# Using a genetic algorithm to identify common structural features in sets of ligands

**John D. Holliday and Peter Willett**

*Krebs Institute for Biomolecular Research and Department of Information Studies, University of Sheffield, Sheffield S10 2TN, UK*

*This article describes a program for pharmacophore mapping, called MPHIL (Mapping PHarmacophores In Ligands). Given as input a set of molecules that exhibit some common biological activity, MPHIL identifies the smallest 3D pattern of pharmacophore points that has at least* m *(a user-defined parameter) points in common with each of the input molecules. The program thus differs from existing programs for pharmacophore mapping in that it does not require all of the molecules to share exactly the same pattern of points, although it will find such a common pattern if it does, indeed, exist. MPHIL uses a genetic algorithm (GA) approach in which an initial, and very rapid, GA is used to suggest possible combinations of points that are then processed by the second GA to yield the final 3D pattern. © 1998 by Elsevier Science Inc.*

*Keywords: clique detection, genetic algorithm, graph matching, maximal common substructure, pharmacophore mapping*

## INTRODUCTION

There is much interest in the development of systems for pharmacophore mapping, where the availability of a small number of bioactive molecules enables the identification of the pharmacophoric pattern that is responsible for the observed activity.[1] Once a putative pattern has been identified, it can be searched against a database of either rigid or, preferably, flexible 3D structures to identify further, previously untested molecules that contain the query pharmacophore and that should hence be submitted for biological testing.[2] The results of these tests can then be used to validate, or to modify, the pharmacophore.

An example of such a molecular design tool is DISCO (DIStance COmparison).[3] This package for automatic pharmacophore detection was developed by Abbott Laboratories (Abbott Port, IL), and identifies both the bioactive conformers and an appropriate superposition rule given an input set of active compounds. DISCO uses a clique detection algorithm to identify the maximal common substructure (MCS), i.e., the largest pattern of pharmacophore features (such as donors or ring centroids) in 3D that is common to all of the input molecules.[4] An MCS algorithm provides an obvious basis for pharmacophore identification but its use can result in the generation of low-precision queries that retrieve large numbers of matching structures when a pharmacophore search is carried out on a database. Specifically, the requirement that the same set of features occur in the same geometric arrangement in all of the molecules can lead to MCSs that contain only a pair of features or that involve large interfeature distance range tolerances, even if many conformations are considered for each of the input molecules. More generally, the use of the MCS involves the inherent assumption that there is a single, common pharmacophore that is responsible for the observed activity, and this assumption may not be correct. Barnum et al.[5] address this problem by relaxing the requirement that all of the molecules under consideration must contain all of the features; instead, certain molecules may be permitted to miss a feature as long as the total number of molecules missing a feature, or some specific feature, is below a user-defined threshold. An alternative strategy is adopted in the work reported here, where we seek a minimal set of features that suffices to cover all of the molecules. Specifically, we describe a program, called MPHIL (Mapping PHarmacophores In Ligands), that identifies a $K$-point site, a pattern consisting of $K$ point-like features and the associated interfeature distances, with which all of the input molecules have at least some minimal number of features, $m$, in common, subject to the constraint that $K$ should be as small as possible. The next section of the article gives a broad overview of the design of the program, and the following sections detail the two principal components of the method we have devel-

Address reprint requests to: Prof. Peter Willett, Krebs Institute for Biomolecular Research and Department of Information Studies, University of Sheffield, Sheffield S10 2TN, United Kingdom.

Received 28 May 1997; revised 12 August 1997; accepted 27 August 1997.

655 Avenue of the Americas, New York, NY 10010

1093-3263/97/$17.00
PII S1093-3263(97)00080-6

oped for generating *K*-point sites. We then report the results of an extensive series of tests with published data sets and discuss our principal conclusions.

## PROGRAM OUTLINE

The input to MPHIL consists of the 3D coordinates of the features in each of the *N* bioactive molecules (typically *N* is in the range 5 to 20) that are to be analyzed, from which the interfeature distances are calculated; the minimal number of points, *m*, that each molecule must have in common with the *K*-point site; and the interpoint distance tolerances that must be met if two 3D patterns are to be regarded as geometrically equivalent.

The process involves the identification of a subset of *m* points from each of the *N* molecules, giving a total of *Nm* points. The geometry of each *m*-point subset may allow the superimposition of points such that one or more points may be common to more than one molecule when the user-defined interpoint distance tolerances are taken into account, this leading to a reduction in the total number of unique points, *K*. The optimum situation is the case in which all molecules contain the same pattern of *m* points, in which case $K = m$; this corresponds to the identification of an *m*-point substructure that is common to all of the molecules, as in existing programs for pharmacophore mapping. The worst possible situation, albeit a rather unlikely one given the types of data set for which pharmacophore mapping is carried out, is one in which none of the input molecules have any points in common at all, in which case, $K = Nm$. In general, then, $m \le K \le Nm$, and MPHIL tries to ensure that *K* is as small as possible. The resulting *K*-point site is thus analogous to the hyperstructure representations that have been suggested for reducing the computational requirements of 2D substructure searching, wherein a hyperstructure is the minimal set of atoms and bonds that encompasses all of the molecules in a data set.[6]

The identification of the smallest *K*-point site is an extremely demanding combinatorial problem: If each of the *N* molecules contains *P* points, then there are no less than

$$\left[\frac{P!}{m!(P-m)!}\right]^N$$

possible combinations of *m*-point subsets that may need to be considered. The matching of such subsets is effected in MPHIL in the normal way, i.e., by means of an MCS algorithm based on clique detection.[4] This algorithm identifies cliques of size *m* that are common to a number of molecules and that may thus be contained within the *K*-point site; the principal novel idea in MPHIL is the use of a genetic algorithm (hereafter a GA) to generate subsets that can be submitted to these clique-based matching procedures so as to minimize the final value of *K*. GAs provide an extremely cost-effective way of identifying good, although generally suboptimal, solutions for combinatorial problems with search spaces that are too large for exploration by deterministic search algorithms. Such problems occur in many areas of chemical structure handling, and GAs have thus been successfully applied to a wide range of problems such as conformational analysis, ligand docking, and *de novo* design, among others.[7,8]

The GA that lies at the heart of MPHIL uses a chromosome that encodes the 3D coordinates of the *K* points that comprise the current *K*-point site. For each molecule, a size-*m* clique must be found in both the molecule and the *K*-point site represented by the chromosome. This requires *N* invocations (one for each molecule) of the clique detection algorithm for each chromosome in each iteration of the GA, and our initial experiments showed this to be far too time-consuming for practical purposes, given the huge number of *m*-point subsets that would need to be considered. Accordingly, an initial, precursor GA has been developed that reduces the computational requirements of the second, clique detection stage. Specifically, the first GA identifies sets of *m* points, as defined by the corresponding interpoint distances, that are common to many of the input molecules, and the second GA then assembles these *m*-point subsets by means of the clique detection procedure to give a *K*-point site that is as compact as possible. The first GA is fast in operation and can thus be viewed as performing a role analogous to the screening methods that are used to maximize the efficiency of the graph-matching stage of 2D or 3D substructure searching.

## IDENTIFICATION OF THE INITIAL POPULATION

The first GA, hereafter referred to as GA-1, identifies a combination of *m* points from each molecule, such that the resulting set of points can be maximally superimposed (corresponding to minimizing the number of distinct points in the final *K*-point site). This GA is a standard, steady-state-with-no-duplicates algorithm[9] and is typically run for 100 000 generations, which is quite sufficient to generate a suitable population for the second GA, hereafter referred to as GA-2.

Each chromosome contains *N* sets, one for each of the *N* input molecules, of *m* integers, each integer representing one of the points within a molecule. An initial population of chromosomes is generated randomly subject to the constraints that the same point does not appear more than once in any set and that no duplicate chromosomes occur. Thus a chromosome in which $N = 8$ and $m = 3$ might be as shown in Table 1a, with the *i*th column representing the *m*-point subset for the *i*th molecule so that, for example, the three-point subset for the first molecule comprises the first, fourth, and sixth atoms in this molecule. Each integer in the chromosome indexes the 3D coordinates of a single atom in one of the set of molecules under investigation, and these sets of coordinates provide the basis for the processing that is represented in the rest of Table 1 (as described further below).

The population of chromosomes is processed using crossover and mutation operators, with roulette-wheel selection identifying the parent chromosomes for each such operator. Two parents are selected during crossover. This is a standard one-point crossover operator in which the crossover point is a randomly generated boundary between sets, and the sets of points following the crossover point are interchanged. Mutation involves one point in a chromosome being removed and replaced with a new point from the same molecule. In both cases, checks are made to ensure that no duplicate chromosomes occur, and the probability of each type of genetic operator being invoked is controlled by user-defined weights. The fitness function is then calculated for the new chromosomes that have been produced, with the chromosomes that have the lowest fitness values (see below) being replaced by fitter chromosomes derived from the parent(s).

**Table 1. Creation of sets of points by GA-1 that subsequently act as the chromosomes that are input to GA-2[a]**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| a | | | | | | | |
| 1 | 5 | 3 | 5 | 1 | 2 | 6 | 5 |
| 6 | 2 | 5 | 4 | 7 | 3 | 1 | 2 |
| 4 | 4 | 2 | 3 | 2 | 1 | 5 | 3 |
| b | | | | | | | |
| N:O<br>9.612 | N:O<br>6.554 | N:O<br>9.327 | N:N<br>4.255 | X:O<br>3.110 | N:O<br>6.911 | N:N<br>3.944 | O:O<br>4.283 |
| N:N<br>4.284 | O:O<br>8.995 | N:N<br>4.146 | N:O<br>9.346 | X:O<br>4.855 | N:O<br>3.157 | N:O<br>6.284 | X:O<br>3.459 |
| N:O<br>6.257 | N:O<br>3.548 | N:O<br>6.542 | N:O<br>6.642 | O:O<br>4.385 | O:O<br>8.644 | N:O<br>9.244 | X:O<br>5.123 |
| c | | | | | | | |
| N:N<br>4.284 | N:O<br>3.548 | N:N<br>4.146 | N:N<br>4.225 | X:O<br>3.110 | N:O<br>3.157 | N:N<br>3.944 | X:O<br>3.459 |
| N:O<br>6.257 | N:O<br>6.554 | N:O<br>6.542 | N:O<br>6.642 | O:O<br>4.385 | N:O<br>6.911 | N:O<br>6.284 | O:O<br>4.283 |
| N:O<br>9.612 | O:O<br>8.995 | N:O<br>9.327 | N:O<br>9.346 | X:O<br>4.855 | O:O<br>8.644 | N:O<br>9.244 | X:O<br>5.123 |
| d | | | | | | | |
| N:N<br>4.284 | N:O<br>3.548 | — | — | X:O<br>3.110 | — | — | — |
| N:O<br>6.257 | N:O<br>6.554 | — | — | O:O<br>4.385 | — | — | — |
| N:O<br>9.612 | O:O<br>8.995 | — | — | X:O<br>4.855 | — | — | — |

[a] See text for details.

The aim of GA-1 is to identify a maximal overlap of the points of the chromosomes, and the fitness function is thus an inverse measure of how badly the points overlap, called the fitness penalty. The fitness penalty is derived is as follows. The interpoint distances (and point descriptors for the connected points) of the *Nm* points are calculated from the input coordinate data and copied to a temporary matrix, as exemplified in Table 1b where, for example, the three points for the first molecule are assumed to be two nitrogens and one oxygen and the three interpoint distances are 9.612, 4.284, and 6.257 Å.

For ease of explanation, we have reordered each set with the lowest value first, as shown in Table 1c. Each *m*-point subset in a chromosome is compared with every other *m*-point subset in that chromosome. If, during the pairwise comparison of two such subsets, each of the three interfeature distances in the second subset lies within the matching tolerance of each of the respective distances of the first subset, and the point descriptors in each case are the same, then the second subset is removed from further consideration as an equivalent subset has been found. For example, the first and third columns of the matrix shown in Table 1c are equivalent if a tolerance of 0.5 Å is used, and it is thus possible to remove the third column from further consideration; similar comments apply to the first and fourth, first and seventh, second and sixth, and fifth and eighth col-

**Table 2. Use of MPHIL on a data set of 19 angiotensin-converting enzyme and thermolysin inhibitors**

| m | Initial tolerance | GA-1 time (min) | GA-2 time (min) | GA-2 iterations | K | Final tolerance |
|---|---|---|---|---|---|---|
| 3 | 0.50 | 3.2 | 64.7 | 2 044 | 3 | 0.10 |
| 3 | 0.50 | 3.2 | 66.5 | 2 369 | 3 | 0.20 |
| 4 | 0.50 | 4.4 | 40.4 | 1 344 | 5 | 0.35 |
| 4 | 0.50 | 4.7 | 68.2 | 2 382 | 5 | 0.30 |
| 5 | 0.50 | 5.1 | 76.9 | 2 119 | 9 | 0.60 |
| 5 | 0.75 | 5.3 | 80.6 | 2 336 | 7 | 0.75 |

**Table 3. Use of MPHIL on a set of 18 CCK-A antagonists**

| m | Initial tolerance | GA-1 time (min) | GA-2 time (min) | GA-2 iterations | K | Final tolerance |
|---|---|---|---|---|---|---|
| 3 | 0.50 | 2.7 | 24.0 | 719 | 3 | 0.35 |
| 4 | 0.50 | 3.5 | 119.0 | 4 398 | 5 | 0.40 |
| 4 | 0.50 | 3.7 | 46.1 | 1 826 | 5 | 0.30 |
| 5 | 0.50 | 4.6 | 53.0 | 1 978 | 9 | 0.55 |
| 5 | 0.50 | 4.6 | 22.7 | 710 | 9 | 0.45 |
| 5 | 0.75 | 4.9 | 37.4 | 1 292 | 7 | 0.85 |
| 5 | 0.75 | 6.2 | 68.0 | 2 596 | 6 | 0.90 |
| 5 | 0.75 | 6.1 | 34.3 | 1 136 | 7 | 0.70 |
| 5 | 0.75 | 4.5 | 41.2 | 1 300 | 7 | 0.75 |

**Table 4. Use of MPHIL with a dataset containing 18 *N*-methyl-D-aspartate antagonists**

| m | Initial tolerance | GA-1 time (min) | GA-2 time (min) | GA-2 iterations | K | Final tolerance |
|---|---|---|---|---|---|---|
| 3 | 0.50 | 5.2 | 25.3 | 831 | 3 | 0.45 |
| 3 | 0.50 | 4.8 | 42.3 | 1 240 | 3 | 0.45 |
| 4 | 0.50 | 6.2 | 66.0 | 1 982 | 9 | 0.60 |
| 4 | 0.50 | 5.9 | 65.2 | 1 936 | 10 | 0.55 |
| 4 | 0.75 | 5.8 | 109.8 | 3 544 | 7 | 0.80 |

**Table 5. Use of MPHIL with a dataset containing 10 antifilarial antimycin analogs**

| m | Initial tolerance | GA-1 time (min) | GA-2 time (min) | GA-2 iterations | K | Final tolerance |
|---|---|---|---|---|---|---|
| 3 | 0.50 | 1.8 | 20.4 | 1 221 | 3 | 0.10 |
| 3 | 0.50 | 1.6 | 24.4 | 1 560 | 3 | 0.10 |
| 4 | 0.50 | 2.4 | 27.4 | 1 721 | 5 | 0.30 |
| 4 | 0.50 | 2.4 | 29.6 | 1 852 | 5 | 0.30 |
| 5 | 0.50 | 4.0 | 23.8 | 1 448 | 9 | 0.35 |
| 5 | 0.75 | 3.8 | 18.4 | 983 | 8 | 0.70 |
| 5 | 0.75 | 4.1 | 17.7 | 1 076 | 8 | 0.85 |

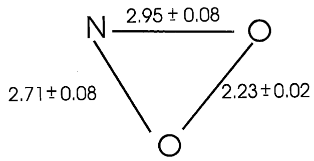

*Figure 1. Solution for MPHIL on a data set of 19 angiotensin-converting enzyme and thermolysin inhibitors using an* m *value of 3.*

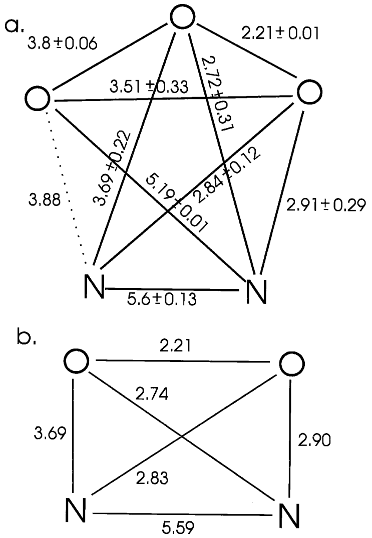

*Figure 2. Solution (a) and sample four-point subset (b) for MPHIL on a data set of 19 angiotensin-converting enzyme and thermolysin inhibitors, using an* m *value of 4.*

umns. The result of these pairwise comparisons, using this tolerance value, is shown in Table 1d.

It will be realised that this procedure shows a marked preference for the low-order *m*-point subsets; indeed, *m*-point subsets from the first molecule would always be represented in the final matrix, because the left-hand column above is never removed. A random ordering of the pairwise comparisons is hence used to reduce this order dependency.

The fitness penalty is calculated by comparing each of the remaining sets in the matrix with every other remaining set. During each such comparison, pairs of equivalent values (i.e., those within a given tolerance of each other and describing equivalent point descriptors), one from the first column and one from the second, are ignored. The discrepancy between (i.e., the modulus of the difference between) the two lowest value distances not yet considered from each set is then accumulated. Therefore, for columns 1 and 2 in the preceding example we have

$$|4.284 - 3.548| + |9.612 - 8.995| = 1.353$$

(the second row of both columns is not considered because the two distances and descriptor sets are equivalent); for columns 1 and 5 we have

$$|4.284 - 3.110| + |6.257 - 4.385| + |9.612 - 4.855| = 7.803$$

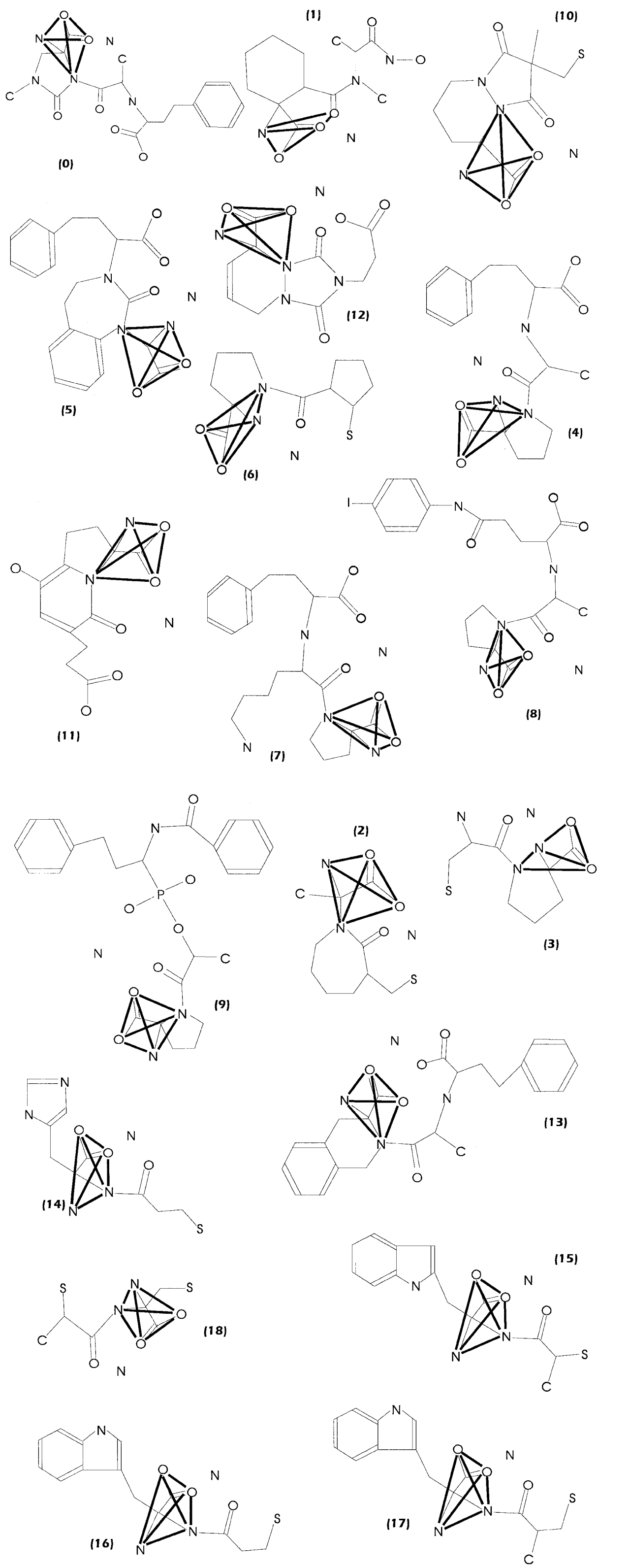

*Figure 3. Four-point subsets for MPHIL on a data set of 19 angiotensin-converting enzyme and thermolysin inhibitors.*

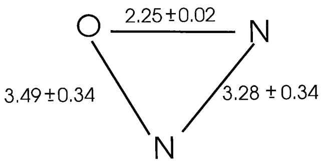

*Figure 4. Solution for MPHIL on a set of 18 CCK-A antagonists, using an* m *value of 3.*

and for columns 2 and 5 we have

$$|3.548 - 3.110| + |6.554 - 4.385| + |8.995 - 4.855| = 6.747$$

The resultant fitness penalty is the sum of these three values, i.e.,

$$1.353 + 7.803 + 6.747 = 15.903$$

and the fitness function for this particular chromosome is then the reciprocal of this fitness penalty.

## REFINEMENT OF THE POPULATION

Each chromosome output by GA-1 encodes a set of points that, taken together, represent a possible *K*-point site. The crossover and mutation operators of GA-2 are applied to each such chromosome to try to achieve a better fit between its constituent points and those of the input molecules, with the fit being checked using a clique detection procedure to confirm that a *m*-point match exists between the encoded *K*-point site and each of the input molecules. As with GA-1, the probability of each type of genetic operator being invoked is controlled by user-defined weights. The algorithm runs until there is no notable improvement in the fitness of the new chromosomes that are being generated. GA-2 is a standard steady state algorithm that uses roulette-wheel parent selection; duplicate chromosomes are not checked for because there is little chance of duplicate coordinates at all points in a pair of chromosomes.

Each chromosome for GA-2 contains the 3D coordinates of the *K*-point site encoded in a high-fitness chromosome output by GA-1, together with the point descriptors and the interpoint distances for that site. These distances are recalculated each time that a new chromosome is generated because mutation may move some of the points (as described below). Each chromosome is also assigned an initial tolerance level that is used to match distances in the clique detection stage. This level is initially the user-defined tolerance, but may be systematically increased when the *K*-point site from GA-1 requires a slightly larger tolerance for a fit. The initial fitness function (see below) for each chromosome is also calculated at this stage.

A standard one-point crossover technique is used in which coordinates following the crossover point are interchanged between two parent chromosomes to produce children. Mutation occurs in several different forms as follows:

- A new point is selected randomly from one of the molecules and added to the chromosome.
- A random point is removed from the chromosome.
- A random point is moved (or creeps) in a random direction by a user-defined distance.
- The tolerance for that chromosome is reduced by some user-defined value.
- The midpoint between two points is calculated and a new point is created at this new position, with the original two points being removed from the chromosome.

The last of these types of mutation is used when two points are so close together that a point located between them, in this case their midpoint, would be located within the tolerance distance of both points, i.e., if the points are separated by a distance that is not greater than twice the user-defined tolerance. A single point located at this midpoint would therefore encompass much of the 3D space covered by the two points, and replacing them hence reduces *K* for that chromosome. In the operations in which a point is removed (including the midpoint mutation), the chromosome tolerance is reinitialized to *T*, where *T* is the initial, user-defined tolerance, but may be systematically increased until clique detection is successful for all molecules, up to a maximum of *2T*. This value usually decreases again during processing, but sometimes results in a solution with a final tolerance that is greater than the initial tolerance.

Clique detection is carried out using the algorithm described by Bron and Kerbosch.[10] For each molecule–chromosome match a check is made to see whether any size-*m* cliques exist, i.e., whether that molecule and the *K*-point site encoded in that chromosome have a set of *m* points in common. If no size-*m* cliques are found in any one molecule–chromosome match, then the routine returns 0 and the new chromosome is discarded. If the new chromosome contains a common clique for each chromosome–molecule comparison, then the routine returns 1 and it replaces an unfit chromosome in the population.

The fitness function is given by the inverse of the sum of the number of chromosome points, *K*, and the current chromosome tolerance, $T_c$ i.e., $1/(K + T_c)$, so that the GA seeks *K*-point sites for which both *K* and $T_c$ are as small as possible. Once convergence has been achieved, the GA determines the value for the tolerance on each edge of the *K*-point site. The clique detection is carried out again, and for each edge–edge mapping between the solution and the compounds, the maximum discrepancy between the two is assigned to the edge of the solution. All mappings for this edge will then lie within this upperbound, which is effectively a tolerance on the edge.

## EXPERIMENTAL DETAILS AND RESULTS

Five sets of structures were used for testing the program; these are described below. For the first four data sets, the points that were used in the construction of the *K*-point sites were the constituent heteroatoms, each being labeled with their respective atom types. The last data set contained coordinate information for sites of activity, labeled AS (acceptor site) and DS (donor site), which were used in the *K*-point site construction. The 3D coordinates and point descriptors (i.e., the elemental types) are read in and the interpoint distances calculated. The parameter values used in a particular run are specified in a parameter file; the values used for all of the experiments reported here are as follows:

- For GA-1, a population of 1 000 chromosomes, 100 000 iterations, and a crossover probability of 0.25

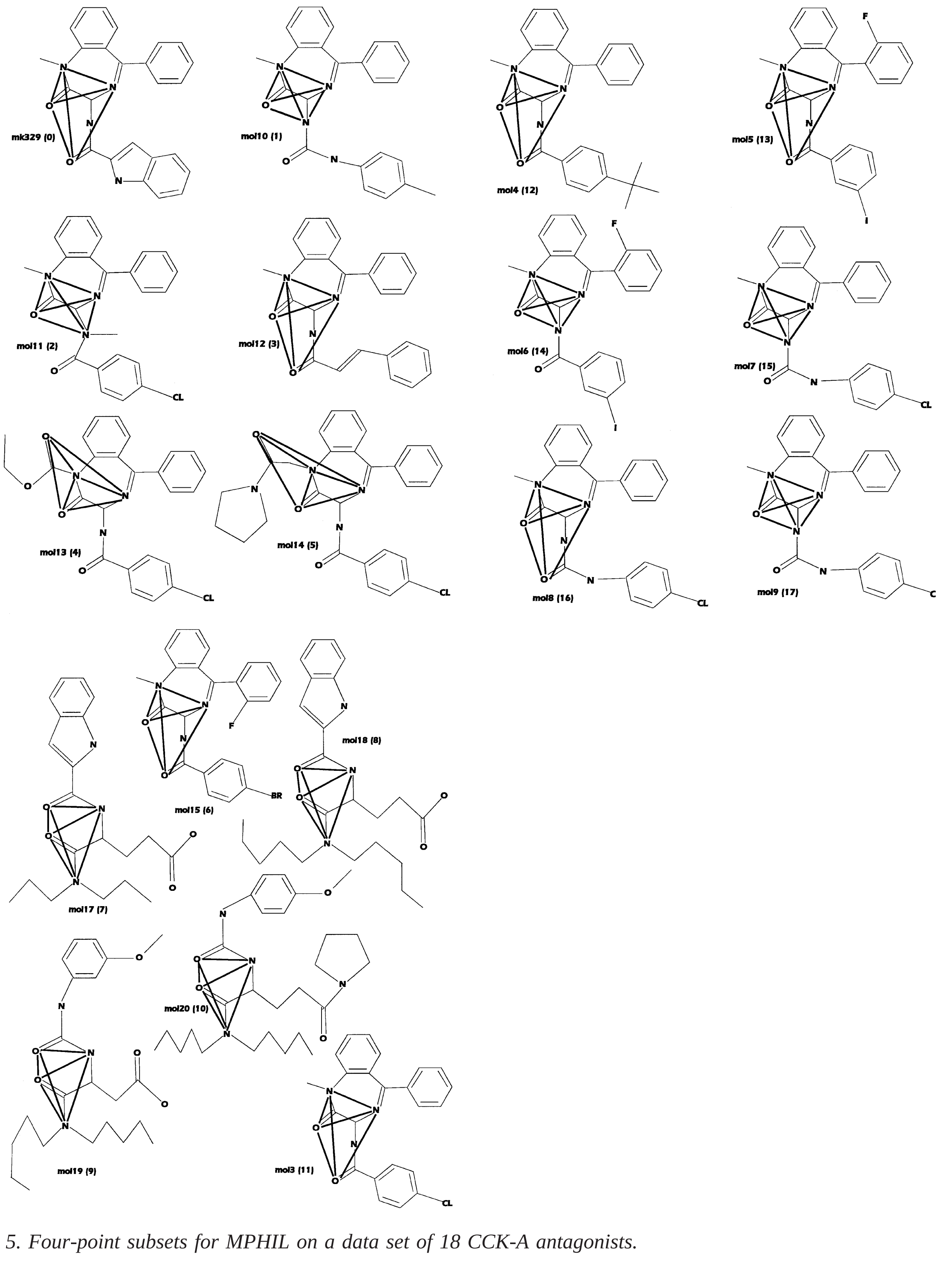

*Figure 5. Four-point subsets for MPHIL on a data set of 18 CCK-A antagonists.*

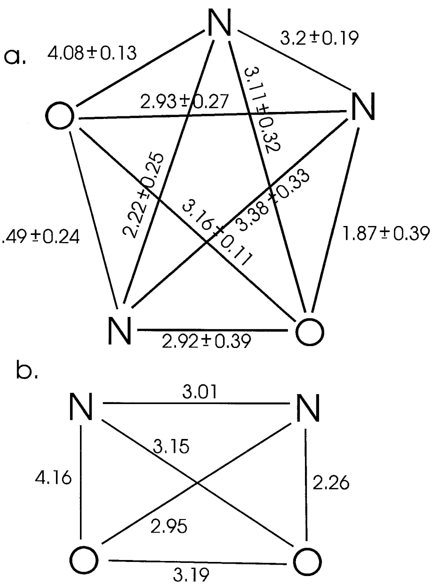

*Figure 6. Solution (a) and sample four-point subset (b) for MPHIL on a data set of 18 CCK-A antagonists, using an* m *value of 4.*

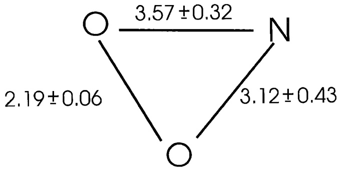

*Figure 7. Solution for MPHIL on a set of 18* N*-methyl-*D*-aspartate antagonists, using an* m *value of 3.*

- For GA-2, a population of 100 and a crossover probability 0.25. Within the mutation there is a 0.33 probability of a reduction in tolerance, and the four remaining mutation operation probabilities are in the ratio of 0.1 for a midpoint operation, 0.3 for a point removal, 0.3 for a new point selection, and 0.3 for a point creep. The tolerance reduction is 0.05 Å and the distance moved during a creep is 0.2 Å

The five data sets are detailed below. Where a subset of the original data is involved, this was created by taking every third compound from the original data set:

- Angiotensin-converting enzyme and thermolysin inhibitors (ACE): A subset of 19 compounds, containing between 7 and 12 points, selected from a set of 58 described by Scott et al.[11]
- A set of 18 CCK-A antagonists (CCKA), containing between 5 and 8 points, described by Rault et al.[12]
- A set of 18 *N*-methyl-D-aspartate antagonists (NMDA), containing between 5 and 8 points, described by Ortwine et al.[13]
- Antifilarial antimycin analogs (SELW): A subset of 10 compounds, containing between 5 and 12 points, selected from a set of 31 described by Selwood et al.[14]

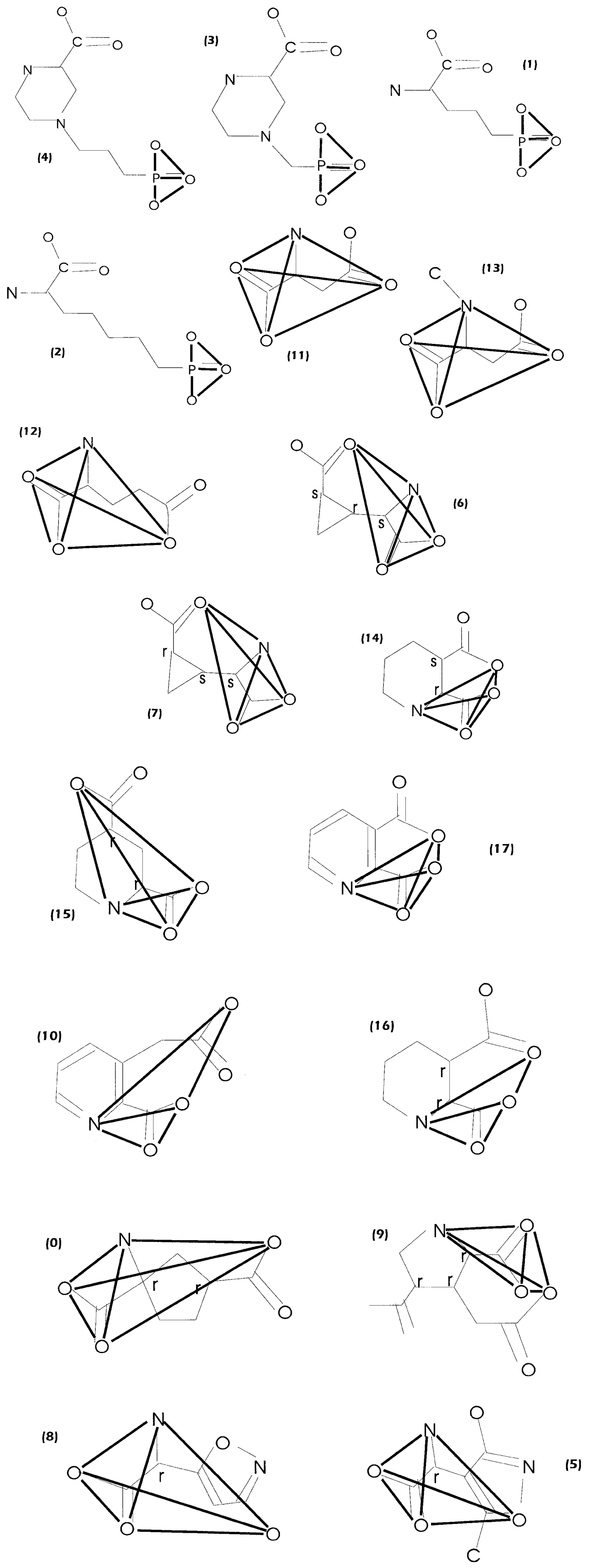

*Figure 8. Four-point subsets for MPHIL on a data set of 18* N*-methyl-*D*-aspartate antagonists.*

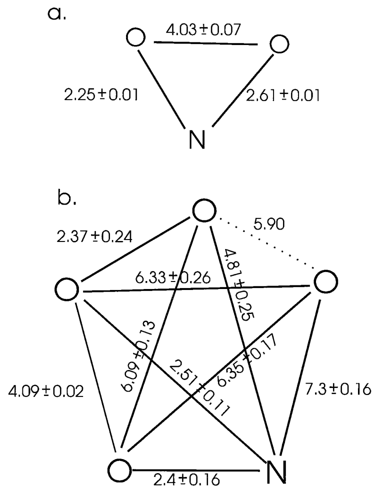

*Figure 9. Solutions for MPHIL on a data set of 10 antifilarial antimycin analogs using* m *values of 3 (a) and 4 (b).*

- A set of 15 heterogeneous ligands selected from the Brookhaven Protein Data Bank (PDB),[15] containing between 3 and 27 data points labeled as acceptor site (AS) or donor site (DS): One of these compounds contains just three donor sites and no acceptor sites, another contains just four acceptor sites and no donor sites

In each case, the 3D structures of the molecules were obtained by running the CONCORD structure generation program, followed by minimization using the Tripos force field in SYBYL 6.3.

Several runs were carried out for each data set, as detailed in Tables 2–5, which describe the user-defined $m$ values and tolerances, the run times in CPU minutes for C programs running on a DEC Alpha 3000 workstation running under Unix, the number of iterations performed by GA-2, and the final $K$ values and tolerances.

***ACE*** The 19 compounds were tested using $m$ values of 3, 4, and 5 with an initial tolerance of 0.5 Å and using an $m$ value of 5 with an initial tolerance of 0.75 Å. Several runs were carried out; the best solutions are summarized in Table 2.

***m* = 3** All 19 compounds contain a common substructure composed of the two oxygens in a carboxylic acid group and an amine. As a result, the same three-point subset is repeated throughout all of the compounds giving a $K$ of 3. This solution, together with the edge tolerances, is shown in Figure 1.

***m* = 4** A common four-point substructure is found in 18 of the compounds, this being made up of the carboxylic acid, the amine, and a further nitrogen in close proximity to the three-point subset. One compound does not contain the second nitrogen, however, and a further point is required to cover this compound. The solution is then the five-point site shown in Figure 2a. The absence of a tolerance on one edge of the solution is due to the fact that this edge is not contained within any of the individual four-point subsets. This is indicated by a broken line in Figure 2a. The 4-point subsets that contribute to the 5-point solution, together with their mappings to the 2D structure diagrams of the 19 compounds, are shown in Figure

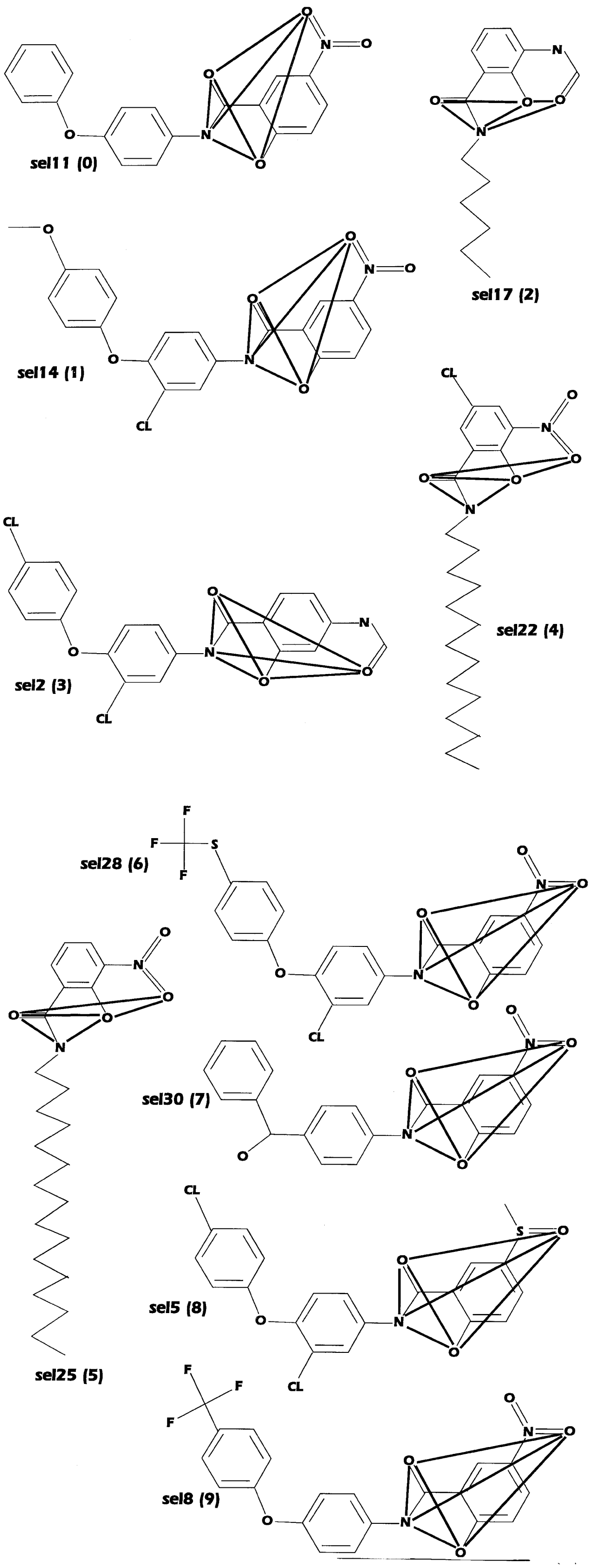

*Figure 10. Four-point subsets for MPHIL on a data set of 10 antifilarial antimycin analogs.*

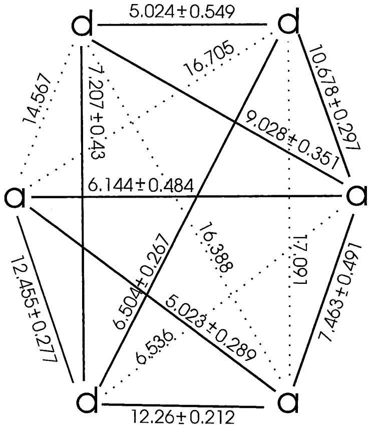

*Figure 11. Solution for MPHIL on a data set of 15 heterogeneous ligands selected from the Brookhaven Protein Data Bank, using an* m *value of 3.*

3, with the interpoint distances for the 4-point subset found in the first of these compounds being shown in Figure 2b.

***m* = 5** Initial tests using an *m* value of 5 and an initial tolerance of 0.5 Å resulted in a solution containing nine points and a final tolerance of 0.6 Å. This was repeated using an initial tolerance of 0.75 Å, producing a solution containing just seven points. Individual tolerances on the edges of the solution varied from 0.17 to 0.75 Å.

***CCKA*** As with the ACE inhibitors, an initial tolerance of 0.5 Å was used for *m* values of 3, 4, and 5 and an additional set of runs were carried out using an initial tolerance of 0.75 Å with an *m* value of 5. The results are shown in Table 3.

***m* = 3** The 3-point solution shown in Figure 4 was common to all 18 compounds tested. A final tolerance value of 0.35 Å was sufficient for this solution.

***m* = 4** As with the ACE inhibitor dataset, the same three-point subset was found to be included in all four-point subsets, as shown in Figure 5. The final solution, the five-point site of Figure 6a, contains three nitrogens and two oxygens with a final tolerance of 0.4 Å. A further run produced the same set of mappings but with slightly different edge distances and edge tolerances. Figure 6b illustrates the four-point subset found in the first compound (mk329) of Figure 5.

***m* = 5** A nine-point site was the best result that could be achieved with an initial tolerance of 0.5 Å. An increase in the initial tolerance to 0.75 Å gave solutions containing seven or even six points, although the final tolerance in the latter case was 0.9 Å.

***NMDA*** This set of 18 compounds was quite structurally diverse, and useful results, i.e., runs with compact *K*-point sites, were only obtained from *m* values of 3 and 4 (at 0.5 Å tolerance), with the latter requiring a further run with an initial tolerance of 0.75 Å. A summary of the results is shown in Table 4.

***m* = 3** These structures all contain a common carboxylic acid in close proximity to a nitrogen, as with the ACE inhibitors. It was thus easy to find the three-point site shown in Figure 7, with a final tolerance at 0.45 Å.

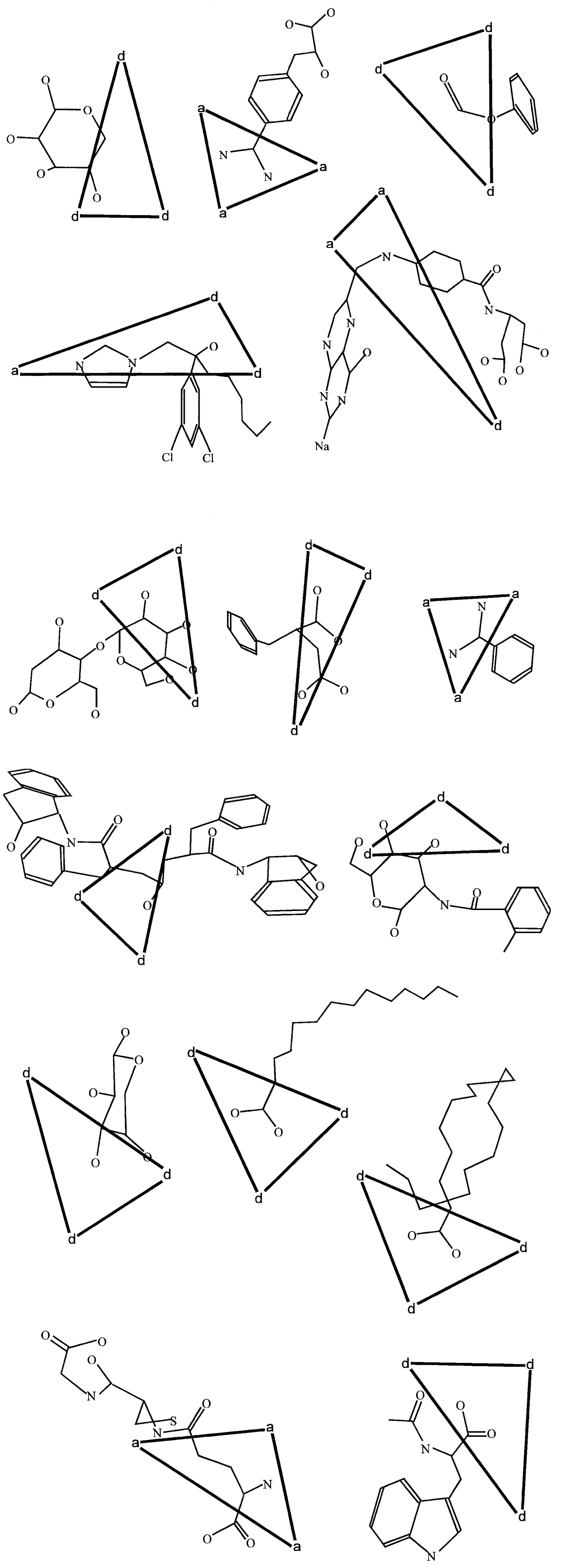

*Figure 12. Three-point subsets for MPHIL on a data set of 15 heterogeneous ligands selected from the Brookhaven Protein Data Bank.*

***m* = 4** An initial tolerance of 0.5 Å gave final solutions with *K* as high as 9 or 10. An increase in the tolerance to 0.75 Å produced a seven-point solution with a final tolerance of 0.8 Å. The 18 four-point subsets that contribute to the 7-point site are shown in Figure 8.

***SELW*** The program was tested using *m* values of 3, 4, and 5 with initial tolerances of 0.5 and 0.75 Å ($m = 5$ only), as shown in Table 5.

***m* = 3** The presence of a common substructure in all 10 compounds resulted in a 3-point solution with a low final tolerance value of 0.1 Å, as shown in Figure 9a.

***m* = 4** An initial tolerance of 0.5 Å resulted in the five-point solution of Figure 9b. As with some of the previous datasets, the three-point solution for an *m* value of 3 (which is shown in Figure 9a), is found in all of the four-point subsets that contribute to the five-point solution. The 10 four-point subsets that contribute to the 5-point site are shown in Figure 10.

***m* = 5** An increase in the initial tolerance was again required to reduce the solution from nine points at 0.5 Å to eight points at 0.75 Å.

***PDB*** The previous data sets have mainly consisted of sets of close analogs. With the heterogeneous structures in this data set, the program was tested using an *m* value of 3 (because one compound contained only three points) and an initial tolerance of 0.5 Å. This produced a *K*-point solution containing six points and a final tolerance of 0.55 Å. This is an exceptional result when considering that one compound contained just three donor sites and another compound contained just four acceptor sites. The solution and the corresponding mappings are shown in Figures 11 and 12, respectively, in which *a* represents an acceptor site and *d* represents a donor site.

## CONCLUSIONS

In this article, we have reported the development of an algorithm for finding the smallest pattern of points in 3D space that has at least some user-defined number of points in common with each of a set of molecules. The program implementing this algorithm, called MPHIL, thus differs from many existing programs for pharmacophore mapping in that it does not require all of the molecules to share exactly the same pattern of points, although it will find such a common pattern if it does, indeed, exist.

The two GAs that lie at the heart of MPHIL have proven to be effective in operation using a range of literature data sets, and also to be surprisingly efficient given the inherently combinatorial nature of the processing that is required. A run typically takes some tens of CPU minutes for a C implementation running on a medium-level Unix workstation, and the program is thus sufficiently rapid to enable several different solutions to be obtained without undue effort. That said, there are obvious limitations. Firstly, most of our experiments have considered only patterns of heteroatoms, and the algorithm needs to be run with more realistic pharmacophore definitions. For example, DISCO represents compounds by the following classes of pharmacophoric features: hydrogen bond donors and acceptors, the centers of hydrophobic regions, and extensions of hydrogen bonds and electrostatic interactions to binding sites in the receptor. Second, and more seriously, the work to date has considered only the limiting case in which all of the molecules are considered to be rigid, and represented by a single low-energy conformation. Possible approaches to encompass conformational flexibility include storing several different low-energy conformations for each molecule, as is done with pharmacophore mapping programs like DISCO[3]; encoding rotatable torsion definitions in the chromosome, as is done in the pharmacophore mapping program GASP[16]; or using a torsional optimization procedure, as is done in some systems for flexible pharmacophore searching.[17,18] The relative merits of these, or other, approaches remain to be investigated.

## ACKNOWLEDGMENTS

We thank Tad Hurst for bringing this problem to our attention, Akbar Naeem and Trevor Heritage for providing data, the referees for their comments on an earlier draft of this paper, and Tripos Inc. for software support and funding. The Krebs Institute for Biomolecular Research is a designated biomolecular sciences center of the Biotechnology and Biological Sciences Research Council.

## REFERENCES

1 Martin, Y.C. Pharmacophore mapping. In: *Designing Bioactive Molecules: Techniques and Applications* (Y.C. Martin and P. Willett, eds.). American Chemical Society, Washington, D.C., 1997 (in press)

2 Bures, M.G., Martin, Y.C., and Willett, P. Searching techniques for databases of three-dimensional chemical structures. *Topics Stereochem.* 1994, **21,** 467–511

3 Martin, Y.C., Bures, M.G., Danaher, E.A., DeLazzer, J., Lico, I., and Pavlik, P.A. A fast new approach to pharmacophore mapping and its application to dopaminergic and benzodiazepine agonists. *J. Comput.-Aided Mol. Design* 1993, **7,** 83–102

4 Brint, A.T., and Willett, P. Algorithms for the identification of three-dimensional maximal common substructures. *J. Chem. Inf. Comput. Sci.* 1987, **27,** 152–158

5 Barnum, D., Greene, J., Smellie, A., and Sprague, P. Identification of common functional configurations among molecules. *J. Chem. Inf. Comput. Sci.* 1996, **36,** 563–571

6 Brown, R.D., Downs, G.M., Willett, P., and Cook, A.P.F. A hyperstructure model for chemical structure handling: Generation and atom-by-atom searching of hyperstructures. *J. Chem. Inf. Comput. Sci.* 1992, **32,** 522–531

7 Willett, P. Genetic algorithms in molecular recognition and design. *Trends Biotechnol.* 1995, **13,** 516–521

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

9 Goldberg, D.E. *Genetic Algorithms in Search, Optimisation and Machine Learning.* Addison-Wesley, New York, 1989

10 Bron, C., and Kerbosch, J. Algorithm 457. Finding all cliques of an undirected graph. *Commun. ACM* 1973, **16,** 575–577

11 Scott, A., DePriest, S.A., Mayer, D., Naylor, C.B., and

Marshall, G.A. 3D-QSAR of angiotensin-converting enzyme and thermolysin inhibitors: A comparison of CoMFA models based on deduced and experimentally determined active site geometries. *J. Am. Chem. Soc.* 1993, **115,** 5372–5384

12 Rault, S., Bureau, R., Pilo, J.C., and Robba, M. Comparative molecular field analysis of CCK-A antagonists using field-fit as an alignment technique. A convenient guide to design new CCK-A ligands. *J. Comput.-Aided Mol. Design* 1992, **6,** 553–568

13 Ortwine, D.F., Malone, T.C., Bigge, C.F., Drummond, J.T., Humblet, C., Johnson, G., and Pinter, G.W. Generation of *N*-methyl-D-aspartate agonists and competitive antagonist pharmacophore models. Design and synthesis of phosphonoalkyl-substituted tetrahydroisoquinolines as novel antagonists. *J. Med. Chem.* 1992, **35,** 1345–1370

14 Selwood, D.L., Livingstone, D.J., Comley, J.C.W., O'Dowd, A.B., Hudson, A.T., Jackson, P., Jandu, K.S., Rose, V.S., and Stables, J.N. Structure–activity relationships of antifilarial antimycin anologues: A multivariate pattern recognition study. *J. Med. Chem.* 1990, **33,** 136–142

15 Bernstein, F.C., Koetzle, T.F., Williams, G.J.B., Meyer, F., Bryce, M.D., Rogers, J.R., Kennard, O., Shimanouchi, T., and Tasumi, M. The Protein Data Bank: A computer-based archival file for macromolecular structures. *J. Mol. Biol.* 1977, **112,** 535–542

16 Jones, G., Willett, P., and Glen, R.C. A genetic algorithm for flexible molecular overlay and pharmacophore detection. *J. Comput.-Aided Mol. Design* 1995, **9,** 253–264

17 Moock, T.E., Henry, D.R., Ozkabak, A.G., and Alamgir, M. Conformational searching in ISIS/3D databases. *J. Chem. Inf. Comput. Sci.* 1994, **34,** 184–189

18 Hurst, T. Flexible 3D searching: The directed tweak technique. *J. Chem. Inf. Comput. Sci.* 1994, **34,** 190–196