RAPID: Randomized Pharmacophore Identification for Drug Design*

P. FINN† L. E. KAVRAKI‡ J.-C. LATOMBE R. MOTWANI§
C. SHELTON S. VENKATASUBRAMANIAN A. YAO¶

Department of Computer Science
Stanford University
Stanford, CA 94305

{latombe,motwani,cshelton,suresh,yao}@cs.stanford.edu

Abstract

This paper describes a randomized approach for finding invariants in a set of flexible ligands (drug molecules) that underlies an integrated software system called RAPID currently under development. An invariant is a collection of features embedded in $\Re^3$ which is present in one or more of the possible low-energy conformations of each ligand. Such invariants of chemically distinct molecules are useful for computational chemists since they may represent candidate pharmacophores. A pharmacophore contains the parts of the ligand that are primarily responsible for its interaction and binding with a specific receptor. It is regarded as an inverse image of a receptor and is used as a template for building better pharmaceutical drugs. The identification of pharmacophores is crucial in drug design since the structure of the targeted receptor is frequently unknown, but a number of molecules that interact with the receptor have been discovered by experiments. It is expected that our techniques and the results produced by our system will prove useful in other applications such as molecular database screening and comparative molecular field analysis.

---

*This research is supported by a grant from Pfizer Central Research.

†Pfizer Central Research. E-mail: finn@sandwich.pfizer.com.

‡Partially supported by startup funds from Rice University. Currently at Department of Computer Science, Rice University, Houston, TX 77005. E-mail: kavraki@cs.rice.edu

§Partially supported by an Alfred P. Sloan Research Fellowship, an IBM Faculty Partnership Award, an ARO MURI Grant DAAH04-96-1-0007, and NSF Young Investigator Award CCR-9357849, with matching funds from IBM, Mitsubishi, Schlumberger Foundation, Shell Foundation, and Xerox Corporation.

¶On leave from Princeton University. E-mail: yao@cs.princeton.edu

1 Introduction

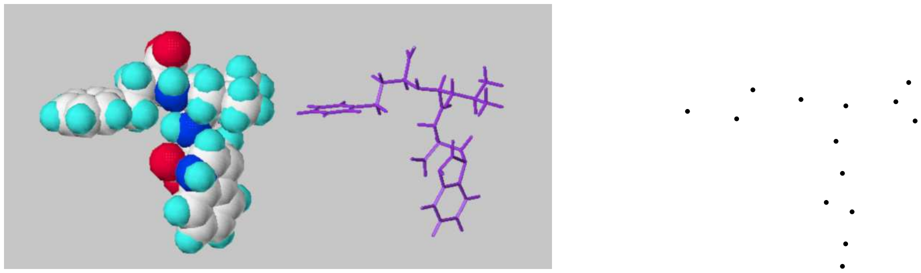
Computational chemists working in the area of structured-based drug design consider both chemical and geometric properties of the interacting molecules when developing new pharmaceutical drugs [6, 23, 28]. The underlying assumption is that drug activity, or pharmacophoric activity, is obtained through the molecular recognition and binding of one molecule (ligand) to a pocket of another, usually larger, molecule (receptor). This assumption is supported by a number of experimental results showing molecules with geometric and chemical complementarity in their active, or binding, conformations [9].

When the three dimensional structure of the receptor is known, docking methods [8] exploit both the geometric and the chemical information available. However, the geometric structure of relatively few molecules has been obtained via X-ray crystallography or NMR techniques. As a result, computational chemists often try to develop pharmaceutical drugs for receptors whose structure is unknown [7, 32]. The starting point in this case is a collection of ligands that have been experimentally discovered to interact with the considered receptor. By examining the chemical properties and the possible shapes of these ligands, chemists seek to identify a set of features embedded in $\Re^3$ that is contained in some active conformation of each (or most) of the ligands. This is called the pharmacophore and it is considered responsible for the observed drug activity. The features of the pharmacophore interact with features of the receptor, while the rest of the ligand acts as a scaffold. Once a pharmacophore has been isolated, it can be used to further improve the activity of pharmaceutical drugs [20].

This paper considers the following problem: Given a set of ligands that interact with the same receptor, find geometric invariants of these ligands, i.e., a set of features embedded in $\Re^3$ that is present in one or more valid conformations of the ligands. We refer to this problem as the pharmacophore identification problem since the pharmacophore is such an invariant. Solving this problem requires dealing efficiently with large amounts of spatial data and shape information. Ligand molecules are very flexible and can assume many distinct potentially valid conformations. A valid conformation is a rigid spatial realization of the atoms of a molecule whose energy is below a predefined threshold [9]. It is not known, a priori, which of the many possible valid conformations of a ligand is the one that contains the desired pharmacophore.

Besides providing starting points for designing effective pharmaceutical drugs, there are several important applications where geometric invariant identification is useful. For example, the geometric invariants identified by our system can help in the formulation of database queries that may retrieve functionally equivalent, but structurally novel molecules from molecular databases [39]. Geometric invariants can also suggest alignments of molecules for input to CoMFA (Comparative Molecular Field Analysis) and other 3D QSAR (Quantitative Structure-Activity Relationship) methods [7, 23]. These are methods that predict the activity of hypothetical compounds on the assayed activity of previously synthesized ones.

This paper describes our efforts to prototype an integrated software system, called RAPID (RAndomized Pharmacophore Identification for Drug design), for addressing the pharmacophore identification problem. We present briefly the overall structure of RAPID in Section 2, and outline related work in Section 3. The two main modules of RAPID, conformational search and identification of invariants, are described in Sections 4 and 5. In this abstract, we focus on the latter module and describe in some detail a randomized



(a) (b) (c)

Figure 1: 1TMN: (a) space-filling, (b) stick, and (c) set-of-points models

approach for solving an abstract geometric problem lying at the core of pharmacophore identification – the search for common subsets in a collection of point sets embedded in $\Re^3$. Our system attempts to deal with the numerous complications that arise in real examples. We report in Section 6 some preliminary experimental results. Finally, in Section 7, we conclude with a discussion some open issues that merit consideration in future work.

2 Overview of RAPID

RAPID tries to identify geometric invariants among a collection of small ligand molecules like the one shown in Figure 1. This ligand is called 1TMN and it is one of the several inhibitors of the protease thermolysin. Figure 1(a) shows the space filling model of 1TMN. In this model a Van der Walls sphere [9] is considered around each atom center. Figure 1(b) shows the corresponding stick model in which only chemical bonds are drawn. The degrees of freedom of ligands include bond lengths, bond angles (angles between two consecutive bonds), and dihedral or torsional angles (angles formed by the first and third of three consecutive bonds, viewed along the axis of the second bond). In practice, only the torsional degrees of freedom are considered since these are the ones that exhibit large variations in their values [9]. Figure 1(c) shows the conformation of 1TMN of Figure 1(a) as a set of points in $\Re^3$. These points may represent atom centers or groups of atoms aggregated to one point endowed with a feature common to all these atoms (e.g., a rigid benzene ring) [32]. We assume that once a conformation is given, one can automatically transform it to a unique collection of points. For the purposes of this paper, a conformation C may represent a valid geometric embedding of a molecule where bonds are retained, or a set of points without bond information.

In RAPID the identification of geometric invariants in a collection of flexible ligands denoted by $M = \{M_1, M_2, \ldots, M_N\}$ is treated as a two-stage process[1] addressing the two following problems:

Problem 1 (Conformational Search) *Given a collection of ligands $M = \{M_1, M_2, \ldots, M_N\}$, the degrees of freedom of each of them, and an energy function E. For each M_i, find a set of conformations $\{C_{i1}, C_{i2}, \ldots, C_{ik_i}\}$, such that for each $i = 1, \ldots, N$ and $j = 1, \ldots, k_i$,*

$$E(C_{ij}) \leq \textit{THRESHOLD}, \text{ and } d(C_{ij}, C_{il}) > \textit{TOLERANCE} \text{ for } l \neq j$$

[1]A third module of RAPID, currently under development, involves the computation of molecular surfaces (see Finn et al [17]).

where THRESHOLD and TOLERANCE are pre-specified values, and $d(.,.)$ is a distance function.

Problem 2 (Invariant Identification) *Given a collection of ligands $M = \{M_1, M_2, \ldots, M_N\}$, where each M_i is a set of conformations $\mathcal{C}(M_i) = \{C_{i1}, C_{i2}, \ldots, C_{ik_i}\}$. Determine a set of labeled points S in $\Re^3$ with the property that for all $i \in \{1, \ldots, n\}$, there exists some $C_{ij} \in \mathcal{C}(M_i)$ such that S is congruent to some subset of C_{ij}. A solution S, if it exists, is called an* invariant *of M.*

Although at this stage the two modules of RAPID work independently, we plan to support their interaction as the system develops. For example, the invariant identification module will be able to request from the conformational search module conformations that contain certain features of the molecule in specified spatial positions. RAPID also needs to deal with a host of complicating factors. In practice, the input may contain ligands that do not bind firmly to a receptor and do not contain the pharmacophore. This requires us to consider a relaxation of Problem 2 above, where a geometric invariant need only be present in conformations of some K of the N molecules. Furthermore, an invariant may be perturbed in its spatial layout in distinct ligands' conformations, or there may be a large number of spurious invariants of differing sizes that do not correspond to a pharmacophore. Each of these contributes to a combinatorial explosion in the search process for the invariants. We defer a detailed discussion of such issues to the final version of this paper.

3 Related Work

As far as conformational search is concerned, both systematic and randomized techniques are being investigated [8, 27, 28]. Systematic search methods sample each torsional degree of freedom of the ligand at regularly spaced intervals and minimize all conformations produced [29]. These techniques can be prohibitively expensive [27] and several heuristics are employed to quickly prune away conformations that are "close" to previously generated conformations [38]. Randomized methods work as follows: conformations are obtained by applying random increments to torsional degrees of freedom of the molecule starting with a user-specified initial conformation [19], or with a previously discovered low-energy conformation [12]. Recent articles, which attempt to compare different methods, emphasize the superior quality of the results obtained with randomized methods [19]. Other considered methods produce low-energy conformations, which obey distance constraints, using inverse kinematics [31], algebraic methods [21], or distance geometry techniques [14]. Note that the *protein-folding* problem is also a conformational search problem but its large size prohibits the use of the above techniques (see Dill [15]).

Invariant identification is related to the well-studied problem in geometric optimization [1, 3, 2, 25] of finding common point sets. While determining the congruence of two point-sets in $\Re^3$ is tractable [1, 3] *in the absence of complications such as noise*, invariant identification is more closely related to the problem of identifying the *largest common point set* (LCP). Unfortunately, the LCP problem turns out to exceedingly difficult; in fact, even for m collections of n points on the real line, the LCP cannot be approximated to within an n^ϵ factor unless $P = NP$, and only weak positive results are known [2, 25]. Of course the problem is polynomially solvable when $m = 2$ [3] but that is not good enough for our purposes. Also, there are constraints in the molecular structures, but there are also various complications as discussed earlier.

In the computational chemistry literature, the most popular algorithms are based on clique-detection. For instance, DISCO [32] initially considers a pair of conformations c_1 and c_2 belonging to different molecules and constructs a "correspondence graph" G. The nodes of G are all node pairs, and an edge is created if the pairs in each of the connected nodes can be matched simultaneously. A clique detection algorithm [11] is then used to find cliques in G. These correspond to invariants in c_1 and c_2, and thus to candidate pharmacophores. Maximum clique detection is NP-hard [18], but this algorithm seems to work well in practice [32, 39]. The approach has been generalized to n conformations by choosing a reference conformation and comparing it with the other $n - 1$ conformations. Unfortunately, if a large number of conformations per molecule are to be considered, there is a tremendous blow-up in the number of primitive operations performed by such algorithms [7], rendering them impractical. Given this situation, several new approaches are under development. For instance, one idea is to start with small invariants (2-3 features) and gradually expand them [7]. Our approach is fairly different and involves heavy use of randomized search techniques [34].

4 Conformational Search

The goal of conformational search as defined in Problem 1 (Section 2) is to produce a number of distinct low-energy conformations of a given ligand. We proceed as follows. Initially a large number of conformations are generated at random. In contrast with previous randomized search methods, we obtain a random conformation by selecting each degree of freedom from its allowed range according to a user-specified distribution. This distribution is frequently the uniform distribution. However, if some a priori information is available about the preferred values of a particular degree of freedom [26], then the corresponding values are selected according to a distribution that reflects the a priori information (i.e. a Gaussian distribution). An efficient minimizer [10, 35] is then used to obtain conformations at local energy minima. Minimization is the most time-consuming step during conformational search, so we have carefully optimized this procedure.

To obtain a representative set of conformations from our sample, we partition it into sets that reflect geometric similarity as captured by the distance measure DRMS. We define DRMS(C_i,C_j) as the square root of the mean of the squared distances of the corresponding atoms of C_i and C_j, after C_i is transformed to C_j. This transformation is computed by using a basis of three predefined atoms a_1, a_2, and a_3 (see Section 5). We perform a greedy clustering of the conformations by placing a given conformation in an existing cluster if its distance from the "center" of that cluster is less than the predefined value TOLERANCE. If no such cluster is found, a new cluster is created. The center of a cluster is the conformation with the lowest energy in the cluster.

We omit further details of the clustering, but note that the implementation employs heuristics for improving the quality of output, e.g., it is ensured the cluster center is close to the average cluster conformation and a post-processing step checks that all cluster centers are sufficiently far apart. While we obtain reasonable experimental results, we are investigating optimizations via incremental clustering algorithms [13].

5 Identification of Invariants

The set of cluster centers, denoted by $\mathcal{C}(M) = \mathcal{C}(M_1) \cup \ldots \cup \mathcal{C}(M_N)$ is the input for the invariant identification module. Each conformation in $\mathcal{C}(M)$ is now represented as a set of labeled points in $\Re^3$ (see Section 2). We wish to determine a structure S that is congruent to a substructure of some conformation in every molecule. The congruence relation is with respect to 3-D rotations and translations that preserve equality of labels.

Our formulation of the pharmacophore identification problem in Problem 2 (Section 2) ignores the issue of error in the data. In practice, atom positions are fuzzy, and it may not be possible to align points exactly. Therefore, we adopt the convention that two points p_1 and p_2 are said to *match* when $\mid p_1 - p_2 \mid \leq \epsilon$, where ϵ is the *point location error*.

The pharmacophore identification problem is a variant of the *largest common point set problem (LCP)* in three dimensions, which is the following. Given s point sets $P_1, P_2, \ldots, P_s$ in $\Re^3$, determine the point set of maximum cardinality congruent to some subset of each point set. For convenience, we assume that each point set P_i has cardinality exactly n. For arbitrary s and dimension d, LCP is hard to approximate within a factor of n^ϵ, for some $\epsilon \geq 0$ [2]. In the sequel, we consider the following version of LCP, called LCP-α: determine a point set S of size $|S| \geq \alpha n$ congruent to some subset of each $P_i, 1 \leq i \leq s$. The motivation for focusing on this subproblem is that it better captures the application where it is known that a pharmacophore has to be a reasonably large invariant. We begin by focusing on the case $s = 2$, and later indicate how we generalize to arbitrary values of s.

5.1 Phase 1: Pairwise Matching

In this section, we focus on the invariant identification problem for two point sets, denoted by MATCH. This problem has been studied extensively in the literature [36, 37]. For the case when $\alpha = 1$, i.e., determining the congruence of two point sets, Atkinson [5] obtains a solution in time $O(n \log n)$ in two and three dimensions. However, for general α, the problem is much harder, relating to combinatorial distance properties of $\Re^3$ and is examined by Akutsu, Tamaki, and Tokuyana [3], in which a worst-case bound of $O(n^{4.6})$ is obtained for unknown α, and $O(n^{2.6}/\alpha^2)$ randomized when α is known. For two dimensions, the corresponding bounds are $O(n^{3.2})$ and $O(n^{2.2}/\alpha)$.

In this section, we describe two random-sampling schemes for solving LCP-α. The first is quite natural, and the second uses more careful random sampling. Under certain assumptions about the distribution on distances between points in a set, we prove loose bounds on the running time of these schemes, illustrating the superiority of the second approach. Our results are confirmed by experimental evidence in Section 6.

For a set of points S, let D_S be the multi-set of all inter-point distances in S. For a given *cluster width* δ, we can partition D_S into L *buckets*, where bucket i contains all entries $d \in D_S$ such that $i\delta \leq d < (i+1)\delta$. For the purposes of theoretical analysis, we assume that given a point set P, each such distance bucket is of constant size, for fixed δ; our experimental results will later justify this assumption. The notation $g(n) = \tilde{O}(f(n))$, where f and g are functions, implies that $g(n) = O(f(n) \log n)$. Note that in three dimensions, a unique transformation T (upto reflection) between two point sets P_1 and P_2 is determined by matching three points p, q, and r in P_1 with three points s, t, and u in P_2. Furthermore, is it known that [16]:

Proposition 1 *Given a transformation T from P_1 to P_2, in time $\tilde{O}(n)$ we can determine for each point $p \in P_1$ a corresponding point $p' \in P_2$ such that $|T(p) - p'|$ is minimized.*

The basic random sampling method is as follows.

BASIC-SAMPLE: For some constant c, perform c/α^3 iterations of the following sampling process: sample a triplet of points $< p_1, p_2, p_3 >$ randomly from P_1; determine three points in P_2 congruent to this set; compute the resulting induced transformation and determine the number of points in P_1 matching corresponding points in P_2; and, if the number exceeds αn, declare SUCCESS.

Theorem 2 *Given a common subset S of size $|S| \geq \alpha n$, the probability that BASIC-SAMPLE fails to declare SUCCESS is at most $1/n$.*

The proof of this theorem is straightforward. Observe that the probability of picking three points that belong to the common substructure is at least $1/\alpha^3$.

Theorem 3 *BASIC-SAMPLE runs in time $\tilde{O}(n/\alpha^3)$ using space $O(n^2)$.*

The running time can be written as $T = n\sum_{i=1}^{1/\alpha^3} \Delta_i$ where Δ_i is the number of candidate transformations generated during iteration i. Using the notation of [3], let $H_P(t)$ denote the number of triangles congruent to t in point set P. It is the case that $\Delta_i = H_{P_1}(t_i) * H_{P_2}(t_i)$, where t_i is the triangle induced by the triplet chosen at iteration i. ¿From our assumption on the size of distance buckets, we produce a constant number of transformations in each stage, and the above bound holds.

Besides BASIC-SAMPLE, we consider the following more sophisticated sampling scheme.

PARTITION-SAMPLE: For some constant c, perform c iterations of the following sampling process: randomly select two subsets A and B of size $1/\alpha$ from P_1; also select a subset C of size $1/\alpha$ from P_2; store all distances $d(p, q)$, for all $p \in C$ and $q \in P_2 - C$, in a hash table; for every triangle (a, b, p) with $a \in A, b \in B$, and $p \in P - (A \cup B)$, probe for $d(p, a)$ and $d(p, b)$ in the hash table to determine all matching triplets (c, p_1, p_2) with $c \in C$ and $p_1, p_2 \in P_2 - C$; finally, as before, if the resulting transformation induces a match of more than αn points, declare SUCCESS.

Theorem 4 *Given a common subset S of size $|S| \geq \alpha n$, the probability that PARTITION-SAMPLE fails to declare SUCCESS is at most $e^{-c\alpha n}$.*

We defer the proof of the preceeding theorem to the final version of the paper.

Theorem 5 *PARTITION-SAMPLE runs in time $\tilde{O}(n^2/\alpha^2 - n/\alpha^3)$ using space $O(n/\alpha)$.*

As before, the running time of the algorithm can be written as $T = n \sum_{i=1}^{n/\alpha^2 - 1/\alpha^3} \Delta_i$. If our assumption uniform hashing assumption is valid, then the bound follows. Figure 5 illustrates experimental indications that this is the case. Intuitively, this partitioning favors "large" solutions over small ones. Given a solution S of size K in P_1, the probability that each of A, B, and $P_1 - A - B$ contain at least one point of S increases with K.

Eliminating Redundant Solutions. Any MATCH algorithm could return many solutions that satisfy a containment relationship with respect to each other. To prevent such redundant solutions from propagating through the complete search procedure, we need to *prune* them. Given invariants S_1 and S_2, we can check if $S_1 \subseteq S_2$ by invoking MATCH on these two sets with α set to 1. Clearly, since α is 1, a solution will be produced only if S_1 is contained in S_2. This pruning is done in a pair-wise fashion on the set of invariants before these invariants are used as input to the next phase.

Inaccuracies in Computing the Transformation. The module at the core of the substructure search algorithm is the following: Given points $p_1, p_2, p_3 \in P_1$, and points $p'_1, p'_2, p'_3 \in P_2$, compute a transformation T from P_1 to P_2 that maps $p_i \rightarrow p'_i$, for $i = 1, 2, 3$. For our purposes we need to compute a transformation that, given this initial set of correspondences, matches the largest number of points from P to Q. Such a transformation can be computed in many different ways. The simplest such way is: **(i)** align p_1 and q_1; **(ii)** align the vectors $\overrightarrow{p_1 p_2}$ and $\overrightarrow{q_1 q_2}$; and, **(iii)** align p_3 and q_3 by a rotation about $\overrightarrow{p_1 p_2}$.

This approach over-constrains the fit between the two triangles. In general, it is sufficient that these pairs match within distance ϵ, at such an orientation that the largest number of points from P_1 can be made to match points in P_2. However, we do not know these other points in advance, and as a result of over-constraining, it is possible that we may not be able to force as many points to match as we should.

If we knew the correspondences between points in P_1 and P_2, then a more "balanced" transformation could be obtained by computing the transformation minimizing DRMS(P'_1, P'_2), P'_1 and P'_2 being the subsets of P_1 and P_2 that correspond [4] (see Section 4 for the definition of DRMS). This suggests the obvious refinement: Use the procedure above to generate a plausible correspondence list, and then use a DRMS minimizing procedure to compute a refined transformation.

However, this approach is more time-consuming, and it is possible that DRMS minimization merely replaces one form of over-constraining by another. In general, if we consider the six-dimensional space of all rigid transformations in $\Re^3$, the set of transformations satisfying the given correspondences is actually an enclosed volume. Choosing any transformation from within this volume will produce an acceptable transformation between the two given point sets. However, when we generate candidate solutions and attempt to match them against other point sets, the "right" choice of transformation could be crucial in preventing points from being lost. Our implementation employs various heuristics to try and minimize the effect of this problem. One such heuristic is the following. Determine a *seed* transformation T by any of the above methods, and then sample three random pairs from the set of correspondences that T induces, using these pairs to construct a new transformation. Clearly, in a perfect world, we will obtain T again. However, given the inaccuracies in point location, it turns out that some choices of triplets may yield a larger number of correspondences than before. We defer experimental results in this regard to a final version of the paper.

5.2 Phase 2: Multiple Matching

The previous phase yields methods for solving LCP on two sets. In Phase 2, candidate solutions obtained from the previous phase are tested against the remaining molecules to determine the invariant. Each MATCH call operates on two conformations. Since each molecule is represented by many conformations, we extend MATCH to two molecules by doing all pair-wise matches between the sets of conformations.

There are various strategies one could use to process multiple molecules. A simple strategy does the obvious linear merging. We take each solution and compare it with the next molecule. We do this for all current solutions, concatenate and prune the results, and repeat with a new molecule. Another approach, which is in some ways less order-sensitive, is a tree-based merge. Here, we run the two-molecule algorithm on distinct pairs of molecules, and recursively combine the results in a binary tree-like fashion. Both of these approaches are simple and are correct, if point location error is not permitted.

In addition to this, we may wish to find an invariant that does not exist in *all* the molecules, but in some fixed number of them. Our current strategy would fail to do this, because all candidate solutions are compared against every molecule. We use a modified merging strategy here to keep track of the number of times an invariant fails to match against a molecule, and only reject those which exceed the maximum allowed number of failures.

If an invariant S is contained in a point set P, then a call to MATCH(S, P) should return a set of solutions containing S. This can easily be checked, which therefore means that we can verify whether an invariant matches against a conformation. Our marking algorithm associates *marks* with each candidate solution. All original conformations are initialized with zero marks. All invariants that result from a MATCH acquire the sum of the marks of the input structures. Every time a solution fails to match against a new molecule (which means that it fails to match with *any* conformation of that molecule), we add a mark to it and continue to propagate it. We reject any solutions for which the number of marks exceeds the prescribed maximum.

6 Experimental Results

The section reports preliminary experimental results; more detailed results and analysis will be included in the final version of the paper. All reported timings are on an SGI Indigo2 with a 175 MHz MIPS R8000 processor and 384MB RAM. Code was written in C/C++, and compiled using SGI CC.

Input. In Figure 2, we show four different inhibitors of the protease thermolysin. These molecules fit into the same cavity of thermolysin and by their presence inhibit the activity associated with that cavity. This example was chosen because all the inhibitors have been crystallized with thermolysin and their active conformations are known and recorded in the PDB database. Note that 1TLP has 69 atoms and 10 torsional degrees of freedom, 4TMN has 68 atoms and 15 degrees of freedom, 5TMN has 64 atoms and 13 degrees of freedom, and 6TMN has 63 atoms and 12 degrees of freedom.

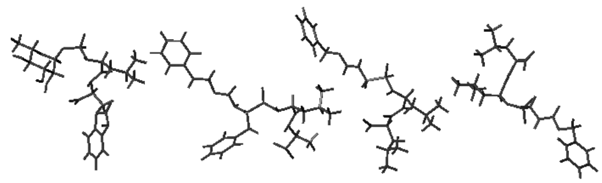


Figure 2: 1TLP, 4TMN, 5TMN, and 6TMN are inhibitors of thermolysin.

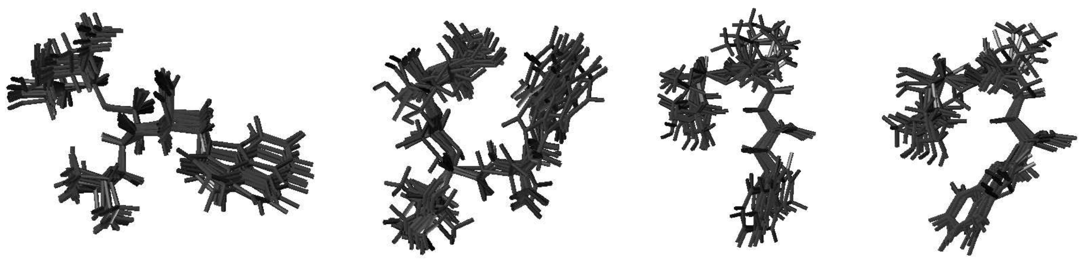


Figure 3: Different clusters of 1TLP.

Conformational Search. Each of the molecules in Figure 2 was run through our conformational search software. A THRESHOLD value of 20 Kcals/mol was used for the energy of the valid conformations. It took 4.3 hours to to produce 4000 valid conformations for each molecule and reduce these to a set of representatives. The timings reported are very reasonable if one takes into account that 10, 15, 13, and 12 dimensional spaces are searched. The number of representatives produced was 850, 1192, 1024, and 955 for 1TLP, 4TMN, 5TMN, and 6TMN, respectively. The TOLERANCE value was set to 1.23 $\AA$ for all runs. A few clusters of 1TLP are shown in Figure 3. Since we know the active conformations of all the molecules, we performed a consistency check at the end of conformational search and we confirmed that the active conformations were close to one of the produced representatives.

An important issue in conformational search is to decide the number of valid conformations to produce. At this stage, this number is determined experimentally. We plan to interleave the generation of conformations and their clustering in the future and stop producing new conformations when these fall into existing clusters and do not increase significantly the overall number of clusters.

Identification of Invariants. To find the invariants in these four molecules, the conformations produced by the first stage were given as input to the search algorithm described in Section 5. For these tests, we considered each non-hydrogen atom to be a separate feature or point. Thus, each conformation had approximately 30 features drawn from 6 feature classes. We define the "solution" to be the overlapping portions of the molecules when aligned as shown in Figure 4. This is the upper right hand T-shaped portion of this diagram. The entire invariant consists of roughly 7 atoms and an additional 7 atoms of "scaffolding," or connecting atoms with no pharmacophore functionality.

Table 1 details the parameters used for invariant identification. The parameters ϵ, δ, and α are as described in Section 5. We set at 5 the number of times the MATCH algorithm is run per comparison, or the maximum number of invariants found per MATCH. These parameters were chosen by experimentation. For example, δ was selected to be much lower than ϵ. We found that higher values of δ increased the running time without improving the quality of the solution.

In Section 5 we assumed that the distance distribution for molecules satisfies certain properties, and showed certain performance bounds based on this. Here, we show experimental validation of these bounds. For the sake of brevity, we omit detailed experimental analysis of variants of Multiple Search and use a linear search to compute invariants, with no failures in matching invariants permitted.

Figure 5(a) plots the number of candidate transformations produced ($\sum_{i=1}^{n/\alpha^2} \Delta_i$) by using the procedure PARTITION-SAMPLE against α. Superimposed on this is a plot of $n/\alpha^2 - 1/\alpha^3$. We do the same for BASIC-SAMPLE in Figure 5(b). Here we superimpose the plot of $1/\alpha^3$. For both algorithms, the experimental results follow the predicted behavior. In Figure 6, we compare the two schemes experimentally. The comparison clearly shows that PARTITION-SAMPLE performs better experimentally, although its asymptotic performance seems worse.

This discrepancy is partly explained in Figure 7. For each molecule in the input set, we plot its distance distribution function (which BASIC-SAMPLE samples from), as well the distribution function induced by the subset of distances that PARTITION-SAMPLE samples (for varying values of α).

We finally present in Table 2 a comparison between the two schemes on large data sets, where we vary the number of conformations per molecule. We include the active conformation in our sets. In all cases, the quality of solutions (in terms of the largest solution found) is comparable, and PARTITION-SAMPLE consistently runs faster than BASIC-SAMPLE. When the number of conformations increases, more invariants are produced because some of the added conformations had additional "scaffolding" which also could be matched. Yet, in every case, the invariant of the largest size was the "correct solution." Many non-trivial solutions of size $8 - 14$ were also produced.

7 Discussion

We briefly discuss some enhancements currently underway that we hope to report in the final version of the paper. Our goal is to optimize the modules of the system to perform experiments that involve 5-20 ligands and a large number of conformations per ligand. As far as conformational search is concerned, we are investigating dynamic clustering techniques. By looking at the relative changes in the number of clusters found, we hope that we will be able to automatically end the conformational search stage. For the invariant identification stage, we will try to reduce the dependence of the implementation on the ordering of the input, a phenomenon observed in the current system. Last but not the least, our short-term plans include the implementation of an algorithm that will extract features from conformations in the way the computational chemists understand them. Currently, each atom gives rise to a feature that bears as a label the corresponding atom type. Computational chemists, however, have rules to group atoms into features that describe the chemical behavior of parts of the molecule, e.g., hydrophobic parts and positively charged parts. Features will reduce the number of points we consider per conformation and will hopefully help us deal with larger examples.

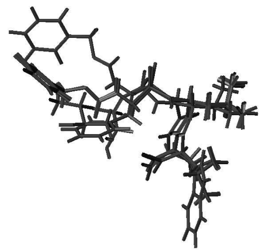


Figure 4: 1TLP, 4TMN, 5TMN, and 6TMN overlapped in their active conformations.

Parameter	Search Value	Prune Value
ϵ	1.3	3.5
δ	0.5	1.0
α	0.3	1.0

Table 1: Parameters for invariant search trials.

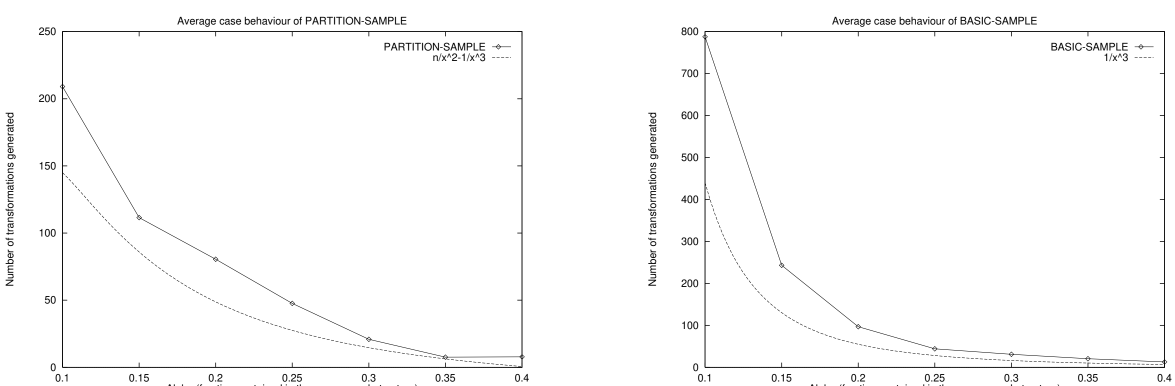
References

[1] T. Akutsu. On determining the congruence of point sets in higher dimensions. *Lecture Notes in Computer Science 834*, page 38.

[2] T. Akutsu and M. Halldórsson. On the approximation of largest common subtrees and largest common point sets. *Lecture Notes in Computer Science 834*, pages 405–413.

[3] T. Akutsu, H. Tamaki, and T. Tokuyama. Distribution of distances and triangles in a point set and algorithms for computing the largest common point set. Manuscript, 1996.

[4] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares Fitting of Two 3-D Point Sets. *IEEE Trans. Pattern. Anal. Machine Intell.*, volume PAMI-9, No. 5, September 1987.



(a) (b)

Figure 5: Experimental results vs. theoretical predictions.

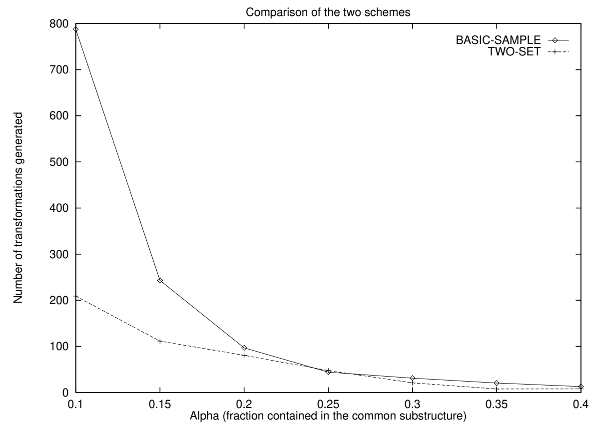
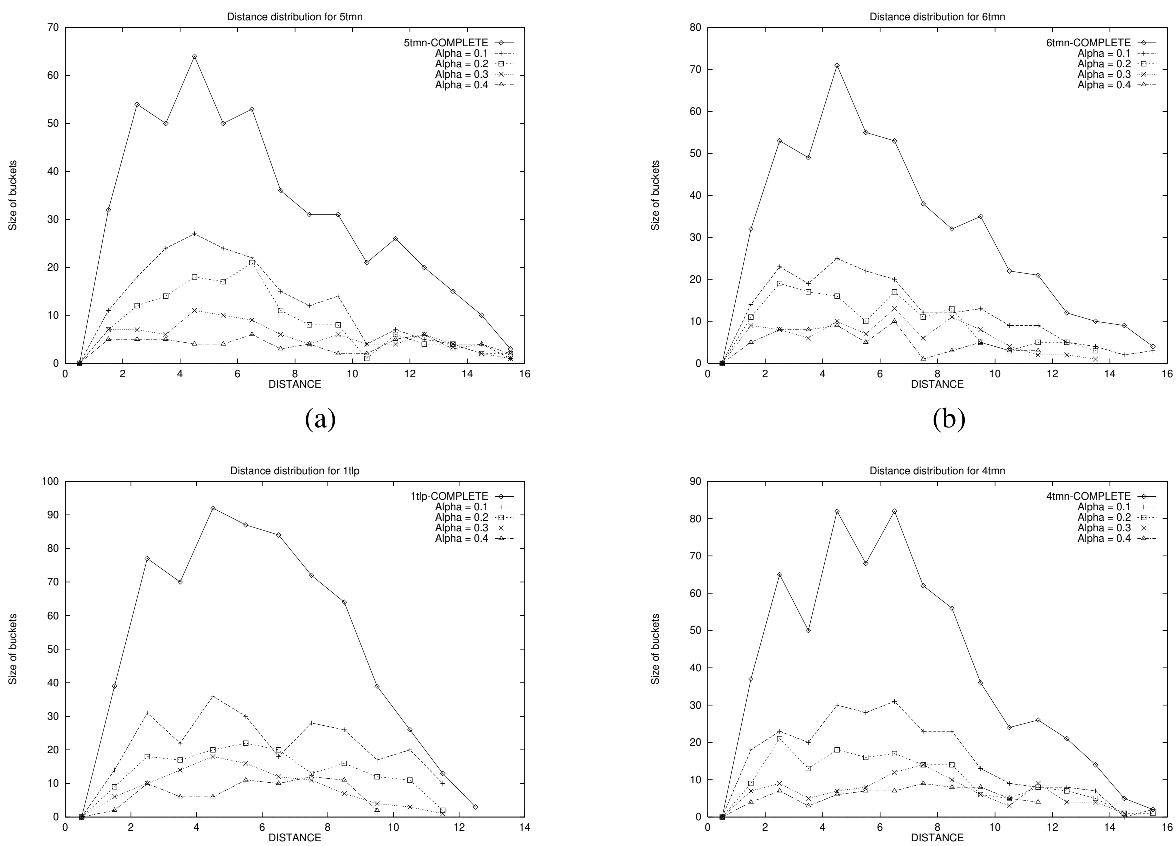


Figure 6: Relative performance of BASIC-SAMPLE and PARTITION-SAMPLE.



(a) (b) (c) (d)

Figure 7: Distance distribution functions of (a) 5TMN, (b) 6TMN, (c) 1TLP, (d) 4TMN.

Number of	Time to	Histogram of Solution Sizes												
Conformations	Completion (in s)	4	5	6	7	8	9	10	11	12	13	14	15	total
BASIC-SAMPLE														
1	4.62		2	1								1		
11	2127.66	90	34	20	5	6	4	1	1	1				
21	24175.42	289	110	46	35	42	18	3	6	2	2	1		
PARTITION-SAMPLE														
1	2.85			1		1					1			
11	656.47	20	9	8	11	11	3							
21	8060.00	73	55	53	30	21	25	7	8	1	1	4	1	

Table 2: Comparison of BASIC-SAMPLE and PARTITION-SAMPLE

.

[5] M. D. Atkinson. An optimal algorithm for geometric congruence. *J. Algorithms* 8 (1987). pages 134–183.

[6] L. Balbes, S. Mascarella, and D. Boyd. A perspective of modern methods in computer-aided drug design. In K. Lipkowitz and D. B. Boyd, editors, *Reviews in Computational Chemistry*, volume 5, pages 337–370. VCH Publishers, 1994.

[7] D. Barnum, J. Greene, A. Smellie, and P. Sprague. Identification of common functional components among molecules. To appear in *J. Chem. Inf. Comput. Sci.*, 1996.

[8] J. Blaney and S. Dixon. A good ligand is hard to find: Automated docking methods. *Perspectives in Drug Discovery and Design*, 1:301–319, 1993.

[9] D. B. Boyd. Aspects of molecular modeling. In K. Lipkowitz and D. B. Boyd, editors, *Reviews in Computational Chemistry*, volume 1, pages 321–351. VCH Publishers, 1990.

[10] R. Brent. *Algorithms for finding zeros and extrema of functions without calculating derivatives*. PhD thesis, Stanford University, 1971.

[11] C. Bron and J. Kerbosch. Finding all cliques of an undirected subgraph. *Commun. ACM*, 16:575–577, 1973.

[12] G. Chang, W. Guida, and W. Still. An internal coordinate monte-carlo method for searching conformational space. *J. Am. Chem. Soc.*, 111:4379–4386, 1989.

[13] M. Charikar, C. Chekuri, T. Feder, and R. Motwani. Incremental Clustering. Manuscript. December 1996.

[14] G. Crippen and T. Havel. *Distance Geometry and Molecular Conformation*. Research Studies Press, Letchworth, U.K., 1988.

[15] K. Dill. Folding proteins: Finding a needle in a haystack. *Current Opinion in Structural Biology*, 3:99–103, 1993.

[16] D. Dobkin and R. Lipton. Multidimensional search problems. *SIAM J. Computing* 5:181–186 (1976)

[17] P. Finn, D. Halperin, L. Kavraki, J.-C. Latombe, R. Motwani, C. Shelton, and S. Venkatasubramanian. Geometric manipulation of flexible ligands. In M. Lin and D. Manocha, editors, *LNCS Series - 1996 ACM Workshop on Applied Computational Geometry*. Springer-Verlag, 1996.

[18] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1980.

[19] A. Ghose, J. Kowalczyk, M. Peterson, and A. Treasurywala. Conformational searching methods for small molecules: I. study of the sybyl search method. *J. of Computational Chemistry*, 14(9):1050–1065, 1993.

[20] R. Glen, G. Martin, A. Hill, R. Hyde, P. Wollard, J. Salmon, J. Buckingham, and A. Robertson. Computer-aided design and synthesis of 5-substituted tryptamines and their pharmacology at the $5-HT_{10}$ receptor: Discovery of compounds with potential anti-migraine properties. *J. of Medical Chemistry*, 38:3566–3580, 1995.

[21] N. Go and H. Scherga. Ring closure and local conformational deformations of chain molecules. *Macromolecules*, 3(2):178–187, 1970.

[22] W. Guida, R. Bohacek, and M. Erion. Probing the conformational space available to inhibitors in the thermolysin active site using monter carlo/energy minimization techniques. *J. of Comp. Chem.*, 13(2):214–228, 1992.

[23] A. Itai, N. Tomioka, M. Yamada, A. Inoue, and Y. Kato. Molecular superposition for rational drug design. In Kubingi, editor, *3D QSAR in Drug Design: Theory, Methods, and Applications*. ESCOM, 1995.

[24] L. Kavraki. *Random Networks in Configuration Space for Fast Path Planning*. PhD thesis, Stanford University, 1995.

[25] S. Khanna, R. Motwani, and F. F. Yao. Approximation algorithms for the largest common subtree problem. Technical Report STAN-CS-95-1545, Department of Computer Science, Stanford University, 1995.

[26] G. Klebe and T. Mietzener. A fast and efficient method to generate biologically relevant conformations. *J. of Computer-Aided Molecular Design*, 8:583–606, 1994.

[27] A. Leach. A survey of methods for searching the conformational space of small and medium sized molecules. In K. Lipkowitz and D. Boyd, editors, *Reviews in Computational Chemistry*, volume 2, pages 1–47. VCH Publishers, 1991.

[28] T. Lengauer. Algorithmic research problems in molecular bioinformatics. In *IEEE Proc. of the 2nd Israeli Symposium on the Theory of Computing and Systems*, pages 177–192, 1993.

[29] M. Lipton and W. Still. The multiple minimum problem in molecular modeling: Tree searching internal coordinate conformational space. *J. of Computational Chemistry*, 9(4):343–355, 1988.

[30] T. Lybrand. Computer simulation of biomolecular systems using molecular dynamics and free energy perturbation methods. In K. Lipkowitz and D. B. Boyd, editors, *Reviews in Computational Chemistry*, volume 1, pages 295–320. VCH Publishers, 1990.

[31] D. Manocha, Y. Zhu, and W. Wright. Conformational analysis of molecular chains using nano-kinematics. *Computer Application of Biological Sciences (CABIOS)*, 11(1):71–86, 1995.

[32] Y. Martin, M. Bures, E. Danaher, J. DeLazzer, and I. Lico. A fast new approach to pharmacophore mapping and its application to dopaminergic and benzodiazepine agonists. In *J. of Computer-Aided Molecular Design*, volume 7, pages 83–102, 1993.

[33] Y. C. Martin, M. G. Bures, and P. Willet. Searching databases of three-dimensional structures. In K. Lipkowitz and D. B. Boyd, editors, *Reviews in Computational Chemistry*, volume 1, pages 213–256. VCH Publishers, 1990.

[34] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, 1995.

[35] D. Pierre. *Optimization theory with applications*. Dover, NY, 1986.

[36] P. Raghavan and S. Irani. Combinatorial and experimental results for randomized point matching algorithms. *Proc. 12th. ACM. SCG*, pp. 68-77, 1996.

[37] P. J. Rezende and D. T. Lee. Point set pattern matching in d-dimensions. *Algorithmica,* 13, 387-404, 1995.

[38] A. Smellie, S. Kahn, and S. Teig. Analysis of conformational coverage: 1. validation and estimation of coverage. *J. Chem. Inf. Comput. Sci.*, 35:285–294, 1995.

[39] P. Willet. Searching for pharmacophoric patterns in databases of three-dimensional chemical structures. *J. of Molecular Recognition*, 8:290–303, 1995.