Improving Conformational Searches by Geometric Screening

Ming Zhang[1,2*] R. Allen White[1,2] Liqun Wang[1]
[1]Department of Biostatistics and Applied Mathematics
The University of Texas M.D. Anderson Cancer Center, Houston TX 77030 USA

[2]Graduate School of Biomedical Sciences
The University of Texas Health Science Center at Houston, Houston, TX 77225 USA

Ronald Goldman[3] Lydia Kavraki[3,4,6] Brendan Hassett[5]
Departments of [3]Computer Science, [4]Bioengineering, [5]Mathematics
Rice University, Houston TX 77251 USA

[6]Graudate Program in Structural & Computational Biology
Baylor College of Medicine, Houston TX 77030 USA

ABSTRACT

Motivation: Conformational searches in molecular docking are a time-consuming process with wide range of applications. Favorable conformations of the ligands that successfully bind with receptors are sought to form stable ligand-receptor complexes. Usually a large number of conformations are generated and their binding energies are examined. We propose adding a geometric screening phase before an energy minimization procedure so that only conformations that geometrically fit in the binding site will be prompted for energy calculation.

Results: Geometric screening can drastically reduce the number of conformations to be examined from millions (or higher) to thousands (or lower). The method can also handle cases when there are more variables than geometric constraints. An early-stage implementation is able to finish the geometric filtering of conformations for molecules with up to nine variables in one minute. To the best of our knowledge, this is the first time such results are reported deterministically.

Contact: mzhang@mdanderson.org

INTRODUCTION

The properties and possible interactions of molecules are intimately related to their accessible conformations. Conformational searches seek to solve the problem of identifying reachable conformations of molecules with low energy. Such conformations determine molecular flexibility and hence functionality, which are important in understanding a variety of biological phenomena at the molecular level. Deep understanding of molecular flexibility will greatly help the design of synthetic materials, the synthesis of drugs, the mechanism of surface catalysis, and the development of biological sensors (Cavasotto & Abagyan 2004; Perola & Charifson 2004; Henry & Ozkabak 1998; Lengauer 2002).

Conformational searches are common in many applications involving pharmacophore modeling, molecular docking, protein folding, and three-dimensional quantitative structure-activity relationships (Baker & Sali 2001; Diller & Merz 2001; Klebe 2000; Samudrala 2000; Song 2002). In this paper, we investigate a new geometric screening method to improve conformational searches in computer-assisted drug design.

Most drug discovery programs start from identification of a biomolecular target of potential therapeutic value. Drug-like compounds (leads) binding to the molecular target and interfering with its activity as a receptor or an enzyme are then sought. High throughput screening is usually performed on molecular libraries of known or constructed compounds. The resulting leads then undergo a cycle of chemical refinement and testing until a drug is developed for clinical trials.

When the structure of the biomolecular target is known, the most common virtual screening approach

*To whom correspondence should be addressed.

is molecular docking. In a successful ligand-receptor docking, the molecules exhibit geometric and chemical complementarity, which are essential for successful drug activity. Very often the binding site is specified by a pharmacophore, a set of 3-dimensional features. The features can be specific atoms, centers of (benzene) rings, positive or negative charges, hydrophobic or hydrophilic centers, hydrogen bond donors or acceptors. The pharmacophore reflects the prevailing idea in computer-assisted drug design that ligand binding is due primarily to the interaction between the features of the ligand and the complementary features of the receptor (Lavalle *et al.* 2000; Rarey *et al.* 1996). Identifying the conformations of the ligands whose features reach the target positions while still maintaining low energies gives rise to conformational search problems.

Earlier work The methods currently available for conformational searches fall into two categories: forward searches and inverse searches. Forward search methods are mostly energy oriented. That is, a large number of conformations are generated and their energies are examined. These methods assign values to the variables (torsional angles) systematically, randomly, or deterministically, and then the energies of these conformations are calculated (Bursulaya *et al.* 2003; Brooijmans & Kuntz 2003). Systematic search algorithms are based on grid values for each variable. The number of conformations to be examined increases exponentially when the number of variables increases. An example of a systematic search is the incremental construction algorithm (Ewing *et al.* 2001; Kramer 1999). Randomized search algorithms assign random values to the variables, thus avoid examining all conformations in the huge conformational space. One of the major concerns with randomized searches is the uncertainty of convergence. Usually multiple and independent runs are performed to improve the convergence. Examples of randomized searches are Monte Carlo (MC) methods and evolutionary algorithms (Jones *et al.* 1997; Lavalle *et al.* 2000). For deterministic searches, in each step, the current state determines the next state, whose energy (scoring function) is no more than that of the current state. Deterministic searches starting from the same point will always produce the same final state. Deterministic algorithms, on the other hand, may often get trapped in local minima that are surrounded by energy barriers. Examples of deterministic methods are molecular dynamics (MD) simulations (Nakajima *et al.* 1997; Pak & Wang 2000).

Inverse search methods try to solve systems of polynomial equations derived from the constraints in order to compute the values for the variables. Only the solutions of these equations generate conformations that are geometrically favorable for the binding. Usually the conformational space has a high dimension, but geometrically favorable conformations form only a low-dimensional locus. Thus the conformational space is dominated (almost everywhere) by geometrically unfavorable conformations. While forward search methods may suffer either from the huge number of possible configurations to be examined, or from the uncertainty of convergence, or from getting trapped in local minima (Verkhivker *et al.* 2000; Vieth *et al.* 1998), these concerns are irrelevant for inverse search methods. However, for inverse searches solving the system of derived equations, either analytically or numerically, is itself a difficult problem with immense computational complexity, especially when the number of solutions is infinite. There has been dramatically increased research on this topic in the past two decades (Aubry *et al.* 2002; Coutsias *et al.* 2004; Emiris & Mourrain 1999; Pedersen *et al.* 1993; Rojas 2000; Zhang & White 2003), though the algorithms developed are still far from practical. Solving systems of polynomial equations with more than six variables is still considered beyond the capabilities of modern computer algebra packages. Currently there are no good, general solvers to solve multi-variable (nonlinear) polynomial equations (Manocha 1998; Press *et al.* 1990).

Our approach We present a new approximation approach aiming to fill the gap between the capabilities of available conformational search algorithms and the demand of real applications. Our approach adds a geometric screening phase before a standard energy minimization procedure to improve conformational searches. Unlike current inverse search methods, the geometric screening phase approximates the solutions rather than solving for the solutions themselves and hence reports approximations of the geometrically favorable conformations. A subsequent minimization procedure can quickly identify the conformations favorable to ligand-receptor docking.

Contributions and significance There are several advantages of this approximation approach compared with currently available search methods: (1) The number of conformations to be examined for energy consideration is drastically reduced from millions (or higher) to thousands (or lower). Moreover, this reduction also prevents energy minimization techniques from getting trapped in local minima on the energy landscape. (2) Since only approximations are sought in the geometric screening, the computational time is much lower than that of computing the exact solutions. (3) The geometric screening phase no longer needs the assumption (as in most inverse search methods) that the number of the solutions to the equations is finite. In molecular docking, it is frequently seen that the number of variables (degrees of freedom) is more than the number of the

equations in the system. Solving for the exact solutions in this case is too difficult to be practical. Geometric screening reports approximations of solutions and thus avoids this difficulty. (4) The geometric screening is independent from the energy minimization process, and the method can be readily integrated into various conformational search packages currently available.

An early-stage implementation of the geometric screening method is able to finish the geometric filtering of conformations for molecules with up to nine variables in one minute. All the computations are carried on a 2GHz personal computer running Linux. To the best of our knowledge, this is the first time that all geometrically favorable conformations of such molecules can be deterministically reported in a timely manner.

SYSTEMS AND METHODS

In most molecular kinematics studies, the van der Waals radii, electric charges, bond lengths, and bond angles are assumed constant, while the torsional angles are allowed to change (Finn & Kavraki 1999; Henry & Ozkabak 1998). We follow this assumption here, but the algorithm developed in this paper generalizes to circumstances where other parameters may vary.

We further partition atoms into atom-groups. An atom-group is a set of connected atoms such that none of the bonds inside the atom-group rotate. Using atom-groups instead of individual atoms can considerably speed up the calculation of molecular conformations (Zhang & Kavraki 2002). Moreover, with atom-groups, we can focus on the interesting part (i.e., more rotatable bonds are present) using small atom-groups and put the less interesting part (i.e., all bonds are considered rigid such as a side chain) into big atom-groups. For simplicity, we also assume that there are no cycles of atom-groups in the molecule. When one atom-group is chosen as the root (anchor), the molecule becomes a tree with the atom-groups at the nodes.

Molecular Equations

Let us start by deriving the equations which describe atom positions.

First, we attach local frames (coordinate systems) to atom-groups to facilitate calculating atom positions. As in Figure 1(a), a local frame $F_i = \{Q_i; \mathbf{u}_i, \mathbf{v}_i, \mathbf{w}_i\}$ is attached to atom-group g_i as follows: Q_i is the atom of bond b_i in g_i; $\mathbf{w}_i$ is the unit vector along bond b_i pointing toward g_{i-1}; $\mathbf{u}_i$ is an arbitrary unit vector perpendicular to $\mathbf{w}_i$; $\mathbf{v}_i$ is perpendicular to $\mathbf{w}_i$ and $\mathbf{u}_i$ (cross product).

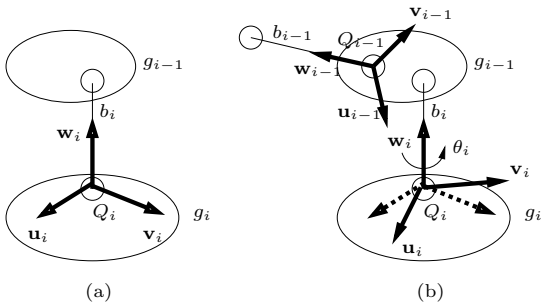

Figure 1: (a) Local frame $F_i = \{Q_i; \mathbf{u}_i, \mathbf{v}_i, \mathbf{w}_i\}$ at atom-group g_i. (b) Local frames $F_i = \{Q_i; \mathbf{u}_i, \mathbf{v}_i, \mathbf{w}_i\}$ at atom-group g_i and $F_{i-1} = \{Q_{i-1}; \mathbf{u}_{i-1}, \mathbf{v}_{i-1}, \mathbf{w}_{i-1}\}$ at atom-group g_{i-1}. θ_i is the torsional angle of bond b_i.

Next we derive the relations between neighboring local frames which will be used to calculate atom positions. Suppose the frame at atom-group g_i is $F_i = \{Q_i; \mathbf{u}_i, \mathbf{v}_i, \mathbf{w}_i\}$ and the frame at its parent atom-group g_{i-1} is $F_{i-1} = \{Q_{i-1}; \mathbf{u}_{i-1}, \mathbf{v}_{i-1}, \mathbf{w}_{i-1}\}$. Let the torsional angle of bond b_i be θ_i (Figure 1(b)). For each atom A in atom-group g_i, the coordinates (x_i, y_i, z_i) in F_i and the coordinates $(x_{i-1}, y_{i-1}, z_{i-1})$ in F_{i-1} are related by

$$(x_{i-1} \quad y_{i-1} \quad z_{i-1} \quad 1)^t = R_i \cdot (x_i \quad y_i \quad z_i \quad 1)^t,$$

where R_i is the product

$$\begin{pmatrix} \mathbf{u}_{i-1}\cdot\mathbf{u}_i & \mathbf{u}_{i-1}\cdot\mathbf{v}_i & \mathbf{u}_{i-1}\cdot\mathbf{w}_i & \mathbf{u}_{i-1}\cdot(Q_i - Q_{i-1}) \\ \mathbf{v}_{i-1}\cdot\mathbf{u}_i & \mathbf{v}_{i-1}\cdot\mathbf{v}_i & \mathbf{v}_{i-1}\cdot\mathbf{w}_i & \mathbf{v}_{i-1}\cdot(Q_i - Q_{i-1}) \\ \mathbf{w}_{i-1}\cdot\mathbf{u}_i & \mathbf{w}_{i-1}\cdot\mathbf{v}_i & \mathbf{w}_{i-1}\cdot\mathbf{w}_i & \mathbf{w}_{i-1}\cdot(Q_i - Q_{i-1}) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\cdot \begin{pmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

We then calculate the position of any atom A in atom-group g_i from the values of the rotatable bonds. Suppose $g_i, g_{i-1}, \ldots, g_0$ is a sequence of atom-groups, where g_j is the parent atom-group of g_{j+1}, $0 \le j \le i-1$, and g_0 is the root atom-group. Then the position of A in atom-group g_i is

$$(x \quad y \quad z \quad 1)^t = R_1 \cdots R_i \cdot (x_i \quad y_i \quad z_i \quad 1)^t,$$

where (x_i, y_i, z_i) are the (constant) coordinates of A in the local frame at atom-group g_i, and (x, y, z) are the coordinates of A in the global coordinate system.

Now we can formulate the conformational search problem analytically. Given (i) a molecule in an initial conformation, and (ii) the target positions (a_i, b_i, c_i) of some features, solve for the values of all the torsional angles so that the features in the final conformation reach their target positions.

For each feature A_i, if the target position is (a_i, b_i, c_i), then

$$(a_i \quad b_i \quad c_i \quad 1)^t = R_1 \cdots R_i \cdot (x_i \quad y_i \quad z_i \quad 1)^t, \quad (1)$$

where (x_i, y_i, z_i) are the local coordinates of feature A_i.

There are three equations in (1) — one for each rectangular coordinate (x, y, z); the last coordinate gives the identity $1 = 1$. Each of the three equations in (1) is linear in $\cos\theta_j, \sin\theta_j,\ j = 1, \ldots, i$. Instead of working with the trigonometric functions directly, we convert the cosine and sine functions into rational functions using the standard transformation

$$\cos\theta_j = \frac{1 - t_j^2}{1 + t_j^2}, \quad \sin\theta_j = \frac{2t_j}{1 + t_j^2}, \quad (2)$$

where $t_j = \tan(\theta_j/2)$. Multiplying both sides of these equations by the common divisors, we obtain three polynomial equations in $t_1, \ldots, t_i$. Each of these equations is quadratic in each of the variables. We call these polynomial equations the *molecular equations*.

Bernstein Bases

The molecular equations derived from the geometric constraints are represented using monomial bases. We are going to rewrite these polynomial equations using Bernstein bases to facilitate approximating the solutions of the molecular equations.

The Bernstein bases are a standard tool in computer graphics and computer aided design (Goldman 2002). Since the molecular equations are quadratic in each variable, we shall use the multi-quadratic Bernstein bases.

Definition The multi-quadratic Bernstein bases are the functions

$$B_{i_1}(t_1) \cdots B_{i_n}(t_n), \quad 0 \le i_1, \ldots, i_n \le 2,$$

where

$$\begin{array}{l} B_0(t_k) = (1 - t_k)^2 \\ B_1(t_k) = 2t_k(1 - t_k), \qquad 1 \le k \le n. \\ B_2(t_k) = t_k^2 \end{array}$$

Any multi-quadratic polynomial $p(t_1, \ldots, t_n)$ (of degree ≤ 2 in each variable) can be written uniquely as

$$p(t_1, \ldots, t_n) = \sum_{0 \le i_1, \ldots, i_n \le 2} c_{i_1,\ldots,i_n} \cdot B_{i_1}(t_1) \cdots B_{i_n}(t_n),$$

where the $c_{i_1,\ldots,i_n}$'s are constant coefficients.

These Bernstein bases have the following two important properties:

(1) When all the parameters $t_1, \ldots, t_n$ are in the range of $[0, 1]$, all the Bernstein basis functions are non-negative. Therefore, if the coefficients of a polynomial are all negative (or all positive), the polynomial has no solutions in the parameter space $[0, 1]^n$.

(2) The sum of all the Bernstein basis functions is identically one. Thus the value of any polynomial (all parameters in $[0, 1]$) will lie in the convex hull of the Bernstein coefficients. Therefore, if the convex hull is small enough, the value of the polynomial (parameters in $[0, 1]$) can be approximated by the coefficients.

Since these two properties of Bernstein bases rely on the fact that all the parameters lie in $[0, 1]$, we need to make changes to the molecular equations derived so that the parameter space is $[0, 1]^n$. This is easily done by shifting the search space $[-r, r]^n$ (r is the search radius) to $[0, 2r]^n$ and then shrinking the range to $[0, 1]^n$.

Subdivision

We now use the Bernstein bases to perform subdivision, aiming to approximate the solutions of the molecular equations. First let us illustrate the subdivision scheme using a simple example.

Example Suppose

$$P(t) = P_0 \cdot B_0(t) + P_1 \cdot B_1(t) + P_2 \cdot B_2(t), 0 \le t \le 1,$$

is a curve (Figure 2).

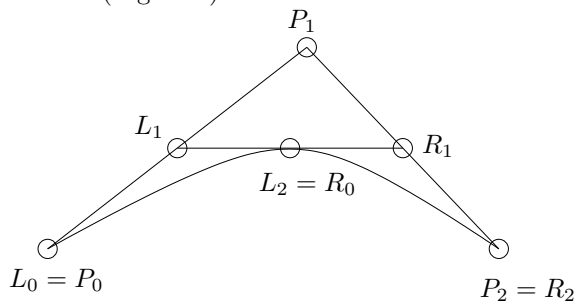

Figure 2: Illustration of simple subdivision. (P_0, P_1, P_2) are coefficients of $P(t)$. The left "wing" (L_0, L_1, L_2) are the coefficients of $P_L(t)$ and the right "wing" (R_0, R_1, R_2) are the coefficients of $P_R(t)$.

Let

$$\begin{array}{rcl} P_L(t) & = & L_0 \cdot B_0(t) + L_1 \cdot B_1(t) + L_2 \cdot B_2(t), \\ P_R(t) & = & R_0 \cdot B_0(t) + R_1 \cdot B_1(t) + R_2 \cdot B_2(t), \end{array}$$

where

$$\begin{array}{ll} L_0 = P_0, & R_0 = L_2, \\ L_1 = (P_0 + P_1)/2, & R_1 = (P_1 + P_2)/2, \\ L_2 = (P_0 + 2P_1 + P_2)/4, & R_2 = P_2. \end{array}$$

Then

$$P_L(t) = P(\frac{t}{2}), \quad P_R(t) = P(\frac{t+1}{2}).$$

That is, $P_L(t)$, $0 \le t \le 1$, is the left half of the polynomial $P(t)$ — from $P(0)$ to $P(1/2)$, and $P_R(t)$, $0 \le t \le 1$, is the right half of $P(t)$ — from $P(1/2)$ to $P(1)$.

Note that after subdivision, the parameter t of $P_L(t)$ and $P_R(t)$ can take any value in $[0, 1]$ (which is necessary to perform further subdivisions using Bernstein bases). However, the parameter ranges of $P_L(t)$ and $P_R(t)$ correspond to half of the parameter range of $P(t)$: $[0, 1/2]$ and $[1/2, 1]$. Thus in subdivision process, the parameter range of a polynomial is usually referred to the corresponding sub-range of the original polynomial.

For any multi-quadratic polynomial $p(t_1, \ldots, t_n)$, the subdivision can be performed as follows. Subdivide $p(t_1, \ldots, t_n)$ with respect to t_1 (while $t_2, \ldots, t_n$ are regarded as constants) and two polynomials are obtained as in the above example. Subdivide these two polynomials with respect to t_2 (while $t_1, t_3, \ldots, t_n$ are regarded as constants) and four polynomials are obtained. Keep this process till subdivision is carried with respect to t_n and 2^n polynomials are obtained.

Therefore, a polynomial $p(t_1, \ldots, t_n)$, whose parameter space is $[0, 1]^n$, can be subdivided into 2^n polynomials, whose parameter spaces (in the original $[0, 1]^n$) are small cubes $[\frac{i_1}{2}, \frac{i_1+1}{2}] \times \ldots \times [\frac{i_n}{2}, \frac{i_n+1}{2}], 0 \le i_1, \ldots, i_n \le 1$. Each of the 2^n resulting polynomials can be further subdivided into 2^n polynomials with even smaller parameter cubes $[\frac{i_1}{4}, \frac{i_1+1}{4}] \times \ldots \times [\frac{i_n}{4}, \frac{i_n+1}{4}], 0 \le i_1, \ldots, i_n \le 3$. A k-level subdivision is generated by repeating this process k times. A subdivision tree is illustrated in Figure 3.

The root node of the subdivision tree contains the original polynomial, whose parameter space is $[0, 1]^n$ – we refer to this cube as the root cube. At level 1 of the subdivision, there are 2^n nodes. These 2^n small parameter cubes do not overlap but they may share a face, an edge, or a vertex. The union of these small cubes is the root cube $[0, 1]^n$. At level j, there are $2^{n \times j}$ cubes with size $1/2^j$ – the root cube has size 1. The union of all level j small cubes is the root cube.

Each node of the subdivision tree is examined to see whether the corresponding small cube contains possible solutions of the original polynomial. A simple criterion is that if the coefficients are all negative (or all positive), then the small cube definitely does not contain any solution of the original polynomial. Such cubes are called *empty cubes*. Empty cubes are immediately eliminated from further subdivision and the branch below is pruned in the subdivision tree. Therefore, identifying as many as possible empty cubes as early as possible is critical to reduce the size of the subdivision tree.

The geometric constraints on the feature atoms generate a set of polynomial equations. We place all these polynomials at the root of the subdivision tree and carry on the subdivision process. A small cube in the subdivision tree will be identified as an empty cube if any one of the polynomials does not have solutions within the cube. Thus more polynomials help to prune the subdivision tree.

This subdivision algorithm does not need the assumption that the number of solutions of the molecular equations is finite. Many inverse search methods need this assumption and hence require that the number of variables does not exceed the number of equations. In molecular docking, frequently there are more variables than equations, and the number of solutions is infinite. The subdivision algorithm can handle conformational searches in such cases.

The small cubes at the bottom level of the subdivision tree have size $1/2^k$. When k is big enough, say 6, i.e., after 6 levels of subdivision, the small cubes become small enough that any point in the small cube can be regarded as an approximation to the possible solutions within the small cube. Therefore, this algorithm reports the small non-empty cubes at the bottom of the subdivision tree as approximations to the solutions of the original molecular equations. These approximations are converted back into approximate values of the variables of the ligand molecules. Thus the output of the geometric screening process generate conformations satisfying the geometric constraints that can be input to a subsequent energy minimization procedure.

ALGORITHM

The geometric screening method is illustrated with the following pseudo code.

(1) Generate equations from the geometric constraints on features (c.f. Equation (1)).

(2) Convert these trigonometric equations into molecular equations (c.f. Equation (2)).

(3) Rewrite these molecular equations from monomial bases to Bernstein bases using

$$\begin{aligned} 1 &= B_0(t_k) + B_1(t_k) + B_2(t_k), \\ t_k &= 0.5 \cdot B_1(t_k) + B_2(t_k), \\ t_k^2 &= B_2(t_k). \end{aligned}$$

Let P = all molecular equations in Bernstein bases and set subdivision level of P to 0.

(4) Subdivide P recursively and report approximate solutions. Set max-subd-level to, for example, 6.

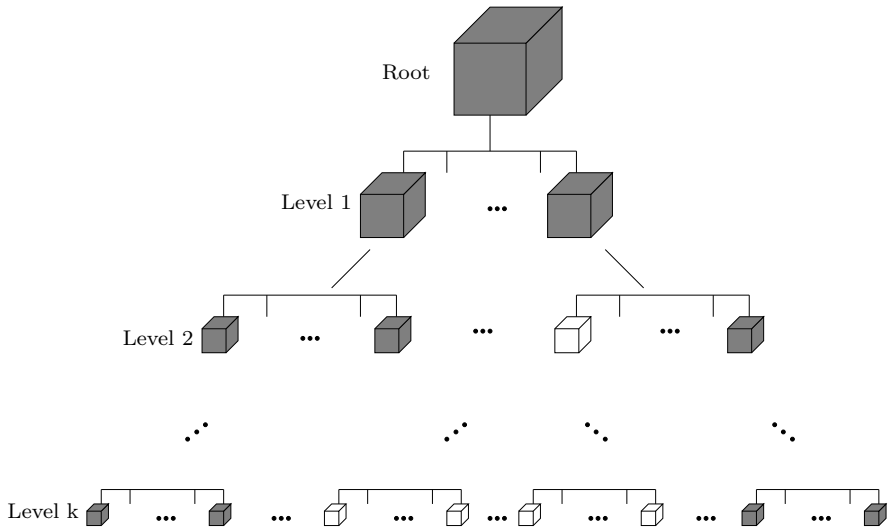

Figure 3: An illustration of the subdivision tree. The root node represents the original cube $[0,1]^n$. Each node at level j represents a small cube of size $1/2^j$ along with the associated molecular polynomials. Shaded cubes contain possible solutions of the original polynomials, while unshaded cubes (empty cubes) contain no solutions inside. All children nodes of empty cubes are empty cubes.

```
Screening(P) {
  if subdivision level < max-subd-level
    if for all p ∈ P the coefficients are mixed (i.e.,
    positive and negative)
      subdivide P into P_1, ..., P_{2^n};
      add subdivision level of P_1, ..., P_{2^n} by 1;
      Screening(P_1), ..., Screening(P_{2^n})
  if subdivision level ≥ max-subd-level
    if for all p ∈ P the coefficients are mixed,
    report the corresponding cube
}
```

DISCUSSION

Complexity Estimate The output of the geometric screening is a set of small cubes covering the solutions of the molecular equations. It is easy to see that the more levels of subdivision we perform, the more nodes the subdivision tree has. It is also easy to see that the number of non-empty cubes at the bottom of the subdivision tree is equal to the number of isolated solutions if each such cube contains a single isolated solution. So the computational complexity of the geometric screening mainly depends on the number of the levels of subdivision and isolated solutions.

If the number of solutions is infinite, for example, when there are more variables than constraints, the solutions may form a curve or hyper-surface. Then the number of small cubes covering the solutions increases exponentially as the level of subdivision increases. In this case, the level of subdivision will be limited to a smaller number so that relatively bigger cubes (coarse approximations) are reported in a timely manner.

Another factor to the complexity is the steepness of molecular equations near the solutions. If the molecular equations are steep around a solution, only a small number of cubes will be reported. Otherwise, a large number of cubes near the solution will be reported when the values of the molecular equations in these cubes are under a pre-defined threshold (and hence regarded as 0). It follows that non-empty cubes at the bottom of the subdivision tree may not necessarily contain solutions of the molecular equations, though they may provide good starting points for the subsequent energy minimization process.

Thus the computational complexity (in the worst case) is exponential on the level of subdivision and number of variables when the number of solutions is infinite or the molecular equations are "flat" near the solutions.

Results We have implemented the above subdivision algorithm to approximate solutions of the molecular equations. The program has been tested on molecules with up to 9 degrees of freedom [c.f. Figure 4]. (Note that each ellipse in Figure 4 represents an atom-group rather than an atom.)

Our program correctly reports the approximate real solutions of the molecular equations. (Verification is simple: conformations generated from these approximations should have their features near their target positions.) For 3 molecular equations, the execution time is less than 0.1 seconds; for 6 equations, the execution

time is less than 10 seconds; for 9 equations, the execution time is less than 1 minute. We also run the program with 9 variables but 8 or less equations. The execution time can go up to 10 minutes to report a much larger number of cubes. All the computations are carried on a 2GHz personal computer running Linux.

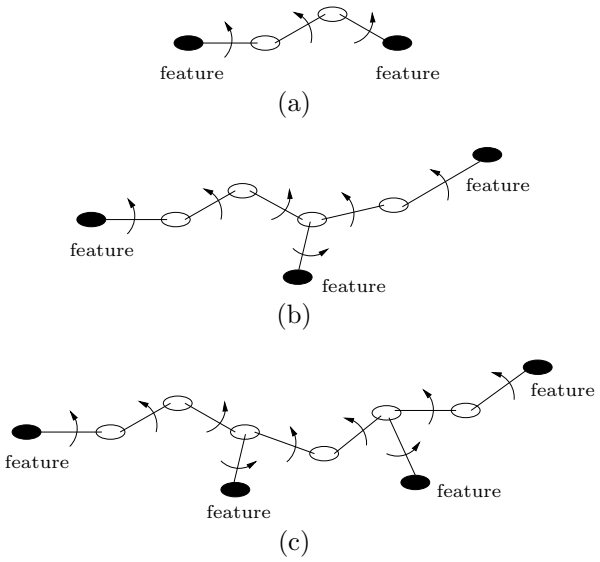

Figure 4: Schematic illustration of conformational searches with 3, 6, 9 rotatable bonds. Each ellipse represents an atom-group. The black ellipses represent atom-groups where a feature lies. One feature is chosen as the anchor and always remains fixed, while other features have pre-specified target positions. There are 3, 6, 9 equations generated respectively from (a), (b), (c).

The example molecular equations are too big to be included in this paper, but are available at the website "http://odin.mdacc.tmc.edu/∼ming/invermatics". These systems of equations are generated by specifying reachable target positions for the features – thus real solutions exist. We have circulated these equations around and no other group is able to report all the real solutions. A molecule that was used to generate 9 molecular equations is shown in Figure 5.

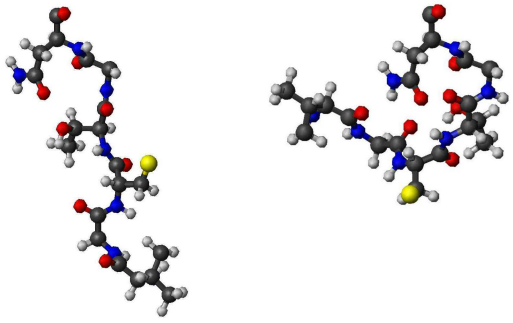

Figure 5: An example molecule used to generate 9 molecular equations. Nine bonds are allowed to rotate while the others are rigid. The part at the top is chosen as the anchor atom-group and remains fixed. The initial conformation is shown at the left hand side and one valid conformation satisfying the geometric constraints is shown at the right hand side.

Applicability Usually the ligands in drug design are small molecules. Figure 6 shows a histogram of the number of rotatable bonds of compounds in a screening library from Specs, which is representative of other chemical database suppliers (Baurin *et al.* 2004). Most of the compounds ($> 80\%$) have no more than 9 rotatable bonds and can be handled by geometric screening at the current implementation.

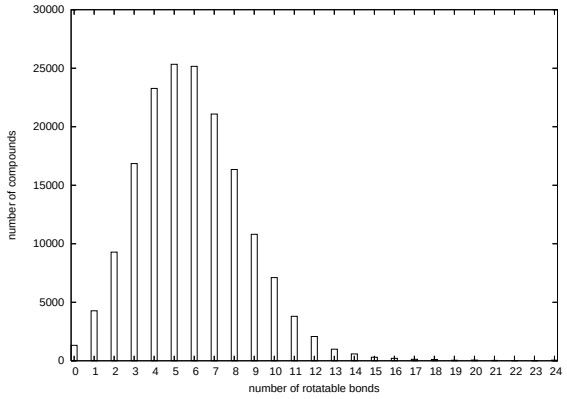

Figure 6: Distribution of number of rotatable bonds of screening compounds in the Specs library released in May 2004 (www.specs.net). The values were calculated with the "dbanalyze" command available in the Unity package of Sybyl.

For compounds with more variables, a combination of geometric screening and other conformational search methods can be used: geometric screening can be used with respect to 9 of the variables while the values of the rest variables can be determined by other systematic, randomized, or deterministic search methods. Also improving geometric screening to handle more variables with collision checking and more accurate empty cube detection (to help pruning the subdivision tree) is under investigation.

For conformational search methods, an interested user may consider the following table based on the number of rotatable bonds of the ligand.

≤ 3	systematic enumeration
4–9	geometric screening followed by energy minimization
> 9	combination of geometric screening and forward searches; but no good method performing complete searches yet

Table 1: Recommended conformational search methods based on number of rotatable bonds.

Parameter Space The values of the rotatable bonds are the torsional angles while the solutions to the molecular equations are the tangent of half angles. If a torsional angle θ is in $[-\pi/2, \pi/2]$, the mapping from tangent to angle is straightforward: $\theta = 2 \cdot \arctan(\tan(\theta/2))$. If θ is also in $[\pi/2, 3\pi/2]$, let $\theta' = \theta - \pi$ and θ' is in the range $[-\pi/2, \pi/2]$. The molecular equations will be re-written as polynomials in $\tan(\theta'/2)$. Thus the value of θ in $[\pi/2, 3\pi/2]$ can be readily recovered from θ'.

If there are k angles have values beyond $[-\pi/2, \pi/2]$, one system of molecular equations becomes 2^k systems of molecular equations where all angles are within $[-\pi/2, \pi/2]$. The union of solutions of these 2^k systems of molecular equations are the solutions of the original molecular equations.

The geometric screening uniformly exploits the parameter space of the tangent of half angles, not the space of the angles. Thus at the subsequent energy minimization process, torsional angles near 0 (or π)will be perturbed in a narrower neighborhood, while torsional angles near $-\pi/2$ or $\pi/2$ will be examined in a wider neighborhood.

Improvements The improvement of geometric screening on conformational searches has two folds. First, geometric screening reduces the computational time as the number of conformations to be examined is reduced. Second, geometric screening performs a complete search in the conformational space with respect to the geometric constraints: if a solution exists, it will be reported. Moreover, the improvement is independent on the energy minimization process.

Conclusion Using subdivision scheme, geometric screening can efficiently isolate and locate all real solutions of the molecular equations, hence compute all the geometrically favorable conformations for ligand-receptor docking. To the best of our knowledge, this is the first time that all real solutions of such systems (up to nine variables) have been deterministically reported and in a timely manner.

Acknowledgment

This research was supported by funds from the University Cancer Foundation at the University of Texas, M.D. Anderson Cancer Center. R. Allen Write is partially supported by SPORE grant 5P30CA016672-27. Ronald Goldman is partially supported by grant NSF/INRIA 0421771. Work on this paper by Lydia Kavraki has been supported in part by a Whitaker Biomedical Engineering Grant and a Sloan Fellowship. Brendan Hassett is partially supported by NSF grants DMS-0196187 and DMS-0134259, and by the Sloan Foundation. We would like to thank Dr. David Maxwell for helpful discussions and the histogram of compound libraries, Dr. John McMurray for useful advice, and the referees for insightful comments and suggestions.

References

Aubry, P., Rouillier, F., and El Din, M.S. (2002) Real Solving for Positive Dimensional Systems, *J. of Symbolic Computation*, Volume 34, Issue 6, 543-560.

Baker, D. and Sali, A. (2001) Protein structure prediction and structural genomics, *Science*, Vol. 294, 93-96.

Baurin, N., Baker, R., Richardson, C., Chen, I., Foloppe, N., Potter, A., Jordan, A., Roughley, S., Parratt, M., Greaney, P., Morley, D., Hubbard, R.E. (2004) Drug-like Annotation and Duplicate Analysis of a 23-Supplier Chemical Database Totalling 2.7 Million Compounds. *J. Chem. Inf. Comput. Sci.*, 44:643-651.

Brooijmans, N. & Kuntz, I.D. (2003) Molecular recognition and docking algorithms, *Annual Review of Biophysis and Biomelecular Structure*, Vol. 32, 335-373.

Bursulaya, B.D., Totrov, M., Abagyan, R., Brooks, C.L. (2003) Comparative study of several algorithms for flexible ligand docking. *J. Computer-Aided Molecular Design* 17:755-763.

Cavasotto, C.N., Abagyan. R.A. (2004) Protein flexibility in ligand docking and virtual screening to protein kinases. *J. Mol. Biol.* 12, 337(1):209-225.

Coutsias, E.A., Seok, C., Jacobson, M.P., Dill, K. (2004) A Kinematic View of Loop Closure, *J. of Computational Chemistry*, Volume 25, Issue 4, 510-528.

Diller, D.J., Merz, K.M. Jr. (2001) High throughput docking for library design and library prioritization. *Proteins*, 43:113-124.

Emiris, I., Mourrain, B. (1999) Computer Algebra Methods for Studying and Computing Molecular Conformations, *Algorithmica*, Vol. 25, Issue 02, pp 372-402.

Ewing, T.J.A., Makino, S., Skillman, A.G., Kuntz, I.D. (2001) DOCK 4.0: search strategies for automated molecular docking of flexible molecule databases. *J. Comput. Aid. Mol. Des.* 15:411-428.

Finn, P.W., Kavraki, L.E. (1999) Computational Approaches to Drug Design. *Algorithmica*, 25, 347–371.

Goldman, R. (2002) *Pyramid Algorithms: A Dynamic Programming Approach to Curves and Surfaces for Geometric Modeling*, Morgan Kaufmann.

Henry, D.R., Ozkabak, A.G. (1998) Conformational Flexibility in 3D Structure Searching. *Encyclopedia of Computational Chemistry*. Schleyer,P.v.R., Ed., Wiley, New York.

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

Klebe, G. (2000) Recent developments in structure-based drug design. *J. Mol. Med.* 78:269-281.

Kramer, B., Rarey, M., Lengauer, T. (1999) Evaluation of the FlexX incremental construction algorithm for protein-ligand docking. *Proteins*, 37:228-241.

Lavalle, S.M., Finn, P.W., Kavraki, L.E., Latombe, J.-C. (2000) A Randomized Kinematics-based Approach to Pharmacophore-Constrained Conformational Search and Database Screening. *J. of Computational Chemistry*, Vol. 21, No. 9, 731–747.

Lengauer, T. (Ed.) (2002) Bioinformatics – From Genomics to Drugs. Weinheim: Wiley-VCH Verlag GmbH.

Manocha, D. (1998) Numerical Methods for Solving Polynomial Equations, Proceedings of Symposium in Applied Mathematics, Vol. 53, 41–66.

Nakajima, N., Nakamura, H., Kidera, A. (1997) Multicanonical ensemble generated by molecular dynamics simulation for enhanced conformational sampling of peptides. *J. Phys. Chem. B* 101:817-824.

Pak, Y., Wang, S.(2000) Application of a molecular dynamics simulation method with a generalized effective potential to the flexible molecular docking problems. *J. Phys. Chem. B* 104:354-359.

Pedersen, P., Roy, M.-F. and Szpirglas, A. (1993) Counting Real Zeros in the Multivariate Case, in *Computational Algebraic Geometry* (F. Eyssette and A. Galligo, eds), Birkhauser, Boston, 203–224.

Perola, E., Charifson, P.S. (2004) Conformational analysis of drug-like molecules bound to proteins: an extensive study of ligand reorganization upon binding. *J. Med. Chem.* 6;47(10):2499-510.

Press,W., Flannery,B., Teukolsky,S., Betterling,W. (1990) *Numerical Recipes: The Art of Scientific Computing*, Cambridge U. Press, Cambridge.

Rarey, M., Wefing, S., Lengauer, T. (1996) Placement of medium-sized molecular fragments into active sites of proteins. *J. Comput. Aid. Mol. Des.* 10:41-54.

Rojas, J.M. (2000) Some Speed-Ups and Speed Limits for Real Algebraic Geometry, *J. of Complexity*, FoCM 1999 special issue, vol. 16, no. 3, pp. 552-571.

Samudrala, R., Huang, E.S., Koehl, P. and Levitt. L. (2000) Constructing Side Chains on Near-Native Main Chains for ab initio Protein Structure Prediction. *Protein Eng.*, 3, 453-457.

Song, G., Amato, N.M. (2002) Using Motion Planning to Study Protein Folding Pathways. *J. of Computational Biology*, 9(2), 149-168.

Verkhivker, G.M., Bouzida, D., Gehlhaar, D.K., Rejto, P.A., Arthurs. S., et al. (2000) Deciphering common failures in molecular docking of ligand-protein complexes. *J. Comput. Aid. Mol. Des.* 14:731-751.

Vieth, M., Hirst, J.D., Dominy, B.N., Daigler, H., Brooks, C.L. (1998) Assessing search strategies for flexible docking. *J. Comput. Chem.* 19:1623-1631.

Zhang, M. and Kavraki, L.E. (2002) A New Method for Fast and Accurate Derivation of Molecular Conformations, *J. of Chemical Information and Computer Sciences*, 42 (1), 64 -70.

Zhang, M. and White, R.A. (2003) A Molecular Inverse Kinematics Problem: An Approximation Approach and Challenges, proceedings of ASCM 2003, 276-287.