# Multiobjective Optimization of Pharmacophore Hypotheses: Bias Toward Low-Energy Conformations

Eleanor J. Gardiner,*,† David A. Cosgrove,‡ Robin Taylor,||,§ and Valerie J. Gillet†

Department of Information Studies, University of Sheffield, Regent Court, 211 Portobello Street, Sheffield S1 4DP, United Kingdom, AstraZeneca, Mereside, Alderley Park, Macclesfield, Cheshire, United Kingdom SK10 4TG, and Cambridge Crystallographic Data Centre, 12 Union Road, Cambridge, CB2 1EZ, United Kingdom

Received July 31, 2009

Two methods are described for biasing conformational search during pharmacophore elucidation using a multiobjective genetic algorithm (MOGA). The MOGA explores conformation on-the-fly while simultaneously aligning a set of molecules such that their pharmacophoric features are maximally overlaid. By using a clique detection method to generate overlays of precomputed conformations to initialize the population (rather than starting from random), the speed of the algorithm has been increased by 2 orders of magnitude. This increase in speed has enabled the program to be applied to greater numbers of molecules than was previously possible. Furthermore, it was found that biasing the conformations explored during search time to those found in the Cambridge Structural Database could also improve the quality of the results.

## INTRODUCTION

A pharmacophore describes the three-dimensional (3D) arrangement of chemical features required for a small molecule to bind to a protein.[1] The aim of pharmacophore elucidation is to deduce the requirements for binding to enable the identification of potentially active small molecules which could be taken forward for biological testing. While automated methods have been developed to determine the pharmacophore using protein−ligand complexes,[2] the more challenging task, and the focus of this paper, is pharmacophore elucidation in the absence of structural information of the protein. Thus, the aim of our method is to deduce the pharmacophore from a series of compounds which are assumed to bind to the same active site of a protein.

There are two main components to pharmacophore elucidation: one is the identification and placement of functional groups within the molecules that can form interactions with a receptor; the other is the alignment of the molecules such that groups with similar properties are overlaid. The functional groups are usually generalized into interaction types such as hydrogen-bond acceptors and donors, aromatic centers, and charge−charge interactions. A recent review of the different types of pharmacophore features that are typically encoded is provided by Wolber et al.[3] The alignment step involves superimposing the molecules such that the pharmacophoric features are overlaid while also taking into account the conformational flexibility of the molecules. Fully automated pharmacophore elucidation remains a challenging task for several reasons. Typically there are several features in each molecule which are candidates for the pharmacophore; there are multiple ways in which they can be mapped from one molecule to another, and the binding conformations are not usually known.

The active analogue approach is an early, semiautomated approach to pharmacophore identification in which the mapping of the pharmacophoric features is defined manually, and a systematic conformational search is used to generate potential alignments.[4] The first fully automated pharmacophore elucidation programs, in which different mappings between features are explored together with conformational search, appeared in the early- to mid-nineties. Perhaps the most widely known of these programs are DISCO,[5] GASP,[6] and Catalyst.[7] In common with other applications that involve 3D structures, such as docking and 3D database searching, there are essentially two different ways in which conformational space is explored. One is to precompute a set of conformers for each ligand and to consider each conformer of each ligand in turn (as done in Catalyst and DISCO), and the other is to use an optimization algorithm, such as an evolutionary algorithm, to explore conformation on-the-fly, as implemented in GASP. A comparison of these programs on sets of ligands extracted from crystal structures in the Protein Data Bank (PDB)[8] for which the pharmacophore can be deduced gave disappointing results, with all of the programs performing poorly on some of the data sets.[9] Thus, pharmacophore elucidation continues to be an active area of research, and several new approaches have been reported in the literature since the Patel study, including the GALAHAD,[10] PharmID,[11] and PHASE[12] programs. A recent review of pharmacophore programs is provided by Martin.[13]

Our own recent work in this area has been based on the use of multiobjective optimization techniques which are designed to evolve a family of different trade-off solutions.[14,15] We contend that, in the absence of the receptor, it is rarely possible to overlay a series of molecules unambiguously, since, as discussed above, there are usually many different ways in which the pharmacophoric features in the molecules can be matched and there are typically several different

* Corresponding author. Email: e.gardiner@sheffield.ac.uk.
† Department of Information Studies, University of Sheffield.
‡ AstraZeneca.
§ Cambridge Crystallographic Data Centre.
|| Current address: Taylor Cheminformatics Software, 54 Sherfield Avenue, Rickmansworth, Hertfordshire WD3 1NL, United Kingdom.

10.1021/ci9002816 CCC: $40.75 © 2009 American Chemical Society
Published on Web 11/12/2009

conformations available to each molecule. Our aim, therefore, is to generate a series of plausible pharmacophore hypotheses that represent different trade-offs in the goodness of the alignment and the conformational energies of the molecules. As in the GASP program, we explore conformational space simultaneously with the alignment. Our initial algorithm was subject to the limitation that all molecules must contain all pharmacophore points, on the assumption that all the molecules bind to the receptor in exactly the same way.[14] We subsequently relaxed this constraint to allow partial matches to be specified, which enables the handling of more diverse sets of molecules and reported results on a set of CDK2 ligands extracted from the PDB where the known alignment of the ligands can be deduced.[15] We were able to generate an alignment close to the crystal structure alignment for six of the molecules, however, this required a large population (2 000 chromosomes), and it required 200 000 operations with an execution time of around 15 h. Furthermore, it was evident that the MOGA was unable to deduce the correct conformations of ligands for features that occur in flexible regions in all ligands. Thus, it was evident that the increase in functionality of the algorithm had led to a large increase in the search space to be explored and a consequent increase in computational cost. Furthermore, it was also evident that the conformational search could result in unrealistic conformations of the molecules being generated since, in common with GASP, no conformational preference information was taken into account when exploring the conformational space of the ligands. Although the resulting conformers are scored on energy, this is done using a crude calculation and is of limited accuracy.

In this paper, we describe the continuing development of our multiobjective optimization algorithm for pharmacophore identification. Specifically, we have implemented two mechanisms to ensure that the conformational search is biased toward low-energy conformations. The first is the introduction of a preprocessing step which is used to ensure that the initial population consists of good starting points for subsequent optimization. This is an iterative procedure in which a clique detection algorithm is used to define an initial mapping between precomputed conformers of the ligands, with the mapping and the conformational information then used to initialize a chromosome. This process is repeated, for both different mappings and conformers, to generate the population of chromosomes. This step has led to a drastic reduction in run times so that the program can be applied to larger and more diverse data sets than was possible previously. The second way in which conformations are biased is through the implementation of a new mutation operator, whereby torsion angles that occur in the Cambridge Structural Database[16] are preferred. Thus, although the preprocessing step is based on precomputed conformers, the subsequent optimization continues to allow the input conformations to be modified on-the-fly albeit with a preference toward statistically preferred conformations.

The paper is organized as follows. A brief outline of the basic algorithm is given first, with the reader referred to our earlier papers for full details.[14,15] We then describe the preprocessing step to set up the population of chromosomes and the use of statistical information within the MOGUL program to bias the search toward energetically favorable conformers. We present results on various literature data sets that are commonly used to evaluate pharmacophore methods.

**Table 1.** Chromosome Structure[a]

| | mapping columns | | | | | | | | conformations | |
|---|---|---|---|---|---|---|---|---|---|---|
| ligand | donors | | acceptors | | hydrophobes | | | | torsion angles | |
| 1 | Du | 1 | 1 | Du | 3 | Du | 1 | 2 | T1 | T2 |
| 2 | 1 | Du | 2 | Du | 2 | 3 | 4 | 1 | T1 | T2 |
| 3 | 1 | Du | 1 | Du | 2 | 3 | 1 | Du | T1 | T2 |

[a] "Du" are dummy features and allow the possibility of a ligand being excluded from the pharmacophore point. Full mapping columns are shaded dark-grey, and partial mapping columns, which represent meaningful mappings, are shaded light-grey. Columns which contain insufficient features to specify a pharmacophore point are unshaded.

## METHODOLOGY

**Overview of Algorithm.** Our approach is based on a multiobjective evolutionary algorithm in which the conformations of the input molecules are varied simultaneously with mappings between pharmacophoric features. Pharmacophoric features are identified for each ligand with the algorithm currently able to recognize donors, acceptors, and hydrophobic groups (as ring centroids and as user-defined lists of atoms). Each donor and acceptor is represented by a pair of points: the heavy atom itself and the virtual point. For a donor, a virtual point is created 2.9 Å from the heavy atom attached to the hydrogen-bond donor proton, and for an acceptor, the virtual point is placed 2.9 Å from the heavy atom associated with each acceptor lone pair. Hydrophobic rings are represented by a centroid and a normal.

The chromosome consists of two parts: a conformational part, which encodes the conformations of the ligands, and a mapping part, which encodes a putative mapping between the ligands (see Table 1). The conformational part of the chromosome consists of $N$ strings to encode the conformations of $N$ ligands, with each rotatable bond of each ligand encoded as an 8-bit number to give a resolution of 1.4°. The mapping part of the chromosome consists of a table with one row per molecule and a user-defined number of columns to encode potential mappings between features in the ligands. The columns are divided by feature type: that is, donors, acceptors, hydrophobes. A full mapping is one comprising a feature in all of the ligands, so that all elements in the column represent real features in the corresponding ligands. In a partial mapping, some of the elements in the column represent "dummy features", that is, they do not correspond to real features and indicate that the corresponding ligands are excluded from that pharmacophore point. A mapping column with fewer than two real features has no physical significance.

A valid chromosome must contain at least three real mappings for every ligand. A chromosome is scored by first building a conformation for each ligand according to the values encoded for its rotatable bonds. The ligands are then aligned incrementally using a framework method. The framework, consisting of $M$ points where there are $M$ mapping columns, is initialized with the first molecule by

setting the coordinates of the points in the framework equal to the coordinates of the real features encoded in the mapping of the first ligand; a dummy feature results in the point in the framework being set to null. Each subsequent ligand is then fit to the framework using a least-squares fitting procedure. The following actions are possible for each feature of a new ligand: (i) If the ligand contains a real feature where previously there were only dummy features, then a new feature is added to the framework; (ii) If the current ligand contains a dummy feature, then there is no change to the framework; (iii) If the feature represents a mapping to an existing feature, then the coordinates of the corresponding point in the framework are modified to be a weighted centroid of the previous point and of the feature in the current ligand (the centroid is weighted toward the existing feature in the ratio $r$:1, where $r$ is the number of molecules already mapped to the feature). The calculation of the new coordinates is carried out using the Kabsch least-squares fitting algorithm.[17] We also require that at least three points in the framework map to all ligands, otherwise the chromosome is rejected.

Once an alignment has been generated, it is scored on each of three objectives: a feature score (estimatimg how well pharmacophore points of similar types are aligned), an energy score (estimating ligand conformational strain) and a volume score (estimating overlap volume). Details of these are provided in the earlier publications.

**Initializing the Population.** When a chromosome is initialized with random values, the absence of a direct link between the two parts of the chromosome can result in mappings and in conformations that are incompatible. In the previous work, an attempt was made to address this issue through the use of distance constraints that were used to prevent solutions that contain infeasible mappings appearing in the initial population, for example, a pair of donor points which are always close (no matter what the molecular conformation) in one molecule cannot match a pair of donors which are always distant in a second molecule. Here we exploit the dependency between mapping and conformation directly by the implementation of a preprocessing step to ensure that the two parts of a chromosome are compatible on initialization. Thus, the fitness of the initial population should be much improved.

There are several instances, in the literature, of improvement in the performance of a genetic algorithm (GA) by use of heuristics to generate an initial population, which is in some sense a 'good' starting point. For example, Puente at al. saw a GA for scheduling problems with heuristic knowledge and reported improvements in performance.[18] Similarly, Todorovski and Rajicic initialize a GA for optimizing electrical power flow using chromosomes which do not violate voltage angle constraints,[19] reporting results of comparable quality in drastically reduced run times.

The preprocessing step implemented here has three aims: (i) to ensure that the alignments encoded in the initial population are good; (ii) to ensure compatibility between the mapping columns and the conformers encoded in the chromosome; and (iii) to ensure that the conformations considered on initialization are energetically reasonable.

The preprocessing step is based on identifying maximal common subgraphs (MCSs) in pregenerated conformers of the ligands. For each ligand, the pharmacophoric features are identified, and a set of conformers is generated with each one represented as a pharmacophore point graph. For each chromosome, clique detection is used to identify an MCS (as detailed below) based on the pharmacophore point graphs of a set of conformers, one for each molecule. The MCS provides a mapping between the corresponding pharmacophore points of the molecules and can be used to align them provided it contains more than three points. An MCS consisting of at least three points is used to initialize the mapping part of a chromosome, and the conformer of each ligand is used to provide the torsion angles for the conformation part of the chromosome. Thus, the two parts of the chromosome are compatible, and the MCS should provide a good starting point for further exploration by the MOGA. This procedure is repeated using different ligand conformers to ensure diversity in the population of chromosomes. The MCS calculation is a widely used method and is often used to compare the atomic graphs of pairs of small molecules.[20,21] Here, an MCS of more than two molecules is generated in a pairwise manner, as detailed below.

**Scheme 1.** Generate a Chromosome

In order to generate a single chromosome member:-
- Pick a molecule $m_1$ and conformer $c_1$ at random (with $n_1$ pharmacophore points).
- Generate initial subset NewS = pharmacophore points of $c_1$.
- While there remain molecules
  - S = NewS
  - Pick a molecule $m_2$ at random, with $n_2$ pharmacophore points.
  - For **each** conformer $c_i$ of $m_2$,
    - FindCliques between $c_i$ and S.
  - *Choose conformer $c_{max}$ from amongst those with largest maximum clique.
  - If maximum clique < 3 Abort.
  - Else
    - Pick a clique, C†, between $c_{max}$ and S.
    - Create new subset S, composed of those pharmacophore points of NewS which are in C.
- End while
- Encode the chromosome mapping columns with the sets of pharmacophore points which have been mapped to S.
- For each molecule, copy the torsion angles from the selected conformer into the relevant portion of the chromosome angles.

* The conformer $c_{max}$ is chosen at random from among the conformers with largest maximum cliques.

† C is chosen using random selection. A size, $n$, is chosen from among the different sizes of clique present, with larger sizes more likely to be chosen. This is achieved using a user-defined linear weighting factor, $f$, with the largest clique size $f$ times more likely to be chosen than the smallest size. (A factor of 1 means all sizes are equally likely to be chosen.) Then a clique, $C$, is chosen randomly from all cliques with $n$ nodes.

The MCSs between the pharmacophore point graphs of a pair of molecules are equivalent to the maximal cliques of their correspondence graph. Note that a graph may have several *maximum* cliques, which are cliques containing the largest possible number of nodes. This number is called the *clique number* of a graph. We also compute all maximal cliques, of at least three nodes, of the graph. A *maximal clique* is a clique which is not contained within a larger clique but does not necessarily contain the maximum number of nodes. The set of all maximal cliques encodes the complete set of feasible overlays of one molecule onto the other.

The algorithm for generating the initial population of chromosomes is outlined in Scheme 1 and is based on the Bron−Kerbosch clique detection algorithm.[22] The first molecule and conformer are chosen at random, and their pharmacophore points are used to define an initial set of features, called the feature subset. A second molecule is chosen, again randomly, and each conformer of the second

molecule is compared with the chosen conformer of the first molecule, using the MCS algorithm and a note made of the clique numbers. A conformer of molecule two is then selected from those conformers with the highest clique number, and a maximal clique is selected. The feature subset is then updated to include only those pharmacophore points that are in the maximal clique. All conformers of the third molecule are then compared with the first molecule, considering only those points remaining in the feature subset; the pharmacophore points in the first molecule are used to approximate the alignment to avoid the time-consuming step of calculating a transformation and of moving the molecules. The subset of pharmacophore points that remains after all molecules have been considered is used to initialize the mapping columns of the chromosome together with the corresponding points in the other molecules. Columns representing features outside the clique are initialized with dummy points and then are filled further in a subsequent rebuilding process, which is described below. The conformer of each ligand used to derive the clique then provides the torsion angles for the conformation part of the chromosome.

It can happen that the conformer selection procedure leads to graphs which have no cliques of at least three points. In this case, the generation of the chromosome is aborted, and a new chromosome is begun.

Finally, the chromosome is checked to see if it is feasible according to distance constraints, as for the previous MOGA. Fewer than 1% of chromosomes are rejected when the approximate mapping does not generate an actual mapping which satisfies the distance constraints. If a chromosome is rejected, then a replacement chromosome is generated as above.

Flexible molecules can have many conformers and many performing-all−against-all clique detection, even for two molecules, may be too time-consuming. We have, therefore, also implemented an option to choose a random subset of conformers for each molecule when building each chromosome. Our experience, as discussed further in this section, is that selecting either all or random subsets of the conformers makes little difference to either the quality or the diversity of the initial population, but it does make a substantial difference to the time taken to generate the population.

As well as random conformer selection, the preprocessing step contains several other random elements: (i) the order in which the molecules are added to the mapping columns is determined at random and so varies between chromosomes; and (ii) the size of clique chosen varies both within and between chromosome generation (see Scheme 1). The fact that population generation remains a largely random (but focused) process is demonstrated by the number of clusters of chromosomes present in the initial population. We cluster the chromosomes based on their full mapping columns. For population sizes of up to 300 individuals (the largest we have needed), we observe that the number of clusters is usually very close to the population size and that the individual chromosomes are mostly singletons and represent different alignments.

**Rebuilding a Chromosome.** The feature score assigned to the alignment encoded by a chromosome is based on explicitly encoded mappings only, so that features that overlay but that are not present in the mapping columns do not contribute to the score and are not identified as features

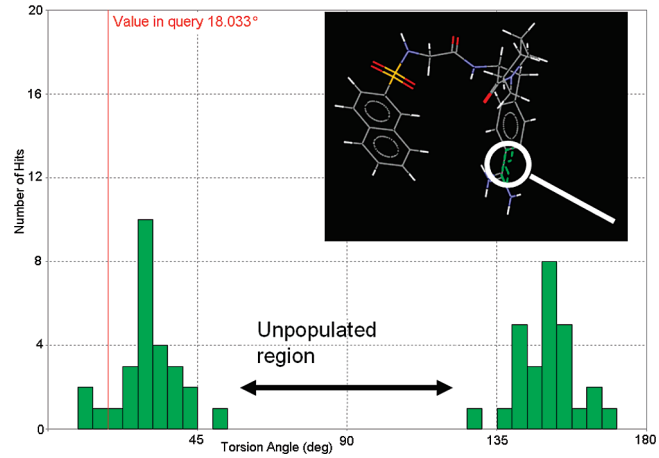


**Figure 1.** MOGUL histogram for selected torsion of thrombin inhibitor, 1dwd.

of the resulting pharmacophore. In such cases, the feature score is an underestimate of the goodness of the alignment. We have, therefore, implemented an option to 'rebuild' the chromosome after the generation of the alignment to include such chance overlays by replacing dummy mapping column entries with genuine pharmacophore points, where possible. This is a two-stage process. First, we try to add points to columns which already contain two or more nondummy entries (light-gray columns in Table 1). Then we try to fill columns which do not represent mappings (the nonshaded mapping columns in Table 1). If the rebuilt chromosome results in a substantially lower feature score than that of the original (since the new mapping may mean that matching points are now moved outside the allowable distance tolerances), then the rebuild is canceled. During a MOGA run, chromosomes can be rebuilt whenever they are altered or all the chromosomes can be rebuilt at user-specified intervals. By default, all chromosomes are rebuilt following initialization in the preprocessing step and are also rebuilt at the end of the run.

**Using Conformational Preferences from MOGUL.** MOGUL provides access to information on molecular geometries present in crystal structures in the Cambridge Structural Database (CSD)[23] and can give valuable insights into the preferred conformations adopted by a molecule.[16] Thus, for a given geometric feature in an input molecule, MOGUL can be used to retrieve all CSD entries that contain the feature with the results displayed as a histogram. For example, Figure 1 shows the histogram of torsion values for the bond highlighted in the thrombin inhibitor, 1dwd, and indicates that this bond is unlikely to adopt an angle in the range of 60−120°. The use of statistical information on conformational preferences to improve the effectiveness of conformational analysis programs is well established.[24,25]

New torsion angles are introduced into the MOGA via the mutation operator, which previously consisted of random mutations only. Now, two different mutation operators have been implemented for the conformational part of the chromosome. In random_mutate, a new angle is chosen at random, as before. In mogul_mutate, a new angle is chosen with a bias toward the preferred conformations of the torsion bond, as given by MOGUL. In mogul_mutate, the replacement angle is chosen from among the values present in the relevant histogram, with a bias toward the more heavily

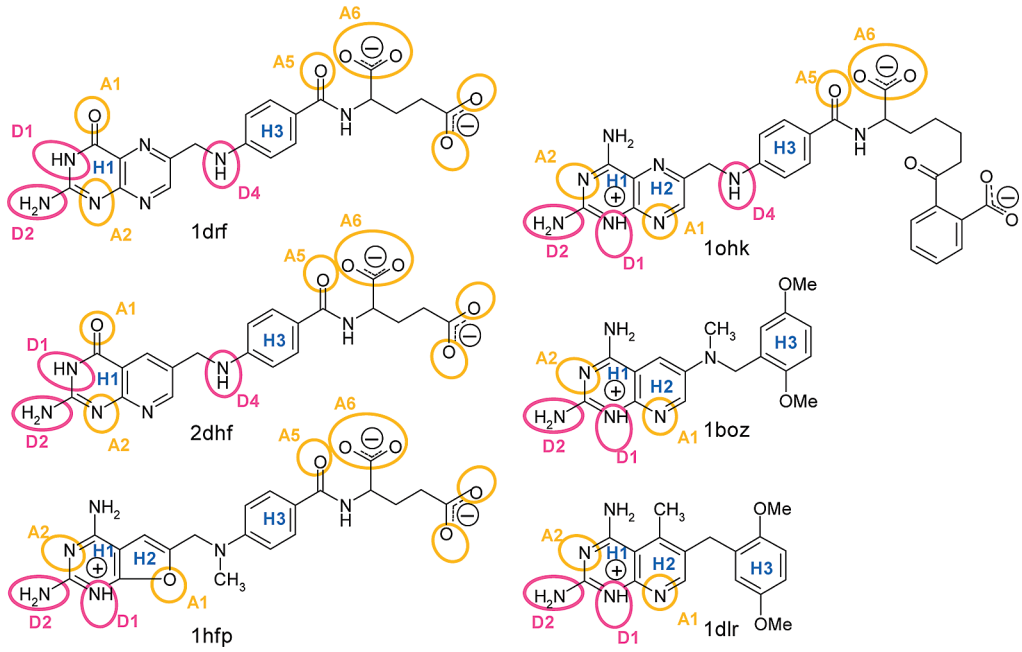


**Figure 2.** The DHFR inhibitors.

populated bins. The use of mogul_mutate is restricted to torsion angles which are sufficiently well represented (the default is 50 occurrences). The likelihood of a particular torsion angle being selected for mutation is dependent on the number of different valued bins it occupies, since there is little benefit in repeatedly mutating a torsion which takes only two preferred values. The ratio of random mutations to MOGUL mutations is user defined (the default ratio is 50:50).

Thus, prior to the MOGA, MOGUL is run for each input molecule with histograms generated for each rotatable bond in each ligand.

**Data.** The performance of the MOGA is demonstrated using three data sets. The first two are taken from Patel et al.[9] and were also used by Richmond et al.[10] in their evaluation of the GALAHAD program. They are a set of six dihydrofolate reductase (DHFR) inhibitors, Figure 2, and a set of six thrombin inhibitors, Figure 3. The third data set is the set of six cyclin-dependent kinase 2 (CDK2) inhibitors used by Cottrell et al.[15] in previous work on the MOGA, Figure 4. To facilitate the discussion, we maintain, where possible, the labeling schemes used by Richmond et al. for the DHFR and thrombin inhibitors (Figures 2 and 3, respectively). The CDK2 molecules are shown in Figure 4. Richmond et al. limited their results to using fixed conformations of the molecules. We also carry out runs using fixed conformations to allow our results to be compared with theirs, and we extend the analysis to consider flexible searches to investigate the effectiveness of our conformational biasing mechanisms.

The DHFR molecules each have a pair of fused aromatic rings which contain a constellation of two hydrogen-bond donors (D1, D2 in Figure 2) and two acceptors (A1, A2). Four of the molecules have a positive-charge center in the same region. Four of the molecules also have one or more negatively charged centers. As in the GASP program, the MOGA treats charged groups as either donors or acceptors, and so the feature alignment will be based only on the hydrogen-bonding features and the rings. One of the challenges for this data set is to overlay the rings correctly so the appropriate donors/acceptors align.

The thrombin data set consists of seven human alpha thrombin inhibitors selected by Patel et al. to represent a diverse data set of small nonpeptidic ligands. Each of the ligands has a region identified by Patel et al. as a basic region, which in the protein−ligand complex makes an interaction with an aspartate residue in the protein. Richmond et al. use this charged region as a potential pharmacophore point. In the MOGA, these regions of the ligands are characterized as having either three (1d9i, 1d6w, 1tom), four (1c4v, 1dwd, 1d49) or five (1fpc) hydrogen-bond donors. Preliminary investigations showed that the MOGA was unable to generate plausible overlays including 1fpc due to it having only a single feature that is common to the remaining six molecules, and so it was omitted from further investigation. Richmond et al. also found 1fpc to be problematic.

The six CDK2 molecules are those used in our earlier development of the MOGA,[15] and the performance of the MOGA on these was part of the motivation for the work described here. With a very large population (2 000 individuals) and a very long run rime (200 000 operations taking 9.5 h on a 3.8 GHz Pentium 4 running Red Hat Enterprise Linux 4), the MOGA was able to generate an overlay that resembled the crystal overlay (with the exception of the ligand 1ke5) although the pharmacophore points generated did not include the acceptor (labeled A1 in Figure 4) that binds to the "hinge" region of the protein.[26] This hinge motif is present in five of the six CDK2 ligands and comprises the points labeled D1 and A1. Note that it is absent from 1ckp.

## RESULTS

We first consider the effect of specific changes to the MOGA in terms of the run times and the ability to generate meaningful pharmacophores. We then analyze the pharamacophore elucidation performance of the MOGA on the three data sets, each comprising six or more molecules.

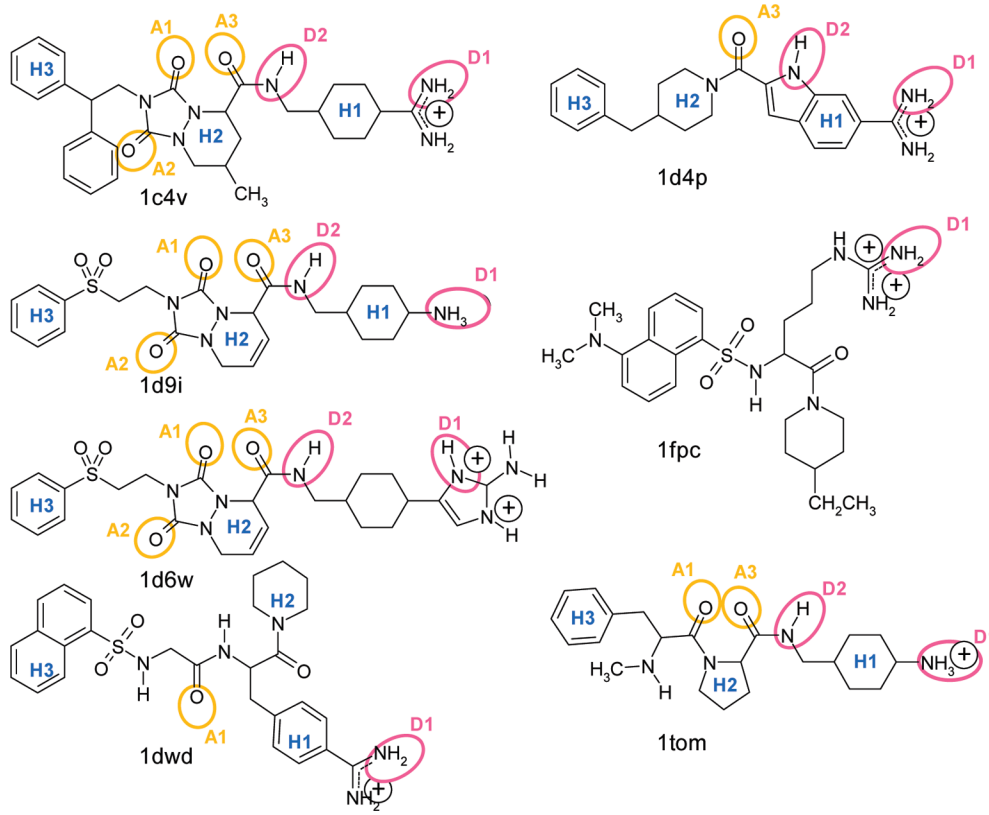


**Figure 3.** Thrombin inhibitors

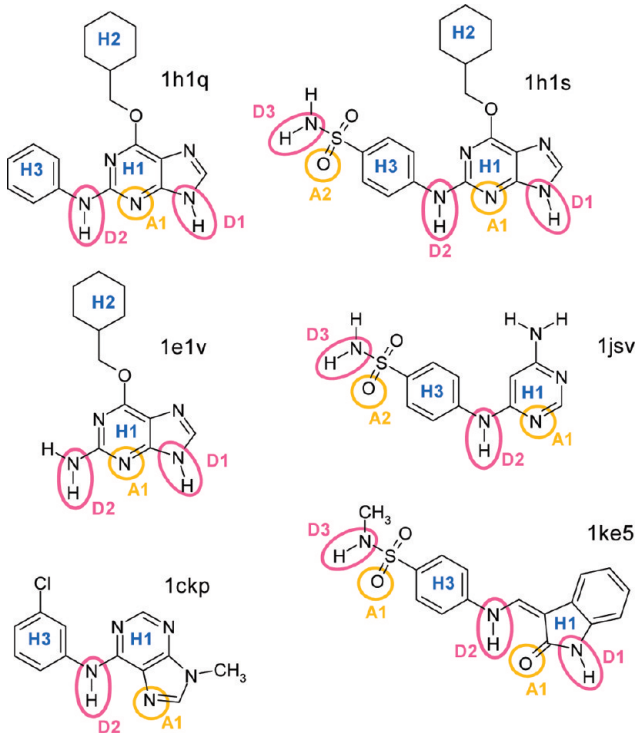


**Figure 4.** CDK2 inhibitors

**Effects of Changes to the MOGA.** First, we made a direct comparison with the previous version of the MOGA (here denoted MOGA1), in order to assess the effect of the preprocessing clique detection on the quality of the initial population (which was previously generated randomly). In addition to the data sets described above, we also used the same data sets we have considered previously.[14] These are four scytalone inhibitors (SCY), three 5HT1D agonists (5HTD), and four carbonic anhydrase inhibitors (CA). In each of these cases, MOGA1 was able to elucidate the known or hypothesized pharmacophores (as is the new MOGA) — our interest here is in the quality of the initial populations. Table 2 shows a comparison of the random chromosomes generated by MOGA1 and by the preprocessing step (MOGA2_P) and the effect of following the preprocessing step by an initial rebuild (MOGA2_PR). The results are averaged over 100 population initializations in which each population consists of 100 chromosomes. In all cases, the feature score is significantly (all $t$ test values $< 8.8 \times 10^{-25}$) higher when the chromosomes are nonrandom. On average, the feature score is improved by almost 9-fold going from random to clique detection chromosome generation and again by 1.2-fold when the chromosomes are rebuilt. In most cases, the volume score is also significantly improved by the preprocessing step but is not further improved by chromosome rebuilding. This is reasonable since a closer alignment of features will not necessarily cause the volumes to be overlaid better. The exception is the thrombin case, where the volume score is significantly worse after the preprocessing step than when using random chromosome generation. This may seem surprising, but it reflects the fact that the initialization step takes no account of the overlaid volume and is based on maximizing the mapping column superposition; therefore, it has most influence on the feature score. The improvement in the DHFR and thrombin feature scores is small but still significant since, when using the original MOGA, the average feature score is negligible (i.e., the molecular features are not sufficiently well overlaid to generate any score at all). Note that, since the energy score can take arbitrarily large

**Table 2.** Effects on the Initial Population[a]

| | MOGA1 | | MOGA2_P | | MOGA2_PR | |
|---|---|---|---|---|---|---|
| data set | *V* | *F* | *V* | *F* | *V* | *F* |
| SCY | 411 (9.1) | 0.09 (0.01) | 613 (12.4) | 0.41 (0.35) | 603 (11.9) | 0.50 (0.40) |
| 5HTD | 421 (3.6) | 0.06 (0.01) | 512 (17.9) | 0.20 (0.28) | 512 (17.7) | 0.25 (0.32) |
| CA | 154 (6.2) | 0.05 (0.008) | 189 (9.9) | 0.33 (0.29) | 189 (9.3) | 0.43 (0.32) |
| CDK2 | 548 (11.8) | 0.01 (0.000) | 583 (12.9) | 0.12 (0.02) | 584 (12.0) | 0.17 (0.02) |
| DHFR | 491 (12.7) | 0.00 (0.00) | 542 (18.0) | 0.08 (0.014) | 542 (19.0) | 0.13 (0.013) |
| thrombin | 809 (21.1) | 0.00 (0.00) | 770 (25.7) | 0.04 (0.01) | 770 (25.6) | 0.07 (0.01) |

[a] MOGA1 indicates randomly assigned chromosomes with no rebuilding. MOGA2_P indicates chromosomes initialized using the clique detection procedure with no rebuilding. MOGA2_PR indicates chromosomes initialized using the clique detection procedure followed by rebuilding. Numbers are mean and standard deviation (in brackets) of average population scores for 100 chromosomes, averaged over 100 population initializations. $V$ = volume in Å$^3$, and $F$ = feature score. The details of the volume and the feature scores are as described in Cottrell et al[15] and are summarized here: the volume score is the mean overlap between the first molecule and each of the other molecules, with each atom considered as a hard sphere. The feature score takes into account the number of pharmacophoric points, the number of molecules that are mapped to each pharmacophore point, and the quality of the overlay.

**Table 3.** Mean Number of Solutions over 100 Runs using 100 Chromosomes and 100 000 Operations per Run[a]

| | MOGA1 | | MOGA2_P | |
|---|---|---|---|---|
| | time (mins) | number of meaningful solutions | time (mins) | number of meaningful solutions |
| CDK2 | 2.1 | 48 | 2.0 | 84 |
| thrombin | 3.5 | 0 | 3.2 | 35 |
| DHFR | 3.3 | 0 | 3.5 | 72 |

[a] Where a meaningful solution is one which has at least three pharmacophore points.

values (either positive or negative), we do not show mean energy scores.

In order to compare the efficiency of our new algorithms relative to MOGA1, we have investigated a number of *meaningful* solutions obtained for a fixed number of operations, using the same population sizes in all cases. We define a meaningful solution to be one which has at least three pharmacophore points, of which two are full (i.e., common to all molecules). In Table 3, we show the number of meaningful solutions for a typical set of runs with a population size of 100 performing 100 000 operations (using all conformers in the preprocessing step). For the original MOGA, (MOGA1) such a run generates no meaningful solutions for the DHFR or the thrombin sets, while the new MOGA (MOGA2_P) generates meaningful solutions for 35% and 72% of the populations, respectively.

The run conditions and the population sizes are standardized in Table 3 for easy comparison. However, for some data sets, smaller populations can be used along with fewer operations. For example, the best results for the CDK2 data set were achieved with a population of only 50 chromosomes run for 5 000−10 000 operations. This takes, in total, less than 30 s both for population generation and MOGA run on a 3.8 GHz linux box (compared with the previous run time of 9.5 h required to give acceptable results).

**Pharmacophore Elucidation.** The MOGA was run in two different modes for each of the three data sets described above. First, the molecules were run with fixed conformations to see how well the MOGA could align the crystal structures. The MOGA preprocessing step was run using the crystal structure conformation as the single 'conformer' for each molecule (all molecules first being randomly reoriented). The initial set of chromosomes, thus, produced had mapping columns generated using clique detection and crystal conformations encoded as the angles. The MOGA was then run in 'fixed conformation' mode, meaning that only the mapping columns were allowed to be altered. Thus, the MOGA is simply trying to overlay the crystal structures based on matching a subset of their pharmacophore points. This mode is very similar to the tests carried out by Richmond et al. in their evaluation of the GALAHAD program.[10] The MOGA was then run in flexible mode, which is the more realistic case, since the bound conformations of the molecules are not usually known. The aim here was to generate a series of alternative hypotheses that are all plausible, and one of which is the true pharmacophore. Conformers for the preprocessing step were generated from CONCORD-generated structures using OMEGA[27] (although in principle any conformer generation program could be used for this step, which is independent of the MOGA itself).

In the discussion which follows, we refer to a pharmacophore point which is present in a subset of molecules as a partial point, thus, we refer to *partial acceptor* and *donor* points.

**Thrombin: Fixed Conformations.** The maximum number of pharmacophore points generated is a user-defined parameter. For these ligands, the MOGA was asked for up to four acceptor, three donor, and four hydrophobic points and was run with a population size of 100 for 20 000 operations, taking approximately 70 s. The crystal overlay is extremely well reproduced. A typical solution produced by the MOGA is shown in Figure 5, along with a view of the crystal structure from a similar viewpoint. We calculate an root-mean-square deviation (rmsd) by treating each overlay as a hypermolecule consisting of all the atoms of the molecules in the overlay and by then superposing the MOGA solution hypermolecule onto the crystal hypermolecule; using this method the heavy-atom rmsd between the MOGA solution and the crystal alignment is 0.53 Å. Notice that the positive-charge center identified in Figure 3 is represented by two hydrogen-bond donor pharmacophore points. In the final population, various different combinations of the possible donors in this region were observed. Three hydrophobic centers and a partial acceptor (present in five of the six molecules and labeled A1 in Figure 3) were also correctly identified. In addition, the MOGA has identified two more partial acceptors, A2 and A3 (present in three and five molecules, respectively). These are not part of the full pharmacophore, but the A2 acceptor atoms will necessarily be overlaid in any alignment which resembles the crystal alignment. The A3 atoms are less closely matched in the crystal overlay, but in the absence of a crystal structure might

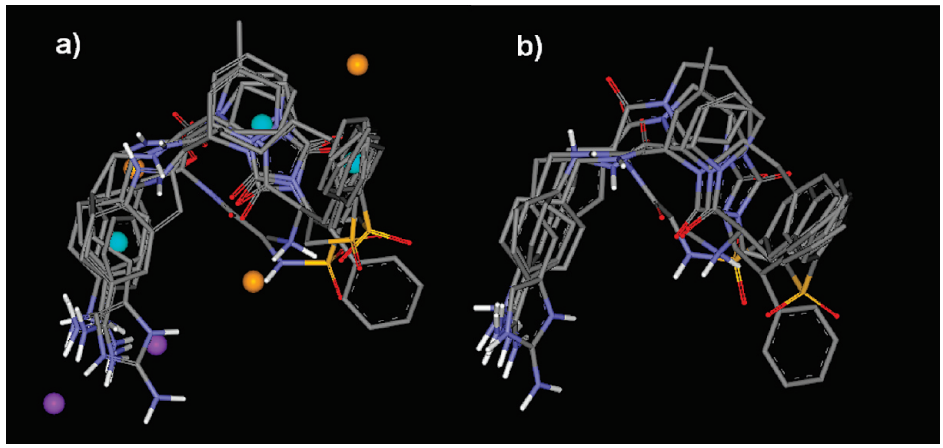


**Figure 5.** Aligning crystal conformations of thrombin data set. (a) Typical alignment produced by MOGA showing the mapped pharmacophore points. (b) A crystal alignment.

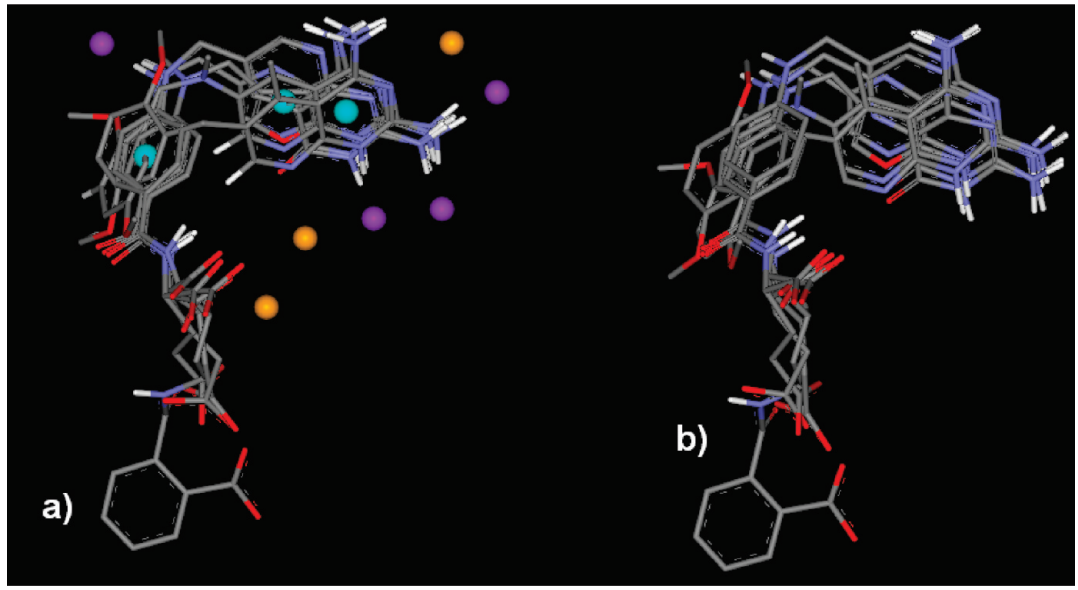


**Figure 6.** Aligning crystal conformations of DHFR data set. (a) Typical alignment produced by MOGA showing mapped pharmacophore points. Donors are purple, acceptors are orange, hydrophobes are cyan. (b) Crystal alignment.

well be deemed part of the pharmacophore. The set of pharmacophore points is very similar to that found by Richmond et al. using GALAHAD. The main difference is that the MOGA does not identify the partial donor point D2. This is because the nitrogen atoms corresponding to this pharmacophore point have not been overlaid sufficiently closely.

**DHFR Data Set. Fixed Conformations.** The MOGA was run with a population size of 100 for 50 000 operations looking for up to six acceptors, four donors, and three hydrophobes, taking approximately 3 min. (NB looking for more pharmacophore features requires more operations and leads to longer run times.) The MOGA was able to find overlays very close to the crystal overlay; one such is shown in Figure 6. The heavy-atom rmsd between the MOGA generated solution and the crystal alignment is 0.49 Å. The hydrophobes H1 and H3 are found in all six ligands, while H2 is correctly identified as a partial pharmacophore present in four of the ligands. The three donor pharmacophore points of the heterocyclic fused rings are all full points. They correspond to the donors D1 and D2 found by GALAHAD, since the MOGA generates points on the donor hydrogens rather than on the nitrogen atom itself, as is the case for GALAHAD. The partial donor D4 is correctly identified in all three molecules in which it appears. Acceptor A2 is also a full point, while A1 is present in only three of the ligands. This reflects the fact that the alignment of the fused rings closely matches that of the crystal structure (rmsd 0.31). The overlay of the carboxylate atoms of the four ligands possessing this feature is also recognized by the MOGA, generating a partial acceptor point A6 which was not found by GALAHAD.

**CDK2 Data Set: Fixed Conformations.** The MOGA was run with a population size of 50 for 10 000 operations, looking for up to four acceptor, two donor, and three hydrophobe points. In this case, the MOGA was unable to reproduce the crystal alignment for all six ligands. In Figure 7, we show a typical MOGA solution which, superficially, looks very similar to that of the crystal overlay, and in fact, the heavy-atom rmsd between the solution and the crystal alignment is 1.8 Å. Five of the ligands are overlaid very well, but 1ckp (which does not have the aliphatic ring) is flipped so that its chlorophenyl ring more or less overlays the aliphatic rings of the other ligands. The main reason for this seems to be that this arrangement allows for a close overlay of the fused rings system present in four of the

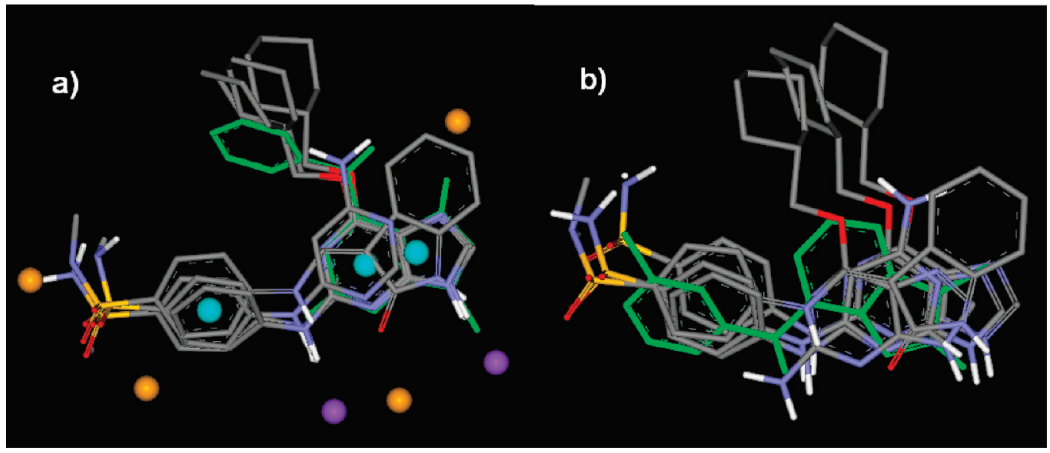


**Figure 7.** Aligning crystal conformations of CDK2 data set. (a) Typical alignment produced by MOGA showing mapped pharmacophore points. Donors are purple, acceptors are orange, hydrophobes are cyan. Misaligned molecule 1ckp is shown in green. (b) Crystal alignment.

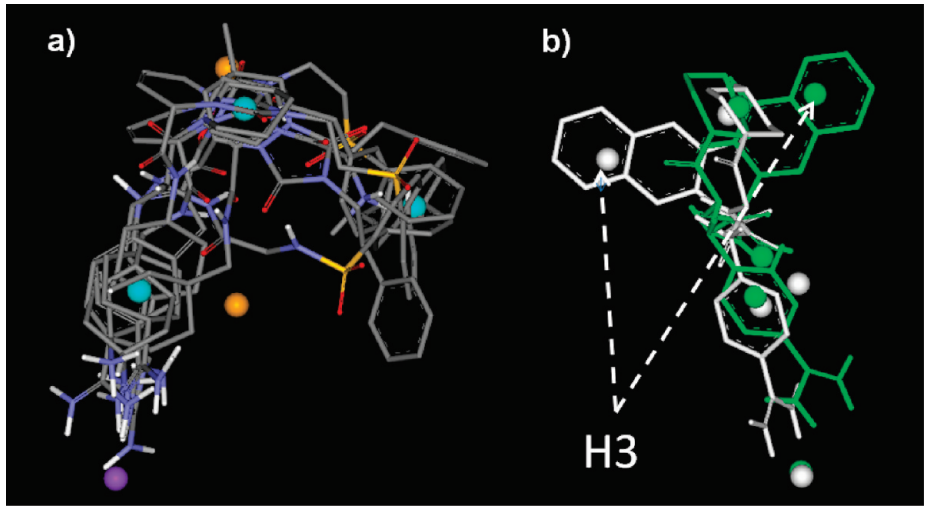


**Figure 8.** Flexible pharmacophore generation for thrombins. (a) Typical alignment produced by the MOGA showing mapped pharmacophore points. Donors are purple, acceptors are orange, hydrophobes are cyan. (b) The crystal (green) and MOGA (white) view of 1dwd in orientation roughly orthogonal to (a). Pharmacophore points in general agreement apart from the hydrophobe H3 are shown by the arrows.

molecules (1h1s, 1h1q, 1e1v, and 1ckp), which is noticeably neater than that of the crystal overlay — the heavy-atom rmsd omitting 1ckp is 0.86 Å. The nitrogen D2 in 1ckp is then aligned with the hinge acceptor pharmacophore point, whereas in the true overlay, 1ckp only has the acceptor of the hinge motif). The correct pharmacophore points have been generated. The acceptor point, A1, and the hydrophobe, H1, are both aligned correctly in all the ligands, as is D2 in all except 1ckp. The hinge donor, D1, has been found by the MOGA in all five ligands which it has correctly aligned. The ligand 1ke5, which the old MOGA failed to align, is now correctly overlaid.

**Thrombin Data Set: Flexible mode.** The thrombin ligands are fairly flexible, with the conformational analysis carried out for the preprocessing step resulting in many input conformations. For example, 94 conformers are generated for the most flexible ligand, 1dwd, using OMEGA with the default parameter settings and using the rms separation between conformers set to 0.8 Å. Therefore, we used random conformer selection in the preprocessing step, selecting 10 conformers of each molecule (making a fresh selection for each chromosome). The best results were obtained when the MOGA was run with a population size of 200 for 60 000 generations, taking about five minutes in total for the population generation and the MOGA run. A typical MOGA solution is shown in Figure 8 above. While being considerably less tidy than either the crystal solution or the fixed conformation solution (Figure 5), it is clearly still acceptably similar to the correct alignment. Two hydrophobes, H1 and H2, and the hydrogen-bond donor D1 are identified as belonging to all six molecules. The remaining hydrophobe and two acceptors are identified as partial matches, largely due to the less precise nature of the alignment. The main difference in the two alignments is that, for all ligands in the MOGA solution, the hydrophobic regions, H3, are rotated approximately 180° from the crystal alignment, while still generating a full pharmacophore point. This is illustrated in Figure 8b using a single ligand for clarity. The differences between the crystal (green) and MOGA (white) view of 1dwd are shown in an orientation roughly orthogonal to that in Figure 8a and also shown are the equivalent pharmacophore points. In the absence of receptor information, this solution is not wrong. There is no good reason for the MOGA to choose the crystal alignment in preference to this one. In general, the MOGA produced very few other plausible alignments for these molecules. Using MOGUL generally gave similar results, but no particular improvement in either alignments or ligand conformations was observed.

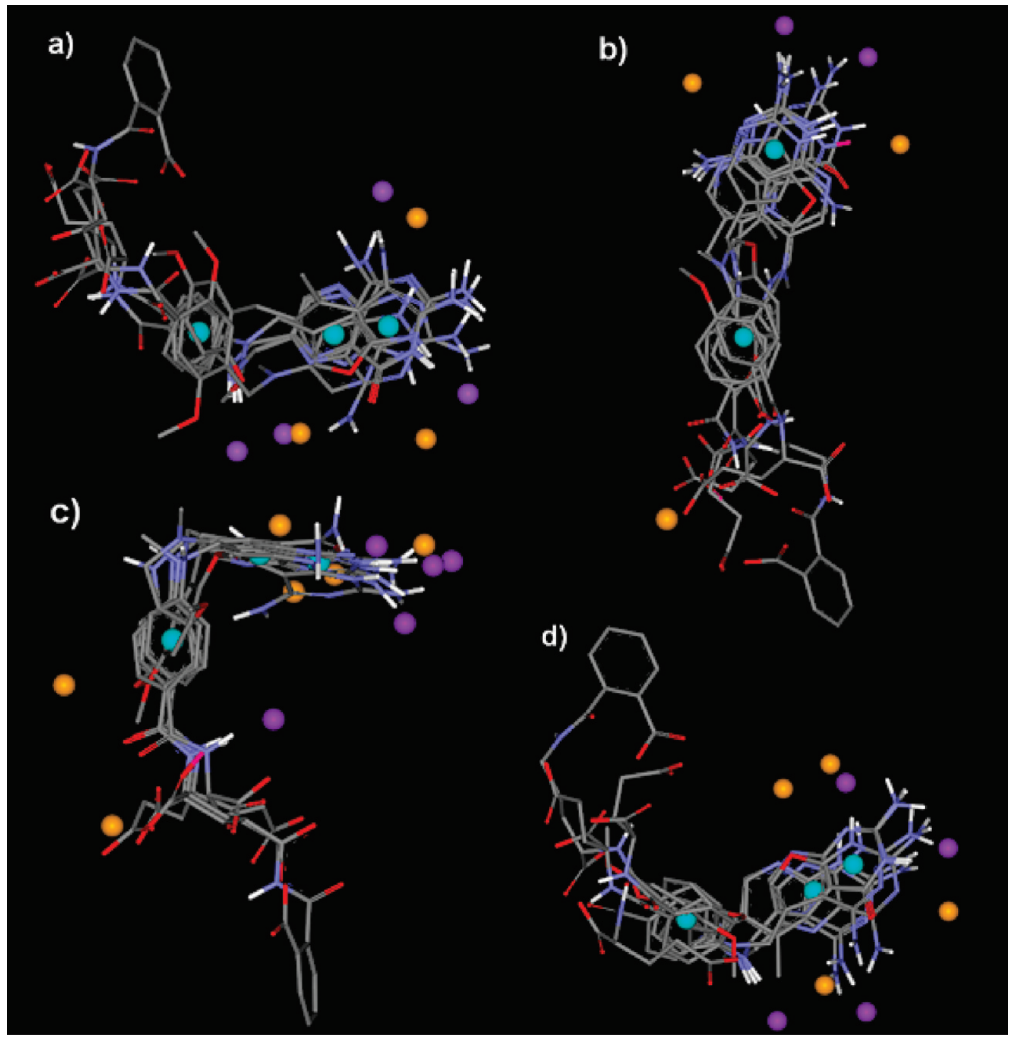


**Figure 9.** Flexible pharmacophore generation for DHFR data set. Donors are purple, acceptors are orange, hydrophobes are cyan. (a−b) Solutions with most molecules overlaid correctly but having one or two molecules flipped with respect to the true alignment. (a) Only 1ohk is flipped. (b) 1hfp and 1ohk are flipped. (c) Fixing 1hfp in crystal conformation means that an alignment and pharmacophore points, very similar to the crystal structure, are generated. (d) Much more extended conformation than crystal structure.

**DHFR Data Set: Flexible Mode.** Some of the DHFR inhibitors are extremely flexible, for example, more than 800 conformers were generated for several of the molecules using the default OMEGA parameters. Even restricting the conformers to those which are more than 1.5 Å apart resulted in 277 conformers for the most flexible ligand, 1ohk. We, therefore, selected 20 conformers at random for each molecule with more than 20 conformers (making a fresh selection for each chromosome) in the preprocessing step. The results shown here were obtained when using a population size of 100 for 50 000 MOGA generations. The preprocessing step and the 50 000 generations took about 70 and 90 s, respectively. The MOGA typically generated alignments based on overlaying the fused ring systems (Figure 9). While the alignments were plausible, the correct overlay of all six molecules was not achieved. For example in Figure 9a, one molecule, 1ohk, is flipped relative to the other five, while in Figure 9b two molecules 1ohk and 1hfp are both flipped. In Figure 9a, the pharmacophore points generated are a large subset of the correct set, with one of the NH donors being omitted. Also, while the alignments are relatively tidy for the rings, the remainder of the molecules are not overlaid nearly as well. We assumed that this was due to their flexibility and tested this hypothesis by using the crystal conformation of one molecule, 1hfp, as its single conformer in the preprocessing step (in a similar manner to that described previously). However, since the MOGA was run in flexible mode, it was allowed to alter this conformation. Under these conditions, an alignment of all six molecules very similar to the crystal structure was found (Figure 9c). The MOGA also had a tendency to generate more extended conformations than the crystal structure (Figure 9d). For the DHFR molecules, using MOGUL angles for mutation did not generally result in any improvement to the overlays generated.

**CDK2 Data Set: Flexible Mode.** Using the clique detection preprocessing step to initialize the starting population produced good results for the CDK2 data set. The best results were obtained using only a small population (50 members) for only a relatively very small (5 000−10 000) number of operations, taking less than 30 s in total (as compared with the population of 2 000 and 200 000 operations reported in Cotrrell et al.).[15] Under these conditions, the MOGA usually produced several alternative plausible alignments. Figure 10a−c shows such a selection. In Figure 10a, the crystal alignment is reproduced very well except for 1ke5 (shown in green), which, although overlaid with the other molecules in the set, is flipped relative to the crystal

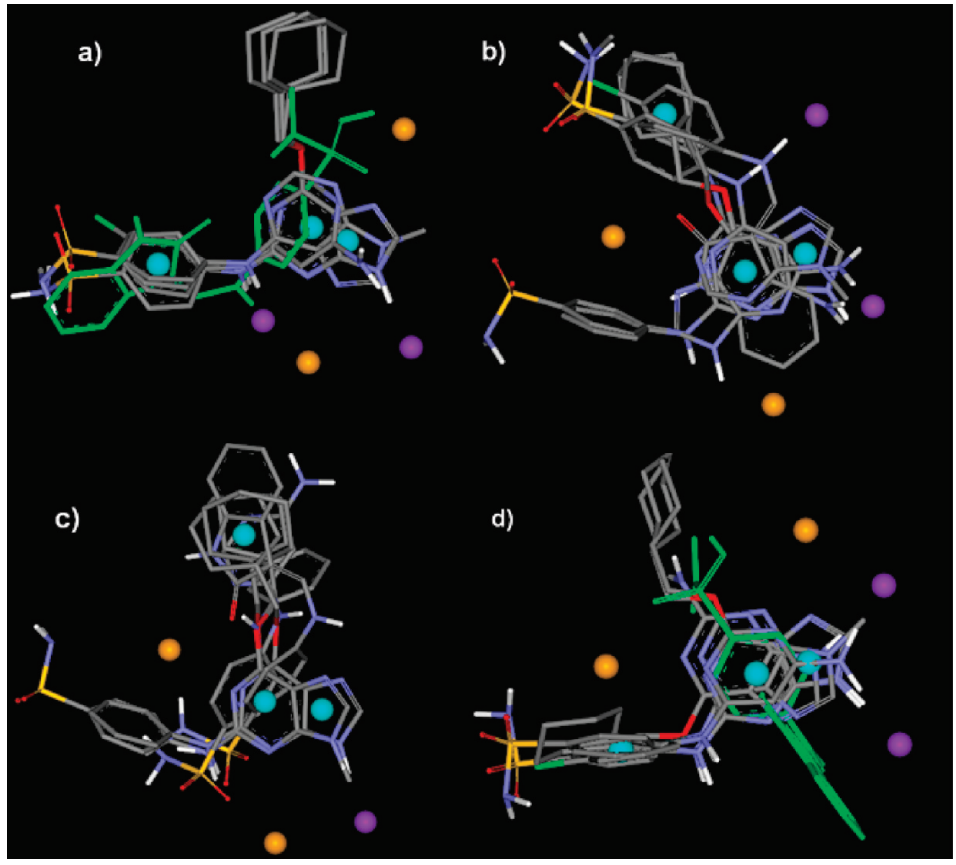


**Figure 10.** Flexible pharmacophore generation for CDK2s. Donors are purple, acceptors are orange, hydrophobes are cyan. Solutions a−c were generated using the MOGA with MOGUL conformational bias. (a) Alignment that is very similar to the crystal alignment, with the exception of 1ke5 (green) and that includes the crystal structure pharmacophore. (b and c) Alternative alignments generated in the same run. (d) Without MOGUL, the MOGA was mostly unable to include 1ke5 (green) in any plausible alignment.

alignment. The hinge pharmacophore, A1 and D2 is also generated together with the three hydrophobes, H1−H3, and both a partial acceptor and donor point. Interestingly, 1ckp, which proved problematic in the fixed case, is aligned here correctly. Alternative hypotheses are shown in Figure 10b and c. For these CDK2 runs, the best results were obtained using MOGUL conformations in 30% of angle mutations (as described in the Methods Section). The MOGA was unable to fit 1ke5 to the alignment using randomly chosen angles in these very short run times. Figure 10d shows a typical solution generated without using MOGUL. Not only is the bulk of 1ke5 not overlaid on the other molecules but also 1h1q is flipped so that its aliphatic ring does not overlay the other two aliphatic rings. The full pharmacophore points include two acceptors, which are not part of the true pharmacophore, and the hinge pharmacophore is not found at all.

## CONCLUSIONS

We have presented two different mechanisms for biasing the alignments toward favorable conformations within our MOGA. The implementation of a preprocessing step to set up a promising initial population is the most effective and has resulted in run times being reduced from several hours to a few minutes (up to five for the data sets explored here), which in turn has allowed us to handle larger and more diverse data sets. A potential drawback of this approach might be a loss of diversity, with the MOGA unable to explore a wide variety of solutions; however, this does not appear to be the case presumably because of the high degree of random elements that are still present during population set up. The main advantage would seem to be directly linking the two parts of the chromosome, which increases the number of feasible solutions so that the MOGA is no longer spending as much time trying to fit infeasible conformers onto an alignment that cannot possibly exist. The preprocessing step is based on precomputed conformers; however, the solutions are not limited to those that exist in the initial population, since the MOGA permits torsion angles to vary during the run.

The second mechanism for conformation bias is introduced through the use of torsion angle distributions, as seen in MOGUL. This resulted in improved (i.e., more crystal-like) conformations for the CDK2 data set but no consistent improvement for either the thrombin or DHFR molecules. One possible reason for this apparently surprising result is that torsion angles present in the crystal structure (which was extracted from coordinates deposited in the PDB) may not be considered likely (or indeed may not be found at all) in the MOGUL database, which has been extracted from data present in the CSD. For example in the DHFR data set, the MOGA was able to align the fused heterocyclic rings but was unable to align the carboxylate tails. Use of MOGUL torsions was not able to improve this result, since many of the 'true' angles were deemed unlikely based on the MOGUL preferences. Thus, for ligand 1ohk, of the seven main-chain torsion angles between the terminal ring of the 'tail' and the next ring, all of which are sufficiently well-represented in

the MOGUL database to be used by the MOGA, in only two cases is the crystal angle considered favorable. (In four cases no instance of the crystal angle is found at all, and in the fifth, it is the least numerous of all the angles present in the database). Similar comments can be made about other DHFR ligands and also some of the thrombin ligands. Since the MOGUL database has been extracted from high-resolution small-molecule crystal structures, it seems probable that the 'failure' of the incorporation of crystallographic information into the MOGA to improve solution quality reflects inaccuracies in the very crystal alignments we are trying to reproduce.

Our results compare favorably with those reported by Richmond et al. for the DHFR and thrombin data sets based on fixed conformations (where a full conformational exploration is not carried out). Furthermore by biasing conformations, we have been able to obtain acceptable results for the more realistic flexible runs. However, it should be noted that one of the difficulties in analyzing the results of a pharmacophore elucidation program is the definition of success. When a target alignment exists, for example, based on a set of ligands extracted from the PDB, as considered here, then starting from the minimized conformers, a program should ideally generate the bioactive conformations aligned exactly as in the crystal structures with the correct set of features identified. However, useful results can also be obtained even if this *gold standard* is not achieved. This was highlighted in the Patel et al. study which included several different criteria for success: the rmsd between the hypothesis and target pharmacophore; the number of misses (either a missing feature or the wrong function group assigned to a feature); and a more qualitative judgment. In the absence of a clear set of objective criteria for validation, the results presented here have been analyzed qualitatively as was done by Richmond et al. However, we are currently developing a more objective way of analyzing the results of pharmacophore elucidation programs, which will be reported in a future publication and which we hope will become adopted by other developers in this area.

A further factor which may be limiting progress in pharmacophore elucidation programs is the sparsity of standard data sets for their validation. The development of such data sets has lagged behind those currently available for protein−ligand docking and ligand-based virtual screening validations.[28−30] One reason for this is that less data is available, since the determination of a single target pharmacophore requires several diverse ligands bound to the same protein. Another is that it is much more time-consuming to develop a high-quality data set since typically this requires a degree of visual analysis and validation of the crystal structures themselves. An early attempt to derive a standard data set was made by Patel et al., who identified 5 data sets with up to 10 ligands. This data set proved very challenging for programs that existed at that time and has been used subsequently by several groups, including ourselves in the development of improved methods. However, the small size of the data set means that there is a real risk that programs will be over trained on it. We are currently working on building a larger test set, which we expect will challenge the limits of current programs and will eventually lead to significant progress being made in what remains a challenging task.

## ACKNOWLEDGMENT

This work was part-funded by AstraZeneca and by the Cambridge Crystallographic Data Centre. We gratefully acknowledge the support provided by Openeye in the provision of OMEGA and Bob Clark for providing us with corrected Patel data sets and by Jason Cole and Simon Cottrell at the Cambridge Crystallographic Data Centre for advice and for programming support.

## REFERENCES AND NOTES

(1) Güner, O. F. *Pharmacophore Perception, Development and Use in Drug Design*; International University Line: La Jolla, CA, 2000.
(2) Wolber, G.; Langer, T. LigandScout: 3-D pharmacophores derived from protein-bound ligands and their use as virtual screening filters. *J. Chem. Inf. Model.* **2005**, *45*, 160–169.
(3) Wolber, G.; Seidel, T.; Bendix, F.; Langer, T. Molecule-pharmacophore superpositioning and pattern matching in computational drug design. *Drug Discovery Today* **2008**, *13*, 23–29.
(4) Dammkoehler, R. A.; Karasek, S. F.; Shands, E. F. B.; Marshall, G. R. Constrained Search of Conformational Hyperspace. *J. Comput.-Aided Mol. Des.* **1989**, *3* (1), 3–21.
(5) Martin, Y. C.; Bures, M. G.; Danaher, E. A.; Delazzer, J.; Lico, I.; Pavlik, P. A. A Fast New Approach to Pharmacophore Mapping and Its Application to Dopaminergic and Benzodiazepine Agonists. *J. Comput.-Aided Mol. Des.* **1993**, *7* (1), 83–102.
(6) Jones, G.; Willett, P.; Glen, R. C. A genetic algorithm for flexible molecular overlay and pharmacophore elucidation. *J. Comput.-Aided Mol. Des.* **1995**, *9* (6), 532–549.
(7) Barnum, D.; Greene, J.; Smellie, A.; Sprague, P. Identification of common functional configurations among molecules. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 563–571.
(8) Berman, H. M. The Protein Data Bank: a historical perspective. *Acta Crystallogr., Sect. A: Fundam. Crystallogr.* **2008**, *64*, 88–95.
(9) Patel, Y.; Gillet, V. J.; Bravi, G.; Leach, A. R. A comparison of the pharmacophore identification programs: Catalyst, DISCO and GASP. *J. Comput.-Aided Mol. Des* **2002**, *16* (8−9), 653–681.
(10) Richmond, N. J.; Abrams, C. A.; Wolohan, P. R. N.; Abrahamian, E.; Willett, P.; Clark, R. D. GALAHAD: 1. Pharmacophore identification by hypermolecular alignment of ligands in 3D. *J. Comput.-Aided Mol. Des.* **2006**, *20* (9), 567–587.
(11) Feng, J.; Sanil, A.; Young, S. S. PharmID: Pharmacophore identification using Gibbs sampling. *J. Chem. Inf. Model.* **2006**, *46*, 1352–1359.
(12) Dixon, S. L.; Smondyrev, A. M.; Knoll, E. H.; Rao, S. N.; Shaw, D. E.; Friesner, R. A. PHASE: a new engine for pharmacophore perception, 3D QSAR model development, and 3D database screening: 1. Methodology and preliminary results. *J. Comput.-Aided Mol. Des.* **2006**, *20* (10−11), 647–671.
(13) Martin, Y. C. Pharmacophore Modeling: 1 - Methods. In *Comprehensive Medicinal Chemistry II*, Triggle, D. J.; Taylor, J. B., Eds. Elsevier: Oxford, U.K., 2006; Vol. 4, pp 119−147.
(14) Cottrell, S. J.; Gillet, V. J.; Taylor, R.; Wilton, D. J. Generation of multiple pharmacophore hypotheses using multiobjective optimization techniques. *J. Comput.-Aided Mol. Des.* **2004**, *18* (11), 665–682.
(15) Cottrell, S. J.; Gillet, V. J.; Taylor, R. Incorporating partial matches within multiobjective pharmacophore identification. *J. Comput.-Aided Mol. Des.* **2006**, *20* (12), 735–749.
(16) Bruno, I. J.; Cole, J. C.; Kessler, M.; Luo, J.; Motherwell, W. D. S.; Purkis, L. H.; Smith, B. R.; Taylor, R.; Cooper, R. I.; Harris, S. E.; Orpen, A. G. Retrieval of crystallographically-derived molecular geometry information. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 2133–2144.
(17) Kabsch, W. Solution for Best Rotation to Relate 2 Sets of Vectors. *Acta Crystallogr., Sect. A: Cryst. Phys., Diffr., Theor. Gen. Cryst.* **1976**, *32* (SEP1), 922–923.
(18) Puente, J.; Vela, C. R.; Prieto, C.; Varela, R. Hybridizing a genetic algorithm with local search and heuristic seeding. In *Lecture Notes in Computer Science*, Mira, J.; Alvarez, J. R., Eds. Springer: Berlin/Heidelberg, Germany, 2003; Vol. 2687, pp 329−336.
(19) Todorovski, M.; Rajicic, D. An initialization procedure in solving optimal power flow by genetic algorithm. *IEEE Trans. Power Sys.* **2006**, *21* (2), 480–487.
(20) Barker, E. J.; Buttar, D.; Cosgrove, D. A.; Gardiner, E. J.; Kitts, P.; Willett, P.; Gillet, V. J. Scaffold hopping using clique detection applied to reduced graphs. *J. Chem. Inf. Model.* **2006**, *46*, 503–511.
(21) Gardiner, E. J.; Artymiuk, P. J.; Willett, P. Clique detection algorithms for matching three-dimensional molecular structures. *J. Mol. Graphics Model.* **1997**, *15* (4), 245–253.
(22) Bron, C.; Kerbosch, J. Finding All Cliques of an Undirected Graph. *Commun. ACM* **1973**, *16* (9), 575–577.

(23) Allen, F. H. The Cambridge Structural Database: a quarter of a million crystal structures and rising. *Acta Crystallogr. Sect. B: Struct. Sci.* **2002**, *58*, 380–388.
(24) Sadowski, J.; Bostrom, J. MIMUMBA revisited: Torsion angle rules for conformer generation derived from X-ray structures. *J. Chem. Inf. Model.* **2006**, *46*, 2305–2309.
(25) Strizhev, A.; Abrahamian, E. J.; Choi, S.; Leonard, J. M.; Wolohan, P. R. N.; Clark, R. D. The effects of biasing torsional mutations in a conformational GA. *J. Chem. Inf. Model.* **2006**, *46*, 1862–1870.
(26) Noble, M. E. M.; Endicott, J. A.; Johnson, L. N. Protein Kinase Inhibitors: Insights into Drug Design from Structure. *Science* **2004**, *303* (5665), 1800–1805.
(27) *OMEGA*, Version 2.1.0; OpenEye Scientific Software: Santa Fe, NM.
(28) Hartshorn, M. J.; Verdonk, M. L.; Chessari, G.; Brewerton, S. C.; Mooij, W. T. M.; Mortenson, P. N.; Murray, C. W. Diverse, high-quality test set for the validation of protein-ligand docking performance. *J. Med. Chem.* **2007**, *50* (4), 726–741.
(29) Verdonk, M. L.; Mortenson, P. N.; Hall, R. J.; Hartshorn, M. J.; Murray, C. W. Protein-Ligand Docking against Non-Native Protein Conformers. *J. Chem. Inf. Model.* **2008**, *48*, 2214–2225.
(30) Huang, N.; Shoichet, B. K.; Irwin, J. J. Benchmarking sets for molecular docking. *J. Med. Chem.* **2006**, *49* (23), 6789–6801.

CI9002816