A Novel Subshape Molecular Descriptor

Santosh Putta,*,† John Eksterowicz,‡ Christian Lemmen,§ and Robert Stanton||

Deltagen Research Labs, 740 Bay Road, Redwood City, California 94063

Received November 14, 2002

Molecules with similar shapes and features often have similar biological activity. Several computational approaches search chemical databases for new leads or templates based on overall molecular shape similarity. However, active molecules often present critical subshapes that are required for binding, which may be missed by comparing overall shape similarity. We present a new approach to compare molecular shapes of different sizes and to calculate subshape similarity. We developed a skeletal representation of the shape which is topologically unrelated to covalent chemical connectivity. This simplifies rotational and translational sampling. We test initial possible alignments by matching similar triangles. This triangle-matching filter rapidly eliminates most geometrically impossible matches. Surviving matches are filtered further in successive stages. These stages involve direction, feature, and shape matching procedures. Our approach is applied to several situations demonstrating lead discovery and evolution.

INTRODUCTION

It is well-known that the molecular shape plays a key role in ligand−receptor binding. Several computational methods, ranging from docking to molecular superposition/alignment, make use of this information. In particular, in the absence of structural data, molecular superposition has been widely used as a technique to understand the ligand/protein binding based on shape and chemical feature similarities. Several of these small molecule alignment methods have been used predominantly to assess similarity in individual cases or at most comparatively small sets of compounds. However, progress in algorithmic efficiency as well as in terms of available hardware permit the application of such tools in the virtual high throughput screening (HTS) arena. More background on several alignment methods can be found in recent reviews[1−3] and the references therein. In these methods the molecular shape is typically modeled as the total volume of a compound via spheres,[4] Gaussians,[5] or other representations of densities such as grid based encoding.[6,7] Further these methods often concentrate on finding overall (or whole) shape similarity between molecules.

While overall shape comparison is sufficient when the molecules are of similar size, it is not useful when molecular size varies greatly. In such cases it is important to be able to align the smaller molecule to a portion of the larger molecule and to reveal key similarities between them. Figure 1 illustrates a case where two molecules of different sizes have several common characteristics. However, aligning molecules of different sizes can be computationally very expensive due to the extent to which translation and rotational spaces need to be covered to obtain correct alignments. The subshape alignment procedure presented here is designed to efficiently address this issue. Other methods designed earlier to address partial shape alignments include the use of quadratic shape descriptors[8] and combining a feature selection procedure with the molecular overlay process.[9]

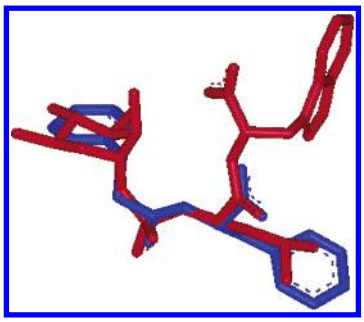

Figure 1. The ligands PPP (blue) and 1TLP (red) are active against thermolysin protein. Shown here is their X-ray crystal alignment in their bound configuration. It is clear from the alignment that only a part of the PPP aligns with 1TLP. It would be very difficult to obtain this alignment using an overall molecular similarity as opposed to subshape similarity.

A successful three-dimensional alignment method must have two important characteristics. First the method needs to be independent of the two-dimensional structure and depend primarily on the three-dimensional shape and the spatial configuration of the chemical features. This property is important in applications such as searching a database to find new leads using existing active compounds. In such cases the goal is often to find new leads that are structurally different from the already known actives. In addition, the method should be able to find all possible alignments between molecules, particularly when the molecules are dramatically different in size. There may be many alignments that are sterically and chemically reasonable between a small molecule and a larger one (Figure 2). With no other prior knowledge in such cases, the method needs to be inclusive (i.e. find all these reasonable alignments).

* Corresponding author phone: (510)742-0486; fax: (510)742-0486; e-mail: sputta@RationalDiscovery.com.

† Current address: Rational Discovery LLC, 555 Bryant Street, #467, Palo Alto, CA 94301.

‡ Current address: Celera Genomics, 180 Kimball Way, South San Francisco, CA 94080.

§ Current address: BiosolveIT GmbH, An der Ziegelei 75, 53757 Sankt Augustin, Germany.

|| Current address: Pfizer Discovery Technology Center, 620 Memorial Drive, Cambridge MA 02139.

10.1021/ci0256384 CCC: $25.00
Published on Web 07/26/2003

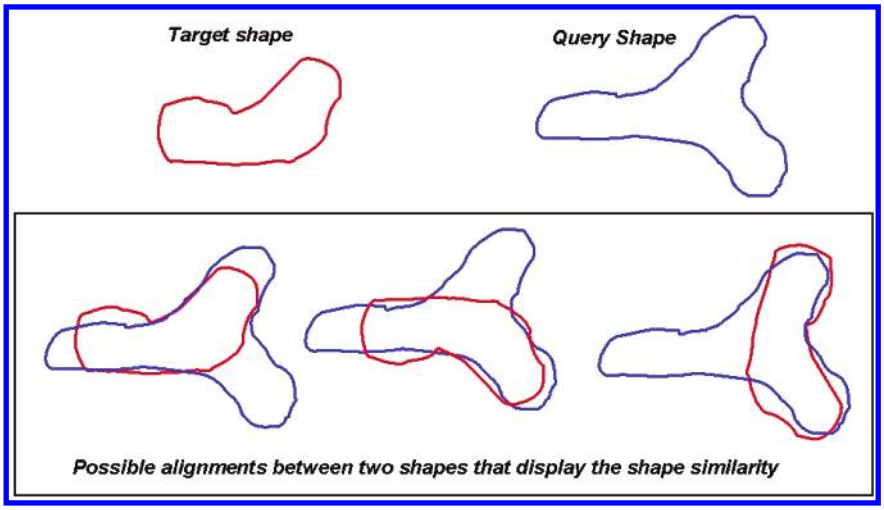

Figure 2. Two-dimensional illustration of possible alignments between two shapes that display the desired steric similarity.

A novel method for calculating subshape alignments with all the characteristics mentioned above is presented here. The subshape alignment algorithm functions through a series of filters so that grossly impossible alignments are eliminated early in computationally inexpensive steps. More reasonable alignments are retained through successively more refined (and computationally more expensive) steps until only a handful of valid alignments remain. The method will be detailed below. The results from a variety of experiments will be shown highlighting some of the key points of the method. These include alignments of (1) macromolecular and ligand sized shapes, (2) similarly sized ligand shapes, and (3) ligand and fragment shapes (roughly 25−50% of the ligand molecular volume). Finally an example of database screening will be given, highlighting the speed and flexibility of the algorithm.

METHOD

The subshape alignment method is designed to operate on two molecular shapes. The shapes are referred to as the *target* and *query*. The target shape is typically derived from one or more biologically active molecules. The three-dimensional conformations of target molecules can be obtained either from X-ray crystal structures or conformational analysis. A typical application involves only a few of these target shapes. On the other, hand query shapes are typically derived from molecules in a database or virtual library that need to be tested for their similarity to the active molecules. Typical applications may involve several thousand such query molecules. Conformational analysis is used to obtain tens of thousands of three-dimensional conformations for these query molecules. Each conformation is then considered as an independent query shape and tested against the target individually.

The main task of the method is to find all possible alignments between the target and query shapes that display *subshape or shape similarity* between them. While our definition of subshape and shape similarity is presented later in the paper, it is important to note here that a pair of target and query shapes may have several ways of aligning to each other. A simple illustration of this concept is shown in Figure 2. A straightforward approach to finding these alignments would be to thoroughly sample the translational and rotational spaces and pick the alignments with the desired properties. However, such an approach is computationally very expensive and not practical for most drug-design problems. Our method overcomes the efficiency issue by going through successive and efficient steps of pruning alignments, starting only with the alignments that are geometrically reasonable.

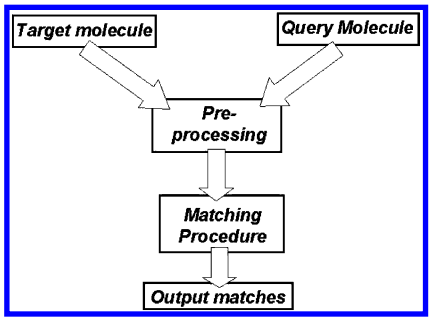

Figure 3. Flowchart for the overall methodology.

The method proceeds in two stages, preprocessing and matching. Figure 3 shows an illustration of this procedure. The preprocessing stage consists of computing and assigning a set of properties to the target and query shapes. These properties are then used in the matching stage to obtain and then prune the alignments between the target and query shapes.

PREPROCESSING

The preprocessing stage consists of computing and assigning a set of properties to the query and target shapes. These properties include grid encoding of the shape, terminal and skeleton points, directions, and feature types. These properties play an important role in the matching stage explained in the next section. The terminal and skeleton points provide the basis for generating initial alignments that are fed through successive stages of filtering. This subsection will define each of these properties and the underlying algorithms used to compute them. Figure 4 gives a flowchart of the procedure followed in this stage. Notice that the *skeleton points* are computed only for the target shape. This allows an exhaustive coverage of one of the shapes, necessary in finding all possible alignments. The remaining steps are common to both the shapes.

Shape Encoding: Volume. A three-dimensional cubic grid is used to encode a target/query shape from a conforma-

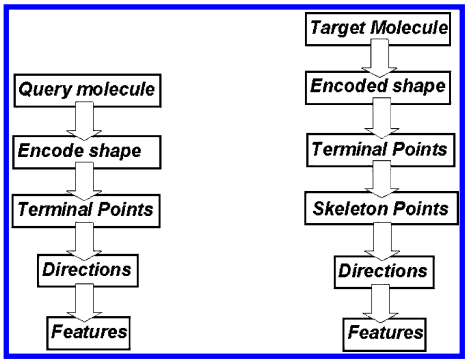

Figure 4. Preprocessing steps for the query and target molecules. Notice here that the procedure followed for the two shapes is slightly different. For the target shape one additional step of computing skeleton points is performed.

tion. The first step in this process is to pose the conformer in a standard way, called the *canonical orientation*. The centroid is calculated from equally weighed coordinates of the heavy (i.e. non-hydrogen) atoms. If $\vec{p}_i$ is the position vector of the *i*th heavy atom, $\vec{p}_c$ the position vector of the centroid is calculated as

$$\vec{p}_c = \frac{1}{n}\sum_{i=1}^{n}\vec{p}_i \tag{1}$$

where n is the total number of heavy atoms in the compound.

The principal axes of the conformer about the centroid are determined by computing the eigenvectors of the following covariance matrix

$$C = \begin{bmatrix} \sum_{i=1}^{n} x_i x_i & \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} x_i z_i \\ \sum_{i=1}^{n} y_i x_i & \sum_{i=1}^{n} y_i y_i & \sum_{i=1}^{n} y_i z_i \\ \sum_{i=1}^{n} z_i x_i & \sum_{i=1}^{n} z_i y_i & \sum_{i=1}^{n} z_i z_i \end{bmatrix} \tag{2}$$

where x_i, y_i, and z_i are the coordinates of the *i*th heavy atom. The conformer is translated and rotated such that the centroid coincides with the origin and the eigenvectors align with the *x*, *y*, and *z* axes. It is important to note that even though this process of canonicalization is used the conformer orientation has no effect on the actual matching procedure described later. Instead canonicalization limits the size of the grid that needs to be used for encoding the shape. One difference between the handling of target and query shapes is that once set, the target shape is never moved from is canonical position.

Once a conformer orientation has been canonicalized, it is superimposed onto a three-dimensional cubic lattice (grid) to facilitate encoding its shape. The grid dimensions are chosen to accommodate all compounds of interest in their canonical orientation. The edge length of each cube in the lattice is specified by the user and is referred to as the *grid spacing*, g_s, which determines the accuracy of the shape encoding. Once chosen the grid remains constant for every shape. The grid occupancy of a conformer is encoded using a bit-vector, whose length is proportional to the number of grid points. Two bits are assigned to store the occupancy of an individual grid point. This allows four distinct values based on whether the grid point is interior, exterior ,or close to the surface of the conformer shape. Table 1 shows the encoding scheme based on the distance of a grid point to the closest heavy atom and the van der Waals' radius of the atom. Figure 5 shows a two-dimensional illustration of the encoding for a typical molecule. The idea of using four levels for encoding a grid point is borrowed from the *anti-aliasing* technique used in computer graphics.[10] The accuracy achieved this way is comparable to a two value encoding on a grid with half the grid spacing. However, in the latter case 8 times as many grid points and 4 times as many bits are required.

Table 1. Occupancy at Each Grid Point Is Assigned Based on Its Distance to the Closest Heavy Atom[a]

range of d	grid occupancy value
$d \le r_w$	3
$r_w < d \le \left(r_w + \frac{r_b}{s_g}\right)$	2
$\left(r_w + \frac{r_b}{s_g}\right) < d \le \left(r_w + \frac{2.r_b}{s_g}\right)$	1
$d > \left(r_w + \frac{2.r_b}{s_g}\right)$	0

[a] d is the distance of the grid point to the center of closest heavy atom. r_w is the van der Waals' radius of the heavy atom. r_b is a user specified parameter. s_g is the grid spacing.

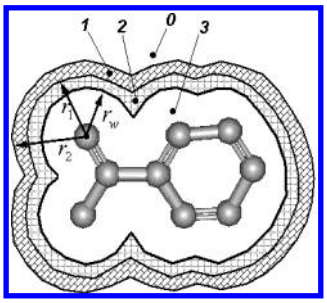

Figure 5. A two-dimensional illustration of the grid encoding of the shape. All grid points in the white region at the center get a value of 3. The value decreases gradually, as shown by the shaded regions, as one moves beyond the van der Waals' surface of the molecule. r_w is the van der Waals' radius. $r_1 = (r_w + r_b/s_g)$ and $r_2 = (r_w + 2.r_b/s_g)$, where r_b is a user specified parameter. s_g is the grid spacing.

Terminal Points. Terminal points are evenly spaced inside a target or query shape. Typically 5 points are computed for a ligand sized shape. As the name indicates the computation of these points starts by defining points that are close to the ends of a shape, and they are used to define vertices for triangles used in the matching stage. Two different algorithms can be used for this purpose. The first algorithm depends only on the conformation that was used to create the shape. The second algorithm depends on the grid encoding of the shape. Though both algorithms yield comparable results, the first algorithm is mush faster than the second one. As a result the first algorithm is used most often except when dealing with some special cases where it fails to apply. One such case is when a target shape is created from a combination of two or more conformations.

The pseudocode from the *first algorithm* is given in Figure 6. For each heavy atom (i.e. non-hydrogen atoms) *i* in the conformation, the number of its neighbors n_i within a distance

```
for each atom i :
      N_i = set of atoms within r_w of atom i (i.e. neighbors of i)
      n_i = number of atoms in N_i (i.e. number of neighbors)
      w_i = 1/n_i
end for loop

for each atom i :
      if (n_i < 7) :
            P = Σ_{j in N_i} w_j • P_j
            add P to list of potential terminal points
      end if
end for loop

cluster the potential terminal points (see Figure 8)
```

Figure 6. Pseudocode for computing initial terminal point from a conformer. r_w is a user specified threshold (typically 4.0 Å). $\vec{P}_j$ is position vector of the jth atom.

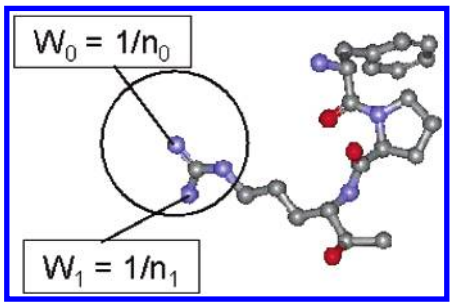

Figure 7. Illustration of the procedure followed to compute initial terminal points based on the conformer. Different weights are assigned to the atoms in the conformer based on the number of neighbors. Then at each atom center with less than a threshold number of neighbors, a weighted centroid is computed from the atomic centers of it neighbors.

r_w of itself is calculated. Where r_w, referred to as the window radius, is a user-specified parameter typically set at 4.0 Å. A weight $w_i = 1/n_i$ is assigned to each atom (Figure 7). For each atom that has less than n_t (a user specified parameter usually set at 7) neighbors, the following weight centroid is computed

$$\vec{P}_k = \frac{\sum_{neighbors} w_i.\vec{P}_i}{n_n} \qquad (3)$$

where $\vec{P}_i$ is the position vector of the ith heavy atom, the neighbors are defined as the atoms that fall within a distance r_w of the current atom, and n_n is the number of neighbors. The points are referred to as the *potential terminal points*. Some of the points $\vec{P}_k$ are very close to each other and are clustered. The pseudocode of the clustering algorithm is shown as Figure 8. For all points that are within $0.75r_w$ (typically 3.0 Å) of each other, the centroid is computed by averaging their coordinates. Figure 9 shows the configuration of the points after the clustering has been performed. These points will be referred to as the *initial terminal points*. A small ligand typically has 2−4 such points.

The aim of the *second algorithm* is same the as the first one; i.e. to compute the initial terminal points. However in this case the grid points used to encode the shape and the shape occupancy values at these points are used instead of the atomic coordinates. Figure 10 gives the pseudocode for

```
T = set of potential terminal points
P = the first point in T
while T is not empty :
      S_Ω = set of points in T within r_t of P
      P_f = centroid of points in S_Ω
      remove S_Ω from T
      add P_f as an initial terminal point
end while loop
```

Figure 8. Pseudocode for clustering potential terminal points obtained from the algorithm shown in Figures 6 and 7. r_t is usually set to be $0.75r_w$. r_w (typically 4.0 Å) being the size of the sphere used while generating the initial terminal points.

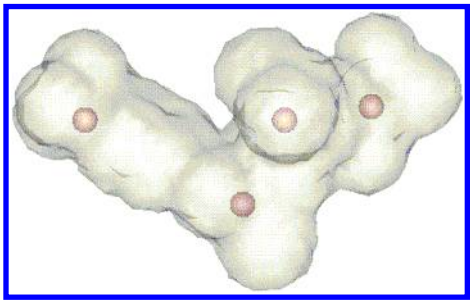

Figure 9. Example of a shape with the initial terminal points. In this case the procedure yields 4 terminal points after clustering.

```
for each grid point i :
      o_i = grid occupancy value at i
      if (o_i = 3) :
            Ω_i = sphere of radius r_w centered at i
            S_Ω = part of the shape inside Ω_i
            v_f = volume of S_Ω / volume of Ω_i
            if (v_f < v_t) :
                  C_i = Centroid of S_Ω
                  Add C_i to list of potential points
            end if
      end if
end for loop
cluster the potential terminal points (see Figure 8)
```

Figure 10. Pseudocode for initial terminal points from shape encoding.

this algorithm. At each grid point i that has grid occupancy of 3 (i.e. all points that are not close to the surface of the shape), a sphere S of radius r_w is considered. The volume fraction v_i is then computed by taking the ratio of the total grid occupancy to the maximum possible grid occupancy in the sphere as shown below

$$v_i = \frac{\sum o_k}{3.n_s} \qquad (4)$$

where o_k is the grid occupancy value at the kth grid point and n_s is the number of grid points in the sphere. The summation in eq 4 is taken only over the grid points inside the sphere S. If $v_i < v_t$, where v_t is a user specified threshold on the volume fraction (typically 0.25), i is considered to be close to the end of the shape. The centroid of the shape inside

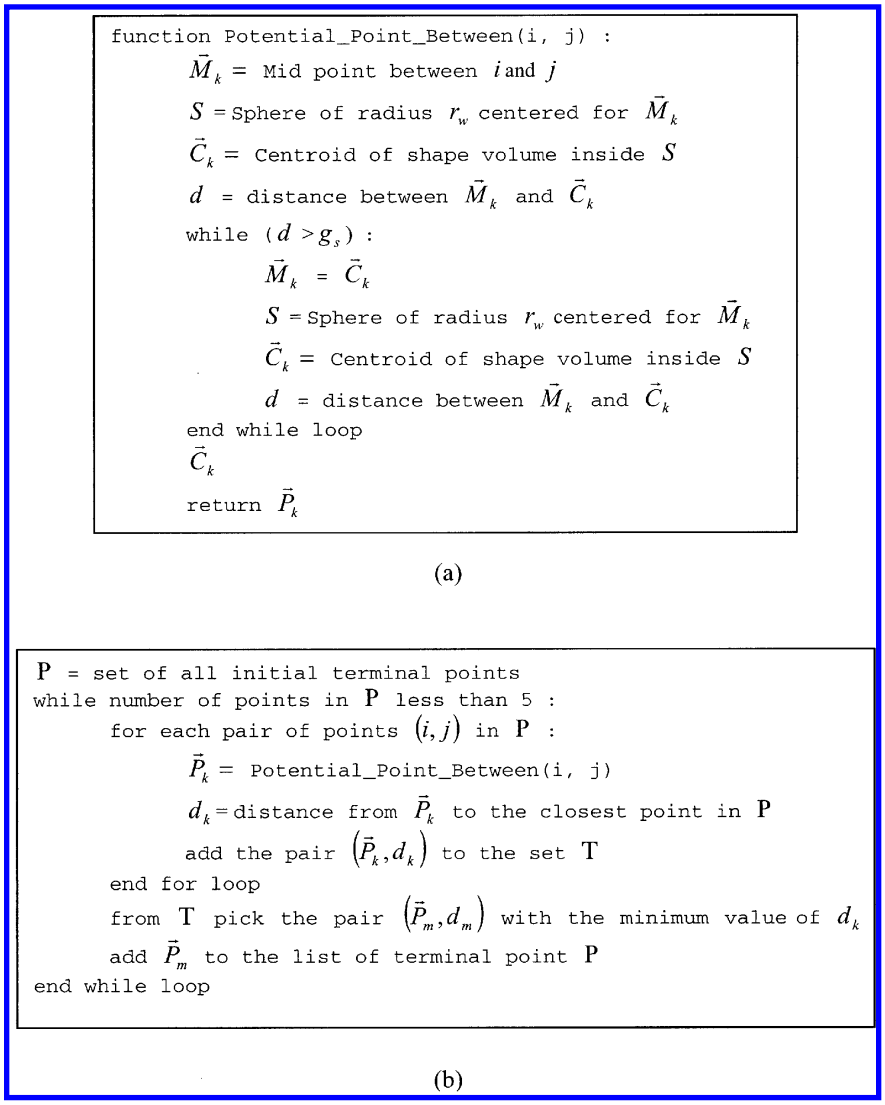

Figure 11. Pseudocode for adding terminal points to the initial points (a) defines a function to add a potential point between two points and (b) uses this function to generate additional terminal points.

S is then computed as shown below

$$\vec{C}_i = \frac{\sum o_k \cdot \vec{p}_k}{\sum o_k} \tag{5}$$

where $\vec{p}_k$ is the position vector of the *k*th grid point. Once again the summation is taken only over the grid points inside *S*. The resulting points are once again referred to as *potential terminal points.* The same clustering method used with the first algorithm (Figure 8) is used to reduce the number of points. A typical ligand shape is left with 2−4 points, and they are considered to be the initial terminal points.

In a case where the query shape is larger than the target shape, experience has shown that about 5 terminal points are needed for the subsequent matching procedure to work correctly. As a result the number of initial terminal points is often not sufficient and additional terminal points are added. The pseudocode for this process is given in Figure 11. For each pair of terminal points ($\vec{T}_i$,$\vec{T}_j$) that already exist, the midpoint $\vec{M}_k$ is computed. A sphere *S* of radius r_w is considered around $\vec{M}_k$. The centroid $\vec{C}_k$ of the shape inside of *S* is computed using eq 5. If the distance *d* between $\vec{M}_k$ and $\vec{C}_k$ is greater than grid spacing g_s (typically 0.5 Å), $\vec{M}_k$ is made to coincide with $\vec{C}_k$ and the procedure is repeated until *d* is less than g_s. At this stage $\vec{C}_k$ is added to a list of potential terminal points. For each potential terminal point $\vec{Q}_k$ its distance d_c to the closest terminal point is computed. Now the potential terminal point with the largest d_c is chosen and added as a new terminal point. This procedure is repeated until a desired number (typically 5) of terminal points has been found for the shape. Figure 12 shows the final configuration of terminal points in a typical shape.

Skeleton Points. As mentioned earlier, the skeleton points are computed only for the target shape. Skeleton points are computed to represent parts of the shape that are not covered by the terminal points. There are typically 25−100 points in a ligand shape and they are 1−2 Å apart. As the name indicates they define the backbone of a target shape. As in the case of the terminal points they form the vertices of triangles used in the matching procedure. They cover the shape more exhaustively than the terminal points. Such exhaustive coverage is necessary on one of the shapes (preferably the bigger shape), so that all possible alignments can be determined. Since target shapes are typically fewer

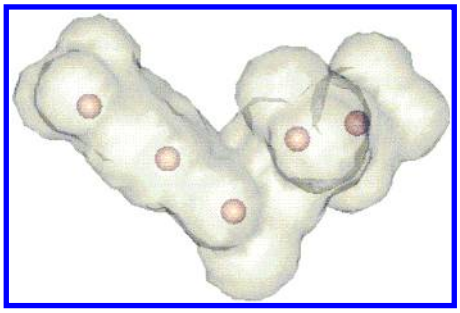

Figure 12. Example of a query shape with all the terminal points. In this case one additional terminal point was added to the initial terminal points (see Figure 9).

```
for each grid point i occupancy value equal to 3 :
      S = Sphere of radius r_w centered at i
      C_k = Centroid of shape volume inside S
      Add C_k to set of potential points S_p
end for loop

P_k = first terminal point
S = Sphere of radius FIX centered at P_k
while S_p is not empty :
      N_k = set of all points in S_p inside S
      compute volume fraction for each point in N_k
      C = point with maximum volume fraction in N_k
      Add C to the list of skeleton point
      Remove all points in N_k from S_p
end while loop
```

Figure 13. Pseudocode for computing skeleton points.

in number (usually reference shape/s), it is more efficient to determine skeleton points only for them. The pseudocode used for the computation of the skeleton points is shown in Figure 13. As in the case of the terminal points, a sphere S of radius r_w is considered around each grid point that has a shape occupancy value of 3. At each of these points the centroid $\vec{C}_i$ of the shape inside S is computed using eq 5. $\vec{C}_i$ is added to a *list of potential skeleton points*. Also at each of these points the volume fraction v_f inside S given by eq 4 is computed. This procedure leads to a large number (100−1000) of points. Several of them are close (under 0.5 Å RMSD) to each other. In the last step of the procedure some of these points are removed from the list. Starting with one of the terminals, any points in the list that are within a user specified threshold t_s (typically 1.5 Å) of it are removed. Among these removed points the point with the largest volume fraction v_f is chosen and added to the list of skeleton points. This procedure ensures that the chosen points are not too far from each other (i.e. the shape is adequately covered) and at the same time close to the central axis of the shape. The focus is now shifted to the latest skeleton point, and the procedure is repeated until the list of potential skeleton points is empty. Figure 14 shows an example of a ligand shape with the skeleton and terminal points.

Directions. Each of the terminal and skeleton points (in the case of target shape) is assigned a direction vector. The direction vector is a unit vector that gives an axis along which the shape is distributed locally at the skeleton/terminal point. Since overall similarity means reasonable similarity locally, these direction vectors can be used effectively in the filtering

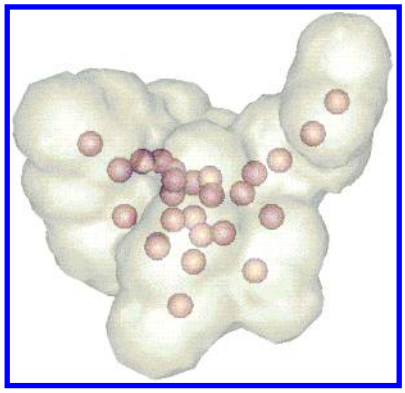

Figure 14. Example of a target shape with the skeleton points.

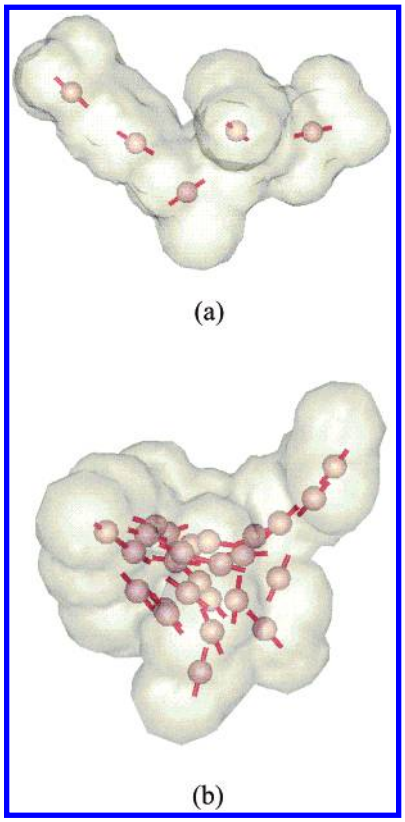

Figure 15. Example of a query (a) and a target (b) shape with the terminal and skeleton points. Also shown in the picture are the direction vectors at each of the points.

alignments as will be explained in the matching stage. Figure 15 shows typical query and target shapes with the skeleton and terminal points with the direction vectors at each point. The direction vectors are computed by considering a sphere of radius r_w at each point and computing the principal axes of the shape volume inside the sphere. The principal axes are the eigenvectors of the covariance matrix

$$M = \begin{bmatrix} \sum_{k=1}^{N_i} w_k.x_k.x_k & \sum_{k=1}^{N_i} w_k.x_k.y_k & \sum_{k=1}^{N_i} w_k.x_k.z_k \\ \sum_{k=1}^{N_i} w_k.y_k.x_k & \sum_{k=1}^{N_i} w_k.y_k.y_k & \sum_{k=1}^{N_i} w_k.y_k.z_k \\ \sum_{k=1}^{N_i} w_k.z_k.x_k & \sum_{k=1}^{N_i} w_k.z_k.y_k & \sum_{k=1}^{N_i} w_k.z_k.z_k \end{bmatrix} \quad (6)$$

where N_i is the set of all the grid points within a sphere of

Table 2. Surface Grid Encoding of a Shape[a]

range of d	grid occupancy value
$d < \left(r_w - \frac{3r_b}{s_g}\right)$	0
$\left(r_w - \frac{3r_b}{s_g}\right) < d \leq \left(r_w - \frac{2r_b}{s_g}\right)$	1
$\left(r_w - \frac{2r_b}{s_g}\right) < d \leq \left(r_w - \frac{r_b}{s_g}\right)$	2
$\left(r_w - \frac{r_b}{s_g}\right) < d \leq r_w$	3
$r_w < d \leq \left(r_w + \frac{r_b}{s_g}\right)$	2
$\left(r_w + \frac{r_b}{s_g}\right) < d \leq \left(r_w + \frac{2.r_b}{s_g}\right)$	1
$d > \left(r_w + \frac{2.r_b}{s_g}\right)$	0

[a] As in the case of the volume encoding, occupancy at each grid point is assigned based on its distance to the closest heavy atom. However, in this case, the occupancy values do not remain at 3 for all points inside the van der Waals' surface. d is the distance of the grid point to the center of closest heavy atom. r_w is the van der Waals' radius of the heavy atom. r_b is a user-specified parameter. s_g is the grid spacing.

radius r_w around the ith skeleton point. (x_k,y_k,z_k) are the coordinates of the kth grid point in the sphere. w_k is the grid occupancy value at this grid point. The direction at the skeleton point is then the first principal axis.

Surface Encoding. In certain situations it is preferable not to consider the entire volume for shape comparison. Instead, determining how well the outer surfaces of two compounds are similar is more beneficial. As mentioned by Klebe[11] "Molecules recognize each other by their surfaces ... not through their underlying bonding skeletons". To study surface matching an additional preprocessing step is performed. This step involves modifying the grid encoding of the volume done in the shape encoding step explained earlier. This surface encoding step is performed after the terminal and skeleton points as well as the directions at these points have been computed, so that these properties are based on the entire volume of the molecular shape. In this step, the core of the shape is *carved out*, and only a thin shell of volume is left behind. This shell serves as an approximation of the surface of the shape. The carving out of the shape is done by first marking all the grid points in the shape encoding that have an occupancy value of 3. At each of these points, the distance to the closest heavy (non-hydrogen) atom is computed. The grid occupancy at each these points is changed based on this distance according to Table 2. Note from the table that the grid occupancy value, unlike in the case of volume encoding, decreases for grid points very close to the center of atoms. This scheme results in grid occupancy values that are large near the van der Waals' surface of the molecule. However, the value gradually reduces to zero if one moves away from the surface either into or out-of the shape. Figure 16 shows a 2D illustration of this encoding.

Features. The last property assigned to each terminal and skeleton point is a set of chemical feature types. Chemical features are chemical substructures that are commonly associated with noncovalent binding: hydrogen bond donors and acceptors, positively and negatively charged groups,

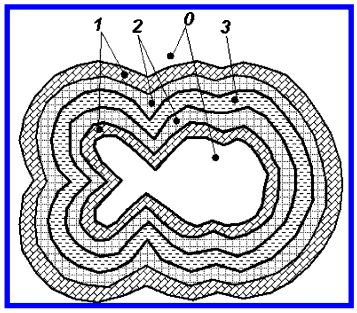

Figure 16. A two-dimensional illustration of grid encoding of the surface of a shape. All grid points that are either buried inside or are far from the surface of van der Waals' surface outside are encoded to be 0. The grid occupancy increases to 3 as one moves close to the van der Waals' surface of the shape either from the inside or outside of the shape. Table 2 gives the details of the encoding scheme.

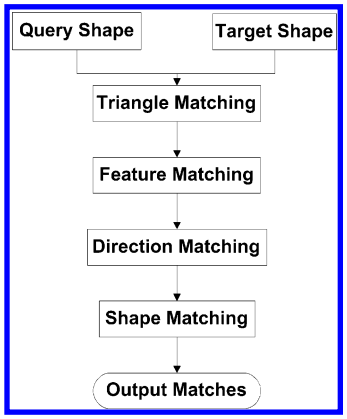

Figure 17. Flowchart showing the matching procedure. This procedure typically consists of four matching steps namely, triangle, feature, direction, and shape. The triangle matching procedure generates a list of alignments between the query and target shape that are geometrically reasonable. The remaining steps systematically remove alignments that are unlikely to display the desired shape similarity.

hydrophobic groups, and aromatic rings. These six chemical feature types are identified in each molecule through a set of queries in a SMARTS-like language.[12] The definitions, including topological definitions for hydrophobicity, are assigned using rules similar to those previously reported.[13] For each conformer that created a target or query shape, the feature locations are noted. If the distance between a terminal/skeleton point and any of the feature locations is less than a user-specified threshold f_t (1.0−2.0 Å as typical values), that feature type is assigned to the skeleton/terminal point. Multiple features of different types may be within the threshold distance from a terminal/skeleton point. In this case all these feature types are assigned to the terminal/skeleton point. Typically 0−3 feature types are found to be within the user specified threshold.

MATCHING PROCEDURE

The matching procedure consists of computing a set of starting alignments and then pruning them to obtain the final set of alignments with the desired properties (an overall flowchart of the matching procedure is shown as Figure 17). The starting alignments are computed using a triangle matching algorithm. This step is followed by a series of pruning steps that remove several of these starting alignments that do not have the desired properties. These steps include

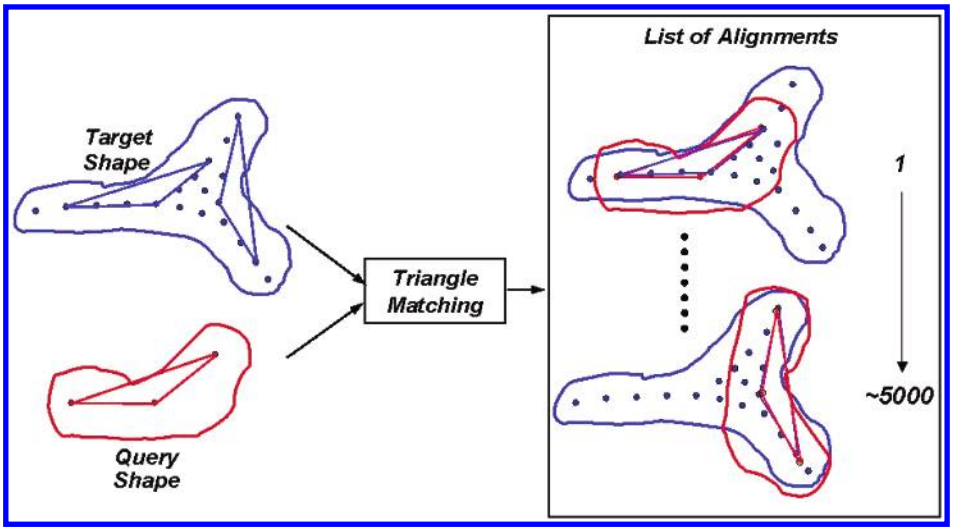

Figure 18. Illustration of the triangle matching procedure. Triangles that are similar to each other are found between the target and query shapes. The overall shape can then be aligned accordingly. Multiple alignments between two shapes may be found using this procedure.

the following: *feature, direction, and shape matching.* Each pruning step removes some of the alignments that are left from the previous step. In general the accuracy of the matching procedure increases from one step to the next, while speed of computation is slower. However, since the number of alignments that has to be dealt with also decreases with each step, the overall computational speed is reasonable. The user has control over the level of pruning at each step via several adjustable parameters. The details of each of these steps are explained below.

Triangle Matching. Triangle matching consists of finding triangles that are similar to each other between the target and query shapes. The triangles on either the target or query shape are obtained by picking all possible combinations of 3 skeleton/terminal points. The distance between the triangles is measured by the minimum root-mean-squared distance r_{min} between the triangle vertices. If this minimum root-mean-squared distance is less than a threshold r_t specified by the user, the triangles are considered to be similar to each other and the resulting alignment is stored. Typically there are about 10^5-10^6 triangle pairs between the target and query shapes. However, computing r_{min} for most of these triangle pairs can be avoided by comparing the corresponding edges between the triangles. If the difference in lengths of any pair of corresponding edges is greater than $\sqrt{6}r_t$, the pair of triangles is not considered for further analysis. For the remaining triangle pairs for which the above edge matching condition is satisfied, the optimal alignment (to minimize root-mean-squared distance) between the target triangle and the query triangle is computed. The procedure followed for this purpose is very similar to the one presented by Kabsch.[14] The query triangle is translated and rotated so that its centroid coincides with the centroid of the target triangle, and it is in the same plane as the target triangle. Finally the in-plane rotation necessary to optimize the root-mean-squared distance is computed analytically. If the resulting r_{min} is less than r_t, the transformation (i.e. translation and rotation applied to the query triangle) corresponding to this optimal alignment is stored. Since the terminal points are fixed to the query shape, this transformation provides an alignment of the query shape to the target shape (Figure 18). Typically 100−5000 triangle matches result from this step.

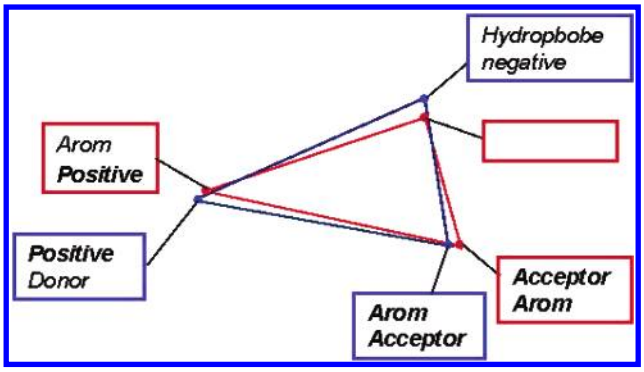

Figure 19. Illustration of the feature matching procedure. At each of the triangle vertices, the procedure checks if at least one of the features types is common.

Feature Matching. The chemical feature similarity at the triangle vertices is checked in this stage. It is designed so that alignments that are chemically unreasonable are removed. Each skeleton and terminal point in target and query shapes have chemical features assigned to them. For a pair of similar triangles (i.e. one query triangle and one target triangle), at each of the corresponding vertices, it is verified if there is at least one common feature type. However, a triangle vertex (i.e. terminal/skeleton) point with no assigned feature type is considered to match a vertex point with any feature type. Figure 19 illustrates this procedure. All alignments that do not satisfy this feature matching constraint are removed from the list of alignments. Typically 30−50% of the alignments are removed. The idea is not to optimize overall chemical similarity of the molecules at this stage. The triangles are mostly representative of only parts of the molecules, as a result being too restrictive in the feature matching stage may remove several matches that have chemical similarities over the entire molecule but are dissimilar locally.

Direction Matching. The alignments remaining after the triangle matching procedure are subjected to the direction matching procedure. This consists of checking, for each triangle pair (i.e. each alignment), the compatibility of the direction vectors at their vertices. As mentioned earlier the direction vectors at the terminal are indicative of the local volume distribution in a shape. Direction matching, therefore,

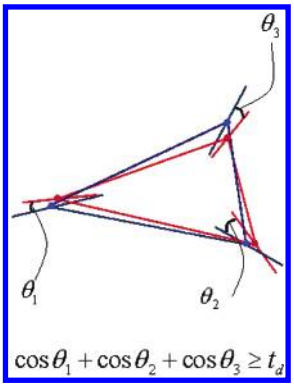

Figure 20. Illustration of direction matching. At each of the triangle vertices, the angles between the direction vectors are computed. The sum of the cosines of these angles is compared to a user specified threshold.

checks for similar volume distribution in the two shapes for the specified alignment. The criterion used to check for the compatibility of directions is given below

$$\cos(\theta_1) + \cos(\theta_2) + \cos(\theta_3) \geq d_t \qquad (7)$$

where $\theta_1,\theta_2,\theta_3$ are the angles between the direction vectors as shown in Figure 20 and d_t is a threshold specified by the user. Figure 20 shows an illustration of this procedure. Typically 20−1000 matches are left after this stage.

Shape Matching. This final step is applied on the remaining matches to perform an overall shape comparison. Since each entry in the bit vector corresponds to a specific grid point, the similarity between two shapes can be evaluated rapidly, by comparing the occupancy bit-vectors. We use the Tanimoto distance, d_t, as a measure of distance or dissimilarity between shapes

$$d_t = 100 \cdot \left(1 - \frac{v_1 \cap v_2}{v_1 \cup v_2}\right) \qquad (8)$$

where, $v_1 \cap v_2$, is the overlap (common) volume occupied by the two shapes, and $v_1 \cup v_2$ is the union volume occupied by the two shapes. Both these quantities are computed directly from the grid occupancy bit-vectors. Another shape overlap measure which we use is termed *protrude*. For this measure, the volumes of the shapes are compared to determine the smaller of the two. The measure is then the percentage of the smaller shape which extends beyond the larger shape. Either of these shape overlap measures can be used with the full shape or surface matching algorithms described previously.

While relatively fast due to our use of occupancy vectors, the shape comparison calculation is by far the slowest of our alignment filtering steps. The final result of this comparison gives a true measure of the shape alignment, which to this point was only approximated in the other filters. Those alignments that survive this final check are considered good and can be saved to a database for future visualization or scoring.

RESULTS

While the subshape matching algorithm was originally designed to handle the specific problem of finding a *fragment* shape within the shape defined by a "ligand" (e.g. benzamidine in a thrombin inhibitor), it was found to be quite robust for handling many shape matching problems. This section will examine examples for a variety of systems for which the subshape matching technology has proved applicable. The examples shown are (1) matching of a "fragment" shape within a larger "ligand" shape, (2) the alignment of similarly sized molecules, (3) docking, and (4) a search of the MDDR using several different shape matching criteria.

Table 3. Number of Matches Present after Each Matching (Filtering) Stage in the Alignment of Benzamidine to NAPAP[a]

(a)

filter	no. of matches	filter	no. of matches
triangle	1595	feature scoring	9
direction	44	RMS pruning	4
shape	20	key feature	1

(b)

	rigid body (crystal structure)	with conformations	closest conformation
NAPAP	0.53	2.26	1.9
PPACK	0.35	1.56	1.35

[a] Following the subshape matching procedure, a feature scoring process was applied to the matches. In addition matches can also be pruned by removing matches that are close to each other measured by RMSD.

Subshape Detection. We examined benzamidine and N-alpha-(2-naphthyl-sulfonyl-glycyl)-DL-P-amidinophenyl-alanyl-piperidine (NAPAP) as an example of the algorithm's ability to distinguish a fragment subshape (roughly 25−50% of the complete molecular volume) within a ligand-sized target shape. Subshape matching of this type would principally be of interest for use with shape descriptors. Interesting fragment shapes can be found from studies of known inhibitors or a generic library and constructed into a binary fingerprint indicating the presence or absence of each shape for a compound. Examples of this binary fingerprint technique using full "ligand" sized shapes in combination with a feature grid have already been presented.[6] Benzamidine and NAPAP provide an excellent example of subshape matches we would like to find. Benzamidine is a component of NAPAP as well as many other thrombin inhibitors. While the full shapes of different thrombin ligands might not match it should be possible to identify subshapes (e.g. benzamidine) which do.

For this test the NAPAP ligand was taken from the 1ETS[15] crystal structure for thrombin, and used as a target shape. For benzamidine a single conformer was generated by our internal conformational analysis software CONAN[16] and was converted into a query shape. Table 3 summarizes the number of matches found after various levels of filtering, with Figure 21 showing the final remaining alignment. The initial 1595 alignments supplied from the triangle matching are efficiently reduced to 44 using a direction filter. This reduction in the number of matches by more than an order of magnitude is achieved using the computationally most expedient filter, passing only a limited number of alignments to the subsequent more expensive filters. The shape, feature, and RMS filters are able to reduce the subshape matches under consideration to only four. If in a final step a key feature filter is applied (tightening the previous generic feature

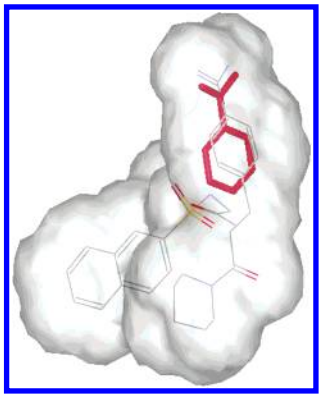

Figure 21. Illustration of the final remaining match of benzamidine fragment to NAPAP.

matching filter to require the overlap of the positive charges), we are left with only one alignment (shown in Figure 21). Visual examination of the resulting alignment shows excellent agreement with the benzamidine fragment of NAPAP.

Alignment. The alignment of compounds is a common computational task for which the subshape matching technology provides a novel approach, providing benefits over more commonly used procedures.[1−3] Typically compounds are aligned using methods which depend on the connectivity of the molecules. Some alignment engines, such as FLEXS,[17] try to create more general alignments by using the projections of binding features to create alignments based upon potential binding motifs. In comparison the subshape matching algorithm should allow for molecules of varying size to be overlapped purely on the basis of their steric volume in a particular conformation. Additional, filtering/ranking of alignments obtained from subshape matching can easily be achieved through feature scoring or quality of overlap.

To test the subshape algorithm in this capacity a subset of the Flex-77 data set from the GMD was used.[17] This set consists of aligned crystal structures for 72 ligands from 13 proteins. The ligands vary widely in size having between 10 and 80 heavy atoms and 0−32 rotatable bonds. Two experiments were conducted using this data set. In the first a standard pairwise rigid-body alignment was run for all pairs of ligands from identical proteins. 662 ligand pairs were identified, and after subshape alignment 76% (502/662) were seen to find matches of less than 2 Å RMSD when compared to their respective crystal structures. In comparison FLEXS was able to find matches of less than 2 Å RMSD for 88% of these same cases.

While the subshape alignment slightly underperformed FLEXS in our initial test, its performance was strong enough to encourage us to investigate additional alignment problems which traditional techniques are unable to address. One such problem is alignment to a composite molecule/shape (experiment summarized in Figure 22). Our combined shape experiment consisted of first, aligning all of the structures/ligands associated with a particular protein (based on the alignment of the entire protein). One of these ligands was then set aside, while a composite shape was created for those remaining (in a "leave-one-out experiment"). The composite shape was defined as the union shape created from the aligned crystal structures. The subshape algorithm was then used to find alignments between the ligand withheld and the composite shape. In this way alignments of novel compounds

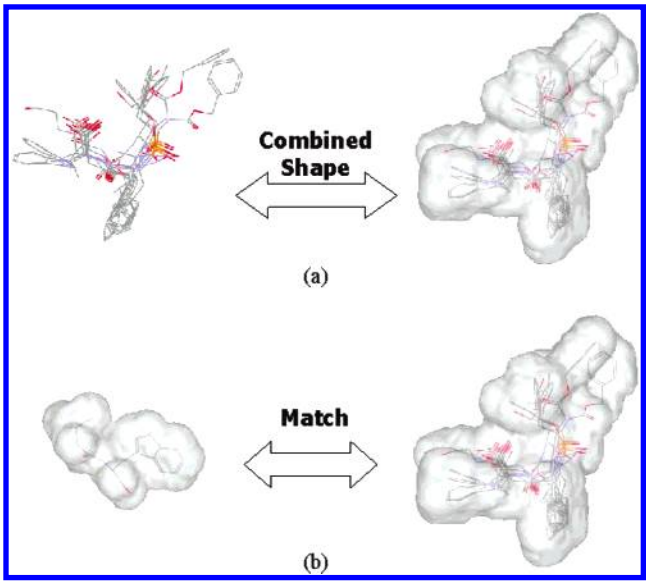

Figure 22. Illustration of the combined shape experiment with 12 thermolysin ligands with X-ray crystal structures. For each round of the experiment, one of the 12 ligands is left out, and the remaining ones are picked to create a combined shape (a). The ligand that has been left out is then used to create a query shape to match the combined shape, which is used as the target shape. The resulting matches are compared to the orientation of the ligand obtained from the crystal structure.

Table 4. Results for the Combined Shape Experiment Conducted Using the X-ray Crystal Conformations of 12 Thermolysin Ligands[a]

ligand name	no. of heavy atoms	best RMSD
1tlp	37	0.72
cbz	22	0.92
4tin	10	1.21
4tmn	36	1.08
ppp	22	0.82
5tmn	32	0.9
5tln	23	0.95
2tmn	13	0.86
thior	17	0.6
rthior	17	0.92
1tmn	35	0.85
3tmn	22	0.72

[a] In each row the first column gives the name of the ligand that has been used to match a combined shape. The combined shape is made out of the remaining 11 ligands. The second column gives the number of heavy atoms in the ligands, as an indicator of the size of the ligand. Finally the last column gives the best RMSD value obtained by the method, as compared to the crystal orientation of the ligand.

can be done to an overall representation of the available/known binding space. Query ligands can easily span the volume representation of several smaller ligands, making use of all the known binding regions, similar in spirit to the MCSS technique.[18−21] An alignment of this type would be impossible using traditional techniques.

Overall 72 "leave-one-out" experiments of this type were done. 83% (60 of 72) were able to find alignments with less than 2 Å RMS to the original crystal alignment after all stages of filtering. As an example Table 4 summarizes the results of these combined shape experiments for thermolysin. The crystallographically determined structures for 12 thermolysin ligands were considered. The ligands ranged widely in sizes having between 10 and 37 heavy atoms. The data in Table 4 shows that the alignments in this case correlated well with

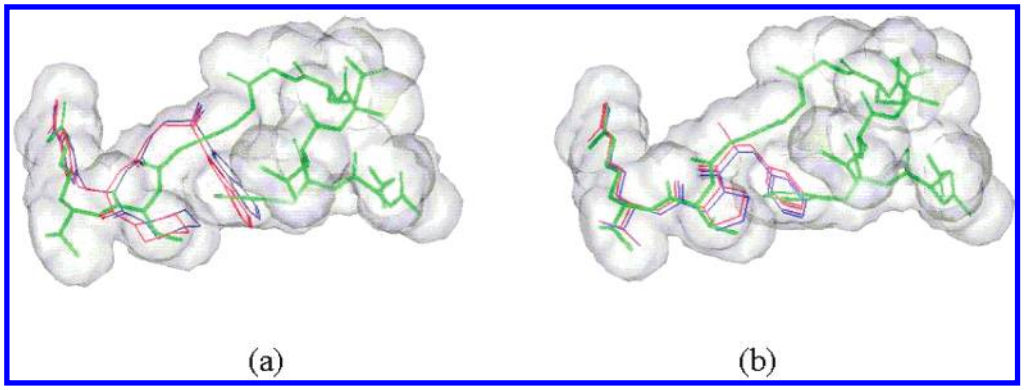

Figure 23. Results of NAPAP (a) and PPACK (b) rigid docking into fibrinopeptide-A. Shown are the fibrinopeptide-A structure (green), the NAPAP/PPACK (red) structure determined by alignment of the thrombin structures, and the subshape alignment (blue).

10 ligands having an RMSD of < 1 Å to their corresponding crystal structure.

Docking. This experiment was designed to test the alignment of whole molecule shapes into large shapes, roughly paralleling the docking of molecules into protein active sites. Three thrombin ligands were selected from the Protein Data Bank (PDB): fibrinopeptide-A, NAPAP, and PPACK (extracted from 1DWE[15] crystal structure for thrombin). The crystal structure of fibrinopeptide-A was made into a target shape. NAPAP and PPACK were then made into query shapes and aligned to fibrinopeptide-A using our method. The protein structures were aligned to provide a reference alignment. These alignments of NAPAP and PPACK with fibrinopeptide-A were used as reference for judging the alignments from our method. Two types of experiments were conducted using these data. In the first, geometries of all ligands (fibrinopeptide-A, NAPAP, and PPACK) were held rigid in the crystal conformations. In the second experiment, conformational analysis was performed on NAPAP and PPACK while leaving the fibrinopeptide-A in its crystal conformation.

The results for the first experiment are shown in Figure 23. The best alignment of the NAPAP shape within fibrinopeptide-A yields a 0.53 Å RMS compared to the crystal alignment. For PPACK, the RMS is 0.35 Å. Initially there are 6438 matches between NAPAP and fibrinopeptide-A. With direction matching, this is reduced to 649 and further reduced to 48 matches when the shape/shape comparison filter is applied. The PPACK results are comparable: 8556 initial matches, 957 matches with direction filtering, and 108 final matches with shape/shape comparison. In both cases the alignment with the lowest RMSD is carried through all of the filtering steps. In addition, other filtering steps such as scoring the matches based on chemical features and removing matches that are very close to each other in terms of RMSD (over heavy atoms) following the subshape matching procedure still leave the desired matches. This scoring process uses a Gaussian function for each chemical feature recognized on the Fibrinogen molecule. A score is then assigned to the each match by summing contributions from these Gaussian functions for all features on the query molecule (NAPAP). Table 3(a) shows these results.

The second experiment was designed to assess the performance of this method in the absence of a crystal structure, and the query shapes were prepared from an ensemble of conformations instead of the crystal structure. For both NAPAP and PPACK three-dimensional conformations were generated using an internal conformational analysis tool CONAN[16] resulting in the order of 100 conformations for each ligand. As shown in Table 3(b) the conformation that is closest to the crystal structure is 1.9 Å for NAPAP and 1.35 Å PPACK. In other words this is the best possible RMSD that can be achieved for this collection of conformations. It is clear from the table that the subshape matching procedure performs quite well on these conformations in duplicating the crystal orientation.

Table 5. Results Are Given for 12 Searches of ~83 000 Compounds from the MDDR, Using Both Volume and Surface Encoding of the Shapes as Well as the Tanimoto and Protrude Distance Measures[a]

overlap type	distance measure	shape threshold	matches	T-act matches	enrichment
overlap	Tanimoto	40	48	16	44.1
overlap	Tanimoto	45	1493	124	11.0
overlap	Tanimoto	50	17213	410	3.1
overlap	protrude	5	10	0	0.0
overlap	protrude	10	2236	2	0.1
overlap	protrude	15	11962	17	0.2
surface	Tanimoto	55	78	30	50.9
surface	Tanimoto	60	3159	188	7.9
surface	Tanimoto	65	34162	518	2.0
surface	protrude	25	464	2	0.6
surface	protrude	30	6235	13	0.3
surface	protrude	35	62371	510	1.1

[a] The shape matching threshold is given along with the number of matches found. Additionally, the number of thrombin actives (T-act matches), along with the method's enrichment in identifying thrombin actives, over random selection are given.

MDDR Search. A search was run on a set of 83 178 small molecules from the MDDR[22] using the subshape matching algorithm and the NAPAP ligand as taken from the 1ETS crystal structure as a target. A search of this type allows one to examine (1) the versatility of the code for handling a wide variety of molecules, (2) approximate timings, and (3) behavior of different shape comparison measures. Table 5 summarizes the results.

For each shape matching algorithm (volume or surface) matches were calculated using three appropriate cutoffs for both the Tanimoto and Protrude distance measures. Specifically, these runs were done using a 19.5 × 15 × 12.5 Å box, with triangle and direction thresholds of 0.5 and 2.6, respectively. Additionally a loose feature matching restriction was used requiring 3 matched features at a distance of 1.25 Å. For each search criteria Table 5 lists the total number of compounds which pass all of the subshape matching filters

along with the number of these compounds which are annotated as "thrombin inhibitors" in the MDDR. The final number presented in Table 5 for each search is the enrichment for activity in the compounds selected for each experiment. The enrichment is calculated as the density of thrombin actives in the selected set over the density in the original pool of compounds $(N_{ap}/N_{aTotal})/(N_{ip}/N_{iTotal})$.

In the 83 178 small molecules of the MDDR, which were studied, 624 have the annotation "thrombin inhibitor". Additionally there are several thousands of related molecules which are annotated as "antiaggretory" and "GB2A/3B" inhibitors. They were not considered for this analysis. Although, many of this second class of compounds undoubtedly have some thrombin activity, there is no way of finding out which ones to validate our results. Not considering these compounds should lower the enrichments reported, underestimating the true therapeutic performance of the technique as compared to what might be obtained for a more homologously assayed data set.

Two general trends emerge from the data (see Table 5). First, the results are highly dependent on the threshold used for the distance measures. Changes of 5 units in the distance measure result in changes of 1 or 2 orders of magnitude in the number of matches. A second trend observed is that the "protrude" distance measure does not work well for this type of searching. In contrast, the Tanimoto measure is able to differentiate thrombin actives from inactives with significant enrichments. This is true for either the overlap or surface algorithms. Particularly, when a very tight distance cutoff is used, enrichments of 44.1 for the overlap algorithm and 50.9 for the surface matching algorithm are seen. The enrichments fall off as the cutoff is raised and more of the pool is picked, but a reasonable enrichment of 11.0 is seen when ~2% of the data set is chosen using the overlap algorithm and the Tanimoto measure.

The calculations were run on a LINUX cluster of 20 1.7 GHz Athlon and 20 1.3 GHz Pentium4 processors. Comparative timings of the jobs on the two processor types resulted in only negligible differences in overall speed. The jobs were run in a trivially parallel fashion, with each processor receiving an equal number of compounds for analysis at the initiation of the run. This process did result in some underutilization of the resource as variability in the number of conformers, stereoisomers, compound size, and number of matches resulted in some processors finishing significantly before others (~15 min for runs that average 6 h). A dynamic load balancing application, which would send individual compounds to processors, as they became free, could easily correct this inefficiency. However, as all the jobs completed overnight we deemed additional load balancing to be unnecessary at this stage.

CONCLUSIONS

We have presented an efficient algorithm to compare molecular shapes of different sizes. The method involves encoding the three-dimensional shapes of the conformations of a molecule onto a grid. Computing representative (terminal and skeleton) points in these shapes then captures the bare skeleton of the shape. These points are then used to efficiently compute and filter possible alignments between the shapes. It is important to note that the early stages of this multistep process (triangle, direction matching) are computationally inexpensive and as a result can deal with a large number of possible alignments. The shape matching process is relatively slower and can therefore be applied only to a small number of alignments. As a result this process efficiently deals with sampling the translational and rotational space extensively.

All of the properties (terminal and skeleton points, directions, etc.) are computed based on the three-dimensional shape, making this alignment procedure reasonably independent of the two-dimensional structure of the compounds. Further, these properties are indicative of the local steric environment, i.e. they capture the subshapes of a molecule, therefore allowing other molecules to be aligned to them. One of the advantages of this method is that the resulting matches include all the possible sterically compatible alignments between the shapes.

The method is applicable to a large number of computational drug design problems as illustrated by the various examples in the results section. The examples include placing a small fragment into the shape of a ligand molecule, comparing a ligand sized molecule to a much larger molecule, and comparing similar sized molecules.

ACKNOWLEDGMENT

We would like to thank the members of the computational sciences team at DuPont Pharmaceutical Reasearch Labs and later Deltagen Research Labs for their help in testing and developing this method. In particular, we would like to note the contributions of Jeff Blaney, Leslie Robinson, and Peter Grootenhuis.

REFERENCES AND NOTES

(1) Klebe, G. *Structural Alignment of Molecules. 3D-QSAR and Drug Design*; Kubinyi, H., Ed.; ESCOM: Leiden, pp 173−199.
(2) Bures, M. G. Recent Techniques and Applications in Pharmacophore Mapping. *Practical Application of Computer-aided Drug Design*; Marcel Dekker: New York, U.S.A., 1997; pp 39−72.
(3) Lemmen, C.; Lengauer, T. Computational Methods for the Structural Alignment of Molecules. *J. Comput.-Aided Mol. Des.* **2000**, *14*, 215−232.
(4) Petitjean, A. M. Geometric Molecular Similarity from Volume-Based Distance Minimization: Application to Saxitoxin and Tetrodotoxin. *J. Comput. Chem.* **1995**, *16*, 80−90.
(5) Grant, J. A.; Pickup, B. T. A Gaussian Description of Molecular Shape. *J. Phys. Chem.* **1999**, *99*, 3503−3510.
(6) Putta, S.; Lemmen, C.; Beroza, P.; Greene, J. A Novel Shape-Feature Based Approach to Virtual Library Screening. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 1230−1240.
(7) Hahn, M. Three-Dimensional Shape-Based Searching of Conformationally Flexible Compounds. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 80−86.
(8) Goldman, B. B.; Wipke, W. T. Quadratic Shape Descriptors. 1. Rapid Superposition of Dissimilar Molecules Using Geometrically Invariant Surface Descriptors. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 644−658.
(9) Robinson, D. D.; Lyne, P. D.; Richards, W. G. Partial Molecular Alignment via Local Structure Analysis. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 503−512.
(10) Foley, J. D.; Van Dam, A.; Feiner, S. K. *Introduction to Computer Graphics*; Addison-Wesley Pub. Co.: 1993.
(11) Klebe, G. Structural alignment of molecules. *3D QSAR in Drug Design*; ESCOM Science Publishers: Leiden, 1993; pp 173−199.
(12) *SMARTS Toolkit: Daylight Chemical Information Systems*: Santa Fe, NM.
(13) Greene, J.; Kahn, S.; Savoj, H.; Sprague, P.; Teig, S. Chemical Functional Queries for 3D Database Search. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 1297−1308.
(14) Kabsch, W. A Solution for the Best Rotation to Relate Two Sets of Vectors. *Acta Crystallogr.* **1976**, *A32*, 922−923.

(15) Berman, H. M.; Westbrook, J. Z.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235−242.
(16) Smellie, A.; Stanton, R. V.; Henne, R. M.; Teig, S. L. Confomational Analysis by Intersection. *J. Comput. Chem.*, accepted for publication.
(17) Lemmen, C.; Lengauer, T.; Klebe, G. FlexS: A Method for Fast Flexible Ligand Superposition. *J. Med. Chem.* **1998**, *41*, 4502−4520.
(18) Evensen, E.; Joseph-McCarthy, D.; Karplus, M. MCSS version 2.1; Harvard University, Cambridge, 1997.
(19) Miranker, A.; Karplus, M. Functionality maps of binding sites: a multiple copy simultaneous search method. *Proteins* **1991**, *11*, 29−34.
(20) Joseph-McCarthy, D.; Thomas III, B. E.; Alvarez, J. C. Pharmacophore based molecular docking. In Proceedings of the 221st ACS National Meeting on Book of Abstracts, San Diego, 2001, CINF-041.
(21) Thomas, B. E.; Joseph-McCarthy, D.; Alvarez, J. C. Pharmacophore based molecular docking. Pharmacophore perception, development and use in drug design (Iul Biotechnology series, 2). In *International University Line*; Guner, O. F., Ed.; La Jolla, CA, 2000; pp 353−367.
(22) *MDDR*; 2000.2 ed.; MDL Information Systems, Inc.: San Leandro, CA.

CI0256384