Elucidating Molecular Overlays from Pairwise Alignments Using a Genetic Algorithm

Gareth Jones,* Yinghong Gao, and Carleton R. Sage

Arena Pharmaceuticals, 6166 Nancy Ridge Drive, San Diego, California 92121

Received March 24, 2009

Inferring the relative bioactive poses between active molecules is a common problem in drug discovery. The use of rapid pairwise alignment algorithms in conjunction with rigid conformer libraries has become a prevalent approach to this problem. These programs can be easily used to compare two molecules or suggest alternatives to a single known active. However, it is not obvious how to combine pairwise alignments between multiple actives into an overlay that reproduces the binding mode of those actives in the target receptor. We describe a new algorithm, DIFGAPE (DIstance geometry Focused Genetic Algorithm Pose Evaluator) that, given pairwise alignments of conformers of active compounds, attempts to reproduce overlays of ligand binding modes. The software was evaluated on 13 test systems from 9 protein targets using associated ligands extracted from the PDB. Starting from 2D ligand structures with no protein information, we were able in 4 systems to approximate the crystallographically observed binding mode. For example, the prediction for a set of 11 ligands targeting FXa had 1.6 Å rmsd to crystal structure coordinates. Finally, the evaluation illustrated current challenges for molecular conformer generators and pairwise alignment algorithms.

INTRODUCTION

The analysis of ligands in the binding site of an experimentally determined target structure has helped drive the design and development of numerous drugs.[1] The computational exploitation of such an analysis has been an important tool in drug discovery and development.[2] However, in the absence of a high resolution target/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. Using a set of input molecules, many tools exist to help derive a hypothesis representing a model for the important features involved in protein−ligand interactions.[3] Key among these tools is the ability to superimpose a pair of molecules to maximize similarity.

Many 3D molecular alignment techniques, encompassing a wide variety of algorithms, have been developed to generate such hypotheses.[3,4] Several of these methods are capable of performing multiple alignments, such as Surflex-Sim,[5] DISCO,[6] Catalyst,[7] and others.[8−11] Multiple alignment algorithms attempt to elucidate ligand-binding modes by superimposing a set of ligands. This is a computationally demanding procedure requiring the combined search of ligand conformational space and available superpositions. As such, these algorithms often have difficulty elucidating binding hypotheses for more than five or six compounds. Other alignment algorithms such as FLEXS,[12] SEAL,[13] SUPERPOSE,[14] SQ,[15] and ROCS[16] are capable of aligning only pairs of structures. In general, pairwise alignment, especially for rigid conformers, is much more tractable and computationally less expensive than multiple alignment.

In this work, we set out to develop a process that would create a 3D overlay from 2D ligand structures, using the intermediate steps of creating a conformer library and generating pairwise alignments for all conformers. The final step in the process involved our algorithm creating a predictive overlay from the pairwise alignments. In order to validate the predicted overlays we attempted to recreate the crystallographically observed alignment of molecules in a target structure (generated by aligning the proteins themselves). The experimental procedure is outlined in Figure 1. It was clear to us that the creation of predicted overlays, which reproduced the observed binding overlay, would only be possible if the pairwise alignment method correctly aligned the bound conformations. Thus the validation was split into two tasks. First, using pairwise alignments of the bound conformations of the ligands in the crystal structure, we determined if we could reproduce the observed binding overlay. If we were successful in this first task, only then did we undertake a second more challenging experiment. In this second experiment, we used the 2D chemical structures of the ligands as inputs to a conformer generator and pairwise alignment method and then determined if we could predict the crystal structure binding mode. The second experiment is representative of working with a set of structures in the absence of an experimentally determined binding site.

DIFGAPE (DIstance geometry Focused Genetic Algorithm Pose Evaluator) is a tool that scores and aligns pairs of rigid conformers to generate binding mode overlays. Starting with a set of 2D active compounds we generated 3D conformer libraries for each compound. Next, using a pairwise alignment tool, we exhaustively created and scored overlays of all pairs of conformers. Finally, using the genetic algorithm (GA) described herein, we created a superposition of the ligands. This was achieved by using a GA scoring function that selected ligand conformers by optimizing the pairwise scores (as generated by the alignment tool) while penalizing inconsistent geometries using distance geometry constraints. In cases where no conformation for a ligand could be

* Corresponding author phone: (858)453-7200; e-mail: gjones@arenapharm.com.

10.1021/ci900109n CCC: $40.75
Published on Web 06/18/2009

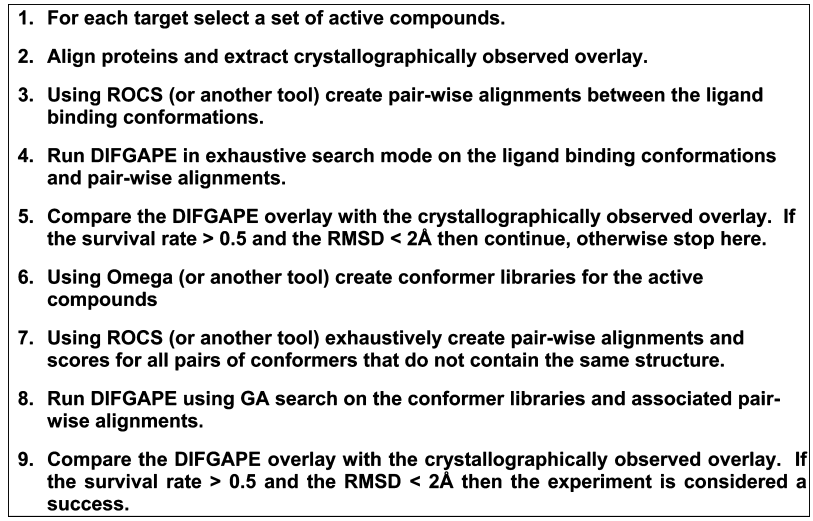

Figure 1. Outline of the experimental procedure.

reasonably incorporated into the overlay, the GA had the option of dropping that ligand from the superposition. Following the selection of conformers by the GA, an overlay of all remaining ligands was created using the pairwise alignments and a consensus procedure. We used software from Openeye Scientific Software to perform conformer generation (Omega[17]) and pairwise alignment (ROCS[16]). However, the algorithm described herein is independent of the choice of conformer generator and alignment method: we required only a conformer library and an alignment tool that also generates a score or some other similarity metric.

In order to stringently validate our approach, we tested a number of sets of ligands for which crystallographically observed binding modes were available in the Protein Databank (PDB).[18] Benchmark coordinates for the bound ligands were obtained by superimposing the proteins. Using least-squares fitting we could then compare the results of our predictions with those observed in the crystal structures. It was found that, starting with 2D ligand coordinates and not using any protein structure, DIFGAPE could create overlays which were comparable to the crystal structure alignments (atomic rmsd < 2 Å). As might be expected, the method struggled to reproduce observed binding modes when the ligands were large and flexible or when multiple binding modes were observed in the crystal structures. In these cases not only was the GA search space large but also the binding mode was less likely to be included in the conformational library, and, even if the binding modes were present for a particular pair of actives, the corresponding pairwise alignment was less likely to be correct.

MATERIALS AND METHODS

Selection of Protein−Ligand Complex Test Systems. We generated 13 test data sets of protein−ligand complexes ranging in size from four to 13 ligands. These data sets were from nine different proteins representing five protein families (Table 1). 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 molecular overlay software.[19] To complement this selection, we added one test system from dihydrofolate reductase complexes present in the PDB (DHFR) and two test systems from Factor Xa complexes present in the PDB (FXa). Given the size of the search space and memory limitations we chose not to run DIFGAPE on more than 13 ligands. (The largest system evaluated all pairs of 1632 conformers and used 7.1 GB of memory, though had we used fewer conformers it would have been possible to evaluate data sets containing more molecules). For the four proteins which had more than 13 protein−ligand complexes available, we used a variety of data set reduction techniques: primarily, we used MOE to both cluster using MACCS keys and perform diverse subset selection on the clustered compounds. For the HIV protease test systems, we chose to investigate the effect of molecular size and flexibility on these peptidic ligands: HIV_Div_MW (8 compounds) was created by selecting compounds with a molecular weight less than 600 from HIV_Div; HIV_Div_MW_RB (four compounds) was generated by selecting compounds with fewer than 20 rotatable bonds from the diverse collection (HIV_Div_MW). Where there were enough compounds of the same structural class we created focused data sets, which could be considered representative of compounds within an SAR series. PDB codes and an SDF file containing the aligned crystallographic conformations and test set designations for the 118 ligands present in our test data set are available in the Supporting Information.

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			13
HIV_Div_MW	HIV1 protease	hydrolase (acid protease)	8
HIV_Div_MW_RB			4
p38	p38	transferase (kinase)	12
rhinovirus	rhinovirus	virus	8
trypsin	trypsin	hydrolase	7

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 2. Operator-based GA.

Generation of Reference Alignments. To create reference crystallographic ligand overlays, multiple alignments of the protein structures of each target were performed using the MOE-Align method.[20] MOE-Align is capable of performing multiple structural alignments of protein chains using a procedure guided by both sequence alignment and 3-dimensional coordinate superpositions. In some instances MOE-Align failed on a set of protein structures: in those instances the problem complexes were identified and removed from the set, and MOE-align was run on the remaining structures. The problem complexes were then added back in using MOE Superpose,[20] which performed pairwise alignment with one randomly chosen protein in the multiple alignment in conjunction with any necessary manual adjustments (such as chain merging or chain deletion).

Following the generation of aligned complexes the ligands were extracted to create reference overlays. If necessary, the structures were manually processed to assign correct bond orders and formal charges. Fragmented, ambiguous, or distorted ligands were discarded.

Generation of Conformer Libraries. Omega 2.2.1[17] was used with default parameters except for the MAXCONFS (maximum number of conformations) setting. Settings of 20, 50, and 150 were evaluated. In practice, little improvement was observed when using 150 conformations. Unless stated otherwise, the results presented here are for MAXCONFS equals 50. We made the conscious decision not to use the Openeye program Flipper, which enumerates stereocenters, and thus used the ligand stereochemistry present in the crystal structure.

Generation of Pairwise Alignments. ROCS 2.3.1[16] was used to generate and output all possible pairwise alignments for each conformer pair using the default parameters with the exception of MAXCONFS, MAXHITS, and BESTHITS parameters which were all set to 10,000, and a similarity cutoff of 0 was used. ROCS provides three similarity metrics: Tanimoto, Combo, and Tversky. While differences were observed between the different metrics, no one metric was found to have superior performance. As such, only combo score results are reported.

Genetic Algorithm. A Genetic Algorithm (GA) is a computer program that mimics the process of evolution by manipulating a collection of data structures called chromosomes. Each of these chromosomes encodes a possible solution, i.e. a selection of molecular conformers, to the molecular overlay problem and may be assigned a fitness score based on the relative merit of that conformer selection. A steady state with no duplicates operator based GA,[9,21,22] was used to search possible conformer selections. This GA is shown in Figure 2.

To select operator parents we used normalized rank-based linear selection[21] with a selection pressure of 1.0001. Selection pressure is the ratio of the normalized and linearly scaled fitness score for the best chromosome to the scaled fitness score for the least-fit chromosome. 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.[21,23] 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. An island model was employed where 5 subpopulations of 100 chromosomes evolved independently.[21]

The GA started with every chromosome in each island created with random values. The GA then iterated over each island applying a total of 60000 genetic operators. The genetic operators available were crossover, mutation, and migration. In any iteration a genetic operator was selected using roulette wheel selection[21,23] 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 which were chosen using roulette wheel parent selection. The operators produced a number of children which replaced the worst individuals in the island.

The GA encoded the problem as an integer string (see below). The crossover operator employed was two-point crossover.[21,23] Here, two (different) cross points are selected at random along the length of the string. The crossover operator took two parents and produced two children. In the first child chromosome the string was identical to the first parent before the first cross-point and the first parent after the second cross-point. In between cross-points it was identical to the second parent. Conversely, the second child was identical to the first parent between the cross points and the second parent otherwise. This two-point crossover is analogous to representing the chromosome as a circular data structure (with the end of the string attached to the beginning) and having the crossover operator exchange a section between parents to produce the children.

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. For each position on

the string the probability of perturbation was $1/l$ (where l was the length of the string). Each position on the integer string was tested in turn. In the event that a position tested for mutation, the value at that position was then set to another allowed value (including the null value) with equal probability. If, after processing the entire string, no perturbation was applied, the entire operation was repeated until at least one perturbation occurred.

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

Chromosome Encoding. Before describing the chromosome encoding used, we define the following: the number of bioactive compounds is n; the number of conformers for molecule x is n_x; and conformer number i for molecule x is $c_{x,i}$. The set of conformers used is $\{C_1, C_2, ...C_n\}$, where $C_x = \{c_{x,0}, c_{x,1}...c_{x,n_x}\}$.

The chromosome used in DIFGAPE was an integer string of length n. Let V be a chromosome, $V = \{v_1, v_2... v_n\}$, such that $0 \leq v_i \leq n_i$. If v_i was greater than 0, then we selected conformer number v_i for molecule i (c_{i,v_i}). Thus, the chromosome encoded a set of conformers with at most 1 conformer per molecule.

For example, suppose we had a problem instance of 4 molecules with 20 conformers per molecule. The integer string chromosome {2,13,0,9} resulted in selecting conformer 2 for molecule 1, conformer 13 for molecule 2, and conformer 9 for molecule 4 (with no conformer being selected for molecule 3). Here there were 21 choices for each molecule: 20 conformers and the absence of the molecule from the overlay. Thus the total number of different conformer selections (and consequently chromosomes) for this problem instance was 194,481 (21^4). For 10 molecules with 50 conformers we had 1.2E17 (51^{10}) possible conformer selections, illustrating the combinatorial nature of the problem.

Given this chromosome encoding, it was easy to create random integer strings to populate the GA. Also, the application of the genetic operators described above always generated valid conformer selection encodings. Any chromosome created by the GA could be readily decoded to generate a set of molecular conformations which was scored by the fitness function.

Fitness Function. The input to the fitness function consisted of a set of molecular conformations, with at most one conformation from each molecule. Let CF be this set of conformations, where $CF = \{c_1, c_2... c_m\}$ and $m \leq n$. Given that we have pairwise similarity scores for every conformer pair, one possible fitness score was simply the sum of similarities (np is the number of similarity scores evaluated and is used as a normalization term)

$$similarity\_score = \frac{1}{np} \cdot \sum_{i=1}^{m} \sum_{j=i+1}^{m} similarity(c_i, c_j)$$

The issue with this simple score was that the final objective of the program was to predict the binding mode of the molecules from the selected conformers. However, the pairwise alignments may be inconsistent with the global alignment. Figure 3 illustrates the problem. Here we have three conformers and three pairwise alignments. We can see

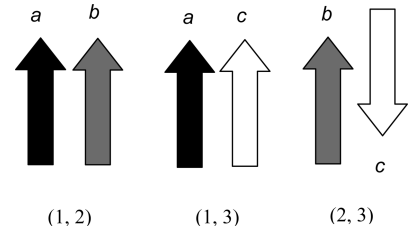

Figure 3. Hypothetical alignments of three conformers.

that the first two alignments expect all three molecules to be "pointing" upward, but the third pair expects either molecule 2 or 3 to be "pointing" down. It is not going to be possible to overlay all three molecules in a manner consistent with the three pairwise alignments.

In order to correct for these discrepancies we incorporate triangle inequalities into our scoring function. Triangle inequalities are commonly used in distance geometry algorithms[24] and utilize the property that the longest side of a triangle must be less than the sum of the two shorter sides. Suppose we have an atom, a, in conformer 1 and an atom, b, in conformer 2 and an atom, c, in conformer 3. From the three pairwise alignments we can determine the three distances ab, ac, and ca which (if we are to have any chance of aligning all three conformers such that these three pairwise alignments are consistent) should conform to the triangle inequality constraint.

Returning to Figure 3, we can see that in the pairwise alignment between conformers 1 and 2 the distance between atoms a and b is 0; in the alignment between conformers 1 and 3 the distance between atoms a and c is 0; whereas in the alignment between conformers 2 and 3 the distance between atoms b and c is clearly greater that 0. In this case the longest distance (bc) is much greater than the sum of the two shorter distances ($ab + bc$) and the triangle inequality constraint is broken.

In order to penalize violations of the triangle constraint we define the following function: $tri\_penalty(a_{i,x}, a_{j,y}, a_{k,z})$ $x \neq y \neq z$, where $a_{i,x}$ is heteroatom number i in molecular conformer x. (In this algorithm we used heteroatoms for determining distance constraints, but the technique could easily be extended to other atoms or to any 3D feature). Using the pairwise alignment for conformers x, y, and z we determined the three distances ($a_{i,x}$, $a_{j,y}$), ($a_{i,x}$, $a_{k,z}$), and ($a_{j,y}$, $a_{k,z}$). Let $diff$ be the difference between the largest distance and the sum of the other two distances. If $diff < 0$ then $tri\_penalty$ returned 0, otherwise the function returned $diff^2$. The penalty for a chromosome was $tri\_penalty$ summed across all conformer triplets and heteroatom triplets. If k_x is the number of heteroatoms in conformer x and nt (used as a normalization term) is the total number of triplets evaluated then

$$penalty = \frac{1}{nt} \cdot \sum_{x=1}^{m} \sum_{y=x+1}^{m} \sum_{z=y+1}^{m} \sum_{i=1}^{k_x} \sum_{j=1}^{k_y} \sum_{k=1}^{k_z} tri\_penalty(a_{i,x}, a_{j,y}, a_{k,z})$$

The final chromosome fitness score was

$$fitness = similarity\_score - penalty$$

In many systems it was not possible to select conformers from all molecules and get a reasonable penalty score. For

this reason the chromosome encoding allows for molecules to be left out of the conformer selection.

The summation in the penalty term scales as n^3. However, it was not a limiting term in the algorithm. Rather, during testing, the limitations of the algorithm proved to be the size of the search space, rather than the time spent evaluating penalty terms.

Generation of Molecular Overlay. At the termination of the GA run we were left with a set of molecular conformations. Using the notation introduced above, when describing the fitness function, let these conformers be $CF = \{c_1, c_2 ... c_m\}$. From this we needed to generate an overlay of the conformations that predicted binding. This was achieved using this procedure.

For each conformer we created an overlay using the pairwise alignments. We illustrate this for conformer c_1. First we retrieved the pairwise alignment for (c_1, c_2) and used this to align c_2 to c_1; next we retrieved the pairwise alignment for (c_1, c_3) and used this to align c_3 to c_1, so that we now had an overlay of c_1, c_2 and c_3; we repeated this for the remaining conformers through c_m giving us an overlay of all conformers based on c_1. This whole process was repeated for each of the other conformers to give us m overlays.

For each conformer the average similarity score with the other conformers was determined. The overlay based on the conformer with the highest average similarity score was chosen as a base overlay. While this base overlay could have been used as the final prediction, it showed considerable bias to the starting conformer and might not represent properly the ROCS alignments between conformer pairs that did not include the starting conformer. For this reason, the following averaging and consensus procedure was performed.

Each of the other overlays was then least-squares fitted onto this base overlay. Following this all m overlays should be commonly aligned so that we had m sets of coordinates for c_1, m sets of coordinates for conformer c_2, and so on.

For each conformer we now generated consensus coordinates. Using the m sets of coordinates we determined average atomic coordinates, for that conformer. We then checked to see if the averaged coordinates represent a distorted structure. If the structure was distorted, then the molecular coordinates that were most distant from the average coordinates were removed, and we regenerated the average coordinates using the remaining $m-1$ sets of coordinates. This process was repeated until the average coordinates were not distorted.

In order to determine if averaged coordinates were distorted we looked at all interatom distances that were separated by a shortest path of at least 5 bonds. Each such distance was compared with the distance between the same atoms in the original undistorted conformer and the absolute distance difference calculated. The average structure was considered distorted if the mean distance difference for these paths was greater than 0.25 Å.

Finally, the consensus overlay was output.

Exhaustive Search. In the cases where we were simply evaluating crystal structure conformations, an exhaustive search algorithm was employed. For example, if we had a system with 10 ligands, each ligand is either present in its crystal structure conformation or missing from the final overlay. This gave only 1024 permutations. We performed a depth-first search and retained the permutation that had the best fitness score.

Implementation. DIFGAPE was coded in Java 1.5.[25] In house benchmarking on clique detection and clustering algorithms indicates that Java programs are likely to be 2 to 10 times slower than C/C++. However, we have also found that Java provides a much more rapid software development cycle and results in highly stable code.

DIFGAPE was encoded in two applications. The first application processed all ROCS results and extracted scores, distances between heteroatoms and created file look-up tables for retrieving full pairwise alignments. This information was then stored on the file-system as a serialized Java object. The second application read the Java object and ran the GA.

The rationale for preprocessing ROCS results was that this was an extremely time-consuming process. By doing this up-front and caching the results we could easily perform numerous GA runs.

Method Validation. DIFGAPE creates an overlay for a set of structures given pairwise alignments between conformers, using a scoring function to select conformer pairs with high alignment scores while penalizing conformer selections that violate distance geometry constraints. The DIFGAPE scoring function also allows for the rejection of structures, which are then dropped from the overlay.

We performed two experiments in the validation. In the first experiment, DIFGAPE was run in exhaustive search mode. ROCS pairwise alignments and scores for the single crystal structure conformers for a given data set were used as input. A search of all alignments was performed to create the best overlay for the compounds in the data set. If this overlay was considered a success (see below) we proceeded to the second experiment.

In the second experiment, DIFGAPE was run in GA search mode. Starting from 2D structures, a conformer library was created for each input structure and alignments for all pairs of conformers were generated. DIFGAPE then used a GA to select conformers for inclusion in the final overlay. The resulting DIFGAPE overlays were evaluated as described below. For the FXa_Focused data set, which comprised 11 compounds, 1620 conformers, and 2.5 million pairwise alignments, the Omega conformer generation (50 conformers per structure) was done in 43 s, the ROCS alignment took 22 min, and the DIFGAPE run took 44 min on a Linux workstation with an AMD Opteron 246 1 GHz processor and 8GB memory. For the DIFGAPE application, memory usage scaled with the number of conformations (and therefore number of pair wise alignments). For the FXa_Focused data set example, 7.1GB of memory on a 64-bit operating system was required.

DIFGAPE overlays were evaluated using two metrics: an rmsd and a survival rate. First the overlay was superimposed using least-squares fitting onto the crystal structure overlay and a mean atomic rmsd determined. The superposition of the DIFGAPE overlay onto the crystal structure overlay and the determination of rmsd distances were calculated using in-house programs that accounted for molecular group symmetries. Next, the survival rate (the proportion of input ligands included in the final DIFGAPE overlay) for the alignment was considered. A DIFGAPE alignment was considered a success if atomic rmsd < 2.0 Å and survival

Table 2. DIFGAPE Results Using Crystal Structure Bound Conformers

data set	rmsd (Å)	no. of input ligands	no. of output ligands	survival rate	pass
CDK2_Focused	0.58	9	5	0.56	yes
CDK2_Diverse	2.81	10	3	0.30	no
DHFR	0.46	12	7	0.58	yes
Elastase	1.49	5	4	0.8	yes
ESR1	0.34	13	7	0.54	yes
FXa_Focused	0.39	11	10	0.91	yes
FXa_Diverse	1.04	8	6	0.75	yes
HIV_Div_MW_RB	0.3	4	2	0.50	yes
HIV_Div_MW	0.67	8	3	0.38	no
HIV_Div	2.81	13	4	0.31	no
P38	1.04	12	4	0.33	no
Rhinovirus	4.56	8	6	0.75	no
Trypsin	1.12	7	5	0.71	yes

rate ≥ 0.5 (meaning at least half the input structures appeared in the final overlay).

RESULTS AND DISCUSSION

Data sets were evaluated as described in Figure 1, and the results are analyzed here.

Overlay Generation from Crystal Structure Conformations. In our first experiment with DIFGAPE we performed an exhaustive analysis of all pairwise alignments of the crystal structure conformations for each data set to yield a consensus overlay. Successful results were achieved for 62% of the data sets (8 of 13) and for 78% of the targets (7 of 9). Individual data set results are summarized in Table 2.

DIFGAPE had mixed results for the two CDK test sets, passing the performance criteria for the CDK_Focused set (lactam core containing) but not the diverse set. The second test appears particularly challenging, with some ligands forming hydrogen bonds to the protein backbone and other ligands apparently forming purely hydrophobic interactions with little or no hydrogen bonding and is likely to prove difficult for any alignment tool. ESR1 survived the first pass analysis, but a closer inspection shows two distinct shapes in the input ligands. As shown in Figure 4 the input ligands can be partitioned into two classes, A and B, where A contains much smaller shapes than B. Figure 5b shows that, while DIFGAPE makes our criteria for success for this data set, it is only able to align the larger compounds from class B. The best overall performance was observed for FXa, and the alignments for both FXa data sets are shown in Figure 6 (1b, 2b). The technique performed relatively poorly on the HIV protease inhibitors, passing only for the simpler HIV_Div_MW_RB test set. Although the data set HIV_DIV_MW of smaller ligands also failed on survival rate, a reasonable alignment was generated for three of 8 compounds.

With a flexible active site P38 presented a challenging test case.[26] Examination of the crystal structure overlay shows considerable variation in the protein around the active site with residues TYR35, MET109, and ASP168 being particularly mobile. While we might expect this variation to impact the ability of DIFGAPE to reproduce the crystallographic binding mode it was difficult to identify a causal relationship between DIFGAPE failure and the variation in the protein active site. Additionally, it can be seen from Figure 7a that there is considerable variability in ligand binding with many different ligand shapes.[27] While DIFGAPE failed the survival rate test for this data set, it does produce a good alignment for four compounds (Figure 7b). This indicates that, even when the algorithm appears to fail, potentially useful information can be extracted from the final alignment.

Figure 8 shows the failed prediction for the rhinovirus set. The DIFGAPE alignment appears acceptable, but a high rmsd was obtained because 2 of the 6 compounds in the overlay were in the reverse orientation from that observed in the crystal structure. However, it is not beyond the bounds of possibility that these nearly symmetrical linear ligands were incorrectly fitted when the crystal structures were resolved or that the ligands can bind in either orientation (there is no little or no polar component to ligand binding, so it is not obvious if either orientation is preferred).

Overlay Generation using Conformer Libraries. For the data sets (8 out of 13) which DIFGAPE reproduced the crystal structure overlay using bound conformers, we attempted to recreate crystal structure overlays using only the 2D structures of the test set compounds as input. Conformer libraries were generated using Omega, and all pairwise alignments were generated using ROCS. The performance of the resultant DIFGAPE overlays from ROCS alignments are shown in Table 3. Success was obtained for the four of eight data sets: FXa_Focused, FXa_Diverse, CDK_Focused, and trypsin. A successful example, for the FXa sets, is shown in Figure 6 (1c, 2c). Conversely, DIFGAPE failed to give good predictions for DHFR, elastase, ESR1, and HIV_Div_MW_RB, despite succeeding with the bound conformations.

It is noticeable in Table 3 that the survival rates for conformer libraries are always higher than those observed using single bound conformations. Given a large number of conformations, the GA was likely to find a single conformation to fit the current overlay. Conversely, this was not possible when we only had a single conformation with pairwise alignments inconsistent with the current overlay. While survival rates improved in all systems, rmsd values were all worse. A number of reasons may be hypothesized for the poorer rmsd values, and we provide four examples. First, it may be partly due to the improved survival rate: while more compounds were fitted into the overlay, compared to the binding conformation study, those extra compounds were likely to have a different conformation to the binding mode. Second, the binding conformation for a ligand may not be present in the conformer library for that ligand. Third, the search space of available conformers is extremely large, and, while the correct conformers may be present, the GA could have been unable to find this solution. Fourth and finally, through some artifact of the alignment and scoring functions, there may be a false optimum solution.

While the effects of most of these factors were hard to quantify, it was easy to identify the member of a conformer library closest to the bound ligand conformation. This is shown in Table 4. For each bound conformer, the rmsd between that bound conformation and the closest conformation in the conformer library generated by Omega was determined and the results binned by rmsd. This procedure was repeated for Omega MAXCONFS (maximum number of conformations) settings of 20, 50, and 150. It was observed that for a MAXCONFS setting of 50, 69% of ligands had the binding conformation contained in their conformer library (assuming a cutoff of 1 Å to determine equivalence between a conformer and the binding mode). Unfortunately, that left

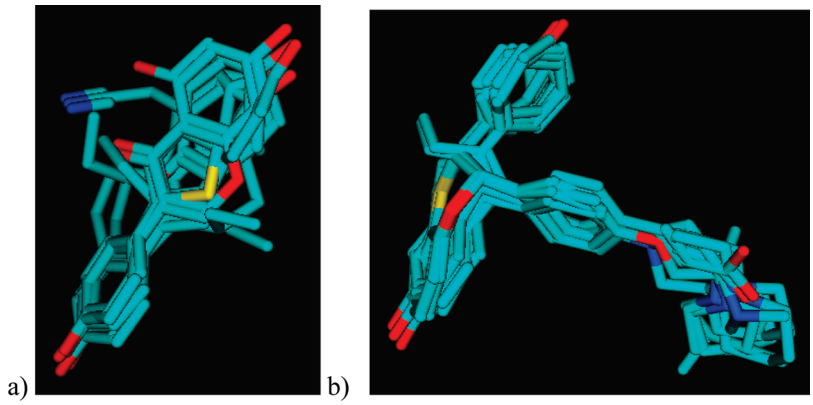

Figure 4. Two distinct shapes among the ESR1 set: a) Class A and b) Class B.

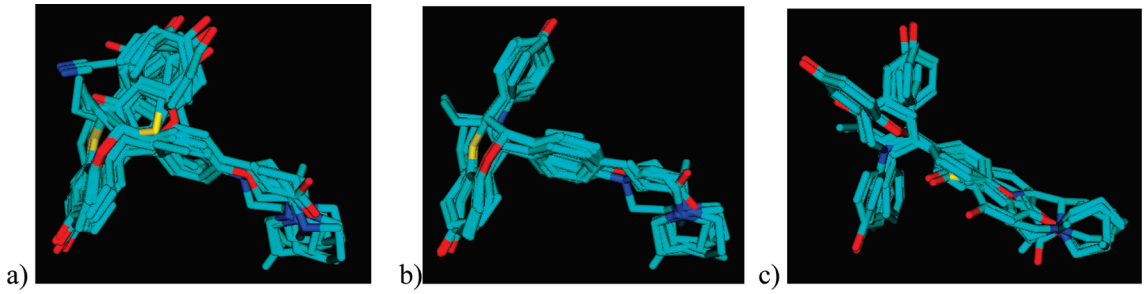

Figure 5. ESR1 set: a) crystal structure overlay, b) DIFGAPE overlay using crystal conformers, and c) DIFGAPE overlay using conformer libraries.

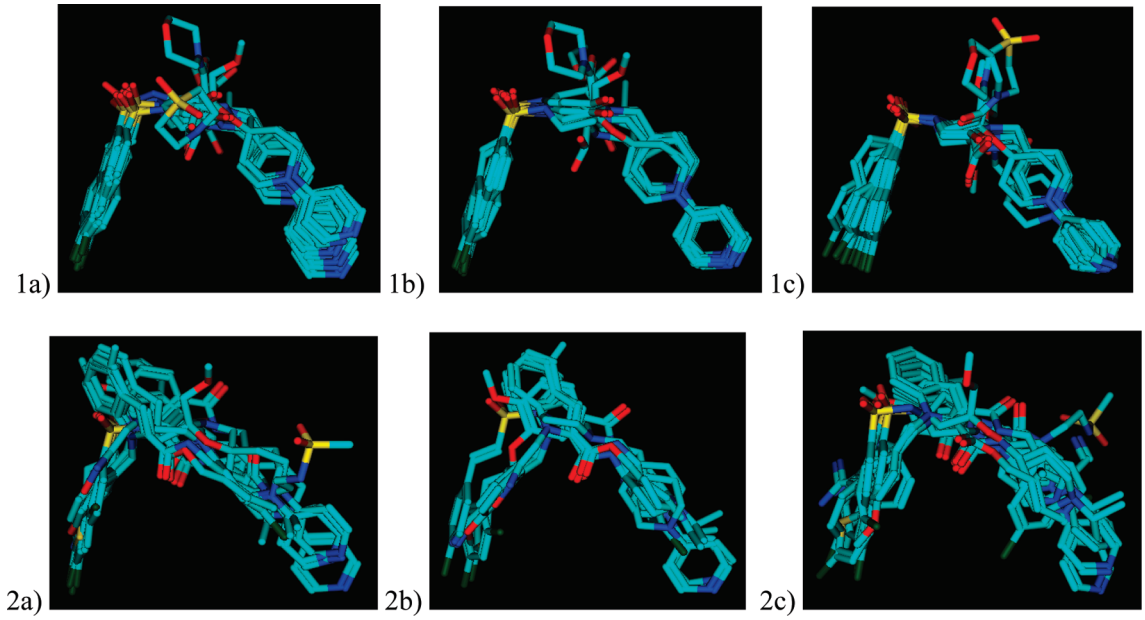

Figure 6. FXa_Focused (1a-1c) and FXa_Diverse Sets (2a-2c): a) crystal structure overlay, b) DIFGAPE overlay using bound conformations, and c) DIFGAPE overlay using conformational libraries.

31% of ligands that did not have the binding mode in their set of conformers and 8% which had no conformer within an rmsd of 2 Å of the binding mode. Interestingly, using a MAXCONFS parameter setting of 20 did not produce a significant deterioration in these numbers nor did a setting of 150 produce a significant improvement.

In the case of ESR1, failure could be attributed to the binding mode being missing from the conformer libraries. It can be seen in Figure 5c that the DIFGAPE alignment has 3 ligands (from PDB complexes 1SJO, 1XP1, and 1XP9) with a misaligned group. This group is axial to the core of the main structure in the bound conformation and equatorial in the DIFGAPE prediction for the conformer library. In fact, using Omega, we were unable to create conformations with this group in the axial position.

Both the elastase and HIV_Div_MW_RB test sets contain large peptidic compounds that are conformationally challenging. While we were successful in generating an overlap with bound conformers we did not succeed with the conformer library. In the case of elastase there were two ligands, from PDB complexes 1ELB and 1ELC, that had unusual binding modes (relative to the other compounds) in the crystal structure. Additionally, the closest conformer to the binding mode generated by Omega for 1ELC was 2.2

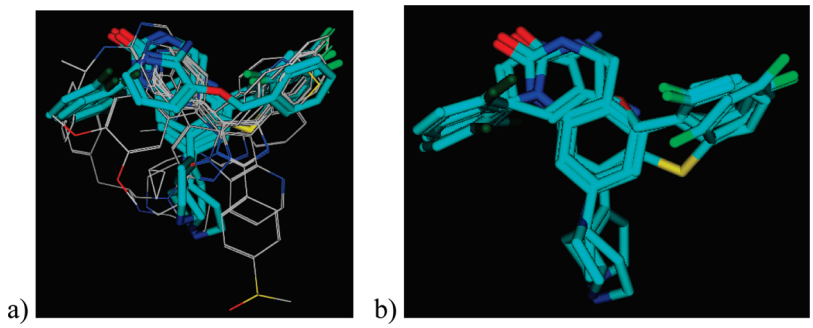

Figure 7. P38 set: a) crystal-structure based alignment of the whole set [Those compounds included in the DIFGAPE overlay using bound conformers are highlighted in cyan.] and b) DIFGAPE overlay using bound crystal conformers.

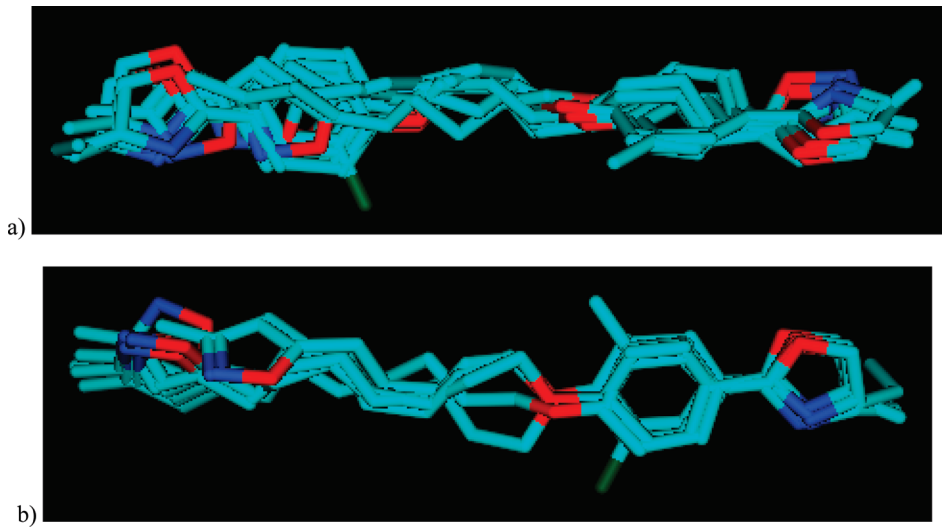

Figure 8. Rhinovirus set: a) crystal structure alignment and b) DIFGAPE alignment using binding mode conformations.

Table 3. DIFGAPE Overlay Results Using Conformer Libraries

data set	rmsd (Å)	no. of input ligands	no. of output ligands	survival rate	pass
CDK2_Focused	1.81	9	6	0.67	yes
DHFR	2.71	12	7	0.58	no
elastase	3.23	5	5	1.00	no
ESR1	2.46	13	9	0.69	no
FXa_Focused	1.61	11	11	1.00	yes
FXa_Diverse	1.84	8	8	1.00	yes
HIV_Div_MW_RB	3.54	4	3	0.75	no
trypsin	1.47	7	5	0.71	yes

Table 4. Counts of Closest Omega Ligand Conformer to Bound Conformation by rmsd Range

	Omega MAXCONFS setting		
rmsd range (Å)	20	50	150
0−1	78	82	91
1−2	28	27	20
2−3	9	8	6
3−4	3	1	1
>4	0	0	0

Å. 1ELC was not included in the overlay created by DIFGAPE using bound conformations. Both 1ELB and 1ELC were included and positioned incorrectly in the second DIFGAPE experiment using conformer libraries. For the HIV_Div_MW_RB set, we were barely successful with bound conformers, and, in the conformer library experiment, all input conformers differed by over 1 Å rmsd from the corresponding bound conformation.

DHFR is the final test that was successful in the case of bound conformers but failed when using conformer libraries. In this case, it appears the increased number of available conformations lead the GA to a false solution. Using a conformer library created with MAXCONFS set to 20, we obtained similar results to the bound conformation experiment (the same survival rate, but with a poorer rmsd of 1.3 Å). It is worth noting that the DIFGAPE scoring function does not include a correction for high energy conformers and perhaps such a term might have helped in this instance.

Finally, it was noted that in most cases the overlays created by DIFGAPE appeared to be much tighter that the crystallographically observed ligand binding modes. This is not surprising as the scoring function is primarily driven by the shape-based overlay term from ROCS. In contrast, on binding to the protein, ligands form hydrophobic interactions with the protein surface which may not result in a tight overlay, especially if the protein is mobile.

CONCLUSIONS

We have presented a methodology to create quality overlays of multiple ligands starting from 2D structures by combining the results from exhaustive pairwise alignments. DIFGAPE takes as input pairwise alignments between ligand conformations together with the scores for those alignments. It then selects conformers using a scoring function that maximizes pairwise scores and penalizes geometric inconsistencies through a distance geometry term. A consensus method was then used to create an overlay from the pairwise alignments.

Using a number of test systems extracted from the PDB we have demonstrated applicability using DIFGAPE in conjunction with the Omega conformer generator and ROCS pairwise alignment tool. In 4 of 13 test systems we were able to produce reasonable approximations of the crystallographically observed binding mode from 2D structures without using any protein structure. Given the difficulty in predicting ligand binding modes in the absence of protein structure this is a particularly encouraging result, and we believe this method is a powerful tool for discovery programs which have active compounds but no protein structure. These overlays may be used as a starting point for 3D-QSAR[28] methods, such as CoMFA,[29] which require an initial alignment of the molecules that are to be analyzed.

The algorithm can be used in conjunction with any fast pairwise alignment tool and conformer generator. However, the final results are very much dependent on the choice of such tools. In 5 of our test systems we did not test using conformer libraries as the use of ROCS pairwise alignments did not allow significant recreation of the binding site overlay. In 3 of the remaining 8 test systems, the absence of binding mode conformations in the conformer libraries created by Omega limited the ability of DIFGAPE to reproduce the observed binding mode.

While effective, the algorithm could benefit from some improvements. For example, there is no term to penalize the selection of high energy conformers. Since most conformer generators (including Omega) include conformer energy in their output this would be a simple change to make. A more difficult (though potentially more useful) change would be to accommodate multiple alignments between conformer pairs. For example, ROCS can be configured to create multiple alignments between the two input structures each with a different score. The GA would then be free to select different alignments between conformers in addition to the conformers themselves. Such a change could be a significant improvement to the algorithm but would require large changes to the GA encoding (but not the scoring function) and would increase the search space dramatically.

ACKNOWLEDGMENT

We thank Dr. Paul Watson for critical review and assistance in proof-reading.

Supporting Information Available: Test set designations for the 103 unique ligands present in our test data set. For each test system: PDB codes of all complexes used, SDF files containing the crystallographic conformations, superimposed PDB files of the protein−ligand complexes, and experimental results for the two DIFGAPE runs fitted to the bound ligand overlay. 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) Bacilieri, M.; Moro, S. Ligand-based drug design methodologies in drug discovery process: an overview. *Curr. Drug Discovery Technol.* **2006**, *3*, 155–165.
(4) Lemmen, C.; Lengauer, T. Computational methods for the structural alignment of molecules. *J. Comput.-Aided Mol. Des.* **2000**, *14* (3), 215–232.
(5) Jain, A. N. Ligand-based structural hypotheses for virtual screening. *J. Med. Chem.* **2004**, *47* (4), 947–961.
(6) 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.
(7) 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.
(8) Labute, P.; Williams, C.; Feher, M.; Sourial, E.; Schmidt, J. M. Flexible Alignment of Small Molecules. *J. Med. Chem.* **2001**, *44* (10), 1483–1490.
(9) 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.
(10) Cho, S. J.; Sun, Y. FLAME: A Program to Flexibly Align Molecules. *J. Chem. Inf. Model.* **2006**, *46* (1), 298–306.
(11) 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.
(12) Lemmen, C.; Lengauer, T.; Klebe, G. FLEXS: A Method for Fast Flexible Ligand Superposition. *J. Med. Chem.* **1998**, *41* (23), 4502–4520.
(13) Kearsley, S. K.; Smith, G. M. An alternative method for the alignment of molecular structures: maximizing electrostatic and steric overlap. *Tetrahedron Comput. Methodol.* **1990**, *3* (6c), 615–633.
(14) Iwase, K.; Hirono, S. Estimation of active conformations of drugs by a new molecular superposing procedure. *J. Comput.-Aided Mol. Des.* **1999**, *13* (5), 499–512.
(15) Miller, M. D.; Sheridan, R. P.; Kearsley, S. K. SQ: A Program for Rapidly Producing Pharmacophorically Relevant Molecular Superpositions. *J. Med. Chem.* **1999**, *42* (9), 1505–1514.
(16) *ROCS, version 2.3.1*; Openeye Scientific Software: Santa Fe, NM, 2007.
(17) *Omega, version 2.2.1*; Openeye Scientific Software: Santa Fe, NM, 2007.
(18) 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.
(19) Chen, Q.; Higgs, R. E.; Vieth, M. Geometric Accuracy of Three-Dimensional Molecular Overlays. *J. Chem. Inf. Model.* **2006**, *46* (5), 1996–2002.
(20) *MOE, version 2007.09*; Chemical Computing Group: Montreal, Canada, 2007.
(21) Davis, L. *Handbook of genetic algorithms*; Van Nostrand Reinhold: New York, 1991.
(22) 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.
(23) Goldberg, D. E. *Genetic algorithms in search, optimization, and machine learning*; Addison-Wesley Pub. Co: Reading, MA, 1989.
(24) Crippen, G.; Havel, T. *Distance Geometry and Molecular Conformation*; Research Studies Press: Taunton, England, 1988.
(25) Lindholm, T.; Yellin, F. *The Java virtual machine specification*; Addison-Wesley: Reading, MA, 1996.
(26) Huang, N.; Shoichet, B. K.; Irwin, J. J. Benchmarking Sets for Molecular Docking. *J. Med. Chem.* **2006**, *49*, 6789–6801.
(27) Soliva, R.; Gelpi, J. L.; Almansa, C.; Virgili, M.; Orozco, M. Dissection of the Recognition Properties of p38 MAP Kinase. Determination of the Binding Mode of a New Pyridinyl-Heterocycle Inhibitor Family. *J. Med. Chem.* **2007**, *50* (2), 283–293.
(28) Kubinyi, H.; Folkers, G.; Martin, Y. C. *3D QSAR in drug design*; Kluwer/ESCOM: Dordrecht, 1998.
(29) 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.

CI900109N