GAPE: An Improved Genetic Algorithm for Pharmacophore Elucidation

Gareth Jones*

Arena Pharmaceuticals, 6166 Nancy Ridge Drive, California 92121, United States

Received May 17, 2010

Prediction of the binding mode for a series of active compounds, in the absence of known protein structure, is a problem of paramount importance in rational drug design. GAPE (genetic algorithm for pharmacophore elucidation) is an automated multicompound overlay creation program, based on the original GASP program, that uses a genetic algorithm to fully explore the conformational space of the input structures and their alignments, so as to elucidate a pharmacophore. The software was evaluated on 13 test systems from nine protein targets using overlaid ligands extracted from the PDB. Using objective rmsd criteria and starting from 2D structures, in the absence of any protein information, GAPE was observed in eight systems to approximate the crystallographically observed binding mode. In the predicted alignments for each of those eight systems, at least half the input structures were within 2 Å rmsd of the crystal structure coordinates. Further analysis, using stricter subjective criteria, showed considerable success in five systems. For example, the prediction for a set of 12 ligands targeting P38 had 11 ligands with a 1.8 Å rmsd to crystal structure coordinates. Finally, the algorithm was favorably compared with the current GASP and Galahad programs.

INTRODUCTION

The analysis of ligands in the binding site of an experimentally determined target structure has helped drive the design and development of many drugs.[1] The computational exploitation of such an analysis has been an important tool in drug design and development.[2] However, in the absence of a high-resolution protein/ligand cocrystal structure, important interactions between the ligand and target must be inferred either from homology modeling of a high similarity target or by the generation of binding hypotheses based on the integration of experimental medicinal chemistry results. The arrangement of chemical features in three-dimensions that is required for a small molecule to interact with a protein is known as a pharmacophore.[3] The purpose of the method described in this paper was to elucidate the pharmacophore from a series of compounds, which are assumed to bind to the same active site of a protein, in the absence of structural information of the protein. In the process of elucidating the pharmacophore, a multiple alignment of the input structures was also created.

Using a set of input molecules, many tools encompassing a wide variety of algorithms have been developed to align active molecules and generate pharmacophore hypotheses.[4,5] In particular, several of these methods are capable of performing alignments of multiple ligands, such as Surflex-Sim,[6] DISCO,[7] Catalyst,[8] GALAHAD,[9] PharmID,[10] PHASE,[11] and others.[12−14] This is a computationally demanding procedure requiring the combined search of ligand conformational space and available superpositions. Genetic algorithms[15,16] (GAs) appear to be suitable for sampling this large search space and have been utilized by a number of methods.[9,14,17] Of particular interest is the program GASP, which uses a GA to explore conformational space and possible superpositions[18] and has been compared favorably to DISCO and Catalyst.[19] Since the publication of the original GASP program, the algorithm has been further developed into a multiobjective genetic algorithm (MOGA), which allows the creation of an ensemble of plausible overlays or pharmacophore hypotheses.[20,21] Improvements in conformational sampling have been made to both the original GASP program[22] and the later MOGA.[23] This paper describes a different a set of improvements that have been made to the original GASP algorithm.

The original GASP algorithm suffered from some important limitations.[18] First, the algorithm required that every molecule contribute to each pharmacophore point. This limits the algorithm to carefully chosen data sets where each molecule contains the entire pharmacophore.[19] Second, the algorithm suffered from the problem of creating valid overlays; in many instances, a chromosome was unable to create an alignment, as there were fewer than three reasonable mappings encoded between a pair of structures. This resulted in great difficulty in creating an initial population in the GA; it prevented the algorithm processing more than four to six input structures and meant that many mutation and crossover operations were wasted, as it was not possible to decode the child chromosomes into overlays. Third, the feature model used in GASP was inconsistent with that observed in small molecule crystal structures. For example, an sp^2 acceptor in GASP was modeled to accept linearly in the directions of its two lone pairs; in fact, such acceptors tend to accept in the plane between the lone pairs.[24] Fourthly, the similarity terms used to score the overlap of features from different molecules was somewhat arbitrary. Fifthly, the performance of GASP was highly dependent on correct atom typing of input structures. Sixthly, the pharmacophore features were all extracted from a designated "base" molecule. Finally, the GASP program was originally developed in tandem with the docking program GOLD,[25−27] with which it shared many features. Following publication, further

* gjones@arenapharm.com.

10.1021/ci100194k © 2010 American Chemical Society
Published on Web 10/26/2010

development of GASP with applicable GOLD improvements was halted. There were many later improvements to GOLD that would have benefited GASP, for example, including restrictions on the values of common torsions, so that they conformed to values observed in small molecule crystal structures; extending the feature mapping portion of the chromosome to include hydrophobic atoms; rewriting the feature mapping portion of the chromosome to exclude (by setting that mapping to a dummy value) those associations that were not satisfied in the created overlay, followed by a rebuilding of the overlay; annealing scoring function parameters so as to permit "fuzzy" solutions early in the GA run; and replacing the Gray coding[16] scheme for binary strings with a binary encoding.

GAPE (genetic algorithm for pharmacophore elucidation) is a new program based on the original GASP tool. Significant enhancements have been made, addressing all seven issues identified above. The scoring function has been altered to remove the restriction that every input structure must contribute to each pharmacophore point. This partial matching was different from the partial matching introduced by Cottrell et al. for MOGA.[21] In GAPE, partial matches are identified in the scoring function, whereas in MOGA, partial matches are coded into the GA chromosome. The resulting pharmacophore features were created from any structure in the overlay; no "base" molecule dependency existed. The scoring of hydrogen-bonding features was changed to reflect the geometries and group probabilities of donors and acceptors as observed in small molecule crystal structures.[24] The shape overlay term in the scoring function was changed to use a common volume term based on atomic Gaussians.[28] GAPE was engineered to read SD files,[29] hydrogens were added as needed, common acids and bases were solvated as appropriate, formal charge centers were identified, and aromaticity perception was performed. All the improvements from GOLD that were identified above were incorporated into the new tool.

Traditionally, molecular alignment programs can be divided into those, such as GASP, that evaluate molecular conformations on-the-fly and permit a near-continuous distribution of conformational space and those, such as Catalyst,[8] that use a number of rigid conformers to represent the conformational space of each ligand. GAPE was able to perform a conformational search by generating molecular conformations on-the-fly or by using a conformer library. In the second case, an encoding similar to that used by DIFGAPE[17] or MEGA-SQ[14] was used to select individual conformers from a library for inclusion into the overlay. By using a GA to select conformers, the conformer library may include many thousands of conformers per structure. It was hoped that the use of conformational libraries would have the advantages of increased speed and would improve algorithmic performance both by reducing the size of the search space and restricting the search space to conformationally accessible regions. In the experiments to validate the use of conformer libraries reported here, the conformer generator Omega from Openeye scientific software[30] was employed (though any conformer generator could have been used). The version of the GAPE program that performed on-the-fly conformational generation will henceforth be referred to as "regular GAPE" and the version that performs conformational search using a conformer library will be referred to as "multiconformer GAPE".

It transpired that the inclusion of hydrophobic atoms in the mapping chromosome and changes to the fitting process were sufficient to prevent bad chromosome creation from becoming an issue as it was in GASP. Coupled with the new partial matches permitted in the pharmacophore, this allowed running the algorithm on much larger sets of compounds.

The author and co-workers have recently used a data set of 13 test systems extracted from the Protein Data Bank (PDB)[31] to validate an unrelated superposition algorithm.[17] The 13 test systems came from nine different protein targets and each contained four to 13 ligands. Using these systems enabled stringent validation of GAPE predictions (determined without any protein structure information) by comparing such predictions to crystallographically observed binding modes. These predictions were validated using two methods: first, using an objective criteria whereby a prediction was considered a success if half or more of the input structures could be fitted to the crystal structure overlay with an accuracy of 2 Å heavy atom rmsd (root-mean-square deviation) or better and, second, a strict subjective inspection of the overlay was performed.

Gardiner et al. point out the difficulty in defining success for pharmacophore alignments.[23] However, for systems where an experimental alignment is available, they identify a *gold standard* whereby a program should generate the bioactive conformations aligned exactly as in the crystal structures together with the correct set of features identified. The objective criteria was an attempt to identify if an alignment was close to the *gold standard*. It did not verify that the correct set of features was identified; the assumption was that, if a majority of structures were correctly aligned, a correct pharmacophore was likely identified.

It should be noted that a GAPE prediction was the best scoring overlay created from a GAPE experiment; it was possible, indeed likely, that there were many overlays created in the experiment that were closer to the observed binding mode than the reported prediction.

Both the regular GAPE program and the multiconformer GAPE program were evaluated against these test systems. Following this, a number of shorter experiments were performed to evaluate the consistency and reproducibility of the results. Next, the applicability of the algorithm in identifying multiple binding modes was investigated and an analysis of GA runs was performed. Finally, the performance of the algorithm was compared to the versions of GASP and Galahad included in SYBYL-X 1.1.[32]

MATERIALS AND METHODS

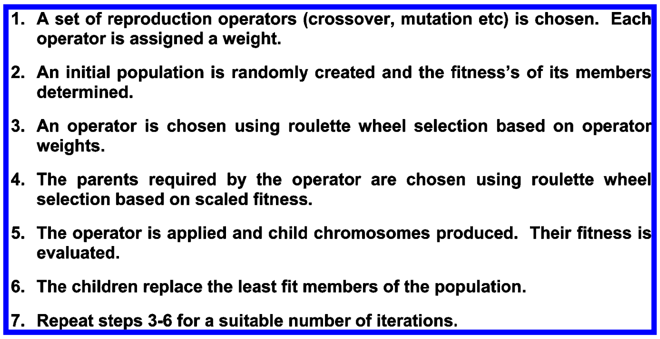
Selection of Protein−Ligand Complex Test Systems. The algorithm was validated by comparing GAPE predictions with experimentally observed overlays. The experimental systems used in the validation of DIFGAPE,[17] were also used to evaluate GAPE. These comprised 13 test data sets of protein−ligand systems, extracted from the PDB[31] and ranging in size from four to 13 ligands (Table 1), which are freely available from the Supporting Information.[17] Ten of the test systems were chosen from protein−ligand complex data sets (CDK2, elastase, ESR1, HIV-1 protease, p38, rhinovirus, and trypsin), described in a previous analysis of

Table 1. Protein Targets and Ligand Data Sets

data set	target	protein family	ligand count
CDK2_Focused	CDK2	transferase (kinase)	9
CDK2_Diverse			10
DHFR	DHFR	oxidoreductase	12
elastase	elastase	hydrolase (serine proteinase)	5
ESR1	ESR1	hormone receptor	13
FXa_Focused	FXa	hydrolase	11
FXa_Diverse			8
HIV_Div	HIV1 protease	hydrolase (acid protease)	13
HIV_Div_MW			8
HIV_Div_MW_RB			4
P38	p38	transferase (kinase)	12
rhinovirus	rhinovirus	virus	8
trypsin	trypsin	hydrolase	7

molecular overlay software.[33] To complement this selection, one test system from dihydrofolate reductase (DHFR) complexes present in the PDB and two test systems from factor Xa (FXa) complexes present in the PDB were added. The DHFR test set is a superset of the DHFR data set used by Patel et al.[19] Given the size and complexity of the search space, GAPE was not run on systems containing more than 13 ligands. For those test systems that had more than 13 ligands, data reduction techniques were used to create subsets.[17] For the CDK2 and FXa targets, there were sufficient ligands of the same compound class to create focused data sets, reminiscent of compounds within an SAR series.[17] Reference crystallographic alignments for the ligands in the test sets are also available in the Supporting Information.[17]

Genetic Algorithms. A GA is a computer program that mimics the process of evolution by manipulating a collection of data structures called chromosomes. Each of these structures encodes a possible solution (i.e., a possible pharmacophore hypothesis and molecular overlay) to the pharmacophore elucidation problem and may be assigned a fitness score based on the relative merit of that solution. A steady-state operator based GA[16] was used to simultaneously explore the conformational space of a set of active compounds and the pharmacophores identified by the overlay of those conformations. This GA is shown in Figure 1. The algorithm is similar to the GASP software already described.[18] However, there have been many significant improvements and developments, including techniques used in the docking program GOLD.[25,26] Two methods are available for the exploration of conformational space: the GA can encode torsion angles directly in the chromosome and create conformers on-the-fly from a reference structure, or a conformer library can be utilized, in which case the GA chromosome contains a conformer number for each structure.



1. A set of reproduction operators (crossover, mutation etc) is chosen. Each operator is assigned a weight.
2. An initial population is randomly created and the fitness's of its members determined.
3. An operator is chosen using roulette wheel selection based on operator weights.
4. The parents required by the operator are chosen using roulette wheel selection based on scaled fitness.
5. The operator is applied and child chromosomes produced. Their fitness is evaluated.
6. The children replace the least fit members of the population.
7. Repeat steps 3-6 for a suitable number of iterations.

Figure 1. Operator-based GA.

In the following sections, the regular GAPE is described. Later, the modifications required for the multiconformer GAPE are detailed.

Initialization of the Active Structures. Input structures were extracted from the test systems used previously.[17] The structures were reduced to 2D smiles then converted to 3D using the program Omega from Openeye.[30]

GAPE was able to read input structures in both SDF and MOL2 file formats. The program automated much of the preparation of input structures. First, valence was filled by adding hydrogens (respecting formal charge settings in SDF or MOL2 files) and aromaticity was assigned on the basis of Hückel's rule. Next, groups were ionized as appropriate and charged groups were identified. Ionizable groups were defined using query SLN patterns[34] and included common acids (the predefined acids were carboxyls, hydroxamates, tetrazoles, acylsulfonamides, phosphates, and sulfonic acids) and bases (the predefined bases were aliphatic amines, guanidinium, amidine, and *o*- and *p*-aminopyridines). Likewise, common charged groups were identified using SLN patterns (the predefined charge groups were charged amine, guanidinium, amidine, *o*- and *p*-aminopyridine, carboxylate, hydroxamate, tetrazole, acylsulfonamide, phosphinyl, sulfonyl, imidazole, and pyridine). Where appropriate, the charge was distributed across more than one donor or acceptor. The patterns that define ionizable and charge groups are configurable by the user.

Following solvation and assignment of charge, pharmacophore features were identified. These were acceptor atoms, donor hydrogens, planar rings, hydrophobic atoms, and user-defined features. Associated with every feature is a fitting point that is used to create an overlay when decoding the chromosome and to score the elucidated pharmacophore (though hydrophobic atoms were used in fitting only and did not contribute to the pharmacophore).

Donor hydrogens and acceptors are identified using the definitions of Mills and Dean.[24] The SLN patterns[34] used are shown in Table 1 of the Supporting Information. The phenyl NH_2 listed by Mills and Dean was not used in GAPE, as these groups are considered planar within GAPE. The fitting point of an acceptor was the center of the acceptor, while the fitting point for a donor hydrogen was 2.9 Å from the donor atom on a vector drawn through the donor hydrogen. Planar ring features are identified by determining the SSSR (smallest set of smallest rings)[35] in a structure then recovering all rings containing all sp^2 or aromatic atoms. The fitting point was the center of the ring. Hydrophobic atoms are simply carbons, with the fitting point being the atom center. User-defined features are specified using query SLNs. The fitting point for one of these features was either the first atom in the query SLN or the center of the matching fragment.

Each input structure was searched for freely rotatable bonds. Any acyclic single bond was considered freely rotatable. Rotatable bonds were next matched against a torsional library. Observation of small molecule crystal structures shows that such bonds often show conformational preferences; in particular, certain torsions are either extremely rare or never observed. The GOLD algorithm took advantage of these preferences by restricting torsional space to torsions observed in crystal structures.[27] GAPE also employed this mechanism. Two libraries of torsional distributions were available; the distributions developed for GOLD[27] and the MIMUMBA[36] distributions. The experiments described here used the GOLD distribution.

Acyclic ring flexibility is sampled in the algorithm using free corners.[37] A free corner is an atom in a ring that does not appear in any other ring and has only cyclic single bonds. Such an atom may flip across the plane defined by its neighbors—in this way a ring may move from a chair to a boat conformation. This method was implemented in a conformational search GA by Payne and Glen[38] and in the GOLD algorithm.[25,26] While ring corner flipping was appropriate for small heterocycles, it was not a suitable method for exploring the conformational space of large rings. In the case that a ring contained more than eight atoms, a single bond was selected at random, broken, and replaced by constraints using the methodology described for the FKBP12 ligand example in the GASP paper.[18,39]

Prior to pharmacophore elucidation, a random translation and rigid body rotation were applied to each input structure. Additionally, random rotations were applied to rotatable bonds. The resulting molecular coordinates were retained as reference conformations.

The Chromosome Representation. Each chromosome comprised a binary string encoding molecular conformations and an integer string encoding a pharmacophore hypothesis. The binary string contained one byte encoding an angle of rotation for each rotatable bond in the overlay and one bit for each free-corner. Binary encoding was used to encode angles of rotation; this was a departure from the GASP algorithm which used Gray-coding.[18] In GAPE (and also in GOLD) binary encoding seemed to give superior results. It was hypothesized that Gray-coding, while designed to improve the performance of mutation, performed poorly in crossover. For those rotatable bonds that matched a torsional distribution, the binary number was linearly interpolated across the observed distribution, otherwise all 360° were sampled.

The compound with the largest number of features was selected as the base molecule. Suppose that this molecule had L features. The integer string encoded associations or mappings between features in the base molecule and each other molecule. The integer string is best understood as a concatenation of $N - 1$ integer strings each of length L, where there were N molecules in the overlay. Each of these smaller strings described how a molecule mapped onto the base molecule. Each feature in every molecule was assigned a sequential feature number. If position P on the string contained the integer value V, then feature number V was mapped onto feature number P in the base molecule. The integer strings could also contain a null or dummy value that indicated that no feature was mapped onto the base molecule feature. Unlike the original GASP algorithm, there was no restriction on duplicates; the same feature in a molecule could be associated with more than one base molecule feature. However, due to the chromosome two-pass fitting procedure described below, this scenario was unlikely and the chromosomes tended to be sparsely populated with non-null values.

The Genetic Algorithm Implementation. As in GASP, GAPE used a steady-state with no duplicates GA.[16] Normalized rank-based linear selection with a selection pressure of 1.001 was employed. Selection pressure is the ratio of the scaled fitness score for the best chromosome to the scaled score for the worst chromosome fitness score. This low selection pressure biased the algorithm toward exploration of the search space rather than rapid optimization to a possibly suboptimal solution. The scaled fitness scores were used to select parents for the genetic operators described below using roulette wheel parent selection[15,16] (here a parent was selected stochastically by spinning a roulette wheel, where each chromosome had a slice of the wheel that was proportional to its scaled fitness score). Like GASP, an island model was employed where a number of subpopulations of 100 chromosomes evolved independently. The number of islands was set to N (the number of molecules) + 1.

The GA started with every chromosome in each island created with random values. The GA then iterated over each island applying genetic operators. A total number of $N \times 15\,000$ operations were applied iteratively. This choice of the number of operations was purely empirical and chosen so as to ensure GA convergence over a range of test systems. That the GA converged was verified by observing the slowing improvement of the fitness score and the increasing number of chromosome niche and duplicate hits as the algorithm progressed. The genetic operators available were crossover, mutation, and migration. In any iteration a genetic operator was selected using roulette wheel selection[15,16] with operator weights of 95 for crossover and mutation and 10 for migration (this meant that migrations were applied 5% of the time and crossover and mutation were each applied 47.5% of the time). Each genetic operator required one or two parents that were chosen using roulette wheel parent selection. The operators produced a number of children that then replaced the worst individuals in the population.

The crossover operation was applied with equal probability to either the binary or integer string chromosome. Here, two parent chromosomes produced two child chromosomes. Simple one-point crossover was used on the binary string.[15] A cross point was selected at random on the string, and child chromosomes were constructed by copying from one chromosome up to the cross point and from the other chromosome after the cross point. For the integer string, the mixing strategy described in GOLD was employed.[27] Where one parent has a nondummy value at a particular position and the other parent had a dummy value at that position, each child inherited the nondummy value. If both parents had nondummy values at that position, the first child would inherit from the first parent and the second child inherited from the second parent. This method outperformed traditional one- or two-point crossover and appeared to be better suited in transmitting from the parents to a child chromosome the maximum information from the sparse mapping encoding feature overlay.

The migration operator chose one population at random and selected a parent chromosome using roulette wheel parent selection which was copied into another randomly selected population.

The mutation operator required one parent and produced one child. The child was a copy of the parent but differed by at least one random perturbation. The perturbation was applied with equal probability to either the binary or integer string. Given a string of length L, the probability of mutation per position was $1/L$. If the string was binary, then mutation at a position entailed switching that bit (converting 1 to 0 and 0 to 1). If the string was an integer, then the value was set to another allowed value (including the null value) with equal probability. If, after processing the entire string, no mutation was applied, the operation was repeated until at least one mutation occurred.

Decoding the Chromosome. Creating a molecular overlay from the chromosome proceeded as follows; the binary string was decoded to generate molecular conformations. Torsions were applied as appropriate around rotatable bonds. Next, free corners were flipped if set in the chromosome.

The integer string was decoded using two passes of least-squares fitting. The integer string encoded mappings between features in the base molecule and every other molecule. As noted above, each feature had an associated fitting point. So each non-null value on the integer string chromosome encoded a pair of fitting points, one in the base molecule and one in another molecule. Using the technique described in the GOLD and GASP algorithms, least-squares fitting was used to superimpose each molecule onto the base molecule in such a way that the distance between fitting point pairs was minimized. The fitting was done in two passes. First, all mapped pairs of fitting points in the chromosome were used to fit the molecule to the base molecule. Next, all mapped pairs that were within 2 Å of each other were selected. If there were fewer than three such points, then the closest three points within 5 Å were chosen (in the event there was less than three such points, the chromosome was considered bad and discarded). A second pass of least-squares fitting was then applied using this smaller set of pairs. Following the second pass, the chromosome was rewritten such that mapped pairs not used in the second pass were set to null in the chromosome; this resulted in a sparsely populated chromosome. This rewriting was employed in GOLD,[27] but not GASP. The rationale here is that, following the least-squares fitting, we have a chromosome that encodes associations that are actually observed in the fitted conformations.

The original GASP algorithm had issues with failure during the least-squares fitting process, as it regularly created chromosomes that contained fewer than three mapped pairs that were within 5 Å. The inclusion of hydrophobic atoms as fitting features seems to have resolved this problem; presumably, given the larger chromosome, it is much easier to find three pairs of points within 5 Å.

The Fitness Function. Following the decoding of a chromosome and the subsequent creation of a molecular overlay, the fitness function assigned a score for use within the GA.

Use of Gaussians. The algorithm made extensive use of 3D Gaussians for scoring. These are convenient in that they are easily integrated and the overlap of two 3D Gaussians is simply a third Gaussian.

A Gaussian is defined by its center (rN), a parameter α, and a normalization term n.

$$G(r) = n\mathrm{e}^{-\alpha(r-rN)}$$

The volume (V) of such a Gaussian is easily calculated

$$V = n\left(\frac{\Pi}{\alpha}\right)^{3/2}$$

The intersection between two Gaussians $G_1(r) = n_1\mathrm{e}^{-\alpha_1(r-rN_1)}$ and $G_2(r) = n_2\mathrm{e}^{-\alpha_2(r-rN_2)}$ is a third Gaussian, $G_3(r) = n_3\mathrm{e}^{-\alpha_3(r-rN_3)}$, such that $\alpha_3 = \alpha_1 + \alpha_2$, $rN_3 = (rN_1\alpha_1 + rN_2\alpha_2)/(\alpha_1 + \alpha_2)$, and $n_3 = n_1n_2 \exp[-\alpha_1\alpha_2/(\alpha_1 + \alpha_2)\|rN_1 - rN_2\|^2]$.

Two types of Gaussians are used in the GAPE scoring functions. First, atomic spheres were represented as Gaussians for the rapid determination of common molecular volumes. Second, pharmacophore or feature points are represented by Gaussians, and the volume of the overlap Gaussian formed between two features is used as a measure of feature proximity.

In the first case, Grant and Pickup[28,40] determined that setting n to 2.7 and fixing the Gaussian volume equal to the atomic hard sphere volume gave results consistent with hard sphere calculations. So, $V = 4\Pi r^3/3$, where r is the van der Waals radius. This gave $\alpha = [(3 \times 2.7)/4\Pi r^3]^{2/3}$. Atomic Gaussians allow the efficient determination of molecular volumes. GAPE determined a molecular overlap volume to two orders; given two molecules and two sets of corresponding atomic Gaussians (A and B), a set of first-order intersection Gaussians was calculated ($C = A \cap B$) and a set of second-order intersection Gaussians that is the intersection of the first-order Gaussians with themselves ($D = C \cap C$). The common molecular volume used in GAPE is then $V_{AB} = V_C - V_D$. This technique can be extended to higher orders; for example, the shape-matching program ROCS[41] can determine molecular volumes via six orders of intersections.

For Gaussians used for scoring feature proximity, it was useful to normalize the Gaussians such that the overlap volume between two fitting point Gaussians is 1 when the two Gaussians overlay completely (i.e., the distance between Gaussian centers is 0). This is achieved by a normalization term, $n = (2\alpha/\Pi)^{3/4}$. The score (or volume overlap) for two fitting points separated by distance d_{12} was then $\exp(-\alpha d_{12}^2/2)$ (given that both fitting Gaussians had identical α values).

The α value used in the fitting Gaussian was a parameter of the algorithm. However, since it is difficult to intuitively understand the meaning of α, a distance parameter named *featureRadius* was used, such that $\alpha = (-2 \ln 0.5)/$*fittingRadius*^{2}. Thus when two fitting points were exactly *featureRadius* apart, the score was 0.5.

The parameter *featureRadius* played an important role in GAPE. Using the technique of annealing important parameters that was employed in GOLD,[26] *featureRadius* was scaled from 3.5 to 1.5 Å. So when the GA started, *featureRadius* was 3.5 Å, and after 60% of the genetic operators had been applied, *featureRadius* was 1.5 Å. In between, *featureRadius* was reset using linear interpolation and the population rescored 24 times. The purpose of this was to initially allow fuzzy and broad pharmacophore points. As the GA progressed, pharmacophore points were required to be much tighter.

General Feature Scoring. The fitness function comprised an arithmetic sum of a number of terms: donor hydrogen similarity, acceptor similarity, aromatic ring similarity, volume similarity, conformational energy, and constraint penalties.

The calculation of the volume score proceeded using the Gaussian method as described above. The conformational score was generated from implementations of the 6−12 van der Waals and torsional energy components of the Tripos force field.[42]

$$\begin{aligned}fitness = 1750 \times donorHydrogenScore + \\ 1750 \times acceptorScore + 2500 \times aromaticRingScore + \\ 100 \times volumeScore + 10 \times conformationScore + \\ 100 \times constraintScore + userfeatureScore\end{aligned}$$

These choices for the values of relative weights of the components of the fitness score were purely empirical. However, they appeared to give a reasonable balance between the different terms and gave good results over a range of test systems.

Of these terms, the *donorHydrogenScore*, *acceptorScore*, *aromaticRingScore*, and *userFeatureScore* were feature-based. Before determining a score for a feature-based term, clusters of features were identified in three-dimensional space. Recall that each of these features contains a fitting point. The following variation of the *k*-means or relocation algorithm[43] was used to determine feature clusters:

(1) Let *Feature* be the current feature type being clustered and suppose that there were *N* features of this type.
(2) A cluster was created using the first of the *N* features.
(3) Each of the remaining features is compared against each cluster in turn. If the current cluster contained no features from this molecule and its centroid was within 2 × *featureRadius* from the feature, then the feature was added to the cluster and the cluster's centroid recalculated and the comparison terminated. If the feature was not added to any existing cluster, a new cluster was created with the feature as a single member.
(4) Following the initialization phase, a series of relocation steps was performed. In one relocation cycle, every feature is compared against all cluster centroids; if a feature is closer to another cluster, then it is moved (or relocated) to that cluster. After examining all features, cluster centroids were recalculated (in relocation clustering this is referred to a batch update, as opposed to online update, where the centroids are recalculated immediately on moving an item).
(5) The relocation cycle (step 4) was repeated until no features were moved.
(6) Any final clusters that contained only one feature were discarded.

The application of this algorithm resulted in a set of feature groupings or clusters in three-dimensional space. These groupings were each considered to be a pharmacophore cluster.

For a given feature type each pharmacophore cluster was scored on the basis of the type and proximity of features and their geometric relationships. For each feature type, a pairwise comparison function was developed. This comparison determined two scores: an overall pairwise comparison score and a geometric score. The geometric score varied between 0 and 1. In order to score a phamacophore cluster, each feature in the cluster was chosen as a base feature and the cluster scored using the pairwise function. The feature that gave the highest score was considered to be the base feature (or representative feature) for that cluster. The process of scoring a base feature was as follows:

(1) A Gaussian of radius *featureRadius* was placed on each fitting point.
(2) The base feature was compared with each other feature in the pharmacophore cluster.
(3) Let *guassianOverlap* be the overlap of the two Gaussians placed on the fitting points. As noted above, the Gaussians were normalized such that 0 ≤ *guassianOverlap* ≤ 1.
(4) Starting with *guassianOverlap*, a geometric score, *geometricScore*, was created. The details for each feature type are described below. For example, when scoring the overlap of donor hydrogens or acceptors, we accounted for compatible geometry and solvent accessibility.
(5) If *geometricScore* ≥ 0.5, the other point is considered matched to the base feature.
(6) The *geometricScore* is used to create an a *featureScore*. The details for each acceptor type are described below. This step attempted to account for similarity between features. For example, if we are looking at donor hydrogens, the group probability from Table 1 in the Supporting Information was incorporated together with an additional term if both donors were charged.
(7) If molecule activities are available, they can be incorporated into the pairwise score. Activities were supplied to GAPE using pKA values (so that 1 nM ∼ 9 and 1 μM ∼ 6). These activities were use to generate a molecule weight such that *weight* = *pKA/averagePKa*. If activity was not set for a compound, its weight was set to 1.
(8) The final pairwise score for the two features was then the *featureScore* scaled by the mean molecule weight.
(9) The pairwise contributions from all pairs containing the base feature and other features in the cluster were summed.
(10) The sum was scaled by a term based on the number of other points matched to the base feature (see step 5). Let *nMatched* be the number of other points that were matched to the base feature. The pharmacophore scaling term was then $\sqrt{nMatched}$. Following scaling by this term, we normalized by dividing by *nMolecules* − 1 to obtain the score for this base feature.

Each feature in the pharmacophore cluster is tested as a base feature. The base feature that gave the highest score was retained as the base feature for the cluster and its score was retained as the score for the cluster.

This process is a departure from the GASP algorithm.[18] In GASP, pharmacophores were built around the base molecule; each pharmacophore point was a base molecule feature and each pharmacophore point had to contain a feature from every molecule. Additionally, activities were not included in the scoring function. The new algorithm is considerably more flexible; representative pharmacophore features can come from any molecule; the pharmacophore cluster does not need to include features from all molecules in the superposition, and activities are incorporated in the pharmacophore score.

One consequence of removing the GASP pharmacophore point constraints was that the algorithm had a tendency to generate a large number of pharmacophore points that contained only two features. In order to encourage the formation of pharmacophore points that contained a large number of features, the scaling factor described in step 10 above was required.

In theory, this new method should have been able to elucidate a pharmacophore that does not contain any features from the base molecule. However, since the base molecule feature fitting points were used to drive overlay creation through the chromosome encoding, in theory, the algorithm was still dependent on a good choice of base molecule. However, in practice, GAPE was able to make good overall predictions while incorrectly predicting the binding mode of the base molecule. Additionally, GAPE was able to correctly overlay trypsin ligands while using a base molecule that was CDK2-active.

On termination of a GA run, the best overlay and associated pharmacophore are written out to structure files. This pharmacophore contains all pharmacophore clusters that contain matched contributions from three or more features (or all clusters if GAPE is performing pairwise alignment).

The generation of the *featureScore* term for each feature type is now described.

Aromatic Rings. For each aromatic ring, normals were determined. Let *n* be a vector of length 3 Å along the ring normal and *c* be the vector for the center of the ring. We then determined two points $c + n$ and $c - n$. Gaussians of radius *featureRadius* were placed on these two points. The *featureScore* term was the intersection volume between these two Gaussians in one ring with the two Gaussians in the other ring. The inclusion of this term ensured that preferential scoring was given to rings whose planes overlapped. For aromatic rings the *geometricScore* is simply the *featureScore*.

Acceptor Atoms. In discussing acceptor atoms and donor hydrogens, the following function is defined: *linear_range(x, min, max)*, such that if the variable *x* has a value less than *min*, the value of *linear_range* is 1, and if *x* is greater that *max*, the value is 0; otherwise, linear interpolation is performed to determine the value of *linear_range*.

As noted in Table 1 in the Supporting Information, acceptors were categorized according to their preference for accepting donor hydrogens as either *dir* (along a lone pair), *plane* (in the plane of two lone pairs), *cone* (within the cone defined by three lone pairs), or *none* (there was no obvious preference for hydrogen bonding). We further defined a solvation point, which was a point at 2.9 Å from the acceptor along a vector that was the mean of all lone pair directions. Let *solvationMean* be the midpoint of the two solvation points. A normalized Gaussian with radius 1.0 Å was placed at *solvationMean* and the overlap with all atoms (except the two acceptor atoms) in both molecules determined. Let this overlap be *solventCorrection*. This term was used to ensure the acceptors were solvent accessible.

Let *s1* be the vector from the first acceptor to its solvation point, *s2* be the vector from the second acceptor to its solvation point, *forwardAngle* be the angle between *s1* and *s2*, and *forwardScore* = *linearScore(forwardAngle, 50, 80)*. This term ensured both acceptors were correctly orientated to accept from the same donor.

Next, a term called *acceptorGeometryScore* was calculated. This term accounted for lone pair preferences of the two acceptors. The following combinations of geometries were possible:

(1) If either acceptor had an acceptor geometry preference of *none* or *cone*, *acceptorGeometryScore* = 1.0. In the case of *cone*, the *forwardScore* term accounted for the geometry preference.

(2) Both acceptors had an acceptor geometry preference of *dir*. For each acceptor the vector between acceptor and lone pair was determined. Let *lonePairAngle* be the angle between the two vectors and *acceptorGeometryScore* = *linearScore(lonePairAngle, 20, 50)*. In the event that the second acceptor had more than one lone pair, *acceptorGeometryScore* was determined for each lone pair and the best score retained. If the first acceptor had more than one lone pair, *acceptorGeometryScore* was determined for each lone pair, summed, and normalized by dividing by the number of lone pairs.

(3) Both acceptors had a geometry preference of *plane*. Using the normals of the two planes, the angle of intersection, *planePlaneAngle*, between the two planes was determined, to give *acceptorGeometryScore* = *linearScore(planePlaneAngle, 20, 50)*.

(4) The first acceptor had a geometry preference of *plane* and the second acceptor had a preference of *dir*. The normal to the plane of the two lone pairs in the first acceptor was determined, as was the vector between lone pair and second acceptor. The angle, *planeLonePairAngle*, between the vector and lone pair plane was then calculated to give *acceptorGeometryScore* = *linearScore(planeLonePairAngle, 20, 50)*. In the event that the second acceptor had more than one lone pair, *acceptorGeometryScore* was determined for each lone pair, summed, and normalized by dividing by the number of lone pairs.

(5) The first acceptor had a geometry preference of *dir* and the second acceptor had a preference of *plane*. The determination of *acceptorGeometry* proceeded as described in item 4.

Finally, we calculated a term called *typeScore*. This term described the hydrogen-bonding strength of the acceptor pair. and was defined as the sum of the group-probabilities for the two acceptors (see Table 1 in the Supporting Information). Additionally, if both acceptors were charged, *typeScore* was doubled.

For acceptors

$$\begin{aligned} \mathit{geometricScore} &= (\mathit{gaussianOverlay} - \mathit{solvationCorrection}) \times \\ &\qquad \mathit{forwardScore} \times \mathit{acceptorGeometryScore} \\ \mathit{featureScore} &= \mathit{geometricScore} \times \mathit{typeScore} \end{aligned}$$

Donor Hydrogens. For each donor hydrogen, a solvation point was defined at 2.9 Å from the donor along the bond vector to the donor hydrogen. Let *solvationMean* be the midpoint of the two solvation points. A normalized Gaussian with radius 1.0 Å was placed at *solvationMean* and the overlap with all atoms (except the two donor hydrogens) in both molecules determined. Let this overlap be *solventCorrection*. This term was used to ensure the donor hydrogens were solvent accessible.

Next, to ensure that the geometry of the two donors was acceptable, we calculated a *donorGeometryScore* term. The angle, *donorDonorAngle*, between the two vectors defined by donor to donor hydrogen was determined. *DonorGeometryScore* was then *linearScore(60, 100)*.

Finally, we calculated a term called *typeScore*. This term described the hydrogen-bonding strength of the donor pair and was defined as the sum of the group probabilities for the two donors (see Table 1 in the Supporting Information). Additionally, if both donors were charged, *typeScore* was doubled.

For donors

$$\begin{aligned} geometricScore &= (gaussianOverlay - solvationCorrection) \times \\ &\qquad donorGeometryScore \\ featureScore &= geometricScore \times typeScore \end{aligned}$$

User-Defined Features. User defined features are specified by the user using SLN patterns. The fitting point for these features is the first atom in the pattern of the center of the fragment matching the pattern (this setting was defined by the user). For these features *geometicScore* = *gaussianOverlay*. The user also defined a weight for each feature type. Let *typeScore* be the sum of the two weights. Then *featureScore* = *geometricScore* × *typeScore*.

Multiconformer GAPE. The section describes the changes that were applied to the algorithm to replace an on-the-fly conformational search of single input structures with a multiconformer library of input structures.

Initialization of input structures preceded largely as above, with the exception that random rotations around rotatable bonds were not applied. Additionally, during loading of the structures, multiple conformers of the same input structure were identified. The only bonds considered as rotatable were those terminal torsions that were required by donors to correctly position their donor hydrogens.

The chromosome encoding was altered to include conformer selections. Let conformer number i for structure x be $c_{x,i}$. A second integer string of length N was created, such that the value v at position p resulted in conformer $c_{p,v}$ being selected for structure p on decoding. Null or empty values were not permitted. This encoding is similar to that used by DIFGAPE[17] and MEGA-SQ.[14] This chromosome was thus comprised of two components, a conformational component (which contained an integer string for selecting conformers and a binary string to rotate terminal donor hydrogens) and a fitting component (which contained the integer string to map features onto base molecule features).

The genetic operators were modified in light of the new component. The mutation operator first selected either the conformational or fitting component with equal probability. If the fitting component was selected, it was mutated as described above. If the conformational component was selected, then the binary string was selected for mutation with a probability equal to the number of torsions encoded divided by the sum of the number of torsions plus the number of molecules (this probability was chosen to prevent excessive sampling of torsions relative to conformers). Mutation on the individual binary and integer strings proceeded as above.

Likewise, the crossover operator selected individual strings in an analogous fashion to the mutation operator. Following the selection of individual strings, crossover proceeded as described above, with the exception of the integer string encoding conformer numbers, for which two-point crossover[16] was applied (the mixing strategy described above for integer strings was not applicable for the encoded conformers).

Upon decoding, the conformational component of the chromosome was decoded to create molecular conformations. First, the integer string was used to select molecular conformers, and next, the binary string was decoded by applying appropriate rotations about any terminal torsions to donor hydrogens. Following this step, the molecular overlay was constructed as described above. Finally, the fitness score was determined as above, with the difference that the conformational energy (*conformationalScore*) was not calculated. Instead, the molecular energy reported by Omega[30] was substituted. This energy was extracted from the sd-file of the input conformer library, and this technique should be applicable to other conformer generators.

In should be noted that because multiconformer GAPE did not perform so many rotations around rotatable bonds and did not calculate steric or torsional energies, it was computationally less intensive than using conformations generated on-the-fly.

In the experiments described here, Omega[30] was used as a conformer generator. In order to attempt to cover conformational space as completely as possible, the parameters MAXCONFS and EWINDOW were set to 10 000 and 20, respectively. The algorithm did not require OMEGA; any conformer generator would likely have been sufficient.

Implementation. GAPE was coded in Java 1.5.[44] In house benchmarking on clique detection and clustering algorithms indicates that Java programs were likely to be 2−10 times slower than C/C++. Furthermore, certain mathematical functions such as arc cosine did not use native libraries and were very slow. Arc cosine was heavily used during rotation around rotatable bonds. Using JNI (the Java Native Interface), this function was replaced by a call to the native library. Despite these obstacles, Java was found to be a worthwhile development language, as it provided a much more rapid software development cycle and more stable code than using C/C++.

Algorithm Validation. As detailed above, the algorithm was validated by comparing GAPE predictions with experimentally observed overlays. For each test set GAPE was run 100 times to create 100 predictions. The overlay with the highest GAPE fitness score was retained as the algorithm prediction. The criteria for success was that at least half the input structures in the prediction could be fitted to the crystal structure alignment with an average heavy atom rmsd of less than 2 Å. As it was not possible to exhaustively evaluate all combinations of input structures, the following procedure was used (where the set of molecules that were currently mapped is called *FittingStructures*):

(1) Start with a single molecule in *FittingStructures.*

(2) Using the only molecules in *FittingStructures*, least-squares fit the GAPE prediction onto the crystallographically observed overlay (this used an in-house least-squares fitting algorithm that processed all isomorphisms and thus accounted for symmetric groups).

(3) Determine the rmsd between the GAPE prediction and the crystal structure overlay for all heavy atoms in *FittingStructures*.

(4) If this rmsd is more than 2 Å, then remove the last structure added to *FittingStructures*. The remaining molecules in *FittingStructures* contain a set of structures for which the GAPE prediction is within 2 Å. Save these molecules and stop.

(5) If *FittingStructures* contains all the molecules in the overlay, stop.

(6) For all molecules not in *FittingStructures*, evaluate the rmsd between the molecule in the fitted GAPE prediction

and the molecule in the crystal structure overlay. Add the molecule with the smallest rmsd to *FittingStructures*.

(7) Continue at step 2.

This process was repeated for all possible starting molecules, and the largest *FittingStructures* set generated was retained (in the event of a tie, the set with the lowest rmsd was chosen). The number of compounds in *FittingStructures* and the associated fitting of those structures onto the crystal structure overlay were used to evaluate the success of the experiment.

RESULTS AND DISCUSSION

Overlay Generation Using GAPE. The following experimental protocol was used to evaluate the performance of GAPE. First, for each test system regular GAPE was run 100 times and the highest scoring solution was retained as the GAPE prediction. Note, that the highest scoring prediction may not necessarily have been the best result in terms of rmsd to the observed binding mode; other solutions from the 100 runs may be closer to the crystallographic alignment but have poorer scores. Second, this procedure was repeated for multiconformer GAPE. In both these experiments, the predictions were evaluated using the rmsd procedure described above. However, rmsd criteria are often flawed; a reasonable rmsd can disguise serious shortcomings in the prediction, and a prediction with poor rmsd may still contain useful information. For this reason, each overlay was examined by hand and a subjective rating of “good”, “bad”, or “partial” assigned. A good prediction was expected to have excellent overall agreement to the crystallographically observed binding modes with only insignificant deviations. A partial prediction had to have at least some significant agreement or overlap with the observed binding mode, whereas a bad prediction bore little or no relationship to the observed binding mode.

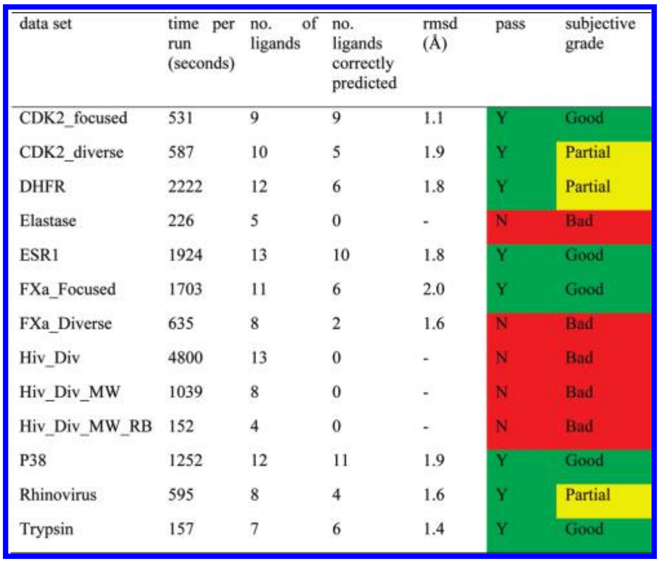
It is worth remembering that GAPE is a pharmacophore elucidation program and, in addition to the overlaid compounds, the most significant output is the predicted pharmacophore. The comparison of elucidated pharmacophore against any possible pharmacophore that could be created from the observed binding mode was not attempted, as this was considered to be a highly subjective process. Nevertheless, in those cases where GAPE successfully reproduced the ligand binding mode, it is reasonable to assume that the predicted pharmacophore contains all the features necessary for binding to the protein.

It was noted that 100 runs may be overkill for many of the test systems. Thus, the third and forth procedures comprised (for each test system) of four separate GAPE experiments of 25 runs for both regular and multiconformer GAPE, respectively. These experiments were used to evaluate the reproducibility of the algorithm and the trade off between CPU resources and predictive power.

It should be remembered that GAPE is a stochastic sampling algorithm where experimental results are dependent on values from a random number generator. Thus, the results reported here may appear inconsistent; for example, in the elastase data set the 100 run GAPE prediction failed to match the observed binding mode, whereas all four of the 25 run experiments were successful.

The various HIV diverse data sets, which contained large peptidic ligands, were particularly challenging and resulted in failure in all experiments. These test systems appear to be intractable to the current algorithm, and the discussion here will involve the other test systems for which some measure of success was observed.

Table 2. GAPE Overlay Results



data set	time per run (seconds)	no. of ligands	no. ligands correctly predicted	rmsd (Å)	pass	subjective grade
CDK2_focused	531	9	9	1.1	Y	Good
CDK2_diverse	587	10	5	1.9	Y	Partial
DHFR	2222	12	6	1.8	Y	Partial
Elastase	226	5	0	-	N	Bad
ESR1	1924	13	10	1.8	Y	Good
FXa_Focused	1703	11	6	2.0	Y	Good
FXa_Diverse	635	8	2	1.6	N	Bad
Hiv_Div	4800	13	0	-	N	Bad
Hiv_Div_MW	1039	8	0	-	N	Bad
Hiv_Div_MW_RB	152	4	0	-	N	Bad
P38	1252	12	11	1.9	Y	Good
Rhinovirus	595	8	4	1.6	Y	Partial
Trypsin	157	7	6	1.4	Y	Good

For the procedure where GAPE was run 100 times, the predictions for the 13 test systems are summarized in Table 2. The CPU timings are for a Linux workstation with an AMD Opteron 246 1 GHz processor. Table 2 lists the maximum number of ligands in the prediction that can be fitted to the observed crystal structure alignment with a heavy atom rmsd of less that 2 Å. It can be seen that objective success was observed for eight of the 13 test systems. In fact, if the three HIV protease data sets are excluded, the only other failures are elastase and FXa_Diverse. On the basis of subjective judgments, GAPE was successful in five data sets, had partial success in three, and failed outright in five. The objective measure was equivalent to the subjective ranking in that the subjective grades of good or partial success equated to subjective success.

In the case of the CDK2_Focused data set, excellent agreement with the experimentally observed binding mode was found, as shown in Figure 2. The only real difference was in the position of 1AQ1, which is unsurprising, as it does share the same volume as the other compounds in the series. Despite the fact that 1AQ1 was chosen as the base molecule, GAPE was able to overlay the remaining compounds in the series.

For CDK2_Diverse, examination of the aligned crystal structure ligands shows that this is a particularly challenging problem, with no easily identifiable pharmacophore. On the face of it, the objective result of half the structures correctly aligned would thus appear to be a success. However, a visual inspection of the five structures that are supposed to be equivalent to those in the crystal structure showed that two of them (the ligands from 1AQ1 and 1P5E) were not correctly aligned. 1AQ1 differed as noted for the CDK2_Focused data set, while the ligand from 1P5E was small and planar, and while the GAPE prediction occupied the same approximate volume, the predicted binding mode was significantly different from the crystallographic binding mode. This left the three ligands shown in Figure 3. While the agreement for these three was reasonable and resulted in assigning

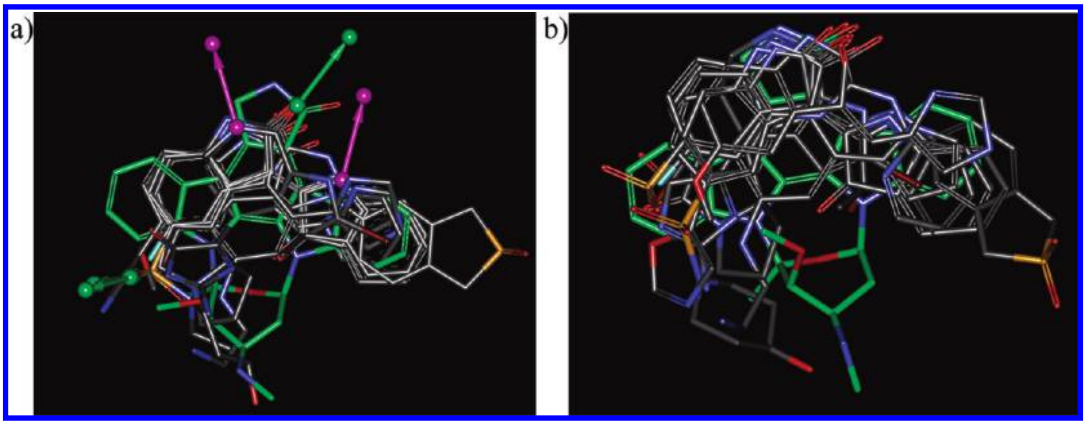


Figure 2. Results for GAPE on the CDK2_Focused data set: (a) GAPE prediction (hydrogen bond donor vectors are shown in purple and hydrogen bond acceptor vectors are shown in green) and (b) crystal structure overlay. In both the ligand from 1AQ1 has carbon atoms colored green.

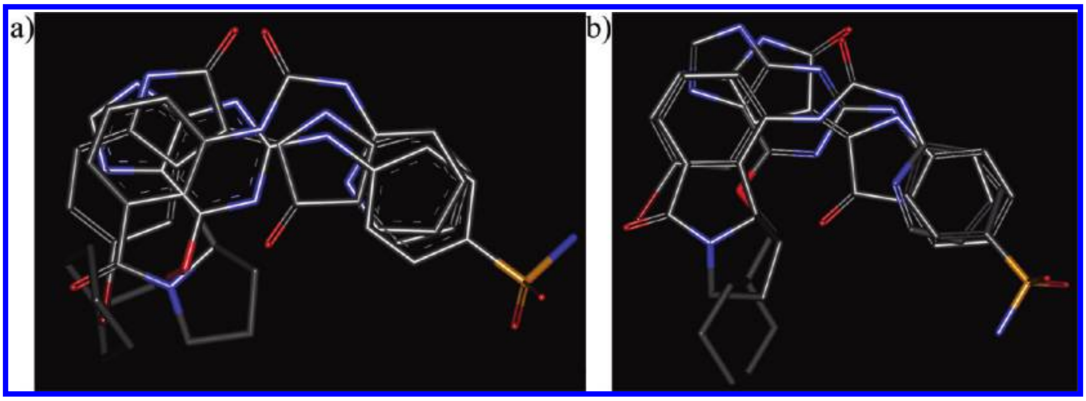


Figure 3. Results for GAPE on the CDK2_Diverse data set: (a) GAPE and (b) crystal structure overlay. In both cases only the three ligands (from 1GIH, 1HIS, and 2BHE) are shown, for which close correspondence between the GAPE prediction and the observed binding mode was seen.

partial success to the test set, the elucidated pharmacophore did not actually represent the interactions observed in the crystal structure complexes.

The test case for DHFR was also challenging, and in this case GAPE was, at least, partially successful, reproducing the binding mode for six of the 12 ligands, as seen in Figure 4. However all six were relatively small compounds; GAPE did not predict the binding mode of any of the five larger ligands (which included the natural ligand folate and the anticancer drug methotrexate), despite a well-defined pharmacophore. In addition, GAPE failed to predict the binding mode of one smaller ligand (from 1S3W) which differed from the other smaller ligands in not having a protonated pteridine ring and consequently interacted differently with residue Glu30 (these two different binding orientations to Glu30 are best observed by comparing the binding mode geometries of folate and methotrexate[25]). In some ways this is a disappointing result, as the common pharmacophore is obvious from an inspection of 2D representations of the structures. It would seem that the number and size of ligands in this test system is close to overwhelming the algorithm.

Elastase is a particularly difficult data set in that structurally similar ligands exhibit different binding modes;[45] despite high structural similarity, the ligands 1ELB and 1ELC bind in a different mode from the other three. Nevertheless, using our rmsd criteria, it is possible to have a successful result if the binding modes of the other three ligands are correctly predicted. Unfortunately, GAPE was unable to do this in the single 100-run experiment (though it was successful in all four of the 25-run experiments).

As shown in Figure 5, ESR1 was a particularly impressive result for GAPE, with 10 out of 13 ligands being correctly overlaid. The algorithm failed for the ligands from 1A52, 1XQC, and 2BJ4, where symmetry effects resulted in an incorrect result. It has been observed that these ligands can be partitioned into two classes: larger "T" shaped ligands and a set of smaller linear ligands that overlay onto the other class on top of the "T".[17] The ligand from 1A52 was one of the smaller ligands; it has an oxygen donor−acceptor at each end and was overlaid incorrectly by 180°, while satisfying the elucidated GAPE pharmacophore. The ligands from 1XQC and 2BJ4 were similarly affected; these both belonged to the larger set of ligands, and the flexible group on top of the "T" was rotated by 180°, relative to the crystallographically observed binding mode.

Two data sets for FXa were analyzed. The GAPE result for the FXa_Focused set is shown in Figure 6. Although only six of the 11 structures could be overlaid onto the crystal structure alignment within 2 Å, all structures conformed to the overall binding mode and the rmsd for all structures was 2.4 Å; therefore, this result was considered a success. Unfortunately, the result for the FXa_Diverse was a failure with no meaningful correlation observed between the GAPE prediction and the observed binding mode. Interestingly, this is the only test system for which DIFGAPE clearly outperformed GAPE.[17]

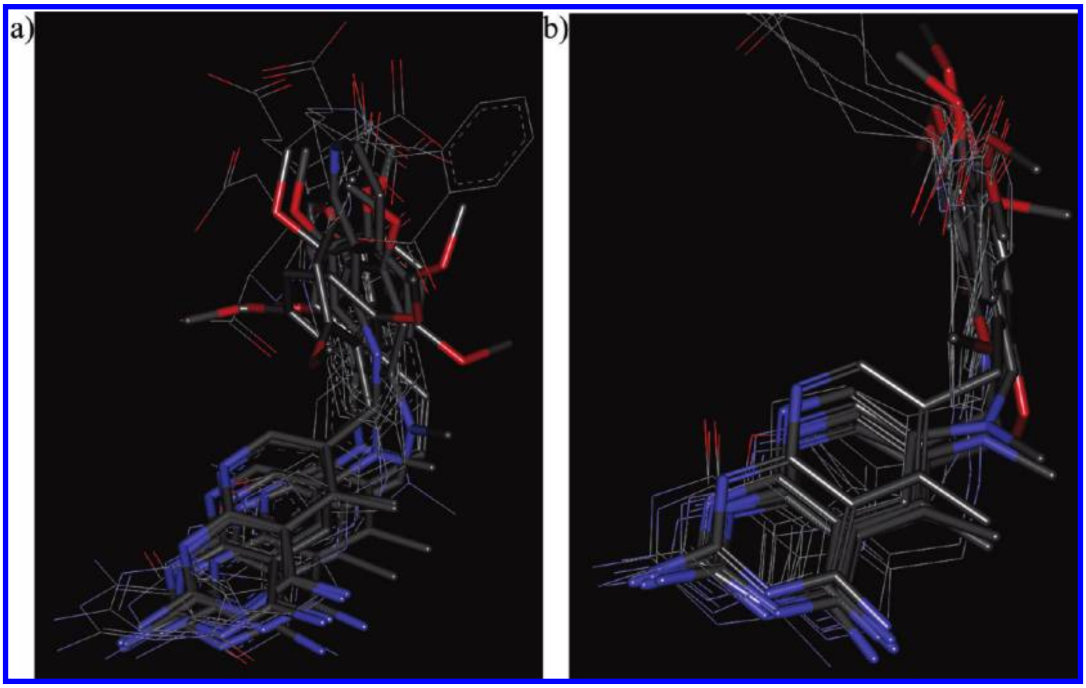


Figure 4. Results for GAPE on the DHFR data set: (a) GAPE prediction and (b) crystal structure overlay. The six structures for which GAPE was unable to successfully overlay onto the crystal structure are depicted with lines. For clarity the pharmacophore features are omitted.

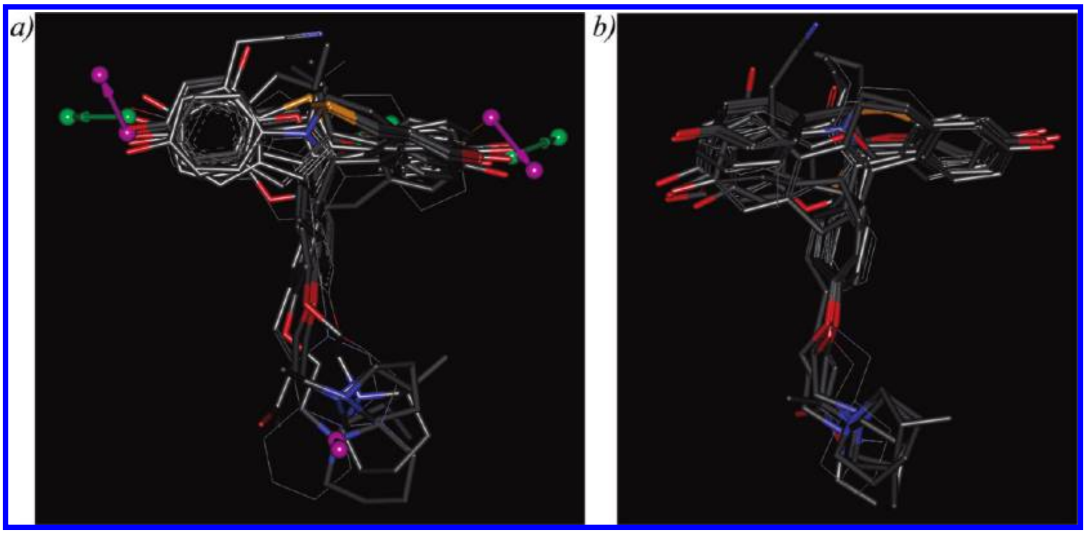


Figure 5. Results for GAPE on ESR1 data set: (a) GAPE prediction (hydrogen bond donor vectors are shown in purple and hydrogen bond acceptor vectors are shown in green; one acceptor is hidden) and (b) crystal structure overlay. The three structures for which GAPE was unable to successfully overlay onto the crystal structure are depicted with lines.

Figure 7 shows the GAPE prediction for the map kinase P38 data set. GAPE successfully aligned 11 of the 12 compounds (it failed on the ligand from 1W84) within the 2 Å criteria. However, closer inspection of the prediction shows that two smaller ligands, from 1W7H and 1DI9, are also incorrect. Additionally, portions of the ligands from 1OZ1 and 1A9U have been slightly displaced so as to overlay the other structures. Nevertheless, this is clearly a successful result.

Results for the rhinovirus data set are shown in Figure 8. This system barely passed the objective criteria, with only half the structures being correctly predicted. The four structures for which GAPE failed were added to the overlay in the reverse orientation from that observed in the crystal structure. However, it has been speculated previously that these near symmetric ligands were incorrectly fitted when the crystal structure was resolved or that they can bind in either orientation.[17]

In the final test set, trypsin, GAPE produced an excellent result, with the caveat that this data set is composed of small and similar ligands. GAPE successfully predicted all ligands with the exception of the ligand from 1PHH (which was identified as the base molecule). This ligand is substantially larger than any other ligand and does not wholly overlay any other structure. GAPE did, however, correctly align the portion of the ligand that interacts with Asp189 in the protein.

Overlay Generation Using Multiconformer GAPE. Before discussing the results for using multiconformer GAPE, it is worth examining the conformational coverage of the multiconformer libraries. In order for the multiconformer mode to be successful, it was clear that the conformers in the library would need to include conformations close to

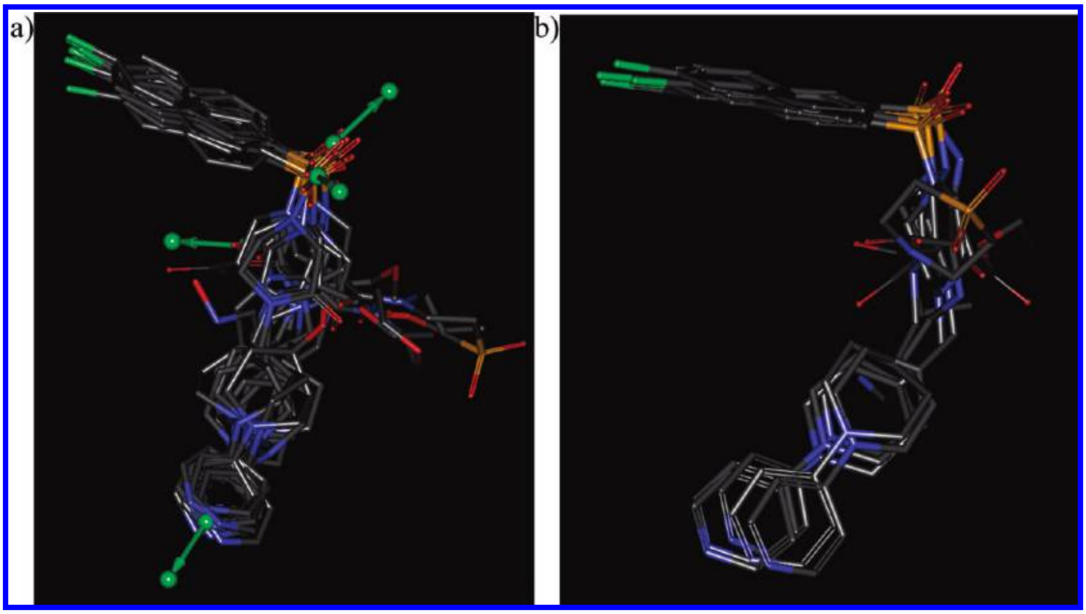


Figure 6. Result for GAPE (hydrogen bond acceptor vectors are shown in green) on the FXa_Focused data set: (a) GAPE prediction and (b) crystal structure overlay. All 11 structures are depicted in stick form.

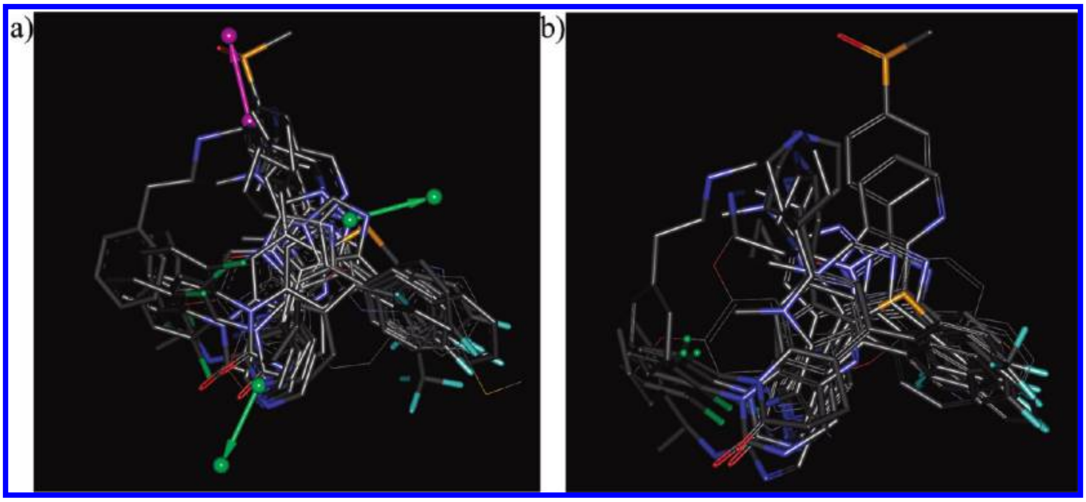


Figure 7. Results for GAPE on the P38 data set: (a) GAPE prediction (hydrogen bond donor vectors are shown in purple and hydrogen bond acceptor vectors are shown in green) and (b) crystal structure overlay. The three structures for which GAPE was unable to successfully overlay onto the crystal structure are depicted with lines.

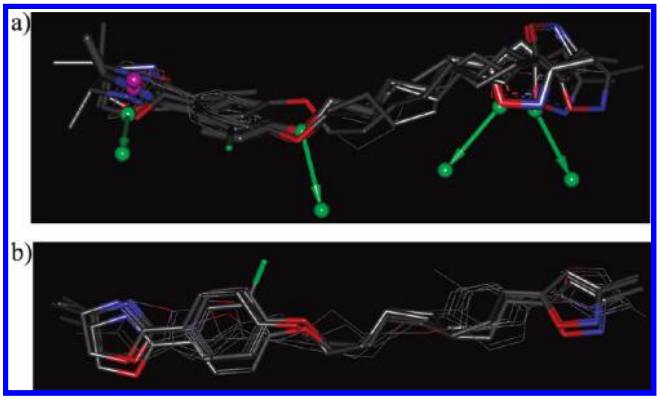


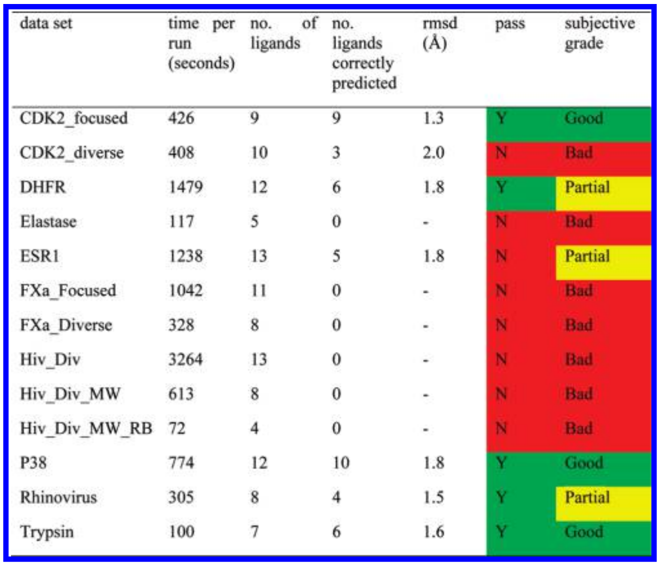
Figure 8. Results for GAPE on the rhinovirus data set: (a) GAPE prediction (the hydrogen bond donor vector is shown in purple and hydrogen bond acceptor vectors are shown in green) and (b) crystal structure overlay. The four structures for which GAPE was unable to successfully overlay onto the crystal structure are depicted with lines.

the observed binding modes. In order to attempt to cover conformational space as completely as possible when using Omega, the parameters MAXCONFS and EWINDOW were set to 10 000 and 20, respectively. For each binding mode structure, the number of library conformations and the rmsd distance in Å between the nearest library conformation and the ligand binding mode was determined. Table 3 details the average, minimum, and maximum values obtained for each data set. It can be seen that the size of the conformational libraries varies considerably, from an average of 21 conformations per ligand (CDK2_Focused) to 6923 conformations per ligand (rhinovirus). With the exception of two HIV data sets and elastase, the average distance per data set between observed binding modes and the closest library conformation was less than 1 Å rmsd. Judging from the maximum distances, there were still nine data sets where at least one ligand did not have a conformer in the library closer than 1 Å rmsd and two data sets where at least one ligand was more than 2 Å rmsd from the closest conformer in the library. Given that an attempt was made to configure Omega to cover conformational space, this is an unfortunate result.

Table 3. Multiconformer Library Statistics

		no. of conformers			closest conformer to binding conformation (rmsd Å)		
data set	no. of ligands	min	avg	max	min	avg	max
CDK2_Focused	9	1	21.0	86	0.06	0.43	1.24
CDK2_Diverse	10	1	330.7	2531	0.26	0.57	1.24
DHFR	12	30	266.0	1410	0.27	0.74	2.17
elastase	5	250	840.8	1191	0.76	1.10	1.50
ESR1	13	1	318.6	1914	0.13	0.62	1.62
FXa_Focused	11	321	1532.3	4449	0.46	0.85	1.85
FXa_Diverse	8	328	2946.1	7765	0.49	0.75	0.93
Hiv_Div	13	3	1508.4	6499	0.55	1.74	3.57
Hiv_Div_MW	8	455	2030.8	6499	0.56	1.19	1.94
Hiv_Div_MW_RB	4	455	1061.8	1466	0.55	0.96	1.93
P38	12	4	419.3	2775	0.13	0.37	0.48
rhinovirus	8	5059	6923.1	8277	0.44	0.58	0.81
trypsin	7	1	42.1	251	0.06	0.27	0.55

Table 4. Multiconformer GAPE Results



data set	time per run (seconds)	no. of ligands	no. ligands correctly predicted	rmsd (Å)	pass	subjective grade
CDK2_focused	426	9	9	1.3	Y	Good
CDK2_diverse	408	10	3	2.0	N	Bad
DHFR	1479	12	6	1.8	Y	Partial
Elastase	117	5	0	-	N	Bad
ESR1	1238	13	5	1.8	N	Partial
FXa_Focused	1042	11	0	-	N	Bad
FXa_Diverse	328	8	0	-	N	Bad
Hiv_Div	3264	13	0	-	N	Bad
Hiv_Div_MW	613	8	0	-	N	Bad
Hiv_Div_MW_RB	72	4	0	-	N	Bad
P38	774	12	10	1.8	Y	Good
Rhinovirus	305	8	4	1.5	Y	Partial
Trypsin	100	7	6	1.6	Y	Good

Table 4 details the performance of multiconformer GAPE. The same experimental protocol that was used to evaluate regular GAPE was applied here. It can be seen that the multiconformer algorithm is over 50% faster than the regular GAPE (overall data sets 782 s versus 1217 s per run). Objective success was observed for five of the 13 test systems. On the basis of subjective judgments, multiconformer GAPE was successful in three data sets, had partial success in three, and failed outright in seven. Given the difficulty of the test systems, this is not such a bad result. However, given that the regular GAPE algorithm appears so much more successful, this was somewhat disappointing. There was no test system where the multiconformer GAPE did better than the regular GAPE. It was hoped that the algorithm would be improved by reducing the size of the conformational search space and biasing the conformational search to conformations of lower energy.

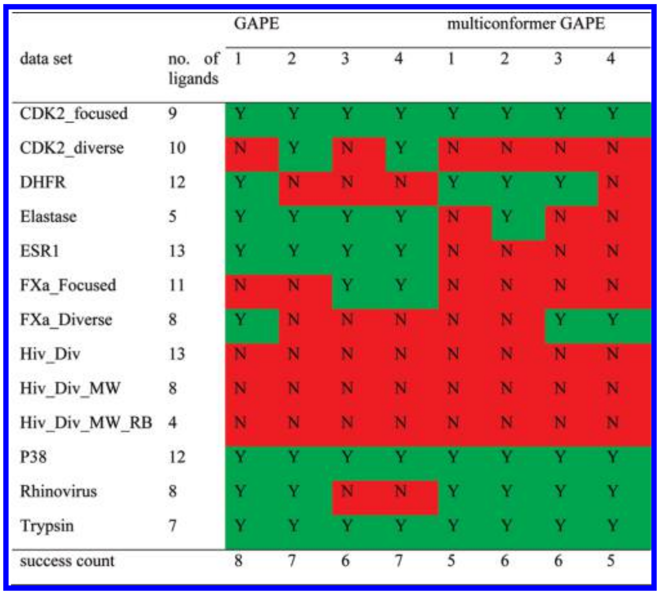
A number of hypotheses may be suggested to explain the differences in performance between the two methods. First, the conformational coverage of the conformer library may not be sufficient; as noted above, there are many binding nodes that were not present in the conformer library. Second, the GA may not be efficiently sampling conformational space in the new encoding. For example, in the traditional generation of conformations on-the-fly, the mutation operator will likely make an incremental difference to the molecular conformation, whereas selecting a new conformer at random could result in a vastly different molecular conformation. Finally, the differences may not be as great as they first appear; due to the stochastic nature of the algorithm, the regular version of GAPE may have performed better than average, while the new multiconformer algorithm may have performed worse than average. The results (presented below) for the shorter number of runs suggest that this may be partially the case.

Overlay Generation Using 25 Runs. The experiments described above are based on the best scoring result from 100 runs. This required a considerable investment in CPU time; in the worst case, the regular GAPE results for the HIV_Div data set took $5^1/_2$ days to produce. For each test system, GAPE was run 25 times and the highest score solution retained as the prediction. This process was repeated four times so as to create four predictions for each data set. This was in order to provide a more realistic evaluation of the algorithm as it might be used in practice and to investigate the reproducibility of the method. While the experimental protocol did not otherwise change, these results were not evaluated subjectively (because of the time that manual inspection of the overlays requires).

The results, using the objective criteria, are presented in Table 5. It can be seen that, on average, over the four experiments the regular GAPE was successful in seven data sets, while multiconformer GAPE was successful 5.5 times. While regular GAPE still outperformed multiconformer GAPE, the difference was not as marked as in the larger experiment of 100 runs. Further inspection gave insights into the reproducibility of predictions. The regular GAPE algorithm was consistently successful in five of the 13 data sets; it consistently failed in three and had mixed results in the remaining five. Interestingly, this means that GAPE was capable of at least partially reproducing the observed binding mode for all data sets except the three HIV systems. Multiconformer GAPE was consistently successful in four data sets, it consistently failed in six, and had mixed results in three. Multiconformer GAPE consistently failed to make successful predictions in the CDK2_Diverse, ESR1, and FXa_Focused data sets, whereas the regular GAPE algorithm was consistently successful in the ESR1 data set and occasionally successful in the other two.

One strange result was that the regular GAPE consistently predicted the observed binding mode for the elastase data

Table 5. Results for Four Experiments Each with the Best of 25 Runs



		GAPE				multiconformer GAPE			
data set	no. of ligands	1	2	3	4	1	2	3	4
CDK2_focused	9	Y	Y	Y	Y	Y	Y	Y	Y
CDK2_diverse	10	N	Y	N	Y	N	N	N	N
DHFR	12	Y	N	N	N	Y	Y	Y	N
Elastase	5	Y	Y	Y	Y	N	Y	N	N
ESR1	13	Y	Y	Y	Y	N	N	N	N
FXa_Focused	11	N	N	Y	Y	N	N	N	N
FXa_Diverse	8	Y	N	N	N	N	N	Y	Y
Hiv_Div	13	N	N	N	N	N	N	N	N
Hiv_Div_MW	8	N	N	N	N	N	N	N	N
Hiv_Div_MW_RB	4	N	N	N	N	N	N	N	N
P38	12	Y	Y	Y	Y	Y	Y	Y	Y
Rhinovirus	8	Y	Y	N	N	Y	Y	Y	Y
Trypsin	7	Y	Y	Y	Y	Y	Y	Y	Y
success count		8	7	6	7	5	6	6	5

set, whereas in the larger experiment of 100 runs it failed to do so. Recall that, in this system, all five ligands possess high structural similarity, but two of them (from the complexes 1ELB and 1ELC) exhibit different binding modes.[45] Figure 9 shows that the GAPE prediction (for the first of the four experiments) predicts three ligands correctly, but fails on the two that show a different binding mode.

GAPE and Multiple Binding Modes. It should be noted that, for the most part, the ligands in each of the 13 test systems contain a common binding motif. From these results it is not clear that GAPE could handle a set of ligands with multiple binding modes. In order to investigate this possibility, GAPE was run on a single test system of 16 ligands that was created by combining the CDK2_Focused and trypsin data sets. GAPE was run 100 times and the prediction taken from the highest scoring run. The GAPE prediction for this combined data set was then split by hand into two predictions: one for the CDK2_Focused compounds and another for the trypsin compounds. Both of these predictions passed the objective criteria for success against their appropriate crystal structures; the trypsin compounds passed the subjective criteria for success and the CDK2 compounds were partially successful under the subjective criteria. In the case of the trypsin ligands, the binding mode was predicted using a base molecule that was CDK2-active (1AQ1).

Analysis of GA Runs. One possible flaw in GAPE is that it may have found a solution close to the observed binding mode early in a GA run, only to have replaced it later by an incorrect solution that the fitness function ranked higher. In order to investigate this, regular GAPE was run 25 times for each of the 13 test systems. In each run, using the objective criteria, the trajectory of the highest scoring individual was examined to determine when the best chromosome in the population was close to the observed binding mode. Of the 325 runs, 140 finished with a hit according to the objective criteria and 37 at one point had the top scoring chromosome come close to the observed binding mode yet failed to finish with a hit. It thus appears relatively rare (11% of runs) for GAPE to rank a solution close to the observed binding mode as the best intermediate solution then go on to select another binding mode as the final solution.

In those 140 cases where GAPE finished with a hit, the first occasion that the top scoring chromosome hit the observed binding mode was (on average) 16% into the total number of genetic operations applied. This suggests that the algorithm converges early and there is potential to reduce the maximum number of genetic operations applied and thus the overall runtime.

Comparison with Tripos GASP and Galahad. The original GASP algorithm was restricted to aligning small data sets[18,19] and would thus have been unable to process these test systems. However, the current version of GASP, supplied by Tripos Associates in the molecular modeling suite SYBYL-X 1.1[32] (henceforth referred to as "Tripos GASP"), was capable of handling larger data sets. Presumably, Tripos GASP has incorporated some or all of the GA improvements that the Sheffield group and others have developed for MOGA[20,21,23] and GASP.[22] Using a MAXOPS setting of 200 000 (with the default settings used for other parameters), an attempt was made to generate 100 GASP solutions for each test system. Where this was possible, the solution with the best Tripos GASP score was taken as the Tripos GASP prediction and compared to the crystal structure using the subjective and objective criteria. Because of technical difficulties, results were not obtained for the two CDK2

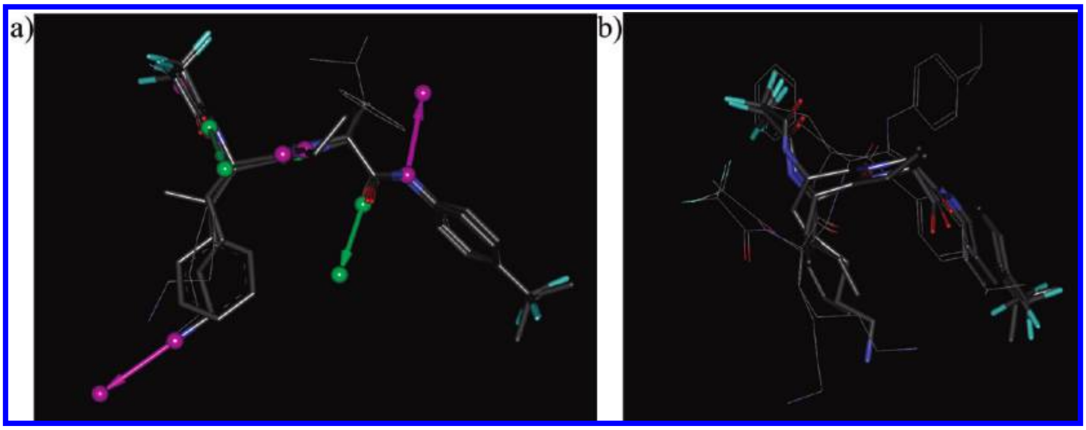
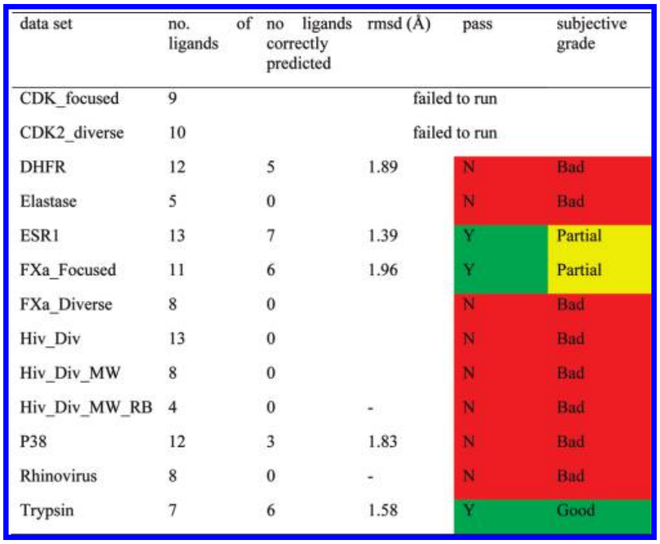


Figure 9. Results for GAPE on the elastase data set: (a) GAPE prediction (the hydrogen bond donor vectors are shown in purple and hydrogen bond acceptor vectors are shown in green) and (b) crystal structure overlay. The two structures, ligands from the complexes 1ELB and 1ELC, for which GAPE was unable to successfully overlay onto the crystal structure are depicted with lines.

Table 6. Tripos GASP Overlay Results



data set	no. of ligands	no ligands correctly predicted	rmsd (Å)	pass	subjective grade
CDK_focused	9			failed to run	
CDK2_diverse	10			failed to run	
DHFR	12	5	1.89	N	Bad
Elastase	5	0		N	Bad
ESR1	13	7	1.39	Y	Partial
FXa_Focused	11	6	1.96	Y	Partial
FXa_Diverse	8	0		N	Bad
Hiv_Div	13	0		N	Bad
Hiv_Div_MW	8	0		N	Bad
Hiv_Div_MW_RB	4	0	-	N	Bad
P38	12	3	1.83	N	Bad
Rhinovirus	8	0	-	N	Bad
Trypsin	7	6	1.58	Y	Good

systems (the program crashed on population initialization). Additionally, a bug in the parent selection procedure prevented creation of all 100 solutions in a single batch, and multiple batch runs were required for each test system. The results for 11 test systems are shown in Table 6. Tripos GASP achieved objective success in three test systems and subjective success in one. The two test systems, ESR1 and FXa_Focused, which achieved objective success, but were considered only partially successful using the subjective criteria, are shown in Figure 10. Figure 10 illustrates a phenomenon that was consistently observed when Tripos GASP was run on data sets containing a larger number of compounds. Although Tripos GASP is able to tightly overlay some structures, others are essentially abandoned and not included in the overlay. It was hypothesized that perhaps the algorithm was not converging or that the population size was insufficient to cover the search space. In order to test this, the experiment was repeated for ESR1 and FXa_Focused, first with MAXOPS = 600 000 and OPS_INC = 13 000 and second with MAXOPS = 600 000, OPS_INC = 13 000 and POPSIZE = 200. Unfortunately, neither of these changes improved the quality of the observed overlays.

A comparison was also performed with the version of Galahad[9] supplied with SYBL-X 1.1.[32] Unlike GAPE or GASP, Galahad is not a predictive program; it uses a multiobjective GA with Pareto optimization to return several alternative hypotheses. Using the default settings (Galahad adjusts parameters based on the input structures) an attempt was made to create hypotheses for the 13 test systems. As with Tripos GASP, the GA failed for the CDK2 data sets and the run on the HIV_Div data set was abandoned when the GA failed to initialize after 48 CPU hours (using an Intel T9900 3.06 GHz processor, which is approximately twice as fast as the processor used for evaluating GAPE). The results for the remaining 10 data sets are shown in Table 7. For each data set, the 20 hypotheses returned by Galahad were compared to the crystal structure, and the one with the best objective score was subjectively examined (in all cases, except trypsin, the 20 hypothesis examined had the top Pareto ranking of 0). Galahad was able to recreate the crystallographically observed binding mode for five data sets using the objective criteria and two data sets using the subjective criteria. If we assume success for the missing CDK2 data sets, these results are not much worse than those obtained from using regular GAPE. However, it is not clear how a Galahad user would be able to identify a hypothesis that matched the crystal structure from one that did not. Table 7 also lists the number of hypotheses that passed the objective criteria. Only for the trypsin and rhinovirus data sets could a user be reasonably certain of finding a hypothesis that comes close to the observed binding mode.

CPU times were not available for either Tripos GASP or Galahad for SYBYL-X on the Windows XP platform used here. However, Tripos GASP was judged to be 3−5 times faster than GAPE and a single Galahad run was approximately 1−5 times as fast as 100 GAPE runs.

In summary, when attempting to predict the observed binding modes of ligands in these test systems, GAPE clearly outperformed both Tripos GASP and Galahad. Of course, the results obtained for each of the three algorithms were dependent on random number seeds and the parametrization chosen.

CONCLUSIONS

An improved methodology using a GA to create quality overlays of multiple ligands has been presented. This GA is based upon the previously described pharmacophore elucidation program GASP.[18] Numerous enhancements and improvements have been applied to the original algorithm, resulting in a substantial increase in the reliability and applicability of the algorithm. A significant enhancement is an improved pharmacophore perception routine that does not require that each pharmacophore point be present in every structure. Additional improvements include hydrogen bond scoring based on observed hydrogen bond distributions and shape scoring based on atomic Gaussian overlap. These improvements have sufficiently increased the scalability of the algorithm that it is now able to accommodate overlays of up to 13 compounds. The program is capable of evaluating molecular conformations on-the-fly, or it can use a supplied conformer library.

A number of test systems extracted from the PDB were used to evaluate GAPE. It has been shown that GAPE is capable of producing reasonable approximations to the crystallographically observed binding mode without using any protein structure. These predictions are made using the highest scoring solution generated by the algorithm; it was possible that where GAPE made a poor prediction it had generated other lower scoring acceptable predictions. The regular GAPE algorithm, which generated molecular conformations on-the-fly, fulfilled our objective criteria for success in eight of 13 test systems. Under a more demanding subjective analysis, GAPE was found to be successful in five cases, partially successful in three cases, and to have failed in five systems. These predictions were shown to be significantly better than the predictions obtained on these data sets using Tripos GASP and Galahad. Given the difficulty in predicting ligand binding modes in the absence of protein structure, this is a particularly encouraging result, and thus, GAPE should be considered a powerful tool for drug discovery programs that have active compounds but no available protein structure.

It was hoped that using a conformer library, as opposed to evaluating conformations on-the-fly, would improve

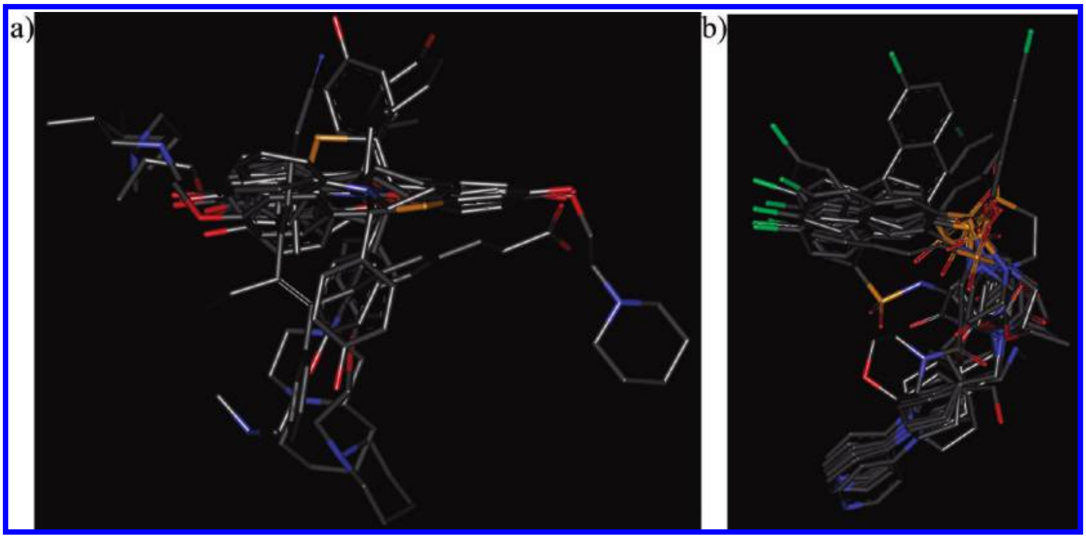
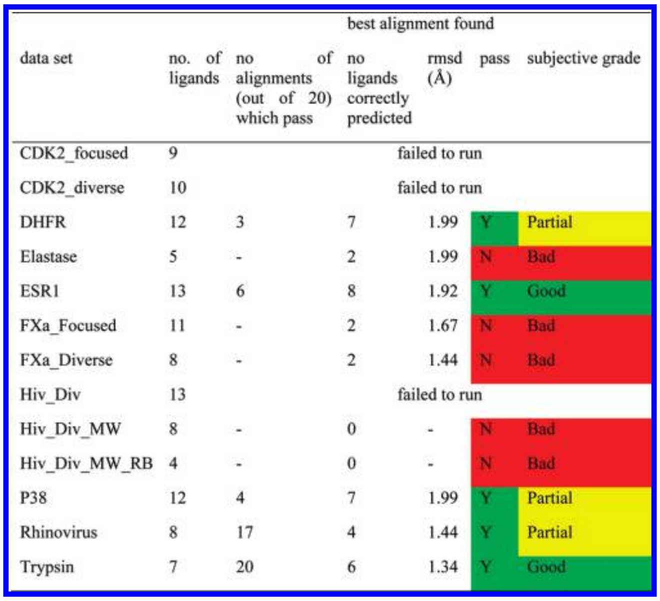


Figure 10. Tripos GASP predictions for (a) ESR1 (shown in the same approximate orientation as Figure 5), and (b) FXa_Focused (shown in the same approximate orientation as Figure 6).

Table 7. Tripos Galahad Overlay Results



			best alignment found			
data set	no. of ligands	no of alignments (out of 20) which pass	no ligands correctly predicted	rmsd (Å)	pass	subjective grade
CDK2_focused	9		failed to run			
CDK2_diverse	10		failed to run			
DHFR	12	3	7	1.99	Y	Partial
Elastase	5	-	2	1.99	N	Bad
ESR1	13	6	8	1.92	Y	Good
FXa_Focused	11	-	2	1.67	N	Bad
FXa_Diverse	8	-	2	1.44	N	Bad
Hiv_Div	13		failed to run			
Hiv_Div_MW	8	-	0	-	N	Bad
Hiv_Div_MW_RB	4	-	0	-	N	Bad
P38	12	4	7	1.99	Y	Partial
Rhinovirus	8	17	4	1.44	Y	Partial
Trypsin	7	20	6	1.34	Y	Good

algorithmic performance both by reducing the size of the search space and restricting the search space to conformationally accessible regions. Unfortunately, multiconformer GAPE was consistently outperformed by regular GAPE. Whether this was due to inherent limitations from using discrete conformers to represent a continuous space or if it was due to issues in adapting the GA to sample conformers was not clear.

The overlays generated by GAPE may be used as a starting point for 3D-QSAR[46] methods, such as CoMFA,[47] which require an initial alignment of the molecules that are to be analyzed. Additionally, the pharmacophoric patterns and common molecular volume from the overlays may be used as a query for search against databases of 3D compounds in virtual screening applications; an internal application has been developed at Arena Pharmaceuticals for this purpose. It would be nice to evaluate the performance of GAPE in the context of virtual screening using the methods described by Kirchmair et al.[48]

While effective, the algorithm could benefit from some improvements. It is intriguing that in most test systems the algorithm successfully identified the binding mode of some compounds in the data set while failing to predict the binding mode of other compounds. It would be an interesting extension to the algorithm to determine the relative contribution of each structure to the overall score. The structures that did not contribute significantly to the overall score could then be excluded from the final overlay. It is possible that the excluded structures might contain the structures whose binding modes were incorrectly predicted.

There are some features of the algorithm that have yet to be formally validated. These include the use of activities in the scoring function, so as to bias relative contributions from different structures, and the use of user defined features and constraints. It has been shown that GAPE has the ability to handle sets of compounds with multiple binding modes. However, this has not yet been fully evaluated on realistic test systems. Furthermore, what GAPE does not currently do is automatically identify the separate pharmacophores or attempt to separate out the different binding modes. Such identification would be a useful extension to the algorithm. It is hoped that these features can be properly validated in future publications.

In addition to the phenyl NH_2 acceptor mentioned in the methods section, there are other weak hydrogen-bonding groups that are not represented in the current algorithm, for example, aromatic ring acceptors and halogen atom acceptors.[49] These could certainly be included in the algorithm, though there would be a cost associated with the resulting increase in size of the search space.

Further improvements could be applied to the issue of conformational sampling. GAPE currently excludes certain torsional angles from generated conformations if those angles are not normally seen in small molecule crystal structures in the same fashion as GOLD[27] does. However, there are more sophisticated mechanisms[22,23] for incorporating the torsional distributions observed in small molecule crystal structures into GA operators whereby the histogram of a torsional distribution can be used to bias the angles generated by the mutation operator.

The algorithm contains a large number of parameters including GA parameters such as population size and operator weights and scoring function parameters such as the relative weights between the different components of the fitness score. For the most part, these parameters have been chosen using ad hoc choices or best guesses. It would be desirable to perform a more systematic search of parameter space to best determine parameter values.

Finally, the algorithm was relatively time-consuming. A considerable improvement in CPU time required could be achieved if the algorithm were recoded in C/C++, though clearly this would be a substantial undertaking.

ACKNOWLEDGMENT

The author thanks Dr. Robin Taylor and the Cambridge Crystallographic Data Centre for providing permission to use the torsional distributions used by GOLD for the purposes of noncommercial validation. The author also thanks Drs. Carleton Sage and Paul Watson for critical review and assistance in proof-reading. The molecular images reproduced here were created using the free version of Accelrys Discovery Studio.

Supporting Information Available: A table listing GAPE donor and acceptor types and a structure file in SD format containing the overlaid ligands for the crystallographically observed binding mode and the predictions for both regular and multiconformer GAPE for each test set. This material is available free of charge via the Internet at http://pubs.acs.org.

REFERENCES AND NOTES

(1) Hardy, L. W.; Malikayil, A. The impact of structure-guided drug design on clinical agents. *Curr. Drug Discovery* **2003**, *15*, 15–20.

(2) Walters, W. P.; Stahl, M. T.; Murko, M. A. Virtual screening: An overview. *Drug Discovery Today* **1998**, *3*, 160–178.

(3) Güner, O. F. *Pharmacophore perception, development, and use in drug design*; International University Line: La Jolla, CA, 2000.

(4) Bacilieri, M.; Moro, S. Ligand-based drug design methodologies in drug discovery process: An overview. *Curr. Drug Discovery Technol.* **2006**, *3*, 155–165.

(5) Lemmen, C.; Lengauer, T. Computational methods for the structural alignment of molecules. *J. Comput.-Aided Mol. Des.* **2000**, *14* (3), 215–232.

(6) Jain, A. N. Ligand-based structural hypotheses for virtual screening. *J. Med. Chem.* **2004**, *47* (4), 947–961.

(7) Martin, Y. C.; Bures, M. G.; Danaher, E. A.; DeLazzer, J.; 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.

(8) Barnum, D.; Greene, J.; Smellie, A.; Sprague, P. Identification of common functional configurations among molecules. *J. Chem. Inf. Comput. Sci.* **1996**, *36* (3), 563–571.

(9) Richmond, N. J.; Abrams, C. A.; Wolohan, P. R.; 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.

(10) Feng, J.; Sanil, A.; Young, S. S. PharmID: Pharmacophore identification using Gibbs sampling. *J. Chem. Inf. Model.* **2006**, *46* (3), 1352–1359.

(11) 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.

(12) Labute, P.; Williams, C.; Feher, M.; Sourial, E.; Schmidt, J. M. Flexible Alignment of Small Molecules. *J. Med. Chem.* **2001**, *44* (10), 1483–1490.

(13) Cho, S. J.; Sun, Y. FLAME: A program to flexibly align molecules. *J. Chem. Inf. Model.* **2006**, *46* (1), 298–306.

(14) Miller, M. D.; Fluder, E. M.; Castonguay, L. A.; Culberson, J. C.; Mosley, R. T.; Prendergast, K.; Kearsley, S. K.; Sheridan, R. P. MEGA-SQ: A method using the SQuEAL function to find the optimal superposition of several quasi-flexible molecules. *Med. Chem. Res.* **1999**, *9* (7/8), 513–534.

(15) Goldberg, D. E. *Genetic Algorithms in Search, Optimization, And Machine Learning*; Addison-Wesley Pub. Co: Reading, MA, 1989.

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

(17) Jones, G.; Gao, Y.; Sage, C. R. Elucidating molecular overlays from pairwise alignments using a genetic algorithm. *J. Chem. Inf. Model.* **2009**, *49* (7), 1847–1855.

(18) 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), 532–548.

(19) 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.

(20) 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.

(21) Cottrell, S. J.; Gillet, V. J.; Taylor, R. Incorporating partial matches within multi-objective pharmacophore identification. *J. Comput.-Aided Mol. Des.* **2006**, *20* (12), 735–749.

(22) Strizhev, A.; Abrahamian, E. J.; Choi, S.; Leonard, J. M.; Wolohan, P. R.; Clark, R. D. The effects of biasing torsional mutations in a conformational GA. *J. Chem. Inf. Model.* **2006**, *46* (4), 1862–1870.

(23) Gardiner, E. J.; Cosgrove, D. A.; Taylor, R.; Gillet, V. J. Multiobjective optimization of pharmacophore hypotheses: Bias toward low-energy conformations. *J. Chem. Inf. Model.* **2009**, *49* (12), 2761–2773.

(24) Mills, J. E.; Dean, P. M. Three-dimensional hydrogen-bond geometry and probability information from a crystal survey. *J. Comput.-Aided Mol. Des.* **1996**, *10* (6), 607–622.

(25) Jones, G.; Willett, P.; Glen, R. C. Molecular recognition of receptor sites using a genetic algorithm with a description of desolvation. *J. Mol. Biol.* **1995**, *245* (1), 43–53.

(26) Jones, G.; Willett, P.; Glen, R. C.; Leach, A. R.; Taylor, R. Development and validation of a genetic algorithm for flexible docking. *J. Mol. Biol.* **1997**, *267* (3), 727–748.

(27) Jones, G.; Willett, P.; Glen, R. C.; Leach, A. R.; Taylor, R. Further development of a genetic algorithm for ligand docking and its application to screening combinatorial libraries. In *ACS Symposium Series 719: Rational Drug Design*; American Chemical Society: Washington, DC, 1999; pp 255−270.

(28) Grant, J. A.; Pickup, B. T. A Guassian description of molecular shape. *J. Phys. Chem.* **1995**, *99*, 3503–3510.

(29) Dalby, A.; Nourse, J. G.; Hounshell, W. D.; Gushurst, A. K. I.; Grier, D. L.; Leland, B. A.; Laufer, J. Description of several chemical structure file formats used by computer programs developed at Molecular Design Limited. *J. Chem. Inf. Comput. Sci.* **1992**, *32* (3), 244–255.

(30) *Omega,* version 2.3.2; Openeye Scientific Software: Santa Fe, NM, 2007.

(31) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28* (1), 235–242.

(32) *SYBYL-X,* version 1.1; Tripos International: St. Louis, MO, 2010.

(33) Chen, Q.; Higgs, R. E.; Vieth, M. Geometric accuracy of three-dimensional molecular overlays. *J. Chem. Inf. Model.* **2006**, *46* (5), 1996–2002.

(34) Homer, R. W.; Swanson, J.; Jilek, R. J.; Hurst, T.; Clark, R. D. SYBYL line notation (SLN): A single notation to represent chemical structures, queries, reactions, and virtual libraries. *J. Chem. Inf. Model.* **2008**, *48* (12), 2294–2307.

(35) Zamora, A. An algorithm for finding the smallest set of smallest rings. *J. Chem. Inf. Comput. Sci.* **1976**, *16* (1), 40–43.

(36) Klebe, G.; Mietzner, T. A fast and efficient method to generate biologically relevant conformations. *J. Comput.-Aided Mol. Des.* **1994**, *8* (5), 583–606.

(37) Goto, H.; Osawa, E. Corner flapping: A simple and fast algorithm for exhaustive generation of ring conformations. *J. Am. Chem. Soc.* **1989**, *111*, 8950–8951.

(38) Payne, A. W.; Glen, R. C. Molecular recognition using a binary genetic search algorithm. *J. Mol. Graphics* **1993**, *11* (2), 74–3.

(39) Sanderson, P. N.; Glen, R. C.; Payne, A. W.; Hudson, B. D.; Heide, C.; Tranter, G. E.; Doyle, P. M.; Harris, C. J. Characterisation of the solution conformation of a cyclic RGD peptide analogue by NMR spectroscopy allied with a genetic algorithm approach and constrained molecular dynamics. *Int. J. Pept. Protein Res.* **1994**, *43* (6), 588–596.

(40) Grant, J. A.; Gallardo, M. A. A Fast method of molecular shape comparison: A simple application of a Gaussian description of molecular shape. *J. Comput. Chem.* **1996**, 1653–1666.

(41) *ROCS, version 2.3.1*; Openeye Scientific Software: Santa Fe, NM, 2007.

(42) Clark, M.; Cramer, R. D.; Van Opdenbosch, N. Validation of the general purpose Tripos 5.2 force field. *J. Comput. Chem.* **1989**, *10*, 982–1012.

(43) Jain, A. K.; Murty, M. N.; Flynn, P. J. Data clustering: A review. *ACM Computing Surveys* **1999**, *31*, 264–323.

(44) Lindholm, T.; Yellin, F. *The Java Virtual Machine Specification*; Addison-Wesley: Reading, MA, 1996.

(45) Mattos, C.; Rasmussen, B.; Ding, X.; Petsko, G. A.; Ringe, D. Analogous inhibitors of elastase do not always bind analogously. *Nat. Struct. Biol.* **1994**, *1* (1), 55–58.

(46) Kubinyi, H.; Folkers, G.; Martin, Y. C. *3D QSAR in Drug Design*; Kluwer/ESCOM: Dordrecht, 1998.

(47) Cramer, R. D.; Patterson, D. E.; Bunce, J. D. Comparative molecular field analysis (CoMFA). 1. Effect of shape on binding of steroids to carrier proteins. *J. Am. Chem. Soc.* **1988**, *110*, 5959–5967.

(48) Kirchmair, J.; Laggner, C.; Wolber, G.; Langer, T. Comparative analysis of protein-bound ligand conformations with respect to catalyst's conformational space subsampling algorithms. *J. Chem. Inf. Model.* **2005**, *45* (2), 422–430.

(49) Schwobel, J.; Ebert, R. U.; Kuhne, R.; Schuurmann, G. Prediction of the intrinsic hydrogen bond acceptor strength of organic compounds by local molecular parameters. *J. Chem. Inf. Model.* **2009**, *49* (4), 956–962.

CI100194K