Similarity Searching in Files of Three-Dimensional Chemical Structures. Alignment of Molecular Electrostatic Potential Fields with a Genetic Algorithm

David J. Wild and Peter Willett*

Krebs Institute for Biomolecular Research and Department of Information Studies, University of Sheffield, Western Bank, Sheffield S10 2TN, United Kingdom

Received May 22, 1995⊗

This paper discusses techniques for similarity searching in databases of three-dimensional chemical structures represented by their molecular electrostatic potential fields. Field-based similarity searching involves the alignment of molecular fields so as to maximize their overlap, and we have used a genetic algorithm for this purpose, in which a chromosome encodes the rotations and translations that are needed to generate such an alignment and in which the fitness function is the value of the similarity measure that results from that alignment. The comparison of a pair of typical small-molecules is sufficiently rapid to enable the algorithm to be used with files of nontrivial size. Experiments with a Kendall Square Research KSR-1 multiprocessor system demonstrate the suitability of the algorithm for use in a parallel environment.

1. INTRODUCTION

Similarity searching involves matching some *target* molecule of interest, such as an initial lead in a drug- or pesticide-discovery program, against all of the molecules in a database to find the *nearest neighbors*, *i.e.*, those molecules that are most similar to the target.[1] Fragment-based approaches to similarity searching in databases of two-dimensional (2-D) chemical structures have become a standard retrieval facility in chemical information systems[2−4] as a complement to the long-established facilities for substructure searching. With the recent development of systems for three-dimensional (3-D) substructure searching, there is interest in the development of analogous procedures for 3-D similarity searching, and several approaches have already been described that are sufficiently fast in operation to allow them to be used with databases of nontrivial size. Many of these approaches have used distance or angular information to define the degree of resemblance between two molecules,[5−9] and there has also been some interest in the use of physical-property information for similarity searching.[10,11] However, few of these approaches take explicit account of the electrostatic, steric, and hydrophobic fields that form the basis of modern approaches to the prediction of biological activity.[12,13]

The use of molecular fields in 3-D similarity searching is the subject of an ongoing project in our laboratory,[14,15] and in this paper we consider similarity searching when a molecule is characterized by its molecular electrostatic potential, or MEP. The calculation of molecular similarity using MEPs has been studied previously by several groups.[16−19] Such methods are effective in operation, and they can also be surprisingly efficient:[20, 21] however, they require a search to be carried out for the best possible alignments of the two molecules that are being compared so as to maximize the calculated similarity measure. An example of such an alignment procedure is reported by van Geerestein *et al.* in their work on the SPERM system.[7,10] The shape of a molecule in SPERM is described by mapping a specified property, such as the MEP, onto the vertices of a tessellated icosahedron, with the precision of the description being determined by the extent of the tessellation. A database structure is aligned with the target structure by rigid-body rotations and translations of the icosahedron representing the former molecule, and the dissimilarity of each such alignment is calculated from the root-mean-squared-difference of the matched vertices in the two icosahedra. This paper reports the use of an alternative approach, based on a genetic algorithm (hereafter a GA), for generating the alignments required for MEP-based similarity searching.

2. A GENETIC ALGORITHM FOR THE ALIGNMENT OF MOLECULAR ELECTROSTATIC POTENTIALS

2.1. Genetic Algorithms. Genetic algorithms are computational problem-solving methods that mimic some of the principal characteristics of biological evolution and genetic reproduction.[22−25] A GA creates a randomly-chosen set, known as a *population*, of individuals, each of which contains a representation of a possible problem solution. This solution is encoded into a linear string that is normally referred to as a *chromosome*. The effectiveness of the solution encoded by each of the chromosomes in a population is measured by the *fitness function*, and the GA manipulates the chromosomes so as to maximize the value of the fitness function. This it does by the creation of subsequent populations that include features from the fitter strings in the previous population, in an iterative procedure that can be thought of as an algorithmic representation of biological reproduction. Parents are selected from the population, and information is taken from their chromosomes to produce one or more child individuals that are inserted into the population. Chromosomes are manipulated by *mutation* (where the chromosomal material may be altered slightly in a random fashion) and *crossover* (where new child chromosomes are created by taking some chromosomal material from one parent, and some from the other) operators. A GA may be considered to have succeeded when *convergence* occurs, *i.e.*, when the members of the population all lie in the same region of search space. However, convergence on suboptimal solutions can

* To whom all correspondence should be addressed. Email: P.WILLETT@SHEFFIELD.AC.UK.

⊗ Abstract published in *Advance ACS Abstracts,* March 15, 1996.

be a problem, particularly with a search space that is difficult to traverse.

There have been several reports of the application of GAs to problems in chemical structure handling.[26−32] It is our belief that they provide both an effective and an efficient mechanism for the investigation of a range of complex chemical matching problems, such as the generation of near-maximal common subgraphs from pairs of large 2-D structures,[30] flexible 3-D substructure searching,[31] and docking flexible ligands into protein active sites.[32] In the following we present a GA for identifying the maximal alignment of a pair of MEPs.

2.2. Calculation of Intermolecular Similarity Using Molecular Electrostatic Potentials. Carbo et al. were the first to suggest that the similarity between a pair of molecules might usefully be calculated by a similarity measure based on their electron densities.[33] The so-called Carbo index, which is actually a form of the long-established cosine coefficient,[34] is defined to be

$$R_{\mathrm{AB}} = \frac{\int P_{\mathrm{A}} P_{\mathrm{B}} dv}{(\int P_{\mathrm{A}}^2 dv)^{1/2} (\int P_{\mathrm{V}}^2 dv)^{1/2}}$$

where P_{A} and P_{B} are the properties (such as the electron densities) of the two molecules that are being compared. Most workers, however, have preferred to use the MEP in preference to electron density in the calculation of the Carbo index.[16−20]

The electrostatic potential P_r at a point r for a molecule of n atoms is calculated from the point charges q_i on each atom i in the molecule, so that

$$P_r = \sum_{i=1}^{n} \frac{q_i}{|r - R_i|}$$

where R_i denotes the position of the ith atom. A molecule is positioned at the center of a 3-D grid, and the potential is calculated at each point in the grid. The similarity between a pair of molecules is then estimated by comparing the corresponding potentials at each grid point and summing over the entire grid, with a suitable normalizing factor to bring the resulting similarities into the range -1.0 to $+1.0$. This numerical approach necessarily involves the matching of very large numbers of grid points, unless very coarse grids are to be used (in which case the calculated similarities are unlikely to reflect accurately the true degree of structural resemblance between the molecules that are being compared). Good et al.[20,21] have developed an alternative approach in which the potential distribution is approximated by a series of Gaussian functions that can be processed analytically, with a substantial increase in the speed of the similarity calculation and with only a minimal effect on its accuracy. The approach hence provides an elegant means of efficiently calculating the similarity between two molecules: however, it is still necessary to identify that alignment which will produce the highest similarity. The GA that is described in the next section attempts to optimize the molecular alignments.

2.3. The Algorithm. The GA seeks to identify a combination of translations and rotations that will align one MEP with another, fixed MEP so as to give the highest possible similarity.

Each chromosome contains five components, two of which encode rotations and the remainder translations. The two rotational components encode rotations in the XY and YZ planes by means of an eight-bit binary number: this allows 256 possible rotations in each plane, *i.e.*, an accuracy of about 1.4°. The translations along the X, Y, and Z axes are also encoded by binary numbers, with the maximum permitted translation along any axis being determined by the size, M, of the larger of the two molecules that are being compared. The smallest unit of translation (S), the stepsize in Å, is a user-definable parameter (although we found a stepsize of one adequate in our tests) and the number of bits used for each of the X, Y, and Z translations was $\log_2 (M/S)$. The chromosomes are initially set to random values and then decoded by applying the indicated rotations and translations to the 3-D coordinates of the atoms in one of the two molecules that are being aligned. The resulting coordinates are passed to the fitness function for the evaluation of the alignment defined by that particular set of rotations and translations: this function is a two-term Gaussian similarity calculation.[20,21]

The simplest, and most widely-used, types of crossover are one-point and two-point crossover. In the former, a crossover point, k, is selected; all of the chromosome elements in the child up to and including point k will come from one parent, and all of the information after the chosen point will come from the other parent. The latter is a simple modification that involves two crossover points, so that the chromosomal information in a child chromosome between the crossover points will come from one parent, with the rest coming from the other. However, we found that slightly better results were obtained using uniform crossover, in which the bit at a particular location in a child chromosome is defined by randomly selecting one of its parents and then taking the bit at the specified location in the selected parent chromosome. A crossover rate of 20% was found to give the best results for this problem, and little noticeable improvement was seen when a dynamic crossover rate was used (in which this percentage was allowed to vary during a run). A simple bit-flip mutation was used in which there was a ($1/l$) probability of each of the bits being flipped, where l was the length of the chromosome in bits. This mutation rate was suggested by Mühlenbein[35] and found to give results here that were at least as good as those obtained with different mutation rates. The chromosomes are hence processed as follows. A random number generator is used to determine whether crossover is to be carried out, and, if it is, then the generator is used to select the two chromosomes that are to be input to the crossover routine. The resulting child chromosome, or a randomly-selected chromosome if crossover is not used in that iteration, is then input to the mutation routine.

The methods used for selecting and mating parents have a significant bearing on the effectiveness of a GA. Two overall strategies for reproduction were considered: *generational replacement* and *steady-state without duplicates*.[22,23] In the former method, successive populations are generated, with each and every member of the new population being derived from parents from the old generation and with the previous generation being completely discarded. In the latter method, a small number of individuals is added to the population on each iteration, and these replace the least-fit members of the old population. A newly-generated chromosome is rejected if it is identical to an existing member of the population. For this application, steady state without duplicates was found to be by far the better method. Since only that single chromosome with the lowest fitness was replaced each time, the number of generations was equal to the total number of crossover/mutation operations that were executed.

The population size, *i.e.*, the number of chromosomes in each generation, is another of the many factors that can affect the operation of a GA. Whilst it is far faster to calculate Gaussian similarities than grid-based similarities, the fitness function is still computationally intensive, and it is hence desirable to keep both the population size and the number of generations (and hence the number of fitness calculations that need to be performed) as low as possible, subject to the need to ensure that the GA can still sample effectively the entire solution space. Tests were carried out with populations containing between 5 and 200 chromosomes and with between 250 and 1000 generations. These parameter settings affect both the effectivess of the GA, *i.e.*, the quality of the alignments that are produced, and its efficiency, *i.e.*, the associated computational requirements. Our experiments suggested that the most cost-effective solution for database searching was achieved with a GA that ran for 250 generations with a population containing just 10 chromosomes. These parameter values are specific to similarity searching, and other values might be more appropriate for other applications; for example, one would probably wish to use a much greater number of generations to create alignments for input to a 3D QSAR analysis.

The GA used conventional roulette-wheel selection. A limitation of this parent-selection method, and of others in which the probability of selection is proportional to fitness, is that it is possible for a few high-fitness individuals quickly to dominate a population, with the possibility of premature convergence to a nonglobal maximum. This problem is addressed by altering the *selection pressure*: lowering the selection pressure means that very fit individuals are less likely to be chosen as parents, whilst increasing the pressure will make it more likely for fit individuals to reproduce. The selection pressure is controlled by means of *linear normalization*. Individuals are ranked in decreasing order of their fitness evaluations, and this ordering is then taken to be the fitness, instead of the value resulting from application of the fitness function to that chromosome.

Two further modifications were considered: the use of *Gray coding*[22] and of *initial population control*. Gray coding is derived from binary coding, but has the property that the representations of adjacent integers differ by only one bit, so that a single bit-flip during mutation of a chromosome will not result in large-scale changes in the calculated fitness. This characteristic has meant that many binary GAs have involved the use of Gray-coded, rather than binary-coded, chromosomes; however, we found that our GA was noticeably less effective when Gray coding was used. The initialization of a population should try to ensure coverage of as much as possible of the potential search space so as to minimize the number of operations that are required for convergence to take place; this is of particular importance when, as is the case here, small populations are being used.[36] Diversity in the initial population was achieved by ensuring that all of the members of the initial population had a large Hamming distance between them, where the Hamming distance between two binary strings is the number of corresponding bits that differ between the two strings. This technique was found to lessen the chances of convergence on severely suboptimal solutions.

The final GA that was used for the experiments reported in the remainder of this paper had a population size of ten, steady-state-without-duplicates reproduction, binary encoding, a static uniform crossover rate of 20%, linear normalization, and a diversely-initialized population. Further details of the very extensive comparative experiments that were carried out prior to arriving at these final parameter values are provided by Wild.[37] The GA was implemented in C and was run on a Silicon Graphics Indigo-2 with a 100 MHz R4000 chip. Since calculation of the fitness function involved repeated exponential calculations, lookup tables were created, containing precalculated exponential values, to minimize the run-times.

2.4. Alternative Approaches. We have compared the performance of the GA for database searching with two alternative approaches to the generation of alignments: these involved the use of *field-graphs*[14] and *bit-climbers*.[38]

A field-graph encodes the most important features of an MEP grid in labeled graphs that typically contain a few tens of nodes at most, these summarizing the potential values at the millions of points that comprise a detailed grid. A field-graph for a molecule is constructed by identifying and isolating MEP grid points which satisfy a certain constraint, *e.g.*, they have a positive (or negative) potential that is greater (or less) than a user-defined threshold value, and then grouping points that satisfy these constraints and that are close to each other in 3-D space. The resulting groups comprise the nodes of the field-graph, with the number of constituent points and the type (positive or negative) being used to label the nodes. The edges of the graph are the geometric distances between the centers of pairs of these points. The MEP similarity between two molecules is obtained by aligning the corresponding field-graphs using a maximal common subgraph isomorphism algorithm and then calculating the Gaussian similarity corresponding to that alignment. The isomorphism algorithm normally identifies several possible alignments, and the overall similarity between the two molecules is the largest of the calculated Gaussian similarities.

The field-graph approach permits a huge reduction in the volume of data that needs to be processed to generate an alignment. It has been shown to identify effective alignments and is now being used for similarity searching on a large corporate database of 3-D structures.[14] The method does, however, have an inherent failure rate of around 6% since the threshold criteria that are used to create a field-graph can result in the identification of less nodes than are necessary for the generation of a unique alignment.

Davis[38] describes a bit-climber, which is a simple hill-climbing algorithm that can be regarded as a simplified GA that involves only mutation. The bit-climber starts with a single, initial chromosome. Then, each bit of the chromosome is flipped in some predefined sequence, and the fitness is evaluated after each such bit-flip. If the modified chromosome is fitter than the original, the original is replaced by the modified chromosome; otherwise the bit is returned to its original state. When each bit has been tested, the process is repeated, possibly with the bits being flipped in a different order. The algorithm terminates when none of the

Table 1. Binding Affinities of 21 Steroids to Human Testosterone-Binding Globulins (TBG)[a]

steroid	TBG affinity	similarity
dihydrotestosterone	9.74	0.78
androstenediol	9.18	0.78
estriol	6.63	0.73
estradiol	8.83	0.72
estrone	8.18	0.65
progesterone	6.94	0.63
androstendione	7.46	0.61
androstanediol	9.11	0.60
pregnenolone	7.15	0.56
hydroxypregnenlone	6.36	0.55
dehydepiandrstrone	7.82	0.55
androsterone	7.15	0.52
deoxycortisol	7.20	0.52
aldosterone	5.32	0.51
hydroxyprogesterone	7.00	0.49
deoxycorticosterone	7.38	0.47
cortisone	6.43	0.44
etiocholanolone	6.15	0.44
cortisol	6.20	0.44
corticosterone	6.34	0.42

[a] The third column shows the best similarity over 20 runs of the GA when it was made to report the similarity between testosterone and each of the steroids. The table is ranked by decreasing similarity.

bit-flips produces a fitter chromosome than the best that has been observed thus far. The bit-climber was implemented so that it either carried out a single run or carried out five runs and then returned the result corresponding to the fittest final chromosome from amongst the five separate runs.

3. RETRIEVAL PERFORMANCE

3.1. Initial Testing. An important characteristic of research into similarity-based retrieval is the need for some quantitative means of evaluating the effectiveness of the similarity measures that are being tested. Several earlier similarity studies, both in our laboratories and elsewhere, have made extensive use of the *similar property principle*.[2,9,11] Here, simulated property-prediction experiments are carried out using datasets for which both structural and biological property (or activity) data are available, so as to ascertain which methods (*e.g.*, which similarity coefficients) result in measures of structural similarity that are most closely correlated with measures of property similarity. Previous work[16−20] has already shown that there is a strong correlation between biological activity and the similarities that result from grid-based MEP calculations, and our initial tests with the GA hence sought to ascertain whether the algorithm was also capable of generating appropriate alignments for a small dataset for which property data were available. The dataset used was a file of 21 steroids, and the binding affinities to their carrier proteins (human testosterone-binding globulins (TBG) and corticosteroid-binding globulins (CBG)).[39] The 3-D structures of the molecules were generated using CONCORD,[40] and the atomic charges were calculated using the MNDO semiempirical molecular orbital calculations in MOPAC.[41] The resulting point charges were then used for the MEP calculations.

Testosterone and corticosterone were used as target molecules against which each of the other molecules were matched using the GA. The resulting similarities were then used to rank the dataset in order of decreasing similarity with the target structure and the extent of the relationship between the calculated similarity and the TBG (or CBG) affinity noted. These relationships are shown in Tables 1 and 2, where it will be seen that there is a qualitative relationship between the calculated similarities and the binding affinities in both cases. The Spearman rank correlation coefficients[42] for the TBG and CBG pairs of rankings were 0.732 and 0.743, respectively ($p < 0.0001$), and we hence conclude that the GA has provided appropriate sets of alignments for these two target molecules.

Table 2. Binding Affinities of 21 Steroids to Human Corticosterone-Binding Globulins (CBG)[a]

steroid	CBG affinity	similarity
deoxycorticosterone	7.65	0.90
cortisol	7.88	0.82
progesterone	7.38	0.79
cortisone	6.89	0.74
deoxycortisol	7.88	0.74
aldosterone	6.28	0.74
pregnenolone	5.26	0.65
androstendione	5.76	0.60
hydroxyprogesterone	7.74	0.59
dehydepiandrstrone	–	0.56
estrone	–	0.55
dihydrotestosterone	5.92	0.53
estradiol	–	0.51
etiocholanolone	5.26	0.50
estriol	–	0.50
androstenediol	–	0.50
testosterone	6.72	0.45
androsterone	5.61	0.44
androstanediol	–	0.44
hydroxypregnenlone	–	0.42

[a] The third column shows the best similarity over 20 runs of the GA when it was made to report the similarity between corticosterone and each of the steroids. The table is ranked by decreasing similarity. The steroids marked with a dash exhibited no measurable activity.

It should be emphasized that any general approach to field-based similarity searching will also need to encompass other, nonelectrostatic types of ligand−receptor interaction, and we are currently developing GA-based approaches for searching steric and hydrophobic fields to complement the work described in this paper.

3.2. Database Searching. The main experiments used a test database of 1000 molecules taken from the Fine Chemicals Database (FCD), from which 100 molecules were chosen at random to act as the target molecules for which the nearest neighbors, *i.e.*, the electrostatically most similar molecules, were required. The structures were processed as for the steroid dataset, *i.e.*, using CONCORD and then the MNDO option in MOPAC.

The steroid searches have demonstrated that the GA is capable of generating appropriate alignments between pairs of MEPs. Accordingly, we can estimate the effectiveness of a search by the magnitude of the similarities resulting from the Gaussian similarity calculation, since a large similarity will be achieved if, and only if, an appropriate alignment has been identified by the GA. Let $S(I)$ be the Gaussian MEP similarity for the Ith most similar molecule to the mth target molecule; then the performance measure E_{mn} is defined by

$$E_{mn} = \frac{1}{n}\sum_{I=1}^{n} S(I) \qquad (1)$$

where n is a user-defined cut-off value (which we have

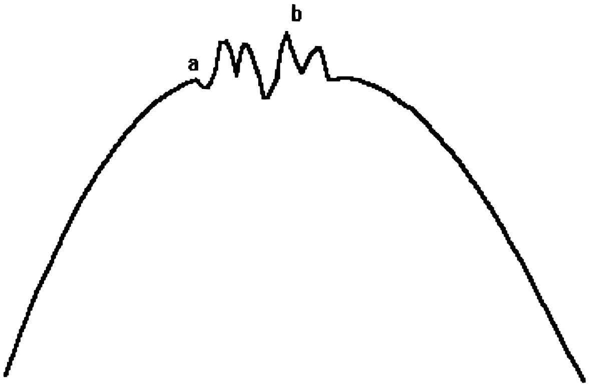

Figure 1. A hypothetical path through a fitness landscape. The global maximum is at (b), and the highest point reached by a simple bit-climber starting on the main left curve is at (a).

chosen to be 20 in these experiments), *i.e.*, E_{mn} is the mean MEP similarity for the n nearest neighbors of the mth target molecule. If there is a total of T target molecules, then the overall effectiveness of the set of searches for the n nearest neighbors of each target molecule is given by

$$E_n = \frac{1}{T}\sum_{m=1}^{T} E_{mn} \qquad (2)$$

The E_{20} values for the 100 searches on the FCD subset using the three search methods were as follows: GA 0.70; field-graphs 0.70; and bit-climber 0.69. The results thus indicate that the three methods give broadly comparable results in terms of the similarities of the top-ranked structures.

The bit-climber result is for the five-run version: that for the single-run version was 0.63. The performance of the bit-climber is surprisingly good, as the one-run version can only forge a single path upwards from the random point at which it starts and will stop when a local maximum is reached, while even the five-run version can only climb a maximum of five hills. This indicates that in many comparisons it may be fairly simple to reach a solution which is close to the global maximum but difficult to find the maximum itself. Consider the hypothetical two-dimensional landscape shown in Figure 1. A bit-climber starting at any point on the main left-hand curve will reliably reach point (a) as a local maximum, which is fairly close to the global maximum (b). The GA may be able to find the global maximum (b) but will not necessarily even reach point (a). In this case, therefore, the bit-climber may perform better than the GA, and even if the GA does reach the global maximum, the result will not be substantially better.

We have noted that all three alignment methods give similar values for E_{20}. The *Wilcoxon signed-rank test*[42] was used to determine the significance, or otherwise, of the differences between each pair of search methods. The Wilcoxon test is a nonparametric two-tailed test that is designed to determine whether two related samples (the E_{20} values for two sets of 100 searches in this case) are significantly different from each other. The test showed that the GA and bit-climber figures are significantly dissimilar at the 0.01 level; in 71 of the 100 searches, the GA gave a result which was higher than that reported by the bit-climber, and in only 29 was the bit-climber result better. Inspection of the results showed that in most cases there was little difference between the two sets of results: the mean rank of the cases where the bit-climber results were better than those of the GA was 45.7, as opposed to 52.4 for those where the GA was better. There were no significant differences at the 0.05 level between either the GA and field-graph results or the field-graph and bit-climber results.

Table 3. Fraction of the Alignments of Identical Structures That Reach or Exceed the Given Threshold Similarity

threshold	=1.0	>0.9	>0.8	>0.7
popsize 10, 250 gens	0.04	0.18	0.38	0.59
popsize 10, 1000 gens	0.19	0.31	0.55	0.74
popsize 50, 5000 gens	0.54	0.68	0.86	0.94

[a] 100 structure comparisons were performed, using structures from the Fine Chemicals Database. The figures given are the mean over ten runs of the GA.

Table 4. Mean and Standard Deviation (in Brackets) Averaged over 10 Runs for Matching the Target Structure in Figure 2 with Itself and with the Three Top Hits

alignment	target	hit 1	hit 2	hit 3
GA 250 gens	0.79 (0.08)	0.79 (0.09)	0.76 (0.04)	0.78 (0.05)
GA 1000 gens	0.90 (0.07)	0.86 (0.03)	0.83 (0.03)	0.85 (0.03)
ASP	0.82 (0.13)	0.82 (0.11)	0.74 (0.13)	0.79 (0.14)

One limitation of the GA is that it can experience considerable difficulties when trying to align a molecule with itself: one would expect the similarity to be 1.0, but this was often not the case in our experiments. A test was carried out in which each of 100 diverse molecules was matched with itself ten times with three sets of parameters: a population of size 10 and 250 generations (the standard parameters used previously); a population of size 10 and 1000 generations; and a population of size 50 and 5000 generations. The results of this test are shown in Table 3, where it will be seen that this a general problem for the GA, even if the most extended search is carried out. It should be noted that these are mean values, and some of the 10 sets of runs gave much better results than those listed: for example, one of the 1000-operation runs had no less than 66 of the 100 searches giving a similarity of 1.0. These results indicate that a large number of operations would be required if the GA was to be used to generate precise alignments for small numbers of molecules; it is comforting to note that an increase in the number of operations had far less effect on the calculated similarities in the context of database searching. For example, running the set of 100 standard queries and 1000 structures mentioned previously with a 10-member population and a maximum of 1000 generations gave a value of 0.72 for E_{20}, which is only slightly greater than the 0.70 value achieved with the standard searching parameters of a 10-member population and 250 generations.

Current approaches to similarity searching are based on features such as fragment substructures or interatomic distances,[1] and there is thus generally a clear structural relationship between a target structure and its nearest neighbors. The very different matching criterion employed in this work means that it is often possible to identify structures that would have only a low degree of similarity with the target when searched using conventional similarity metrics, with the result that field-based approaches have the potential to suggest possible leads that are noticeably more diverse than those resulting from existing similarity-searching systems. This statement is exemplified by the search outputs shown in Figures 2 and 3, which each show one of the target

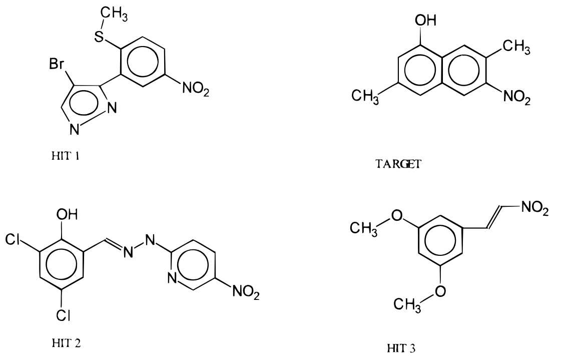

Figure 2. 2D structure diagrams for one of the 100 targets, and the top three hits from a GA search of 1000 molecules from the Fine Chemicals Database.

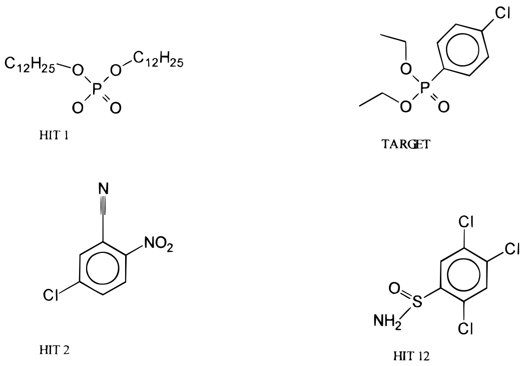

Figure 3. 2D structure diagrams for one of the 100 targets, and three of the top hits (numbers 3−11 were analogues of hit 2) from a GA search of 1000 molecules from the Fine Chemicals Database.

molecules used in these searches, and three of the molecules from the database that were considered by the GA to be most similar to the target.

3.3. Comparison with a Simplex Optimizer. A referee suggested that our results should be compared with those obtained from use of a simplex optimizer. We have thus carried out some experiments with the ASP similarity package,[43] using the molecules shown in Figure 2. The target molecule here was matched with itself and then with each of the three top-ranked hits, with each match being repeated ten times. This procedure was carried out: with the GA running for 250 generations (*i.e.*, as in our searching experiments); with the GA running for 1000 generations; and with ASP set up to return a similarity analogous to that produced by our GA (*i.e.*, the Gaussian electrostatic approximation was used, and no account was taken of conformational flexibility), using the the same initial, randomly-generated alignments as for the GA experiments.

Table 3 details the mean and standard deviation of the calculated similarities for the ten runs that were carried out in each case. It will be seen that the mean ASP similarity was slightly greater than that for the 250-generation GA but less than that for the 1000-generation GA and that both GAs gave noticeably less variable sets of similarity values than did ASP. We also carried out runs in which ASP was allowed to optimize that GA alignment that gave the largest similarity: here, the target gave a similarity of 1.00 (0.99) with itself, *i.e.*, the correct match had been identified, and similarities of 0.92 (0.91), 0.92 (0.88), and 0.91 (0.90) with the three other molecules (where the bracketed figure in each case is the similarity value for the initial, GA-based alignment that was input to ASP).

3.4. Efficiency of Searching. The discussion thus far has considered only the effectiveness of the GA, with little consideration of its computational efficiency. Our inital C implementation, running on an R4000 Silicon Graphics Indigo-2 machine with a 100 MHz R4000 chip, had a mean run-time of 3.7 CPU s to match a pair of molecules using the standard parameters of section 2.3. However, this figure is determined in large part by the specific parameter values that are used and will increase linearly if, for example, a larger number of iterations was to be used. For comparison, the Fortran 77 field-graph program described by Thorner[14] had a mean run-time of 2.7 CPU s for the same set of 100 target molecules and 1000 database molecules, while our C 5-run bit-climber program took 4.3 CPU s for the same dataset. We have implemented a number of optimizations to the C code which means that the current version takes around 1 s for a search: thus, a GA search on a standard workstation of a corporate database containing ca. 200 000 molecules would be expected to take about 3 CPU days for the completion of a search. While this is feasible, one would clearly wish to improve the speed of response, and we have thus developed parallel implementations on the Kendall Square Research KSR-1 multiprocessor system, as described in the following section.

4. PARALLEL IMPLEMENTATION

4.1. The Kendall Square Research KSR-1. The KSR-1 is a Multiple Instruction-stream, Multiple Data-stream (or MIMD) parallel computer that can contain between 8 and 1088 processors, each processor being a 20 MHz RISC-style 64-bit unit operating at 40 MIPS and 40 MFLOPS (peak speeds). Physical memory is distributed throughout the KSR-1, with each processor being attached to a 32 Mbyte local cache and to 0.5 Mbyte of fast *subcache* memory but the KSR's shared-memory programming model allows access to a single, contiguous space that represents the machine's entire memory. An individual processor may wish to access data stored in its own subcache, its own local cache, or elsewhere in the memory, and care must be taken to minimize the number of accesses to nonlocal and nonsubcache memory if fast execution speeds are to be achieved. The KSR-1 used in our experiments contained 64 processors, arranged in two groups each containing 32 processors, and was programmed in KSR C, which uses a standard Unix compiler with parallel extensions implemented as routines in dedicated libraries and with a simple C interface to a parallel run-time library.[44,45]

The underlying mechanism for parallelism is the *pthread*. A pthread is a sequential flow of control within a process that cooperates with other pthreads to solve a problem. They are thus a natural mechanism for implementing parallel applications, with each piece of work being assigned its own pthread (or its own team, which is a group of pthreads grouped together with a common team identifier). In the remainder of this section, we discuss thread-based mechanisms that can be used to provide both coarse-grained and fine-grained parallel implementations of our GA database search.

4.2. Coarse-Grained Searching. The most obvious way of implementing a database-searching program on a parallel architecture is to instruct each processor to process different records from the database. This level of parallelism was effected on the KSR using a *pool strategy*. The molecules that are to be searched are read into memory, forming a pool from which processors can obtain molecules to process when they are ready to perform work. Once all of the molecules have been extracted from the pool, the search is complete, and the results can be collated and reported. More specifically, a team of *N* pthreads is created. Each pthread will check the pool to see if there are any database molecules waiting to be processed; if so, the first available molecule is tagged to prevent another pthread from claiming it. The pthread will then execute the GA, comparing this database molecule with the target molecule. Once the GA has terminated, the best alignment of the target molecule with the database molecule is stored in a global data structure, and the pthread attempts to process another database molecule. This process continues until there are no more molecules left to process, leaving the pthread free to terminate.

Initial experiments showed that about 30% of the total execution time was being spent in bringing data to the processor. This was caused by a large number of *subcache misses*, *i.e.*, the inability to find desired data in a processor's subcache, resulting in an access to local cache memory. The Gaussian-similarity fitness function that lies at the heart of our GA, whether in serial or parallel implementations, involves the extensive calculation of exponentials, and we were able to obtain substantial increases in the speed of the serial program by using lookup tables that contained precalculated exponential values. The high subcache miss-rate on the KSR-1 was found to be due to the large memory requirements of these lookup tables, which are larger than the size of an individual subcache. Experiments were done with smaller lookup tables (meaning that some of the exponentials had to be calculated) and no lookup tables (meaning that all of the exponentials had to be calculated), and the results reported below are those that were obtained with moderate-sized tables, which gave the best results.

This optimized program was used to investigate how the performance of the program changed as the number of processors used was increased. Ideally, the use of *N* processors should result in a speedup of *N* times over the use of one processor, but the overheads in performing the work in parallel usually mean that the speedup is significantly less than this. Experiments were done with searches on the 1000-molecule dataset with 1−32 processors. The speedup with the latter-sized machine was 30.8, indicating that the program was making excellent use of the processors and that the intercache communications had been successfully minimized.

In order that the KSR pool strategy may be compared with the GA running on a serial computer, comparable similarity searches were carried out on a Silicon Graphics R4000 Indigo-2 serial machine and on a 48-processor subset of the 64 processors available in the KSR that was available to us. A database search of our 1000-molecule FCD subset took, on average, 261 CPU s (when averaged over a set of 10 target structures), as against 3741 CPU s on the serial machine.

4.3. Fine-Grained Searching. There is a large amount of parallelism in any GA: not only the inherent parallelism that characterizes the manipulation of the schema that describe the chromosome substrings,[22] but also the explicit parallelism that arises from the manipulation of the chromosomes comprising a population. There has thus been considerable interest in the development of parallel models of genetic computation,[46−49] and we now describe our studies of the applicability of two such models.

In the *island* (or *distributed population*) model, several different subpopulations are maintained, and each population is processed independently of the others. The only communication between populations is by *migration*; at given intervals, an individual will be chosen from one population and moved into another. The migration was implemented by means of a *migration pool*: when a subpopulation wishes to send a migrant, it is placed in the migration pool so that it is ready for another subpopulation to claim it when desired. The use of a migration pool avoids synchronization problems since processors are not locked into waiting for a migrant from a specific subpopulation (which may not produce one for some time), and since subpopulations are not held up waiting for a recipient subpopulation to be ready. If no migrations are available when a subpopulation wishes to receive one, processing of the subpopulation is continued with no new population member. Migration can be effected by using either *newest migration* or *best migration*. In the former, parents from one population reproduce to produce a child that is always passed on to the next population, where it replaces the population member with the worst fitness. In the latter, new members are inserted into their own populations, and then the best population member is chosen for migration to the next population. We found that newest migration gave the better results.

The alternative *centralized model* is very different, in that only a single population is used (as with the normal serial GA), but parallel processes are spawned which each work on the same global population (rather than on different subpopulations as in the island model). Here, each of the *N* pthreads in the basic coarse-grained algorithm creates a team of slave pthreads. These slave threads iteratively pick parents from the population and create new members to be inserted into the population. A new member replaces the population member that currently has the worst fitness.

Wild[37] presents the results of an extensive series of searches carried out using our implementations of these two fine-grained models. Neither of them was comparable in efficiency with the much simpler coarse-grained algorithm, and we have thus not included any of the search results here. However, the experiments suggest that these two models might be more appropriate for GAs that involve larger populations than those found to be effective here.

5. CONCLUSIONS

There is very considerable interest in the use of the MEP as a parameter in the design of drugs and pesticides,[50−52] and this has resulted in the development of several approaches for calculation of intermolecular similarities using the MEP.[16−20] These approaches are effective in operation but require the maximal alignment of the MEPs of a target structure and a database structure if the approaches are to be used for database searching. In this paper we have presented a GA for the alignment of pairs of MEPs so as to maximize the intermolecular electrostatic similarity and described its implementation in both serial and parallel environments. Our experiments demonstrate that the GA leads to similarities that are comparable in effectiveness for database searching to those resulting from the use of our previously-described approach based on field-graphs[14] and superior to those resulting from the use of a bit-climber. We also demonstrate that the GA leads to more robust alignments than does a simplex optimization procedure. A coarse-grained implementation of the GA on a KSR-1 MIMD parallel processor resulted in near-perfect speed-up, but two fine-grained implementations were noticeably less successful.

It is interesting to consider whether we should adopt the GA or the field-graph approach for further development, since both of them have weaknesses. The field-graph approach is far more complex and time-consuming owing to the need to generate the graphs from the electrostatic potential grids before a search can be carried out. While we have been able to devise reasonably effective methods for graph-generation, they do involve a drastic loss of information, and the resulting graphs thus describe only the most important parts of an MEP. Moreover, the field-graphs for some molecules do not contain sufficient nodes to enable those molecules to be aligned with a target molecule. However, with appropriate parameterization, the matching of field-graphs is sufficiently rapid to enable searches to be carried out on databases containing hundreds of thousands of molecules.[14] The GA that we have described here may be slower than a fully optimized field-graph program, even with the small population sizes and limited numbers of operations that we have used; moreover, the nondeterministic nature of a GA means that it can result in very poor alignments in some cases. Indeed, there is no guarantee that it will be able correctly to align a molecule with itself, with the result that the self-similarity can be less than the value of 1.0 for the Carbo index that one would expect; the field-graph approach, conversely, will always produce the correct alignment in such cases. Nevertheless, examination of search hits does show that the GA reliably makes a good set of matches even in a small (1000-molecule) database. In addition, it is possible to perform a search biased toward efficiency, *i.e.*, one that runs quickly but which may miss some good hits, or a more effective search, *i.e.*, one using a larger population size and/or a greater number of generations than in the experiments reported here.

There is a major limitation in the work that we have described thus far since the results presented both here and elsewhere[14] have considered only rigid molecules and have taken no account of the torsional flexibility that characterizes the majority of 3-D structures. The MEP of a molecule is known to be strongly dependent on its conformation,[53] and methods are thus required that can encompass such variations when seeking to identify the optimal alignment of a pair of MEPs (in much the same way that it has proved necessary to augment the initial systems for pharmacophoric pattern-matching with facilities for flexible searching).[31,54,55] We are currently investigating the extent to which the field-graph and GA approaches to field-based searching can encompass conformational flexibility. Previous work has demonstrated the general suitability of GAs for flexible matching procedures,[26,27,31,32] and our initial results suggest that they are to be preferred to the field-graph approach in the present context.[56]

ACKNOWLEDGMENT

We thank the following: the Biotechnology and Biological Sciences Research Council and the Science and Engineering Research Council for funding; Tripos Associates Inc. for hardware and software support; David Thorner and Matt Wright for assistance with the field-graph searches; Bobby Glen, Gareth Jones, and Robin Taylor for helpful discussions on the applications of genetic algorithms and molecular electrostatic potentials; Graham Riley and Robert Sumner for making available to us the KSR-1 at the Centre for Novel Computing in the University of Manchester and for much helpful advice; and Val Gillet and the referees for helpful comments on an initial draft of this paper. The Krebs Institute for Biomolecular Research is a designated Biomolecular Sciences Centre of the Biotechnology and Biological Sciences Research Council.

REFERENCES AND NOTES

(1) Downs, G. M.; Willett, P. Similarity Searching in Databases of Chemical Structures. *Rev. Comput. Chem.* **1995**, *7*, 1−66.

(2) Willett, P. Similarity and Clustering in Chemical Information Systems; Research Studies Press: Letchworth, 1987.

(3) Barnard, J. M.; Downs, G. M. Clustering of Chemical Structures on the Basis of Two-Dimensional Similarity Measures. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 644−649.

(4) Chemical Structure Systems; Ash, J. E., Warr, W. A., Willett, P., Eds.; Ellis Horwood: Chichester, 1991.

(5) Pepperrell, C. A.; Willett, P.; Taylor, R. Implementation and Use of an Atom-Mapping Procedure for Similarity Searching in Databases of 3-D Chemical Structures. *Tetrahedron Comput. Methodology* **1990**, *3*, 575−593.

(6) Bemis, G. W.; Kuntz, I. D. A Fast and Efficient Method for 2D and 3D Molecular Shape Description. *J. Comput.-Aided Mol. Design* **1992**, *6*, 607−628.

(7) Perry, N. C.; van Geerestein, V. J. Database Searching on the Basis of Three-Dimensional Molecular Similarity Using the SPERM Program. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 607−616.

(8) Nilakantan, R.; Bauman, N.; Venkataraghavan, R. A New Method for Rapid Characterisation of Molecular Shape: Applications in Drug Design. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 79−85.

(9) Bath, P. A.; Poirrette, A. R.; Willett, P.; Allen, F. H. Similarity Searching in Files of Three-Dimensional Chemical Structures: Comparison of Fragment-Based Measures of Shape Similarity. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 141−147.

(10) van Geerestein, V. J.; Perry, N. C.; Grootenhuis, P. G.; Haasnoot, C. A. G. 3D Database Searching on the Basis of Ligand Shape Using the SPERM Prototype Method. *Tetrahedron Comput. Methodology* **1990**, *3*, 595−613.

(11) Downs, G. M.; Willett, P.; Fisanick, W. Similarity Searching and Clustering of Chemical-Structure Databases Using Molecular Property Data. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 1094−1102.

(12) 3D QSAR in Drug Design; Kubinyi, H., Ed.; ESCOM: Leiden, 1993.

(13) Molecular Modelling and Drug Design; Vinter, J. G., Gardner, M., Eds.; Macmillan: London, 1994.

(14) Thorner, D. A. Electrostatic Field Searching in Databases of Three-Dimensional Chemical Structures; Ph.D. Thesis, University of Sheffield, 1995.

(15) Turner, D. B.; Willett, P.; Ferguson, A.; Heritage, T. W. Similarity Searching in Files of Three-Dimensional Chemical Structures. Evaluation of Similarity Coefficients and Standardisation Methods for Field-Based Similarity Searching. *SAR QSAR Environmental Res.* **1995**, *3*, 101−130.

(16) Manaut, F.; Sanz, F.; Jose, J.; Milesi, M. Automatic Search for Maximum Similarity Between Molecular Electrostatic Potential Distributions. *J. Comput.-Aided Molecular Design* **1991**, *5*, 371−380.

(17) Burt, C.; Richards, W. H.; Huxley, P. The Application of Molecular Similarity Calculations. *J. Comput. Chem.* **1990**, *11*, 1139−1146.

(18) Richard, A. M. Quantitative Comparison of Molecular Electrostatic Potentials for Structure-Activity Studies. *J. Comput. Chem.* **1991**, *12*, 959−969.

(19) Petke, J. D. Cumulative and Discrete Similarity Analysis of Electrostatic Potentials and Fields. *J. Comput. Chem.* **1993**, *14*, 928−933.

(20) Good, A. C.; Hodgkin, E. E.; Richards, W. G. The Utilization of Gaussian Functions for the Rapid Evaluation of Molecular Similarity. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 188−191.

(21) Good, A. C.; Richards, W. G. Rapid Evaluation of Shape Similarity Using Gaussian Functions. *J. Chem. Inf. Comput. Sci.* **1993**, *33*, 112−116.

(22) Goldberg, D. E. Genetic Algorithms in Search, Optimisation and Machine Learning; Addison-Wesley: New York, 1989.

(23) Handbook of Genetic Algorithms; Davis, L., Ed.; Van Nostrand Reinhold: New York, 1991.

(24) Forrest, S. Genetic Algorithms: Principles of Natural Selection Applied to Computation. *Science* **1993**, *261*, 872−878.

(25) Goldberg, D. E. Genetic and Evolutionary Algorithms Come of Age. *Commun. ACM* **1994**, *37*(3), 113−119.

(26) Blommers, M. J. J.; Lucasius, C. B.; Kateman, G.; Kaptein, R. Conformational Analysis of a Dinucleotide Photodimer with the Aid of the Genetic Algorithm. *Biopolymers* **1992**, *32*, 45−52.

(27) Judson, R. S.; Jaeger, A. M.; Treasurywala, A. M.; Peterson, M. L. Conformational Searching Methods for Small Molecules: a Genetic Algorithm Approach. *J. Comput. Chem.* **1993**, *14*, 1407−1414.

(28) Fontain, E. Application of Genetic Algorithms in the Field of Constitutional Similarity. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 748−752.

(29) Payne, A. W. R.; Glen, R. C. Molecular Recognition Using a Binary Genetic Search Algorithm. *J. Mol. Graphics* **1993**, *11*, 74−91.

(30) Brown, R. D.; Jones, G. J.; Willett, P.; Glen, R. C. Matching Two-Dimensional Chemical Graphs Using Genetic Algorithms. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 63−70.

(31) Clark, D. E.; Jones, G.; Willett, P.; Kenny, P. W.; Glen, R. C. Pharmacophoric Pattern Matching in Files of Three-Dimensional Chemical Structures: Comparison of Conformational Searching Algorithms for Flexible Searching. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 197−206.

(32) Jones, G.; Willett, P.; Glen, R. C. Molecular Recognition of Receptor Sites Using a Genetic Algorithm with a Description of Desolvation. *J. Mol. Biol.* **1995**, *245*, 43−53

(33) Carbó, R.; Leyda, L.; Arnau, M. How Similar is a Molecule to Another? An Electron Density Measure of Similarity Between Two Molecular Structures. *Int. J. Quantum Chem.* **1980**, *17*, 1185−1189.

(34) Sneath, P. H. A.; Sokal, R. R. Numerical Taxonomy; Freeman: San Francisco, CA, 1973.

(35) Mühlenbein, H. How Genetic Algorithms Really Work. I. Mutation and Hill-Climbing; Proceedings of Parallel Problem Solving from Nature 2; Elsevier: Amsterdam, 1992.

(36) Reeves, C. R. Using Genetic Algorithms with Small Populations. Proceedings of the Fifth International Conference on Genetic Algorithms; Forrest, S., Ed.; Morgan Kaufmann: San Mateo, CA, 1993; pp 92−99.

(37) Wild, D. J. Structural and Electrostatic Similarity Searching in Three-Dimensional Chemical Databases Using Genetic Algorithms and Parallel Computers. Ph.D. Thesis, University of Sheffield, 1994.

(38) Davis, L., Bit-Climbing, Representational Bias, and Test Suite Design, Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo CA, 1991, 18−23.

(39) Cramer, R. D.; Patterson, D. E.; Bunce, J. D. Comparative Molecular Field Analysis (CoMFA). 1. Effect of Shape on Binding of Steroids to Carrier Proteins. *J. Am. Chem. Soc.* **1988**, *110*, 5959−5967.

(40) CONCORD is distributed by the University of Texas at Austin and Tripos Associates, St. Louis, Missouri, U.S.A.

(41) Stewart, J. J. M. MOPAC: a Semiempirical Molecular Orbital Program. *J. Comput.-Aided Mol. Design* **1990**, *4*, 1−105.

(42) Siegal, S. Nonparametric Statistics for the Behavioral Sciences, McGraw-Hill: Kogakusha, Tokyo, 1956.

(43) ASP. Automated Similarity Package User's Guide; Oxford Molecular Ltd.: Oxford, 1993.

(44) KSR Parallel Programming; Kendal Square Research Corp.: Waltham, MA, 1992.

(45) KSR C Programming; Kendal Square Research Corp.: Waltham, MA, 1992.

(46) Tanase, R. Distributed Genetic Algorithms. Proceedings of the Third International Conference on Genetic Algorithms; Morgan Kaufmann: San Mateo, CA, 1989; pp 434−439.

(47) Gordon, V. S.; Whitley, D. Serial and Parallel Genetic Algorithms as Function Optimizers. In Proceedings of the Fifth International Conference on Genetic Algorithms; Forrest, S., Ed..; Morgan Kaufmann: San Mateo, CA, 1993; pp 177−183.

(48) Bianchini, R.; Brown, C. Parallel Genetic Algorithms on Distributed-Memory Architectures; University of Rochester Computer Science Department Technical Report 436; Rochester, NY, 1993.

(49) Grefenstette, J. J. Parallel Adaptive Algorithms for Function Optimization. Technical Report CS-81-19; Vanderbilt University Computer Science Department: Nashville, TN, 1981.

(50) Guha, S.; Majumdar, D.; Bhattacharjee, A. K. Molecular Electrostatic Potential: a Tool for the Prediction of the Pharmacophoric Pattern of Drug Molecules. *J. Mol. Struct. (Theochem)* **1992**, *256*, 61−74.

(51) Pépe, G.; Siri, D.; Reboul, J. The Molecular Electrostatic Potential and Drug Design. *J. Mol. Struct. (Theochem)* **1992**, *256*, 175−185.

(52) Burt, C. Molecular Similarity Calculations for the Rational Design of Bioactive Molecules. Molecular Modelling and Drug Design; Vinter, J. G., Gardner, M., Eds.; MacMillan: London, 1994; pp 305−332.

(53) Reynolds, C. A.; Essex, J. W.; Richards, W. G. Atomic Charges for Variable Molecular Conformations. *J. Am. Chem. Soc.* **1992**, *114*, 9075-9079.

(54) Moock, T. E.; Henry, D. R.; Ozkabak, A. G.; Alamgir, M. Conformational Searching in ISIS/3D Databases. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 184−189.

(55) Hurst, T. Flexible 3D Searching: the Directed Tweak Technique. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 190−196.

(56) Thorner, D. A.; Wild, D. J.; Willett, P.; Wright, P. M. Similarity Searching in Files of Three-Dimensional Chemical Structures. Flexible Field-Based Searching of Molecular Electrostatic Potentials. *J. Chem. Inf. Comput. Sci.* Submitted for publication.

CI9500851