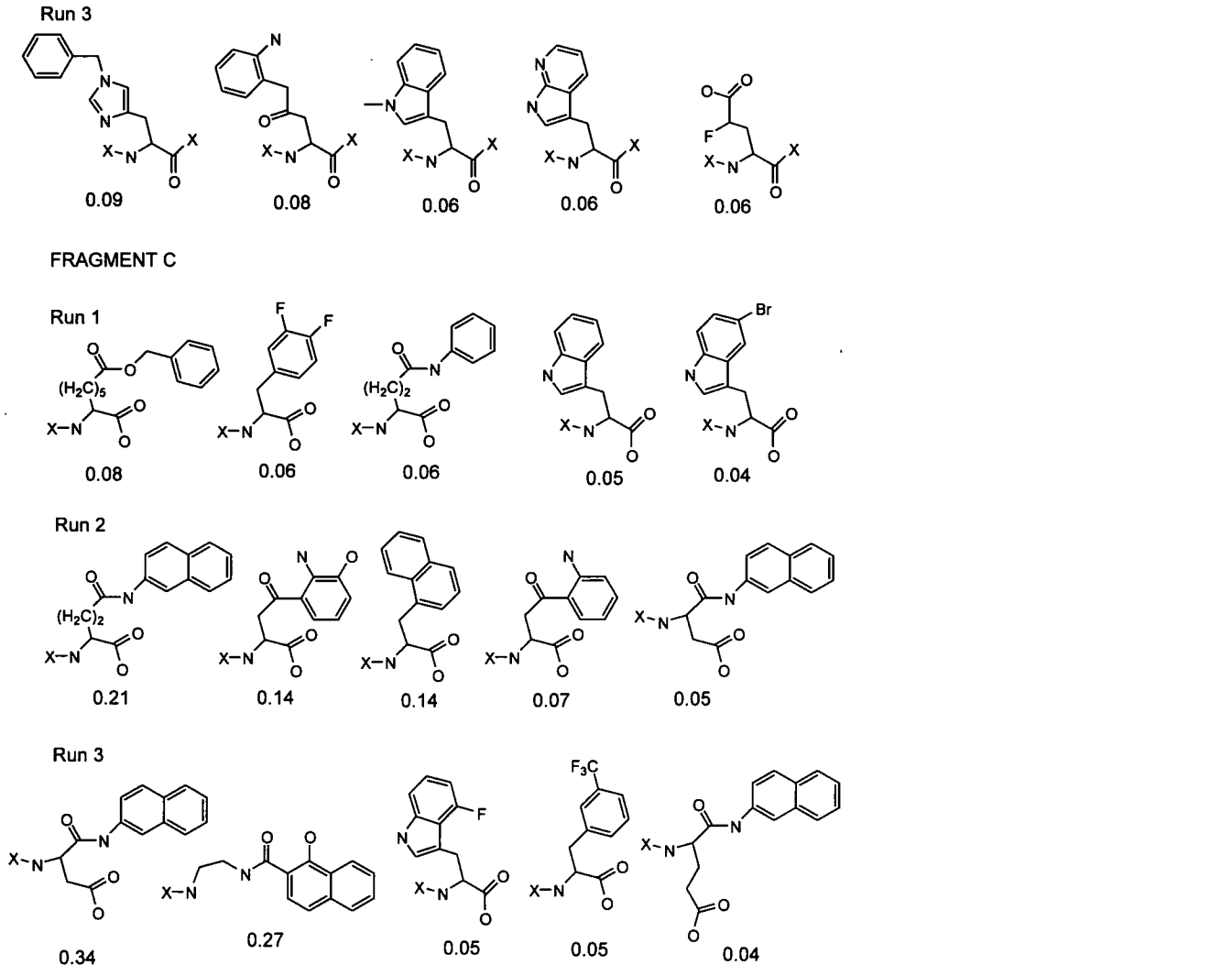


Figure 10. The most frequently occurring fragments in each of the 1SLN runs.

the connection table. There are many methods for flexibly docking single molecules into receptor sites or onto other molecules,[30–35] but currently most would be too slow for our GA application, in which thousands of molecules would have to be scored.

We have tried to address the question of whether the most frequent fragments taken from high-scoring molecules lead to libraries that will also be high scoring. After having tested several examples, 2D and 3D, we find this is a reasonable approximation in the sense that it always produces molecules that score much better than randomly constructed libraries, although not necessarily as good as the original molecules from the GA. In general, it appears to be a better approximation for our 2D scoring methods than our 3D methods. We speculate that this is due to the 3D scoring methods producing a fitness surface that is very "rough," that is, making a small change in structure often leads to a great change in score. This is especially noticeable when docking to a receptor, where a change of a few tenths of an Angstrom can cause a bad steric clash. Another consequence of this roughness is that 3D GA results vary more from one run to another, and are more susceptible to being stuck in nonoptimal solutions. Thus, it might be argued that at least some 3D scoring methods are more suitable for designing single compounds than combinatorial libraries.

Finally, we address the question of molecule- *vs* library-based GAs. The philosophy of our library-based approach is to find high-scoring molecules and simultaneously constrain the number of unique fragments in those molecules. This is analogous to the philosophy used in generating diverse libraries, wherein diversity at the level of molecules is maximized, but the number of unique fragments is constrained.[36] The library-based GA can find the same or similar trimers as the molecule-based GA. However, in our hands it takes three times as many generations to find these molecules when the runs are started from sets of random fragments. Because we have not tried to optimize the GA procedure for library-based GA, we cannot



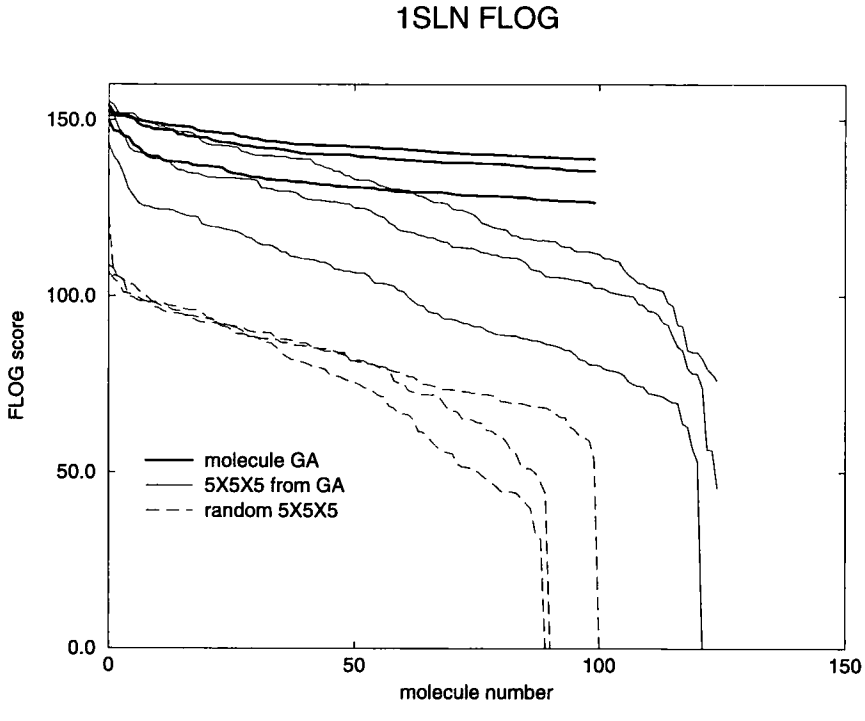


Figure 11. Scores of molecules in decreasing order plotted as a function of rank for the 1SLN example. Shown are the scores of molecules from the GA run, molecules in a 5 × 5 × 5 library built from the fragments in Figure 10, and molecules in the 5 × 5 × 5 library constructed from fragments randomly selected from the fragment universe.

claim that this will always be the case. However, in retrospect there are a number of reasons a library-based GA would take so long. One reason could be that, in our implementation, the "step size" is small; changing a single fragment to a related fragment makes only a very small change to the overall library. However, the most important reason is probably the size of the combinatorial spaces that must be searched. For molecule-based GAs, the number of possible molecules for A-B-C trimers is $N_A \times N_B \times N_C$, where N_A is the number of fragments of A. Given our fragment universe, this is $\sim 1 \times 10^{10}$. The number of possible $5 \times 5 \times 5$ libraries, on the other hand, is:

$$\frac{N_A!}{n_A!\,(N_A - n_A)!} \times \frac{N_B!}{n_B!\,(N_B - n_B)!} \times \frac{N_C!}{n_C!\,(N_C - n_C)!}$$

where $n_A = n_B = n_C = 5$. This is $\sim 5 \times 10^{44}$.

Aside from the increased number of generations, the difference in computational cost per generation is significant. For a library-based GA, the number of score calculations that must be done per iteration is $n_A \times n_B \times n_C$ as large as for a molecule-based GA. For us, this makes library-based GAs with 3D scoring impractically time consuming. Finally, results from the molecule-based GA can be used to design sets of single molecules irrespective of whether the molecules share fragments, whereas the library-based GA forces the molecules to share fragments, as in a combinatorial library. We would argue then, that for designing targeted libraries by GA it would be preferable to use a molecule-based rather than a library-based approach. Of course, predicted activity is not the only consideration in constructing libraries. One often wants to include such factors as molecular weight, cost, "drug-likeness," and distance from some other library. Fortunately, these are definable for single molecules and can easily be added to a supplementary scoring function in a molecule-based GA. Other factors, such as diversity within a library, lack of overlap of molecular weights, and distribution of logP, are not as easily defined for single molecules. For a targeted library, the issue of diversity is different than it would be in a library meant to be tested on a variety of assays. We need not, and probably should not, insist on maximal diversity (as in Gillet et al.[36] for example) because that would work against our selection of compounds to fit a given SAR. On the other hand, we want to insist on a small amount of diversity lest our GA home in on too small a region of chemical space. Methods of imposing a small amount of diversity at the molecule level have been suggested. For, example, Zheng et al. [16] suggest penalizing molecules that are too similar to other molecules in the population. This is an active topic of investigation in our laboratory.

ACKNOWLEDGMENTS

Many parts of the genetic algorithm are assembled from tools in the MIX modeling suite. The authors thank the other members of the MIX team for their tireless work. Dr. Laurie Castonguay, Ralph Mosley, and Dr. Adel Naylor-Olsen provided many useful suggestions that led to improvements in the analysis of the genetic algorithm results.

REFERENCES

1 Clark, D.E., and Westhead, D.R. Evolutionary algorithms in computer-aided molecular design. *J. Comput.-Aided Mol. Design* 1996, **10,** 337–358

2 Maddalena, D.J., and Snowdon, G.M. Applications of genetic algorithms to drug design. *Exp. Opin. Ther. Patents* 1997, **7,** 247–254

3 Bures, M.G., and Martin, Y.C. Computational methods in molecular diversity and combinatorial chemistry. *Curr. Opin. Chem. Biol.* 1998, **2,** 376–380

4 Weber, L. Applications of genetic algorithms in molecular diversity. *Curr. Opin. Chem. Biol.* 1998, **2,** 381–385

5 Walters, W.P., Stahl, M.T., and Murcko, M.A. Virtual screening—An overview. *Drug Discovery Today* 1998, **3,** 160–178

6 Kick, E.K., Roe, D.C., Skillman, A.G., Liu, G., Ewing, T.J.A., Sun, Y., Kuntz, I.D., and Ellman, J.A. Structure-based design and combinatorial chemistry yield low nanomolar inhibitors of cathepsin D. *Chem. & Biol.* 1997, **4,** 297–307

7 Murray, C.W., Clark, D.E., Auton, T.R., Firth, M.A., Li, J., Sykes, R.A., Waszkowycz, B., Westhead, D.R., and Young, S.C. PRO−SELECT: Combining structure-based drug design and combinatorial chemistry for rapid lead discovery 1. Technology. *J. Comput.-Aided Mol. Design* 1997, **11,** 193–207

8 Mason, J.S., Morize, I., Menard, P.R., Cheney, D.L., Hulme, C., and Labaudiniere, R.F. New 4-point pharmacophore method for molecular similarity and diversity applications: Overview of the method and applications, including a novel approach to the design of combinatorial libraries containing privileged substructures. *J. Med. Chem.* 1999, **41,** 3251–3264

9 Li, J., Murray, C.W., Waszkowycz, B., and Young, S.C. Targeted molecular diversity in drug discovery: Integration of structure-based design and combinatorial chemistry. *Drug Discovery Today* 1998, **3,** 105–112

10 Illig, C., Eisennagel, S., Bone, R., Radzicka, A., Murphy, L., Randle, T., Spurlino, J., Jaeger, E., Salemme, F.R., and Soll, R.M. Expanding the envelope of structure-based drug design using chemical libraries: Application to small-molecule inhibitors of thrombin. *Med. Chem. Res.* 1998, **8,** 244–260

11 Makino, S., Ewing, T.J.A., and Kuntz, I.D. "DREAM++: Flexible docking program for virtual combinatorial libraries." *J. Comput.-Aided Mol. Design* 1999, **13,** 513–532

12 Klebe, G., and Abraham, U. Comparative molecular similarity index analysis (CoMSIA) to study hydrogen-bonding properties and to score combinatorial libraries. *J. Comput.-Aided Mol. Design* 1999, **13,** 1–10

13 Murray, C.M., and Cato, S.J. Design of libraries to explore receptor sites. *J. Chem. Inf. Comput. Sci.* 1999, **39,** 46–50

14 Bohm, H.-J., Banner, D.W., and Weber, L. Combinatorial docking and combinatorial chemistry: Design of potent non-peptide thrombin inhibitors. *J. Comput.-Aided Mol. Design* 1999, **13,** 51–56

15 Sheridan, R.P., and Kearsley, S.K. Using a genetic algorithm to suggest combinatorial libraries. *J. Chem. Inf. Comput. Sci.* 1995, **35,** 310–320

16 Zheng, W., Cho, S.J., and Tropsha, A. Rational design of a targeted combinatorial chemical library with opiatelike activity. *Int. J. Quantum Chem.* 1998, **69,** 65–75

17 Liu, D.X., Jiang, H.L., Chen, K.X., and Ji, R.V. A new approach to design virtual combinatorial library with genetic algorithm based on 3D grid property. *J. Chem. Inf. Comput. Sci.* 1998, **38,** 233–242

18 Brown, R.D., and Martin, Y.C. Designing combinatorial library mixtures using a genetic algorithm. *J. Med. Chem.* 1997, **40,** 2304–2313
19 Kearsley, S.K., Salamack, S., Fluder, E.M., Andose, J.D., Mosley, R.T., and Sheridan, R.P. Chemical similarity using physiochemical property descriptors. *J. Chem. Inf. Comput. Sci.* 1996, **36,** 118–127
20 Kearsley, S.K., Underwood, D.J., Sheridan, R.P., and Miller, M.D. Flexibases: A way to enhance the use of molecular docking methods. *J. Comput.-Aided Mol. Design* 1994, **8,** 565–582
21 Gasteiger, J., Rudolf, C., and Sadowski, J. Automatic generation of 3D-atomic coordinates for organic molecules. *Tetrahedron Comp. Method.* 1990, **3,** 537–547
22 Feuston, B., Miller, M.D., Culberson, J.C., Nachbar, R.N., and Kearsley, S.K. Comparison of knowledge-based and distance geometry approaches: For the generation of molecular conformations. *J. Chem. Inf. Comput. Sci.* (submitted).
23 Miller, M.D., Sheridan, R.P., and Kearsley, S.K. SQ: A program for rapidly producing pharmacologically relevant molecular superpositions. *J. Med. Chem.* 1999, **42,** 1505–1514
24 Miller, M.D., Kearsley, S.K., Underwood, D.J., and Sheridan, R.P. FLOG: A system to select quasi-flexible ligands complementary to a receptor of known three-dimensional structure. *J. Comput.-Aided Mol. Design* 1994, **8,** 153–174
25 Available Chemicals Directory is distributed by Molecular Design, Ltd., San Leandro, California, 1997
26 MDL Drug Data Report is distributed by Molecular Design Ltd., San Leandro, California, 1998
27 Becker, J.W., Marcy, A.I., Rokosz, L.L., Axel, M.G., Burbaum, J.J., Fitzgerald, P.M.D., Cameron, P.M., Esser, C.K., Hagmann, W.K., Hermes, J.D., and Springer, J.P. Stromelysin-1: Three-dimensional structure of the inhibited catalytic domain of the C-truncated proenzyme. *Prot. Sci.* 1995, **4,** 1966–1976
28 Bernstein, F.C., Koetzle, T.F., Williams, G.J.B., Meyer, E.F. Jr., Brice, M.D., Rodgers, J.R., Kennard, O., Shimanouchi,T., and Tasumi, M. The protein data bank: A computer-based archive file for macromolecular structures. *J. Mol. Biol.* 1977, **112,** 535–542
29 De Laszlo, S.E., Glinka, T.W., Greenlee, W.J., Ball, R., Nachbar, R.B., and Prendergast, K. The design, binding affinity prediction and synthesis of macrocyclic angiotensin II AT1 and AT2 receptor antagonists. *Bioorg. Med. Chem. Lett.* 1996, **8,** 923–928
30 Oshiro, C.M., Kuntz, I.D., and Dixon, J.S. Flexible ligand docking using a genetic algorithm. *J. Comput.-Aided Mol. Design* 1995, **9,** 113–130
31 Lemmen, C., Lengauer, T., and Klebe, G. FLEXS: A method for fast flexible ligand superposition. *J. Med. Chem.* 1998, **41,** 4502–4520
32 Handshuch. S., Wagener, M., and Gasteiger, J. Superposition of three-dimensional chemical structures allowing for conformational flexibility by a hybrid method. *J. Chem. Inf. Comput. Sci.* 1998, **38,** 220–232
33 Welch, W., Ruppert, J., and Jain, A.N. Hammerhead: Fast fully automated docking of flexible ligands to protein binding sites. *Chem. & Biol.* 1996, **3,** 449–462
34 Given, J.A., and Gilson, M.K. A hierarchical method for generating low-energy conformers of a protein-ligand complex. *Proteins* 1998, **33,** 475–495
35 Baxter, C.A., Murray, C.W., Clark, D.E., Westhead, D.R., and Eldridge, M.D. Flexible docking using tabu search and an empirical estimate of binding affinity. *Proteins* 1998, **33,** 367–382
36 Gillet, V.J., Willett, P., and Bradshaw, J. The effectiveness of reactant pools for generating structurally-diverse combinatorial libraries. *J. Chem. Inf. Comput. Sci.* 1997, **37,** 731–740