Flexible Structural Comparison Allowing Hinge-Bending, Swiveling Motions

George Verbitsky,[1] **Ruth Nussinov,**[2,3] **and Haim Wolfson**[1]
[1]*Computer Science Department, School of Mathematical Sciences, Tel Aviv University, Israel*
[2]*Sackler Institute of Molecular Medicine, Sackler School of Medicine, Tel Aviv University, Israel*
[3]*Intramural Research Support Program-SAIC, Laboratory of Experimental and Computational Biology, National Cancer Institute-Frederick Cancer Reasearch Facility, Frederick, Maryland*

ABSTRACT **We present an efficient method for flexible comparison of protein structures, allowing swiveling motions. In all currently available methodologies developed and applied to the comparisons of protein structures, the molecules are considered to be rigid objects. The method described here extends and generalizes current approaches to searches for structural similarity between molecules by viewing proteins as objects consisting of rigid parts connected by rotary joints. During the matching, the rigid subparts are allowed to be rotated with respect to each other around swiveling points in one of the molecules. This technique straightforwardly detects structural motifs having hinge(s) between their domains. Whereas other existing methods detect hinge-bent motifs by initially finding the matching rigid parts and subsequently merging these together, our method automatically detects recurring substructures, allowing full 3 dimensional rotations about their swiveling points. Yet the method is extremely fast, avoiding the time-consuming full conformational space search. Comparison of two protein structures, without a predefinition of the motif, takes only seconds to one minute on a workstation per hinge. Hence, the molecule can be scanned for many potential hinge sites, allowing practically all C_α atoms to be tried as swiveling points. This algorithm provides a highly efficient, fully automated tool. Its complexity is only $O(n^2)$, where n is the number of C_α atoms in the compared molecules. As in our previous methodologies, the matching is independent of the order of the amino acids in the polypeptide chain. Here we illustrate the performance of this highly powerful tool on a large number of proteins exhibiting hinge-bending domain movements. Despite the motions, known hinge-bent domains/motifs which have been assembled and classified, are correctly identified. Additional matches are detected as well. This approach has been motivated by a technique for model based recognition of articulated objects originating in computer vision and robotics. Proteins 1999;34:232–254.** **Published 1999 Wiley-Liss, Inc.**[†]

Key words: domain motions; hinge–bending motions; flexible protein structure comparison; computer vision; detection of conformational changes

INTRODUCTION

There are numerous approaches to finding recurring substructural motifs in protein structures, where the substructure recurs as a rigid body.[1–16] Yet there are few methods which can detect motifs, allowing hinge(s) within them. The scarcity in such automated methods is owing to the more difficult task which is involved. Automatically finding a recurring substructure between two (whether globally similar or dissimilar) protein structures, without a predefinition of the motif, and doing so at high speed is a very difficult problem. Here we present an algorithm designed to accomplish such a goal. It is very fast: comparison of two protein structures takes only seconds on a Silicon Graphics Indigo 2 R4400, 150 MHZ workstation. As in our previous structural comparison algorithms, it carries out the comparisons in a manner which is independent of the order of the amino acids on the polypeptide chains, hence allowing a change in the directionality of the chains, as well as insertions and deletions. In the implementation illustrated here, only one hinge is allowed. However, the approach is general, and several hinges can be straightforwardly implemented. Currently, hinges are allowed in only one of the molecules. The speed of the comparison enables scanning the molecule for many potential hinge sites, with, for example, every C_α atom serving as a hypothetical trial swiveling point.

A structural comparison technique which allows swiveling motions of domains (or, subdomains), is a very useful tool. Proteins are flexible entities. Domain and subdomain motions have been repeatedly observed.[17,18] A comprehensive survey of domain movements in proteins has been presented by Gerstein et al.[19] The authors have analyzed the ability of different segments of a protein to move with respect to each other with small changes in energy. Their conclusion was that there are two predominant types of motions—*hinge* motions and *shear* motions. Among the frequently noted cases are “open” versus “closed” forms of

Grant sponsor: National Cancer Institute; Grant number: N01-CO 56000; Grant sponsor BSF; Grant number: 95-00208.

*Correspondence to: R. Nussinov, NCI-FCRDC Building 469, Room 151, Frederick, MD 21702. E-mail: ruthn@ncifcrf.gov

Received 22 June 1998; Accepted 28 September 1998

enzymes. Domain motions are essential components in allosteric enzymes, in protein assembly, and in cell motility. Moreover, domain motions have been observed in domain (or part) "swapping."[20] Swapping of structural parts during protein folding can bring about misfolding and aggregation. Detection of hinge-bent motifs between otherwise similar—or dis-similar—protein structures can indicate which residues may play a critical role in substructural swiveling. Hence, a technique that carries out such structural comparisons rapidly is very powerful. The only input are the atomic coordinates and potential hinge location(s) in one of the structures being compared. In a rigid structural comparison between a pair of molecules, one of the molecules is allowed all translational and rotational degrees of freedom in 3-D space. However, in our case here we allow this molecule, in addition to the previous motion, to have all the internal rotational degrees of freedom at the hinge point as well. This is done in an efficient manner avoiding a full conformational space search.

From the computational point of view, methods searching for molecular similarity are closely related to 3-D object recognition techniques developed in the field of Computer Vision.[21] The recognition of three-dimensional objects in cluttered scenes is a classical problem in computer vision. Currently the only successful recognition techniques are model-based. One is given a set of model objects in advance, with the goal of detecting those models which can fit in a geometrically consistent way into the overall shape of the scene. In the most challenging scenario the objects may be partially occluded and the scene may contain additional clutter which cannot be fitted to the models in the database. The best one can hope for is partial matching of the model objects. From a geometric viewpoint the problem of searching for structural molecular similarity is identical to this challenging partially-occluded object recognition goal. The *target molecule* is the visually cluttered scene. We search for a partial fit between the structure of a *target molecule,* with candidate *model molecules.* The occlusion reflects mismatches, insertions and deletions. In a manner analogous to a robot moving its limbs or rotating and tilting its head, we allow molecular parts such as domains, subdomains, and loops, to rotate around preselected point-hinges. By allowing them complete 3D rotations around a point, rather than the simpler case of rotation around a covalent bond, we implicitly take into account several consecutive, or nearby rotatable bonds.

The method presented here was initially conceived for articulated object recognition.[22] An articulated object is defined there as an object consisting of rigid subparts which are connected by rotational or sliding joints. It was developed as an extension of the ideas of the Generalized Hough Transform[23] and the Geometric Hashing[24] paradigms for rigid object recognition. The problem it solves is formulated as follows: Given a target molecule represented by a set of geometric features (e.g. the centers of its C_α atoms) and a database of molecules, each of which can possess hinges, find those molecules that under an appropriate translation and rotation of the whole molecule in addition to appropriate rotations at the hinges, will have a large enough set of its geometric features superimposed with those of the target molecule.

This problem arises frequently in searches for hinge-based motifs in protein structures and in computer-assisted drug design, where one is looking for molecules which have structural similarity with a given lead compound. In the description of the algorithm below, the database consists of a single molecule, and the question reduces to the decision of whether this molecule and the target molecule have a large common substructure, allowing rotations at hinge points.

The major steps of our method can be outlined briefly as follows:

1. Each of the model molecules is represented by means of its transformation independent features (invariants). Recall that the transformations consist of translations and rotations of different parts of the model object. This transformation invariant molecular information, along with any additional details we wish to retain, is stored in a hash table.
2. The target molecule is represented by its transformation independent features as well, which are compared to the information stored in the hash table. Based on this comparison several potential matches are generated. The use of a hash table for storing transformation-independent features of model objects makes the comparison extremely efficient.
3. The potential matches are subsequently verified by transforming the model molecules onto the target and evaluating the extent of the match.

Almost all existing molecular structure comparison methods consider (and compare) the molecules as rigid objects. The method presented here generalizes and extends approaches to molecular structure similarity by considering molecules as articulated objects with predefined *joints* (or *hinges*). To illustrate the logic of our algorithm, let us consider the molecule as an ordered set of its C_α atoms. We may select a specific C_α atom and divide the molecule into two parts. Now, consider the molecule as an object comprised of two rigid parts connected by the selected C_α atom. This C_α atom can be viewed as a *joint connection* or *hinge* between the two parts. Hence, these parts may swivel with respect to each other around that C_α atom. There is no limitation in the selection of the hinge site. The hinge may be positioned anywhere in 3D space and need not be specifically one of the C_α atoms of the molecule. It may be inside a loop, enabling two secondary structure elements to swivel with respect to each other; or it may be anywhere within the interior of the molecule. Alternatively, it may be positioned outside the molecule, for example, at the intersection of vectors describing secondary structure elements, extending the repertoire of allowed motions. Comparing molecules as articulated objects, enables revealing structural similarities which cannot be discovered by methods which consider molecules as strictly rigid objects. Clearly, the rigid case is a particular case, where there is no rotation at the hinge. Consequently, if the optimal match is such that the molecules are superimposed as if they were rigid, with no internal motion involved, it would be detected as well.

The ability of our technique to detect similarities between molecules when these are considered articulated objects illustrates its inherent ability to automatically handle domain movements. Gerstein et al.[19] have indicated two potential ways a protein can accommodate large domain movements while still maintaining its packing. *Shear motions* provide one such possible structural mechanism. These motions involve small sliding movements between closely packed segments of the polypeptide chain, such as helices. The culmination of shear movements both within and between domains may result in large overall movements. A *hinge mechanism* provides an alternative way for protein domains to move while still maintaining their packing. Domains may move as rigid bodies with their deformations confined to their linking hinge regions. Here, we implement our structural comparison method for hinge-bending motions. We apply it to a large number of cases, picked from the protein motions database maintained by Gerstein. In all cases, the movements have been reconstructed satisfactorily. In additiion, we have applied it to a collection of structures taken from the structurally non-redundant dataset generated from the PDB.[25]

Our method can be straightforwardly generalized enabling a molecule to be divided into several parts. In such a case, hinges would be defined for every pair of neighboring subparts (subdomains) of the molecule. In particular, we note that despite the fact that the method provides broader capabilities for discovering structural similarities between molecules, its run-time complexity is only of the order of n^2 operations, although with a big constant factor, where n is the number of C_α atoms in the compared molecules. We have previously applied a similar method for the docking of flexible molecules (Sandak et al.[26,27]). These attributes allow extensive database comparisons. The procedure is automated to scan potential hinge sites in all, or a large portion of the C_α atoms in the model molecule, iteratively picking model molecules from a list.

Recently, Wriggers and Schulten[28] have published a method for the automatic detection of hinges in protein molecules.

DETECTION OF HINGE-BENT MOTIFS: THE ROBOTICS-BASED ALGORITHM

Our technique was inspired by the Generalized Hough Transform method[22,23] which was originally developed for rigid-body matching. Here we describe its rationale in very general terms. We show how it has been elegantly extended to a general algorithm which can detect motifs in proteins, where these are allowed to have swiveling points between their (rigid) parts.[22] Here we consider proteins with only one swiveling point. However, the algorithm can be straightforwardly extended to deal with multiple such points.

Let us consider structural comparison of a pair of proteins, which are modeled as rigid shapes. Let us nickname the first protein the model and the second protein the target. Assume that these proteins are represented by the sets of their C_α atoms. The goal is to detect their best superimposition, namely, the translation and rotation of the model protein which superimposes a maximal number of its C_α atoms on the C_α atoms of the target protein. Since the proteins are not identical, not all their C_α's will be superimposed and we have no a-priori knowledge of the matching set.

The rotation and translation of a rigid shape can be conveniently described as the rotation and translation of a 3-D Cartesian reference frame associated with this shape (see Figure 1a). The initial location and orientation of this reference frame can be arbitrary, as long as it is rigidly positioned relative to the shape. After defining a reference frame on the model protein, the structural comparison is equivalent to the determination of the position of this reference frame with respect to the target protein so that the proteins are maximally superimposed. Since the shapes are not identical we cannot compare them using some global shape characteristics such as the diameter of the shape† and local information should be applied. We consider local shape features, which have characteristics invariant to rotation and translation (e.g. lengths of triangle sides). Similarity of the shape characteristics of two features, one in the model protein and the second in the target protein implies that these features should be superimposed. Such an alignment of the features generates a new position for the model reference frame. Thus, similarity of local feature characteristics can be translated into potential model reference frame locations. One can rank these locations by the number of local feature alignments that contributed to them. Eventually, the high scoring frame locations are translated to favorable candidate superimpositions. The idea presented here can be implemented in an efficient way which allows comparison of a target protein against a database of model proteins.

To illustrate the situation when hinges are introduced, let us consider the model shape in Figure 1a. Assume that we have two target shapes. The first is the original model which has undergone only a translational and rotational transformation (Fig. 1b). The second is the same model. However, it has undergone not only translation and rotation, but additionally, one of its parts has been rotated with respect to the other around a hinge point (Fig. 1c). Let us apply the above mentioned comparison technique to each of the target shapes. In the first case, since the model is identical to the target up to a rigid transformation, we obtain one candidate reference frame whose score of contributed "votes" strongly outnumbers the score of other, random candidate frames. However, in the second case we obtain two high-scoring candidate reference frames, matching either the first or the second of the rigid parts of the model shape. Assuming that these two parts are roughly the same size and have the same number of relevant shape

*http://bioinfo.mbb.yale.edu/MolMovDB

†A diameter of a shape is defined as a vector connecting its most distant points.

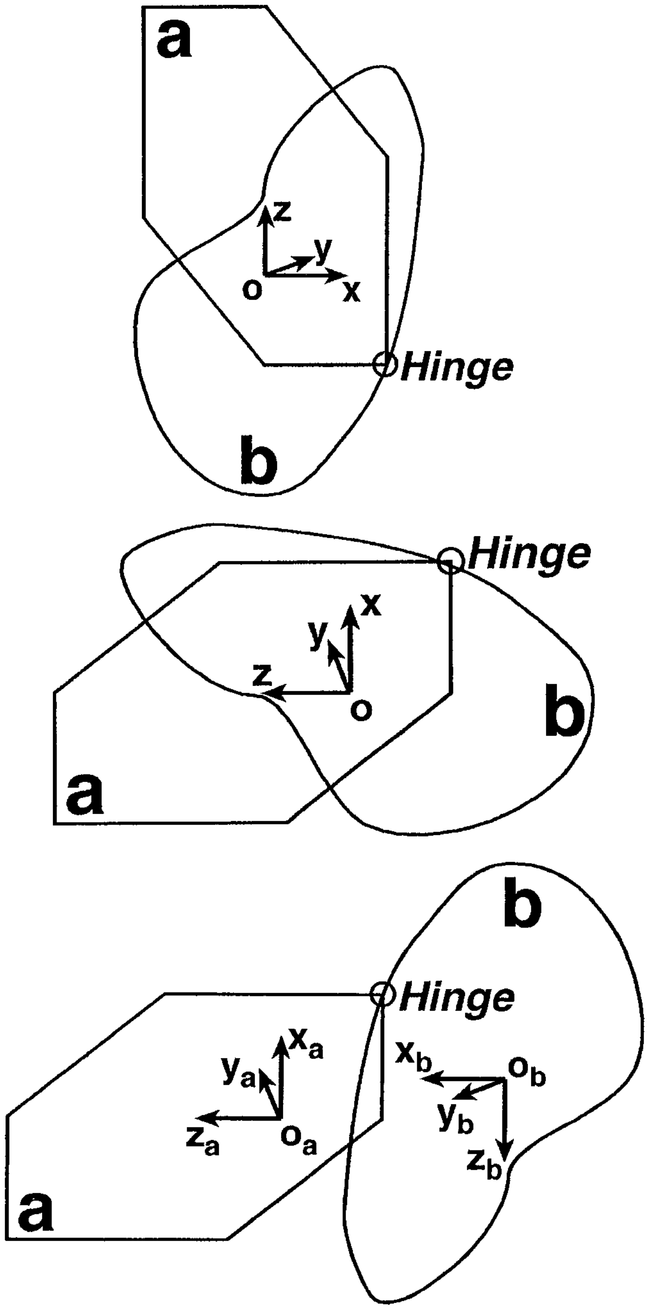

Fig. 1. The Generalized Hough Transform for rigid object recognition. “O” is the model shape. (**top**) The reference frame of “O” is chosen arbitrarily. (**middle**) “O” has undergone a rigid translational and rotational transformation. During the structural comparison phase we receive one candidate reference frame whose score of votes strongly outnumbers the scores of votes of other candidate frames. (**bottom**) “O” has undergone not only translation and rotation, but also one of its parts has been rotated with respect to the other around the hinge. During the structural comparison phase we obtain two candidate reference frames whose score of votes is relatively high. In essence, the positions of these reference frames represent rigid matchings of the target shape with either the first or the second part of the model shape.

features, each of the candidate reference frames obtains votes from about half of the features which contributed to the best candidate reference frame of the first (rigid) case. The situation would be more complicated if there is a large number of different proteins in the database. These would yield many candidate reference frames, some of which may obtain an equal or larger number of votes than either of the two. This is likely to happen if the database contains structures similar to that of our original model shape. In this case, these candidate reference frames would be rejected during filtering the low-scoring reference frames. Hence they would escape identification.

However, the technique we have presented can be further extended to deal with such hinge-bent shapes. Note that in the Generalized Hough Transform algorithm,[23] the protein’s reference frame may be positioned anywhere and its coordinate axes may be defined arbitrarily, without affecting the correctness of the algorithm or its complexity. This unlimited freedom in the definition of one of the algorithm elements points to additional, hidden power in the method which has previously not been explored.[22] Let us see how a judicious choice of this reference frame location allows us to integrate shape information from both parts which are connected by the hinge.

Assume that we are dealing with the same task as described in Figure 1, however, this time we have located the origin of the model reference frame at the hinge point (see Figure 2a). Applying the same recognition technique to the second target shape, we again obtain two different reference frames with a relatively high score of votes, one for each rigid part. However, these reference frames have the same origin (Fig. 2b), although (possibly) at different orientations. Thus if one scores only the locations of the reference frame origin, the information obtained from both parts is combined. These considerations underlie the Articulated Object Recognition technique,[22] which we have adopted here.

METHODS

In this section we provide a detailed description of the flexible hinge-based structural comparison algorithm.

The input is comprised of two proteins which are represented by their set of geometric features. In this study we have represented the proteins as sets of their backbone C_α atoms in 3-D space. Other representations, e.g. by secondary structures,[16] can be used in this scheme as well. We refer to one of the proteins as *the model protein* and to the other as *the target protein*. The model protein is the one possessing a hinge point and hence is comprised of two parts, each of which may be rotated around the hinge point with respect to the other. Based on previous analysis, we choose some point in space—whether encapsulated by the molecular surface envelope and thus within the protein, or outside the protein—to be the hinge point. In the implementation shown here, we have picked a C_α atom to serve as a hinge. This choice divides the single set of points into a pair of ordered sets in 3-D space. The target molecule is also represented by the centers of its C_α atoms. We can associate additional information with each interest point. Here a label of a point is comprised of the protein’s name and the part number to which it belongs. However, addi-

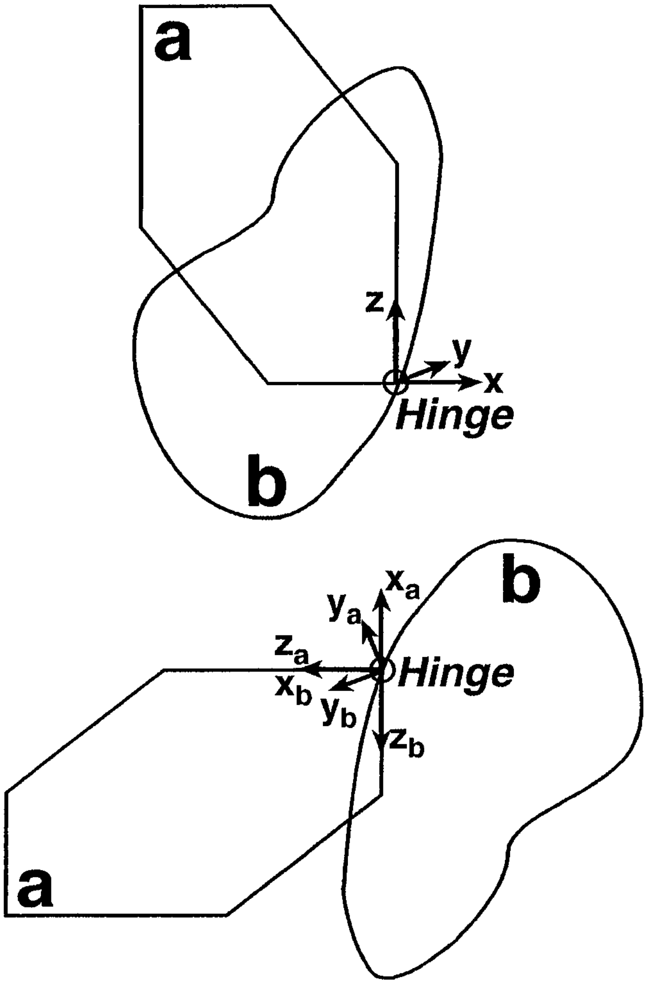

Fig. 2. The extension of the Generalized Hough Transform method for articulated object recognition. (**top**) Unlike in Figure 1, this time the origin of the reference frame of the articulated object has been located at the hinge. (**bottom**) During the comparison phase we again obtain two different reference frames with relatively high scores of votes. However, this time these reference frames have the same origin.

tional information may be added, such as residue name and type, atom type (if atoms other than C_α are utilized), etc.

Partial similarity of protein shapes is captured by the similarity of local shape fragments, which remains conserved under rotation and translation. This local, rigid transformation, invariant shape fragment similarity plays an important role in our approach. We define a **frame-invariant** as a local shape fragment (feature) with the following properties:

i) an unambiguous Cartesian reference frame is defined based on this shape feature;
ii) a numerical vector of indices, which is invariant under rotation and translation of the shape is calculated for this feature. We call this vector the **shape signature** of the feature.

Let us demonstrate this definition for a shape feature that we use—a triplet of C_α atoms, which are not on one line, namely a non-degenerate triangle. One can easily define an unambiguous reference frame which is positioned on this triangle. We take, for example, the origin as the vertex opposite to the shortest triangle side. The x-axis is the direction of the longest side. The y-axis direction is the cross product of the two longer sides. The z-axis direction is the cross product of the x and y axes. If some of the triangle sides are almost equal, one needs to exploit the order of the points or define several redundant frames. The numerical rotation and translation invariant in this example is the triplet of the triangle sides lengths.

A possible superimposition of a model molecule onto the target molecule is determined by accumulating information on the candidate positions of the model's (hinge based) reference frame with respect to the target. This information is based on local comparisons of the frame-invariants which induce such positions. Hence, each candidate position of a reference frame is labeled by the model protein's name and the rigid subpart to which the frame-invariant inducing it belongs. Two labeled *coordinate frames* are defined as consistent if they have the same origin and their labels differ only in their subpart number field of the label.

The flexible hinge-based structural comparison method consists of two major steps:

1. Preprocessing.
   (a) Define the hinge point of the model protein.
   (b) Define the model protein reference frame, with the origin at the hinge point.
   (c) For each frame-invariant of the model protein:
      i. Compute the coordinate frame associated with it;
      ii. Compute the coordinate transformation between this coordinate frame and the (hinge-based) reference frame of the protein and label it with (the protein name, the part number) label;
      iii. Compute the shape signature of the frame-invariant and use it as an address to a table (nicknamed R-Table) for storing the labeled coordinate transformation.

If the database contains more than one model protein the preprocessing step is done for each model protein. Note, that this step can be done off-line without the knowledge of the target protein. Hence, one can prepare the R-table in advance.

2. Recognition.
   (a) For each frame-invariant of the target protein:
      i. Compute the coordinate frame associated with it;
      ii. Compute the shape signature of the frame-invariant and use it as an R-Table entry address. For each record in this R-table entry apply its coordinate transformation on the frame-invariant's coordinate frame to compute a candidate reference frame. Label this candidate reference frame with the label of the applied coordinate transformation. If this coordinate frame with this label has already been obtained, increase its vote score by 1. Otherwise, create a new record to store this coordinate frame.
   (b) Search for consistent, high-scoring pairs of candidate
   (c) Verify the consistent, high-scoring pairs of candidate reference frames.

Verification of hypothetical transformations is done by superimposing the model and target proteins, detecting matching C_α atom centers, and recomputing a transformation which aligns these matching pairs with minimal RMSD. The transformations are ranked again by the number of matching atoms.

Below we describe briefly the implementation of each of the phases.

Preprocessing (R-Table Precomputation)

(a) Interest features

In the current implementation C_α atoms are the interest features of the model protein. We define a frame-invariant to be a triangle with vertices at the interest features that belong to the same part of the molecule. The run-time complexity of the algorithm is heavily dependent on the number of interest features and the number of frame-invariants in the compared objects. To gain increased efficiency in the preprocessing stage, we reduce the number of interest features and the number of frame-invariants in the model molecules by imposing geometrical constraints. A triplet of C_α atoms can be a frame-invariant if:

1. The distance between any two C_α atoms of the triplet is not too large (not larger than **max_triangle_side _length**).
2. The distance between any two C_α atoms of the triplet is not too small (not smaller than **min_triangle_side _length**).
3. All elements of the triplet are sufficiently close to each other in the ordered set they belong to. Namely, the triplet members are not far away from each other on the chain.

Below, these constraints will be collectively referred to as the *proximity constraints.*

(b) Coordinate frames and coordinate transformations

During the preprocessing phase of this flexible hinge-based structural comparison method, we compute the spatial transformation between the local coordinate frames associated with the frame-invariants of the protein and the protein reference frame. Our definitions of the model protein reference frame and the local coordinate frames associated with the frame-invariants of the protein facilitate performing this task. The coordinate frames are defined as orthonormal coordinate frames, i.e. the basis $\{\vec{\mathbf{c}}_1, \vec{\mathbf{c}}_2, \vec{\mathbf{c}}_3\}$ of each coordinate frame meets the following conditions:

$$\forall i \in \{1, 2, 3\} \quad \forall j \in \{1, 2, 3\} \quad (\vec{\mathbf{c}}_i, \vec{\mathbf{c}}_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases} \qquad (1)$$

where $(\cdot, \cdot)$ is the scalar product of vectors in R^3. We exploit this fact during computation of the spatial transformation between the local and the reference coordinate frames. We use $\{\vec{\mathbf{x}}, \vec{\mathbf{y}}, \vec{\mathbf{z}}\}$ instead of $\{\vec{\mathbf{c}}_1, \vec{\mathbf{c}}_2, \vec{\mathbf{c}}_3\}$ to denote the basis of a coordinate frame. Here $\vec{\mathbf{x}}$ denotes a unit vector of the X-axis of a coordinate frame, $\vec{\mathbf{y}}$ is a unit vector of the Y-axis and $\vec{\mathbf{z}}$ is a unit vector of the Z-axis.

Derivation of local coordinate frames. There are a number of ways to define a local coordinate frame for a given frame-invariant. We compute the lengths of the triangle's edges. The vertex joining the longest and the middle–length edge is the origin of a local coordinate frame. We take a unit vector along the longest edge of the triangle, with the beginning at the origin, as an X-axis unit vector $\vec{\mathbf{x}}$. A unit vector along the middle length edge of the triangle, with the beginning in the origin is $\vec{\mathbf{v}}$. The Y-axis unit vector $\vec{\mathbf{y}}$ is defined as the normalized cross product of $\vec{\mathbf{x}}$ and $\vec{\mathbf{v}}$:

$$\vec{\mathbf{y}} = \frac{\vec{\mathbf{x}} \times \vec{\mathbf{v}}}{\|\vec{\mathbf{x}} \times \vec{\mathbf{v}}\|}$$

where $\|\cdot\|$ is a norm in R^3. Note that the length of $\vec{\mathbf{y}}$ is equal to 1, and according to the cross product definition it is orthogonal to $\vec{\mathbf{x}}$. It suffices to store only unit vectors of the coordinate axes X and Y of a coordinate frame. The unit vector $\vec{\mathbf{z}}$ of the coordinate axis Z, can easily be computed as a cross product of the first two unit vectors: $\vec{\mathbf{z}} = \vec{\mathbf{x}} \times \vec{\mathbf{y}}$. The triplet of vectors $\{\vec{\mathbf{x}}, \vec{\mathbf{y}}, \vec{\mathbf{z}}\}$ is an orthonormal basis in R^3.

The reference frame. We define the model protein reference frame as follows: The hinge point of the molecule is chosen to be the origin of the reference frame; vector $\vec{\mathbf{e}}_1 = (1, 0, 0)$ is the X-axis unit vector $\vec{\mathbf{x}}$; vector $\vec{\mathbf{e}}_2 = (0, 1, 0)$ is the Y-axis unit vector $\vec{\mathbf{y}}$; vector $\vec{\mathbf{e}}_3 = (0, 0, 1)$ is the Z-axis unit vector $\vec{\mathbf{z}}$. This reference frame basis complies with the definitions and conditions set above, and hence is orthonormal.

Coordinate transformations. The spatial transformation between two coordinate frames in R^3 has two components, rotational, and translational. Due to the fact that we consider orthonormal bases and the target basis vectors are $\vec{\mathbf{e}}_1 = (1, 0, 0)$, $\vec{\mathbf{e}}_2 = (0, 1, 0)$, and $\vec{\mathbf{e}}_3 = (0, 0, 1)$, the rotational transformation which transforms the orthonormal basis $\{\vec{\mathbf{x}}, \vec{\mathbf{y}}, \vec{\mathbf{z}}\}$, where $\vec{\mathbf{x}} = (x_1, x_2, x_3)$, $\vec{\mathbf{y}} = (y_1, y_2, y_3)$, $\vec{\mathbf{z}} = (z_1, z_2, z_3)$, into $\{\vec{\mathbf{e}}_1, \vec{\mathbf{e}}_2, \vec{\mathbf{e}}_3\}$, can be easily computed from the coordinates of the basis vectors.

The translational transformation may be represented by one vector, *the shift vector.* It is computed by subtracting from the coordinates of the origin of the reference frame, the coordinates of the origin of the local coordinate frame. To apply a spatial transformation on an orthonormal basis, we apply the rotational transformation to each of the basis vectors. New unit vectors of coordinate axes are obtained. The new origin is obtained by computing the *shift vector* in the new coordinate basis and adding it to the origin of the basis.

(c) R-Table structure

A major consideration in the choice of appropriate data structure for the R-Table is efficiency in accessing times during the preprocessing and recognition phases. The R-Table has been implemented as a three-dimensional hash table. Triangles with C_α atom vertices, satisfying the proximity constraints, are defined to be the frame-invariants of the object protein. Their ordered (rounded) edge-lengths serve as address to the R-Table. A triplet (i, j, k) belongs to the address space of the R-Table if its elements satisfy the following inequalities:

$$\textbf{min\_triangle\_side\_length} \le \mathrm{i} \le \mathrm{j} \le \mathrm{k} \le \textbf{max\_triangle\_side\_length}$$

Implementing the R-Table as a 3D hash table with the second dimension size dependent on the first index of the address, and the third dimension size dependent on the first two address indices, saves considerable space. This approach reduces the size of the R-Table. We have approximately six times fewer entries than in a straightforward implementation of the R-Table as an ordinary 3D hash table. Technically, the R-Table is allocated and constructed dynamically.

(d) R-Table entry structure

An R-Table entry is a linked list of records. Each record contains the name of the protein, the part number and the coordinate transformation. During the R-Table precomputation, for each frame-invariant (a triangle with C_α atom vertices, as defined above) of the model protein, we compute its local coordinate frame and the spatial transformation between this local coordinate frame and the previously defined reference frame of the model protein. We insert the molecule name and part number to which the frame-invariant belongs and its spatial transformation into the R-Table, at the address specified by the ordered triangle edges lengths.

Recognition

(a) Space net

During the recognition step, for each frame-invariant (the triangle with C_α atom vertices) of the target molecule we compute its local coordinate frame. The triplet of ordered triangle edge-lengths serves as an address to the R-Table entry, where all relevant coordinate transformations are already stored. Both the respective hash table bin, and its neighbors are visited to allow for some error in the matching. For each of the R-Table entries, we inspect the linked list of records. For each record we apply its coordinate transformation on the frame-invariant associated coordinate frame. The resulting coordinate frame receives the same labels (“protein name” and “part number”) as the applied coordinate transformation. It constitutes a candidate reference frame of that part and protein. We store the triplet (coordinate frame, protein name, part number) in a hash table we call SpaceNet. The (rounded) coordinates of the origin of the triplet coordinate frame divided by the resolution form the address of the SpaceNet entry where this triplet is stored. This type of division enables controlling the bin size in the SpaceNet.

A SpaceNet entry contains two linked lists of records, for each of the (two) parts. Each record contains the triplet data and a counter for votes. This two-part list separation facilitates searching the SpaceNet entry for the pair of candidate reference frames which belong to different parts of the protein. It also reduces the complexity of the new record insertion operation. Insertion of a new record consists of two steps:

1. To store the triplet (coordinate frame, name of a protein, part number) in the hash table SpaceNet, we inspect the relevant linked list and verify consistency with the record.
2. If we find such a record we update it as follows:
   (a) recompute the axis vectors and the origin of the frame;
   (b) orthonormalize the updated coordinate frame;
   (c) increase its votes counter by 1.
   Otherwise, we create a new record, copy to it the triplet’s data and set its votes counter to 1.

After all frame-invariants of the target molecule are processed and all candidate reference frames are computed, we inspect all candidate reference frames and select only those which have accumulated a relatively high vote score. These are likely to constitute matching reference frames.

(b) High-scoring candidate joint (hinge) locations: selection and clustering

Selection. To facilitate subsequent processing we utilize a special data structure for storing the best **candidate_joint_location_list_size** high-scoring candidate reference center locations. To perform the task efficiently and to achieve low space complexity, we have implemented it as a minimum heap of size **candidate_joint_location_list_size.** This takes only *O*(*log* **candidate_joint_location_list_size**) time to insert a new element into the heap, with a constant factor close to 1 (see Cormen et al.[29]).

To select high-scoring candidate center locations, we inspect the SpaceNet hash table. We unite pairs of candidate reference frames occupying the same table entry and associated with different parts of the same model molecule. The votes score that the candidate reference center location receives is the sum of those accumulated by the candidate reference frames. We compare its votes score with “the current maximal votes score,” the maximal score among all candidate reference center locations encountered. If the candidate reference center votes score is greater than the current maximum, we update it and insert this candidate reference center location into the high-scoring candidate reference center locations heap. Otherwise we insert it into the high-scoring candidate reference center locations heap provided that its votes score is greater than a **lower_coefficient** multiplied by the current maximal votes score, and that its score is larger than the score of the head of the heap. The high-scoring candidate reference center locations are clustered.

Clustering. To cluster, for each coordinate frame *F* of high-scoring candidate reference center location, we compute the average of the origins and of the axes vectors of the coordinate frames with the origin and the vectors sufficiently close to those of *F*. The respective candidate votes are summed. Each high-scoring candidate reference center location contains two clusters of candidate coordinate frames, one for each part. Since they have the same origin, their average is computed. The candidate coordinate frames are assigned to that average. For the sake of convenience, we omit the word "cluster" and refer to these clusters of candidate coordinate frames as ordinary coordinate frames.

Any technique for finding a threshold value for filtering low scoring among high-scoring candidate reference center locations may be applied. We discard all candidates whose votes score is less than a specified percentage of the best obtained score. To further reduce the number of candidate reference center locations, we discard those which are close to higher scoring candidates. The selected high-scoring candidate reference center locations are verified.

(c) High-scoring candidate reference center locations verification

Each high-scoring candidate reference center location has a label, the model protein name, and two coordinate frames, considered hypothetical reference frames. Let us label a high-scoring candidate reference center location "A." We label the two coordinate frames associated with it, RF_1 and RF_2. In this step, we verify the similarity between the target molecule and protein "A" following transformation of each of its two parts to RF_1 and RF_2 respectively.

Coordinate transformation computations. We take advantage of the convenient (and simple) choice of the reference frame of the model molecules. As noted, vectors $\vec{\mathbf{e}}_1 = (1, 0, 0)$, $\vec{\mathbf{e}}_2 = (0, 1, 0)$, $\vec{\mathbf{e}}_3 = (0, 0, 1)$, have been chosen as unit vectors of coordinate axes of the reference frame of the model molecule. The hinge, $\vec{\mathbf{h}} = (h_1, h_2, h_3)$, has been chosen as the origin of the reference frame. Given a coordinate frame in R^3 with the basis $\vec{\mathbf{x}} = (x_1, x_2, x_3)$, $\vec{\mathbf{y}} = (y_1, y_2, y_3)$, $\vec{\mathbf{z}} = (z_1, z_2, z_3)$ and the origin $\vec{\mathbf{o}} = (o_1, o_2, o_3)$, the translational and the rotational components of the spatial transformation which transforms the model molecule reference frame into this coordinate frame can be easily computed from the coordinates of the basis vectors.

Molecular comparison. Let us call the image of protein "A," following applications of the two aforementioned spatial transformations on its parts, "$\tilde{A}$." To compare "$\tilde{A}$" with the target molecule we compute matching pairs of C_α atoms of these two molecules. A C_α atom "a" of molecule "$\tilde{A}$" and a C_α atom "b" of the target molecule match if "a" is the closest C_α atom of "$\tilde{A}$" to "b," "b" is the closest C_α atom of the target molecule to "a," and the distance between "a" and "b" is less than or equal to a **matching_parameter.** Owing to the third condition, the matching pairs computation task can be performed in linear time on the size of the input. To this end, we map the target molecule onto the 3-D space which is implemented as a 3-D hash table of points. For the model molecule "A" we compute "$\tilde{A}$," and for each C_α atom of "$\tilde{A}$" we compute its hash address. We access a hash table entry at this address, and its neighbors as determined by the **matching_parameter.** We seek the closest C_α in the target molecule. It is easy to see that the complexity of this procedure depends linearly on the size of the target and model molecules. After all matching pairs between "$\tilde{A}$" and the target molecule have been assembled, we compute the RMSD between these molecules when only matching C_α pairs are taken into account. The number of matching pairs and the RMSD between "$\tilde{A}$" and the target molecule are used to measure the degree of similarity between the molecules.

Complexity Analysis

In the preprocessing, for each frame-invariant of a model molecule, we compute its coordinate frame and spatial transformation which transforms this coordinate frame to the reference frame of the molecule. This computation takes $O(1)$. Insertion of this transformation along with additional data into the R-table takes $O(1)$ as well. Hence, the complexity of the step is $O(\Sigma_m\ k_m)$, where k_m is the number of frame-invariants of the m–th model molecule, and m runs through all model molecules in the database. According to the last *proximity constraint*, the number of frame-invariants of a model molecule is $O(\mathrm{n})$, where n is the number of C_α atoms in the model molecule. The complexity of this preprocessing step is, therefore, $O(\Sigma_m\ n_m)$, where n_m is the number of C_α atoms in the m–th model molecule. m runs through all model molecules in the database.

The complexity of the candidate reference frames computation and insertion into SpaceNet hash-table is $O(P_{RT} + P_{SN})$, where P_{RT} is the sum of products of the form $d_i p_i$, d_i is the number of frame-invariants of all model molecules in the i–th and neighboring bins of the R-Table, p_i is the number of frame-invariants of the target molecule with the same R-Table address as the i–th bin of the R-Table. i runs through all hash-addresses of the R-Table, and P_{SN} is the sum of items of the form l_j^2. l_j is the number of candidate reference frames which have the same SpaceNet address as the j–th bin of the SpaceNet, and j runs through all hash addresses of the SpaceNet.

The complexity of high-scoring candidate joint locations selection is $O(P_{SN})$. As previously, the computation of matching pairs takes $O(n_1 + n_2)$, where n_1 and n_2 are the numbers of C_α atoms in the compared molecules. Thus, the overall complexity of the verification stage is $O(\Sigma_{m'} n_{m'} + Mn_{target})$, when $n_{m'}$ is the number of C_α atoms in the m'–th molecule, n_{target} is the number of C_α atoms in the target molecule, M is the number of high-scoring candidate reference centers and m' runs through high scoring molecules. Hence, the overall complexity of the recognition step of the method is $O(P_{RT} + P_{SN} + \Sigma_{m'} n_{m'} + Mn_{target})$. Accordingly, the complexity of the method is $O(\Sigma_m n_m + P_{RT} + P_{SN} + Mn_{target})$, when m runs through all model molecules in the database.

RESULTS

Our results are summarized in three tables. Table I describes the cases we have used in this study, and Table II presents the results we have obtained for them: the number of matching pairs in each of the parts, the RMSDs which have been obtained, and the rotation angles. Table III lists the CPU times used in the comparisons, in seconds, on a Silicon Graphics Indigo R4400, 150MHz workstation. Some of the cases are described in detail below. We have divided these examples to two types. First we provide a comparison of our results with the ones obtained earlier for the rigid molecule comparison. Next we present results demonstrating the ability of the program to discover domain motions in proteins.

TABLE I. The Molecules Used in This Study

PDBcode	Name	Identifying information	Source
3cts	Oxo-acid-lyase	Citrate synthase (e.c.4.1.3.7)—(co*a, citrate) complex	Chicken (Gallus gallus) heart muscle
1cts	Oxo-acid-lyase	Citrate synthase (e.c.4.1.3.7)—citrate complex	Pig (Sus scrofa) heart
1hkg	Transferase	Hexokinase a and glucose complex (e.c.2.7.1.1)	Yeast (*Saccharomyces cerevisiae*)
2yhx	Transferase (phosphoryl, alcohol acceptr)	Yeast hexokinase b (e.c.2.7.1.1) complex with ortho-toluoylglucosamine	Baker's yeast (*Saccharomyces cerevisiae*)
3wrp	DNA binding regulatory protein	trp aporepressor	*Escherichia coli*
1wrp	DNA binding regulatory protein	trp repressor (trigonal form)	*Escherichia coli*
5er2	Hydrolase (acid proteinase)	Endothia aspartic proteinase (endothiapepsin) (e.c.3.4.23.6) complex with cp-69,799	Chestnut blight fungus (*Endothia parasitica*)
4ape	Hydrolase (acid proteinase)	Acid proteinase (e.c.3.4.23.10), endothiapepsin	Chestnut blight fungus (Endothia parasitica)
1ama	Transferase (aminotransferase)	Aspartate aminotransferase (e.c.2.6.1.1) complex with alpha-methyl aspartate-pyridoxal-5′-phosphate	Chicken (*Gallus gallus*) heart mitochondria
9aat	Transferase (aminotransferase)	Aspartate aminotransferase (e.c.2.6.1.1) complex with pyridoxal-5′-phosphate at ph 7.5	Chicken (*Gallus gallus*) heart mitochondria
1bp2	Hydrolase	Phospholipase (e.c.3.1.1.4)	Bovine (box taurus 1.) pancreas
1pp2	Hydrolase	Calcium-free phospholipase (e.c.3.1.1.4)	Western diamondback rattlesnake (*Crotalus atrox*)
2gd1	Oxidoreductase (aldehyde(d)-nad(a))	Apo-*d-*glyceraldehyde-3-phosphate dehydrogenase (e.c.1.2.1.12)	*Bacillus sterothermophilus/nca* 1503
1gd1	Oxidoreductase (aldehyde(d)-nad(a))	Holo-*d-*glyceraldehyde-3-phosphate dehydrogenase (e.c.1.2.1.12)	*Bacillus sterothermophilus/nca* 1503
1cll	Calcium-binding protein	Calmodulin (vertebrate)	Human (*Homo sapiens*) recombinant form
4cln	Calcium-binding protein	Calmidulin	*Drosophila melanogaster* expressed in (*Escherichia coli*)
2bbm	Calcium-binding protein	Calmodulin (calcium-cound) complexed with rabbit skeletal myosin light chain kinase (calmodulin-binding domain) (nmr, minimized average structure)	Calmidulin: *Drosophila melanogaster;* peptide: synthetic
8adh	Oxidoreductase (nad(a)-chol(d))	Apo-liver alcohol dehydrogenase (e.c.1.1.99.8)	Horse (*Equus caballus*) liver
6adh	Oxidoreductase (nad(a)-choh(d))	Holo-liver alcohol dehydrogenase (e.c.1.1.1.1) complex with nad and dmso	Horse (*Equus caballus*) liver
1196	Hydrolase (0-glycosyl)	Lysozyme (e.c.3.2.1.17) mutant with ile 3 replaced by pro (I3p) (space group p 32 2 1)	Bacteriophage t4 (mutant gene derived from the m13 plasmid by cloning the t4 lysozyme gene)
1197	Hydrolase (o-glycosyl)	Lysozyme (e.c.3.2.1.17) mutant with ile 3 replaced by pro (I3p) (space group p 21 21 2)	Bacteriophage t4 (mutant gene derived from the m13 plasmid by cloning the t4 lysozyme gene)
1lfh	Iron transport	Lactoferrin (apo form)	Human (*Homo sapiens*)
1lfg	Transferrin	Lactoferrin (diferric)	Human (*Homo sapiens*)
2lao	Amino-acid binding protein	Lysine-, arginine-, ornithine-binding protein (lao)	*Salmonella typhimurium*
1lst	Amino-acid binding protein	Lysine-, arginine-, ornithine-binding protein (lao) complexed with lysine	*Salmonella typhimurium*
3gapa	Chain A of gene regulatory protein	Catabolite gene activator protein—cyclic/amp complex	*Escherichia coli*
3gapb	Chain B of gene regulatory protein	Catabolite gene activator protein—cyclic/amp complex	*Escherichia coli*
1ddt	Toxin	Dimeric diphtheria toxin	*Corynebacterium diphtheriae*
1mdt	Toxin	Monomeric diphtheria toxin	*Corynebacterium diphtheriae*
2tbvc	Virus (chain c)	Tomato bushy stunt virus	Tomato bushy stunt virus
2tbva	Virus (chain a)	Tomato bushy stunt virus	Tomato bushy stunt virus

TABLE II. Hinge-bending Matching Results[†]

Model molecule and hinge position	Target molecule	Model size	Target size	First part votes score	Second part votes score	First part matching value	First part RMS	Second part matching value	Second part RMS	Matching value	RMS	Inter-domain rotation angle
1cts(71)	3cts	437	429	11	226	64	1.45	312	1.26	376	1.29	2.65
2yhx(71)	1hkg	457	457	15	27	66	1.50	337	1.31	403	1.34	1.74
1wrp(71)	3wrp	103	101	56	10	62	0.9	30	1.16	92	0.99	1.74
4ape(71)	5er2	330	334	105	249	70	0.67	259	0.87	329	0.83	2.29
9aat(71)	1ama	802	401	4	298	58	1.62	301	0.96	359	1.09	3.84
1pp2(6)	1bp2	244	123	3	54	5	1.1	105	1.41	110	1.4	1.7
1gd1(171)	2gd1	1336	1336	113	451	170	0.88	1142	1.08	1312	1.05	4.12
2bbm(76)	1cll	174	144	12	6	37	1.62	64	1.73	101	1.69	173.78
2bbm(78)	4cln	174	148	27	9	33	1.45	54	1.71	87	1.61	172.01
6adh(171)	8adh	748	374	170	201	167	0.6	201	1.14	368	0.93	7.79
1197(74)	1196	328	162	74	13	67	1.07	89	1.3	156	1.21	30.63
1lfg(250)	1lfh	691	691	183	342	166	1.0	434	1.15	600	1.11	57.53
1lfg(250)	1lfh	691	691	112	333	131	1.49	432	1.04	563	1.16	7.68
1lst(91)	2lao	238	238	177	165	87	1.18	106	0.89	193	1.03	51.91
1lst(91)	2lao	238	238	177	98	90	0.68	72	1.26	162	0.98	1.92
3gapb(130)	3gapa	205	208	45	19	122	0.94	74	1.13	196	1.02	30.01
1mdt(373)	1ddt	1046	523	306	96	366	0.99	143	1.50	509	1.16	179.27
2tbva(165)	2tbvc	287	322	293	261	164	0.4	120	1.09	284	0.77	21.54

[†] The first column gives the PDB file name of the model molecule. The number given in parenthesis is the location of the hinge. The second column lists the corresponding target PDB file name. The next two columns list the sizes of the two corresponding molecules. The 5th and 6th columns give the number of votes scored by each of the two parts. The 7th column lists the number of matching C_α pairs in the first part of the model molecule. The next (8th) column gives the RMSD obtained by the first part. The 9th column tabulates the number of matching C_α pairs in the second part of the model molecule, and the 10th column gives the RMSD obtained by this, second, part. The next (11th) column gives the sum of the matching C_α pairs between the two—model and target—molecules obtained by both parts. The overall RMSD of the match is listed in the 12th column. The interdomain rotation angle obtained following the transformation to obtain the match is given in the last column.

TABLE III. The CPU Times (in Seconds) Used in the Comparisons[†]

	Model	Scene	Prepro-cessing (s)	Recog-nition (s)	Comments
1.	1cts	3cts	0.42	103.38	
2.	2yhx	1hkg	0.44	95.78	
3.	1wrp	3wrp	0.10	16.86	
4.	4ape	5er2	0.27	21.84	
5.	9aat	1ama	0.84	148.58	
6.	1pp2	1bp2	0.23	29.79	
7.	1gd1(O)	2gd1(O)	0.28	24.79	Chain O vs. Chain O.
8.	2bbm	1cll	0.19	75.42	
9.	2bbm	4cln	0.18	63.36	
10.	6adh	8adh	0.63	79.75	
11.	1197	1196	0.32	45.90	
12.	11fg	11fh	0.25	66.73	
13.	11st	21ao	0.21	21.58	
14.	3gapb	3gapa	0.21	22.97	
15.	1mdt	1ddt	0.56	76.30	
16.	2tbva	2tbvc	0.28	22.16	

[†] These are given separately for the **preprocessing** and the **recognition** stages of the algorithm. The runs were performed on a Silicon Graphics workstation (Indigo station, 150MHZ MIPS R4400, 96 MB of RAM). The only exception was 1lfg-1lfh, which needed more memory and hence was run on a stronger SGI machine (4-processors 195 MHZ MIPS R10000 with 1024 of RAM).

Comparisons With Previously Obtained Similarities

(a) Phospholipase A2

We compare two representatives of phospholipase A2 protein family: bovine pancreas (PDB code **1bp2,** Bernstein et al.[25]) and domain R from Crotalus atrox venom (**1pp2**). We define the R domain from Crotalus atrox venom (**1pp2**) to be the model molecule and bovine pancreas (**1bp2**) to be the target molecule. To simulate a rigid comparison, we position the hinge at the sixth C_α atom of **1pp2.** Our program provides an excellent match of these two proteins. During the verification stage we transform the model molecule, each part with its respective transformation. As a result of these transformations, the two parts of **1pp2** are rotated with respect to each other by 1.69 degrees. Since one of the parts has only five C_α atoms, the match may be considered rigid. Moving the hinge along the domain R from the Crotalus atrox venom (**1pp2**) similar matchings are observed in all cases. 110 matching pairs of C_α atoms are obtained (versus 98 pairs by Bachar et al., 1993[30]). Consequently, the RMSD calculated now is larger (1.39 Å *versus* 0.89 Å previously).

The first match we obtain is:

1*pp*2	:	1–14,	15–30,	33–55,	57–67,	72–76,	79–99,	102–115
1*bp*2	:	1–14,	16–31,	34–56,	66–76,	81–85,	89–109,	110–123

The match reported in Bachar et al.[30] is 1–12, 16–30, 34–55, 67–75, 89–109, and 113–126 in both molecules.

(b) Glyceraldehyde dehydrogenase

The reaction mechanism and the structure of d-glyceraldehyde-3-phosphate dehydrogenase have been studied extensively owing to its biological importance and interesting properties regarding cooperativity of coenzyme binding. The amino acid sequences of the enzyme from various

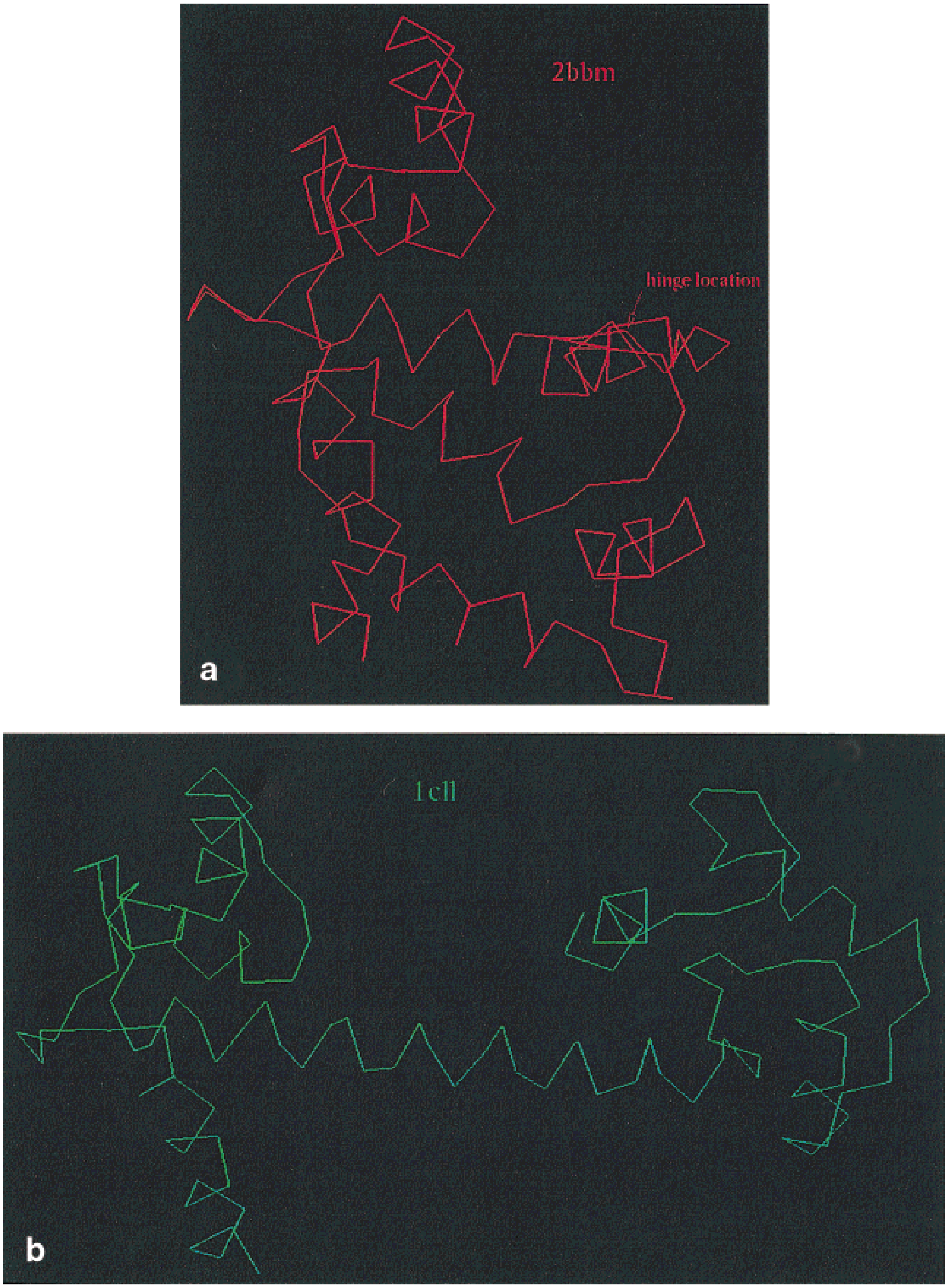

Fig. 3. Calmodulin (calcium-bound) complexed with rabbit skeletal myosin light-chain kinase (PDB code: **2bbm**) versus calmodulin of human (**1cll**). To match to **1cll**, the two parts of **2bbm** have been rotated with respect to each other by 173.78 deg. (**a**) **2bbm**; (**b**) **1cll**; (**c**) the two structures superimposed. (**d**) Calmodulin (calcium-bound) complexed with rabbit skeletal myosin light chain kinase (**2bbm**) versus calmodulin of drosophila melanogaster (**4cln**). To match to **4cln**, the two parts of **2bbm** have been rotated with respect to each other by 172.01 deg.

sources show a high degree of homology between molecules from different species. Here we compare apo-d-glyceraldehyde-3-phosphate dehydrogenase (**2gd1**) and holo-d-glyceraldehyde-3-phosphate dehydrogenase (**1gd1**). Each protein is comprised of four chains, each 334 residues long. The chains in both proteins are labeled "*O*," "*P*," "*Q*," and "*R*." To distinguish between chains belonging to these two proteins, we denote chains belonging to **1gd1** "O_1," "P_1," "Q_1," and "R_1," and "O_2," "P_2," "Q_2," and "R_2" the chains belonging to **2gd1.** We chose holo-d-glyceraldehyde-3-phosphate dehydrogenase 2 to be the model molecule and apo-d-glyceraldehyde-3-phosphate dehydrogenase 2 to be

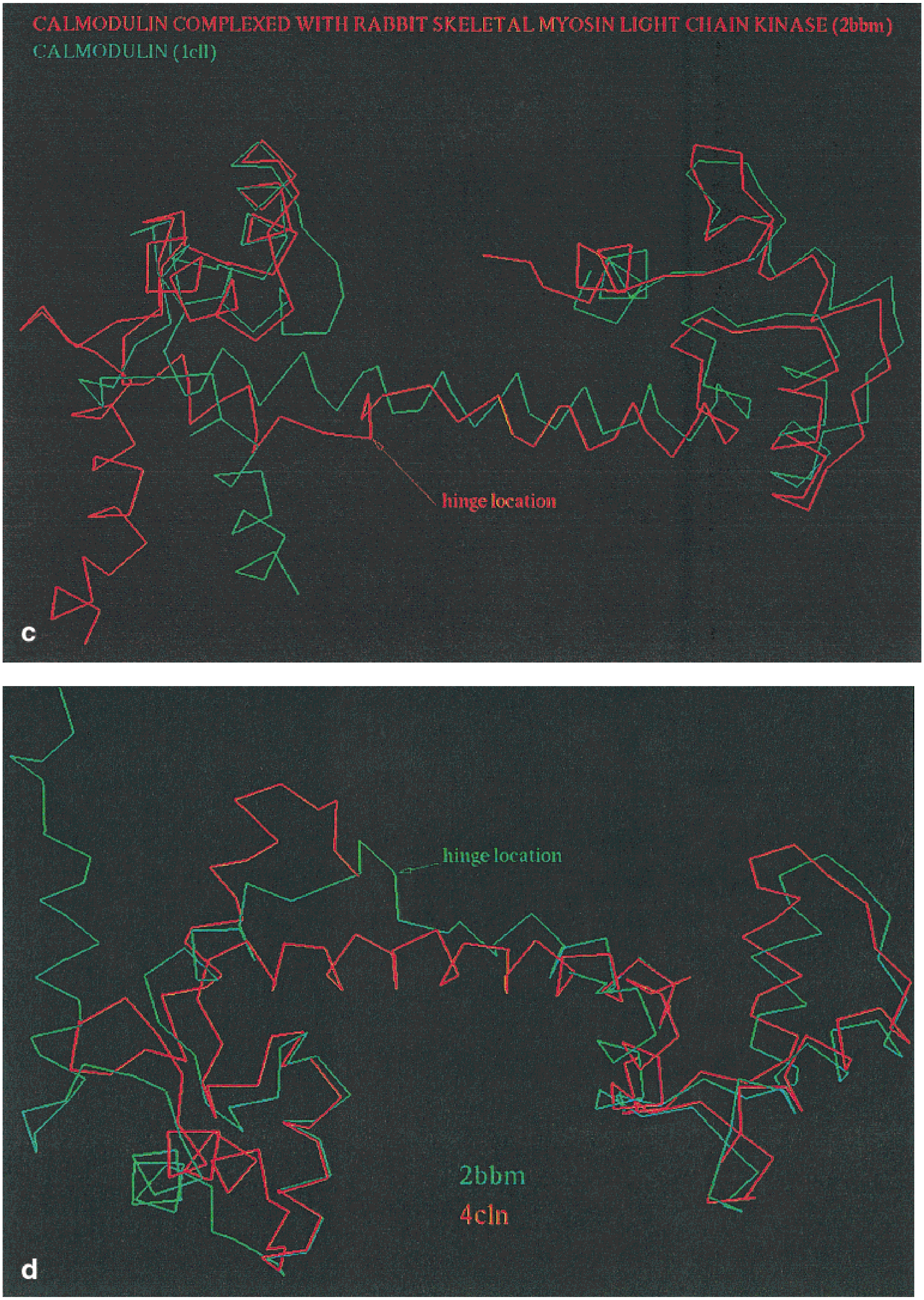
CALMODULIN COMPLEXED WITH RABBIT SKELETAL MYOSIN LIGHT CHAIN KINASE (2bbm)
CALMODULIN (1cll)
hinge location
c
hinge location
2bbm
4cln
d

Figure 3. (Continued.)

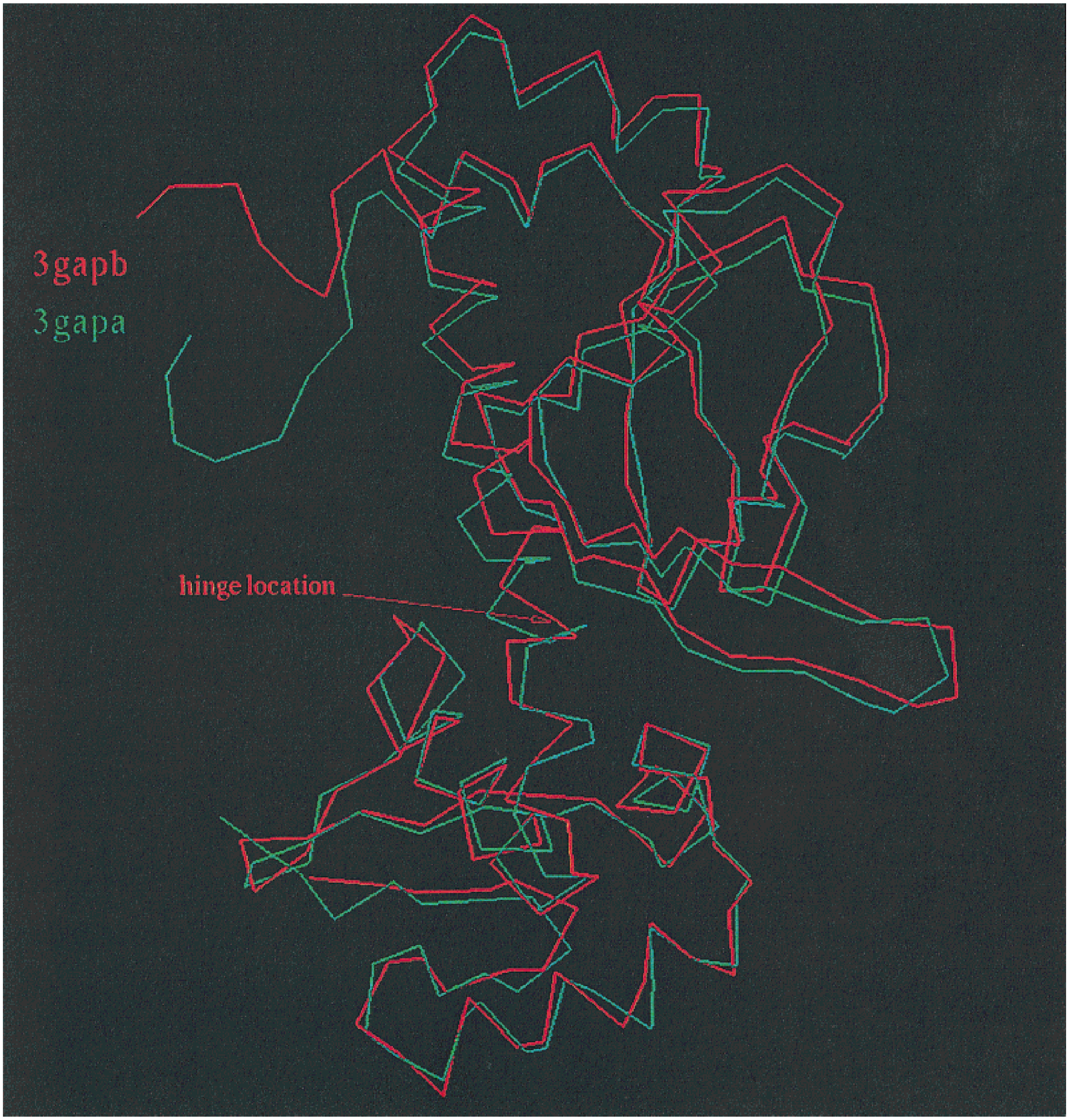

Fig. 4. Chain A of Catabolite Gene Activator Protein (**3gap**) versus chain B of the same protein. To match to chain A, the two parts of chain B have been rotated with respect to each other by 30.0 deg.

the target one. Two runs were performed. In the first the hinge is at the sixth C_α atom of **1gd1,** and in the second at the 151-st C_α. The results we have obtained in these two runs are very similar, although those obtained in the first case resemble the rigid matching more than the ones obtained in the second. In the first case, the two parts of **1gd1** were rotated by about 1–2 degrees with respect to each other, and in the second case by 4–5 degrees. In the first case the size of the first part of **1gd1** is only 5 residues. Here we provide the results of the second case.

We obtained four different matchings (ordered by the number of matching C_α pairs):

1. The number of matching C_α atom pairs is 1312. The RMS distance is 0.95 Å. The main continuous matching fragments are:

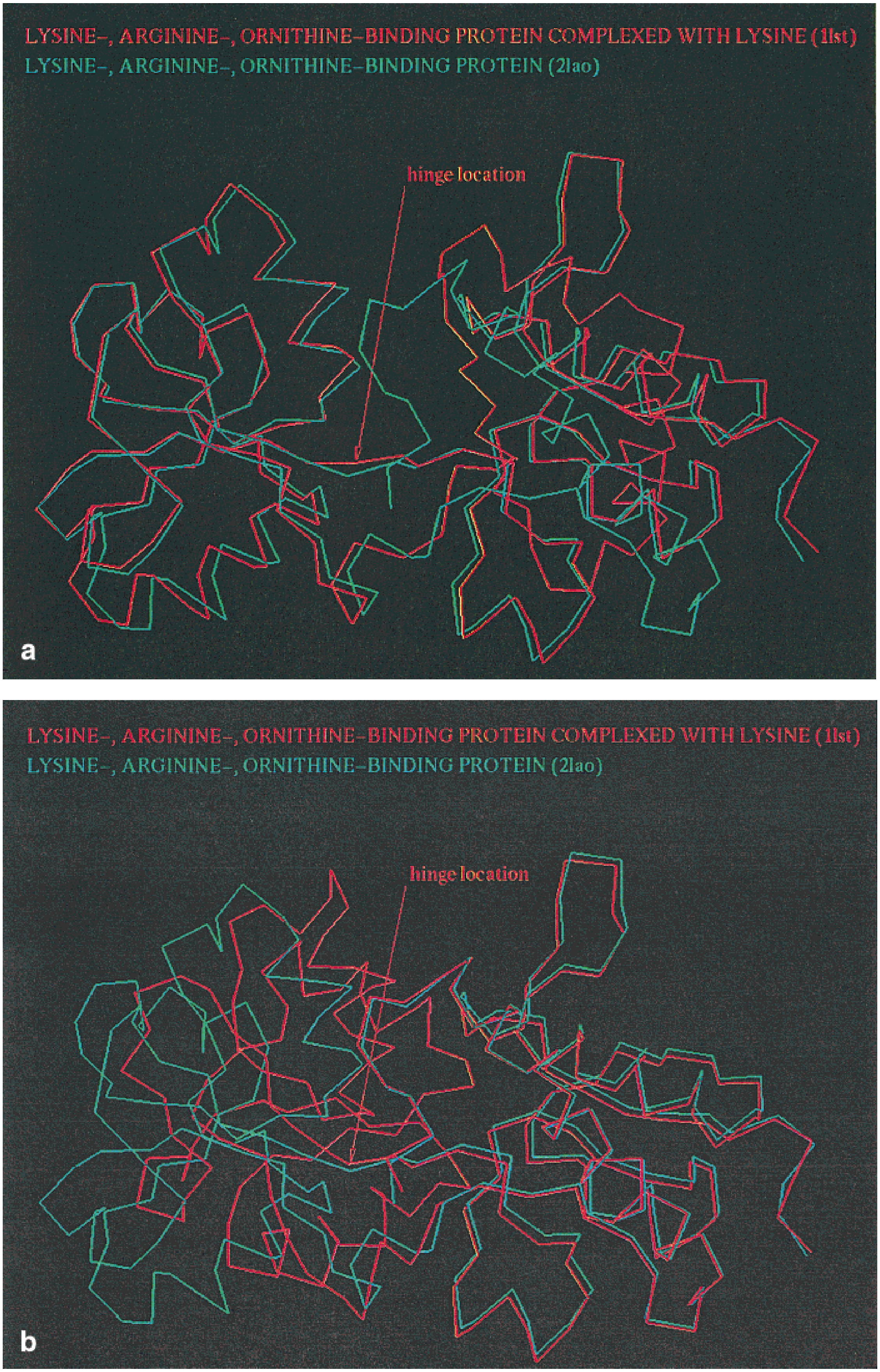
LYSINE-, ARGININE-, ORNITHINE-BINDING PROTEIN COMPLEXED WITH LYSINE (1lst)
LYSINE-, ARGININE-, ORNITHINE-BINDING PROTEIN (2lao)
hinge location
a
LYSINE-, ARGININE-, ORNITHINE-BINDING PROTEIN COMPLEXED WITH LYSINE (1lst)
LYSINE-, ARGININE-, ORNITHINE-BINDING PROTEIN (2lao)
hinge location
b

Fig. 5. Lysine-, arginine-, ornithine-binding protein (LAO) complexed with lysine (**1lst**) versus lysine-, arginine-, ornithine-binding protein (LAO) (**2lao**). (**a**) The first of two matches. To match to **2lao**, the two parts of **1lst** have been rotated with respect to each other by 51.91 deg. (**b**) The second match. To match to **2lao**, the two parts of **1lst** have been rotated with respect to each other by 1.92 deg.

Model(1*gd*1)	:	1–322,	335–410,	413–668,	669–675
Target(2*gd*1)	:	1003–1334,	669–744,	747–1002,	335–341
Model(1*gd*1)	:	677–699,	703–742,	755–765,	783–1000
Target(2*gd*1)	:	343–365,	369–408,	421–431,	449–666
Model(1*gd*1)	:	1003–1078,	1081–1112,	1114–1336	
Target(2*gd*1)	:	1–76,	79–110,	112–334	

As one can easily see, in this match chain “O_1” matches to “R_2,” “P_1” to “Q_2,” “Q_1” to “P_2,” and “R_1” to “O_2.”

2. The number of matching C_α atom pairs is 1,308. The RMSD is 1.03 Å. The main continuous matching fragments are 1–367, 369–408, 420–431, 439–442, 447–699, 703–742, 754–765, 771–778, 781–808, 811–1035, 1037–1075, 1081–1083, 1088–1112, 1115–1334, in both molecules. Clearly, in this match chain “O_1” matches to “O_2,” “P_1” to “P_2,” “Q_1” to “Q_2,” and “R_1” to “R_2.”
3. The number of matching C_α atom pairs is 1,307. The RMSD is 0.97 Å. The main continuous matching fragments are:

Model(1*gd*1)	:	1–332,	335–340,	343–365,	369–408
Target(2*gd*1)	:	335–666,	1–6,	9–31,	35–74
Model(1*gd*1)	:	423–431,	449–668,	669–699,	703–742
Target(2*gd*1)	:	89–97,	115–334,	1003–1033,	1037–1076
Model(1*gd*1)	:	755–765,	772–778,	781–1000,	1003–1336
Target(2*gd*1)	:	1089–1099,	1106–1112,	1115–1334,	669–1002

This time chain “O_1” matches to “Q_2,” “P_1” to “R_2,” “Q_1” to “O_2,” and “R_1” to “P_2.”

4. The number of matching C_α atom pairs is 1,296. The RMSD is 1.04 Å. The main continuous matching fragments are

Model(1*gd*1)	:	1–365,	369–408,	423–431,	449–668
Target(2*gd*1)	:	669–1033,	1037–1076,	1091–1099,	1117–1336
Model(1*gd*1)	:	671–699,	703–741,	755–765,	781–803
Target(2*gd*1)	:	3–31,	35–73,	87–97,	113–135
Model(1*gd*1)	:	806–809,	814–1001,	1003–1025,	1028–1078
Target(2*gd*1)	:	138–141,	146–333,	335–357,	360–410
Model(1*gd*1)	:	1081–1083,	1086–1112,	1115–1334	
Target(2*gd*1)	:	413–415,	418–444,	447–666	

Now chain “O_1” matches to “P_2,” “P_1” to “O_2,” “Q_1” to “R_2” and “R_1” to “Q_2.”

The matches between the chains in all four cases are almost perfect. It is easy to see that these matches reveal a remarkable structural similarity of all chains of both proteins. Furthermore, these matchings indicate the spatial arrangement of the chains of both proteins. Assembling these results together, we get (the sign “⇔” implies “matches to”):

Match #1	Match #2	Match #3	Match #4
“O_1” ⇔ “R_2”	“O_1” ⇔ “O_2”	“O_1” ⇔ “P_2”	“O_1” ⇔ “Q_2”
“P_1” ⇔ “Q_2”	“P_1” ⇔ “P_2”	“P_1” ⇔ “O_2”	“P_1” ⇔ “R_2”
“Q_1” ⇔ “P_2”	“Q_1” ⇔ “Q_2”	“Q_1” ⇔ “R_2”	“Q_1” ⇔ “O_2”
“R_1” ⇔ “O_2”	“R_1” ⇔ “R_2”	“R_1” ⇔ “Q_2”	“R_1” ⇔ “P_2”

Since all chains of both proteins are extremely similar to each other we can consider these matchings as matchings of **1gd1** (or **2gd1**) with itself. Thus, we can divide the four chains to two pairs, {“O,” “Q”} and {“P,” “R”}, such that in each pair a rotation by 180 deg around these axes will transform one chain to another.

Domain Motions in Proteins

(a) Motion in calmodulin

Calmodulin (CaM) is a ubiquitous Ca^{2+} binding protein. It is involved in a wide range of cellular Ca^{2+}-dependent signaling pathways. It regulates the activity of a large number of proteins including protein kinases, protein phosphatases, nitric oxide synthase, inositol triphosphate kinase, nicotinamide adenine dinucleotide kinase, cyclic nucleotide phosphodiesterase, Ca^{2+} pumps, and proteins involved in motility.[31] Here we compare calmodulin (calcium-bound) complexed with rabbit skeletal myosin light-chain kinase (**2bbm**) (Fig. 3a) with human calmodulin (**1cll,** Fig. 3b) and with calmodulin from *Drosophila melanogaster* (**4cln**). We choose **2bbm** to be the model protein in both comparisons and **1cll,** and **4cln** the target ones. Our program reveals a non-rigid matching between these two protein pairs. In the first match, the hinge has been put at the 76-th C_α atom of **2bbm** (Fig. 3c). The two parts of **2bbm** have been rotated with respect to each other by 173.8 degrees. **2bbm** contains 148 residues and **1cll** 144. The number of matching C_α pairs is 101. The main continuous matching fragments of these two proteins are

Model(**2bbm**)	:	26–30,	42–63,	86–112,	115–142
Target(**1cll**)	:	23–27,	39–60,	83–109,	112–139

The RMSD between the transformed **2bbm** and **1cll** is 1.69 Å. Putting the hinge at the 26-th C_α atom of **2bbm** we obtain the following main continuous matching fragments of the molecules

Model(**2bbm**)	:	6–21,	28–32,	56–73
Target(**1cll**)	:	3–18,	25–29,	53–70

Comparing (**2bbm** with **4cln**) we again pick **2bbm** to be the model molecule. The best match is obtained when the hinge is put at the 78-th C_α atom of **2bbm** (Fig. 3d). The parts of **2bbm** are rotated with respect to each other by 172 deg. Both **2bbm** and **4cln** are 148 residues long. The number of matching C_α pairs of transformed **2bbm** and **4cln** are only 87. The main continuous matching fragments between these proteins are 27–29, 43–62, 88–102, 115–129, 134–144, in both model (**2bbm**) and target (**4cln**) molecules. The RMSD between transformed **2bbm** and **4cln** is 1.61 Å.

(b) Motion in alcohol dehydrogenase (ADH)

We compare holo alcohol dehydrogenase complexed with NAD and DMSO (PDB code **6adh**) and apo alcohol dehydrogenase (**8adh**). In his protein motion database, Gerstein et al.[19] classifies the domain motion in alcohol dehydrogenase as a shear mechanism. Our program is still able to reveal an almost perfect match between chain A of **6adh** and **8adh** (not shown). We define **6adh** to be the model molecule and put the hinge at the 171-st C_α atom. The two parts of **6adh** are rotated by 7.8 deg during the verification stage of the algorithm. Both chain A of **6adh** and **8adh** are 374 residues long. The number of matching C_α pairs in the two molecules is 368. The main continuous matching fragments of these two proteins are 4–97, 100–295, 299–352, and 358–374 in both model (**6adh**) and target (**8adh**) molecules. The RMSD between the transformed model molecule and the target molecule is 0.93 Å.

(c) Motion in catabolite gene activator protein (CAP)

We compare chains A and B of catabolite gene activator protein (**3gap**). Chain B has been defined to be the model protein and chain A—the target one. We put the hinge at the 130–th C_α atom of chain A. A non-rigid matching between these two chains is observed. To match chain A, the two parts of chain B are rotated with respect to each other by about 30 degrees. Chain A is 208 residues long and chain B consists of 205. The number of matching C_α pairs is 196. The main continuous matching fragments are residues 6–8 from the model (**3gapb**) with 7–9 from the target (**3gapa**), and 10–53, 55–134, 138–152, and 155–204 from both molecules. The RMSD between the transformed chain B and chain A of **3gap** is 1.02 Å (Fig. 4).

(d) Motion in lysine/arginine/ornithine (LAO) binding protein

We compare lysine-, arginine-, ornithine-binding protein (LAO) complexed with lysine (**1lst**) with lysine-, arginine-, ornithine-binding protein (LAO) (**2lao**). We choose **1lst** to be the model molecule and **2lao** to be the target. The hinge has been put arbitrarily at the 71–st C_α atom of **1lst.** The following two matches have been obtained:

1. The number of matching pairs of C_α atoms of **1lst** and **2lao** is 164. The main continuous matching fragments are 1–91 and 189–238 in both model (**1lst**) and target (**2lao**) molecules.
2. The number of matching pairs of C_α atoms of **1lst** and **2lao** is 141. The main continuous matching fragments are residues 59–61 from the model (**1lst**) with 68–66 from the target (**2lao**), and 91–163 and 165–192 from both molecules.

Three large fragments of the two proteins are matched: 1–91 (first match), 91–192 (second match), 189–238 (first match). Both proteins are 238 residues long. We therefore put a hinge at the 91–st C_α atom of **1lst**. This time the following two matches were obtained (Fig. 5a,b):

1. The number of the matching pairs of C_α atoms of **1lst** and **2lao** is 193. The main continuous matching fragments are 1–7, 10–86, and 88–192 from both model (**1lst**) and target (**2lao**) molecules. To match **2lao**, the two parts of **1lst** are rotated with respect to each other by 51.9 deg. The RMSD between the transformed model molecule and the target molecule is 1.02 Å.
2. The number of matching pairs of C_α atoms of **1lst** and **2lao** is 162. The main continuous matching fragments are 1–91, and 189–238 from both molecules. To match to **2lao**, the two parts of **1lst** are rotated with respect to each other by 1.92 deg, i.e., this is a rigid match. The RMSD between the transformed model molecule and the target molecule is 0.98 Å.

It is easy to see that here we have two hinges. The first is at the 91-st C_α atom and the second close to the 192-nd C_α. This example demonstrates the ability of our program to reveal this type of domain motion.

(e) Motion in lactoferrin

We compare two lactoferrin conformations: lactoferrin difleric (**1lfg**) and lactoferrin apo form (**1lfh**). We pick **1lfg** to be the model molecule and **1lfh** to be the target one. The hinge has been put arbitrarily at the 71–st C_α atom of the model molecule **1lfg**. Both proteins are 691 residues long. Four fairly similar matches have been obtained. The more interesting one has 554 matching C_α pairs with the main continuous matching fragments being 5–83 and 87–91 in both model (**1lfg**) and target (**1lfh**) molecules; 235–237 in the model with 197–199 in the target, and 250–274, 285–292, 295–302 in both. Clearly, two large fragments of the two proteins match each other, 5–91 and 250–691. The two obvious candidates to be the hinge are the 91–st C_α atom and the 250-th C_α atom of **1lfg**. However, if we put the hinge at the 91-st C_α atom we divide the model molecule into two parts such that in one part we have two domains of essentially different sizes (249–90 = 159 residues versus 691–250 = 441 residues, i.e., one domain is 2.8 times larger than the other). In this case the larger part provides substantially more votes to candidate reference frames during the recognition stage of the algorithm. This implies that during high-scoring candidate reference frame selection, the votes for the smaller part will be largely overlooked. On the other hand, if we put the hinge at the 250-th C_α atom of **1lfg**, we divide the model molecule into two parts in a way such that in one part we again obtain two domains of different sizes. But this time the difference is appreciably smaller (one domain is only 1.8 times larger than the other). Putting the hinge at the 250-th C_α atom of **1lfg**, seven matches are obtained. These matches may be divided into two, fairly similar groups:

1. The number of matching pairs of C_α atoms is 600. The main continuous matching fragments are 91–140, 143–280, 283–292, 295–330, 333–417, and 422–691 in both model (**1lfg**) and target (**1lfh**) molecules. To optimally match **1lfh**, the two parts of **1lfg** have been rotated with respect to each other by around 57.5 degrees. The

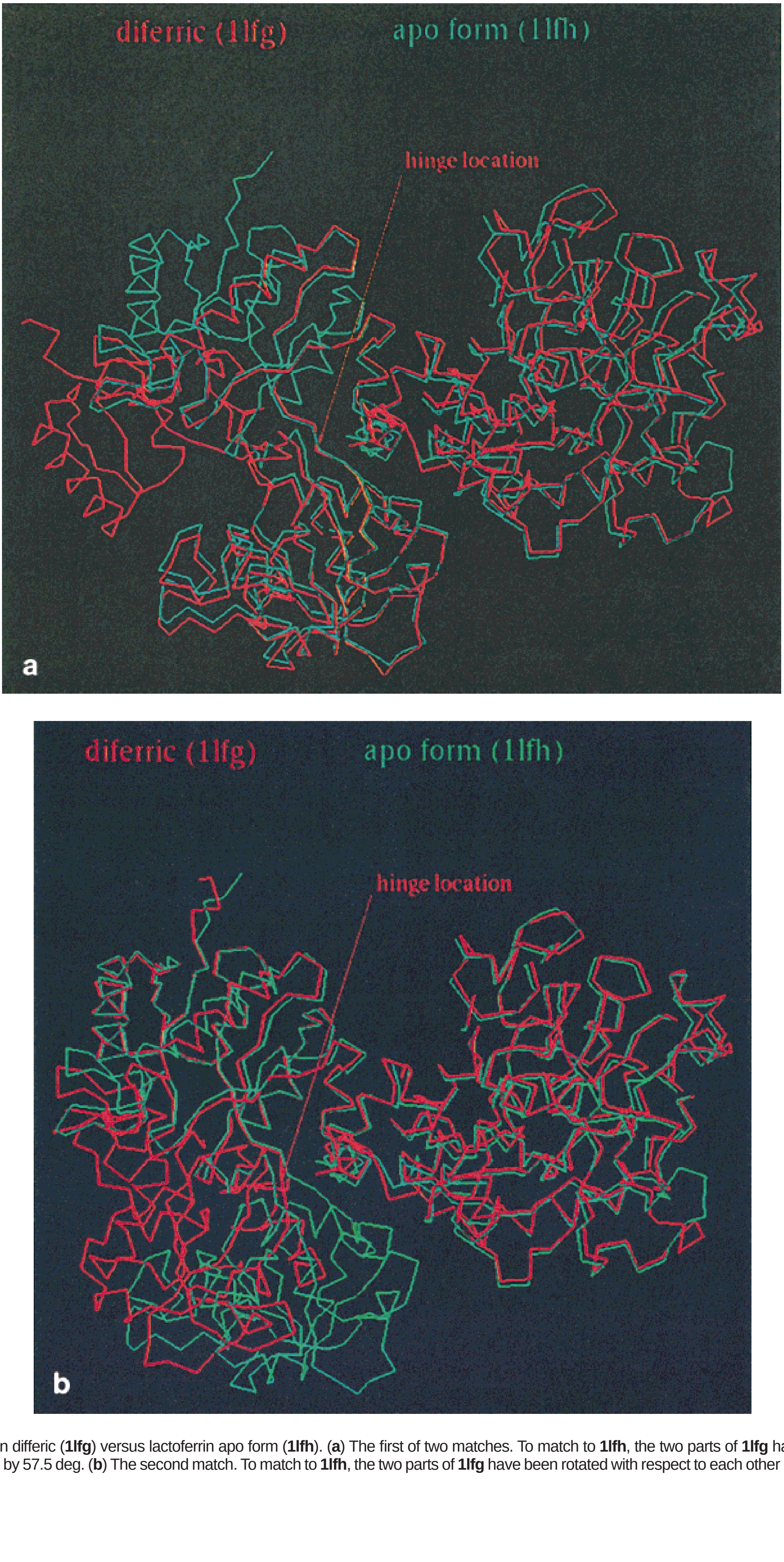

Fig. 6. Lactoferrin differic (**1lfg**) versus lactoferrin apo form (**1lfh**). (**a**) The first of two matches. To match to **1lfh**, the two parts of **1lfg** have been rotated with respect to each other by 57.5 deg. (**b**) The second match. To match to **1lfh**, the two parts of **1lfg** have been rotated with respect to each other by 7.68 deg.

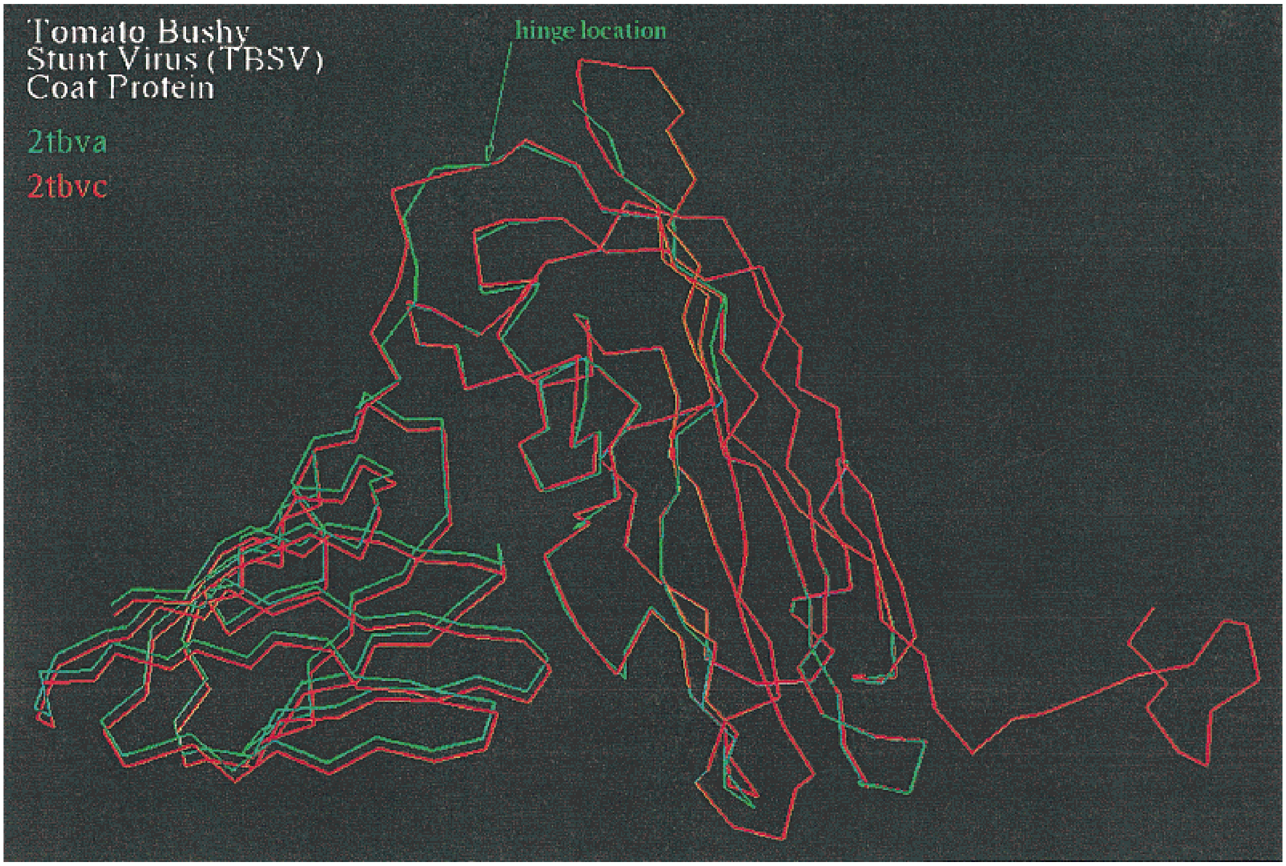

Fig. 7. Chain A of tomato bushy stunt virus (**2tbv**) versus chain C of the same protein. To match to chain C, the two parts of chain A have been rotated with respect to each other by 21.53 deg.

RMSD between the transformed model and the target molecule is 1.11 Å.

2. The number of matching pairs of C_α atoms is 563. The main continuous matching fragments are 5–91, 250–264, 266–272, 285–330, 332–417, and 422–691 in both molecules. To match **1lfh**, the two parts of **1lfg** are rotated with respect to each other by around 7.7 deg. The RMSD between the transformed model molecule and the target molecule is 1.16 Å.

Putting the hinge at the 250-th C_α atom of **1lfg** has shown that these two molecules are comprised of three similar domains: 5–91 (second match), 91–250 (first match) and 250–691 (both matches). Hence, one needs to put two hinges, at the 91-st C_α atom and at the 250–th C_α atom of **1lfg** to obtain a perfect match of these two proteins. The two matches are displayed in Figure 6a,b.

(f) Motion in T4 lysozyme mutants: Ile3 → Pro and Met6 → Ile

We compare a lysozyme mutant with ile3 replaced by pro (I3P) (space group P 32 2 1) (**1l96**) with lysozyme mutant (space group P 21 21 2) (**1l97**) (not shown). We choose **1l97** to be the model protein and **1l96** to be the target protein. Again, we put the hinge at the arbitrarily chosen 71–st C_α atom of the model molecule **1l97. 1l97** is comprised of two chains, each containing 164 residues. **1l96** is 162 residues long. The best of the five matches we have obtained has 157 matching pairs of C_α atoms with the main continuous matching fragments being 12–162 in both molecules. To match **1l96** the two parts of **1l97** have been rotated with respect to each other by about 29.6 deg. The RMSD between the transformed model molecule and the target molecule is 1.19 Å. The results suggest that the first 12 C_α atoms of **1l96** match those of **1l97**. To verify it we put the hinge at the twelfth C_α atom of **1l97**. The two most interesting results are:

1. The number of matching pairs of C_α atoms of the compared molecules is 129. The main continuous matching fragments are 1–14, and 74–162 in both molecules. To match **1l96**, the two parts of **1l97** have been rotated with respect to each other by 13.8 deg.
2. The number of matching pairs of C_α atoms of the compared molecules is 108. The main continuous matching fragments are 1–82 in both molecules. To match **1l96**, the two parts of **1l97** are rotated with respect to each other by 27.3 deg.

These results confirm our assumption. Hence, to obtain a perfect match between **1l96** and **1l97** we have to insert two hinges at the twelfth C_α and at the 71–st C_α atoms of **1l97**.

(g) Motion in tomato bushy stunt virus (TBSV) coat protein

We compare chains A and C of tomato bushy stunt virus (**2tbv**). We choose chain A to be the model protein and chain C to be the target protein. Chain A (**2tbva**) of tomato bushy stunt virus has 287 residues and chain C (**2tbvc**) has 321. The best match has been obtained when the hinge was placed at the 165–th C_α atom of **2tbva**, with 284 matching C_α pairs. The main continuous matching fragments are

Model(**2tbva**)	:	1–168,	170–172,	175–243,	245–287
Target(**2tbvc**)	:	36–203,	205–207,	210–278,	280–322

To match **2tbvc**, the two parts of **2tbva** are rotated with respect to each other by 21.5 deg. The RMSD between the transformed model molecule and the target is 0.77 Å (see Figure 7).

(h) Motion in diphtheria toxin

We compare monomeric diphtheria toxin (**1mdt**) with dimeric diphtheria toxin (**1ddt**). **1mdt** is the model protein and **1ddt** is the target. Monomeric diphtheria toxin (**1mdt**) is comprised of two chains each 523 residues long. Dimeric diphtheria toxin (**1ddt**) is 523 residues long. The best match was obtained when the hinge was placed at the 373-th C_α atom of **1mdt**. The number of matching C_α pairs is 509. The main continuous matching fragments are 1–187, 189–338, 340–367, 376–397, 399–426, 429–484, 491–510, and 514–523 in both model (**1mdt**) and target (**1ddt**) molecules.

To match **1ddt**, the two parts of **1mdt** have been rotated with respect to each other by about 179.3 deg. The RMSD between the transformed model molecule and the target molecule is 1.16 Å (Fig. 8).

FURTHER COMMENTS

Here we make some further comments about the applicability and performance of the hinge-bending flexible structural comparison method.

First, how critical is the choice of the hinge position? As hinges are often imprecisely known, errors in the assignment of the hinge can be expected. Hence this question is highly relevant. Certainly, if the exact location of the hinge is known in advance, predefining it results in an efficient, straightforward matching. However, even in the absence of such exact knowledge we were able to obtain the hinge-bending flexible matching. We have experimented with shifting of the hinges three C_α's to the left or to the right of the “correct” hinge, and still obtained matching, although with a deterioration of the quality of the superposition. However, this initial rough identification may be followed by a refinement of the hinge by successive applications around this site. Second, if the location of the hinge is unknown, the speed of the program allows repeated applications, systematically varying the hinge position. Since a large number of C_α atoms can be tested as potential hinge sites, an a priori knowledge of the hinge location is not a prerequisite. A refinement of the hinge around a trial hinge can follow. Third, by iteratively scanning the model molecule for many trial hinge-sites, and inspecting the quality of the obtained superpositions, several hinge sites can be located.

In the applications presented here similar molecules have been used. The ability of the program to handle some noise is shown by the matching of both crystal and NMR (**2bbm**) structures. Currently we are carrying out extensive database analysis. We automatically scan the C_α atoms in model proteins, iteratively carrying out the matching with a large number of proteins in the set. A particularly interesting set of cases which we are examining are the cytokine receptor superfamily. A straightforward match obtained by the program is of 1a21, the extracellular domain of the rabbit tissue factor and 1hwh, a human growth hormone mutant. The first part was matched with an RMSD of 2.29 Å and the second with an RMSD of 2.0 Å. Similarly, 1a21 was matched with a domain hinge-bending movement with 1hwg.

As with rigid structure comparisons, an inherent limitation of such comparisons is not always being able to judge the biological meaning of an obtained geometrical match of C_α atom-pairs between two proteins. Here, in addition, each of the parts should be large enough to obtain meaningful results. We have not experimented thoroughly enough with the program to uniquely define what is a “large enough” part.

CONCLUSIONS

Here we have presented a powerful novel approach for automatically matching protein molecules, enabling domain, or subdomain swiveling. Since domain motions are known to take place, the existence of such a methodology is very useful. Whereas to date searches for motifs have been strictly rigid-body ones, the existence of such a technique enables instituting searches for hinge–bent motifs. Well known examples are domain (or, part) swapping between meta-stable conformations. Hence, such a technique can be utilized to look for similar configurations, differing only by the introduction of a swiveling hinge, as might be the case in misfolded proteins. Superimposing the two structures can reveal the residues contributing to this conformational flip.

While this method is not as fast as its rigid-body counterpart, it is still very efficient. For the protein–protein cases we have examined, it took between 17 to 148 seconds on an Indigo 2 SGI workstation. In addition, it still possesses all the attributes of our previously presented methods, i.e., it is independent of the order of the residues on the polypeptide chain, and hence disregards insertions, deletions and changes in chain directionality.

This approach is general. It can be utilized to compare drugs, in searches for (flexible) pharmacophores, or motifs in RNA structures. It can be straightforwardly implemented to enable several, simultaneous hinges, and to a

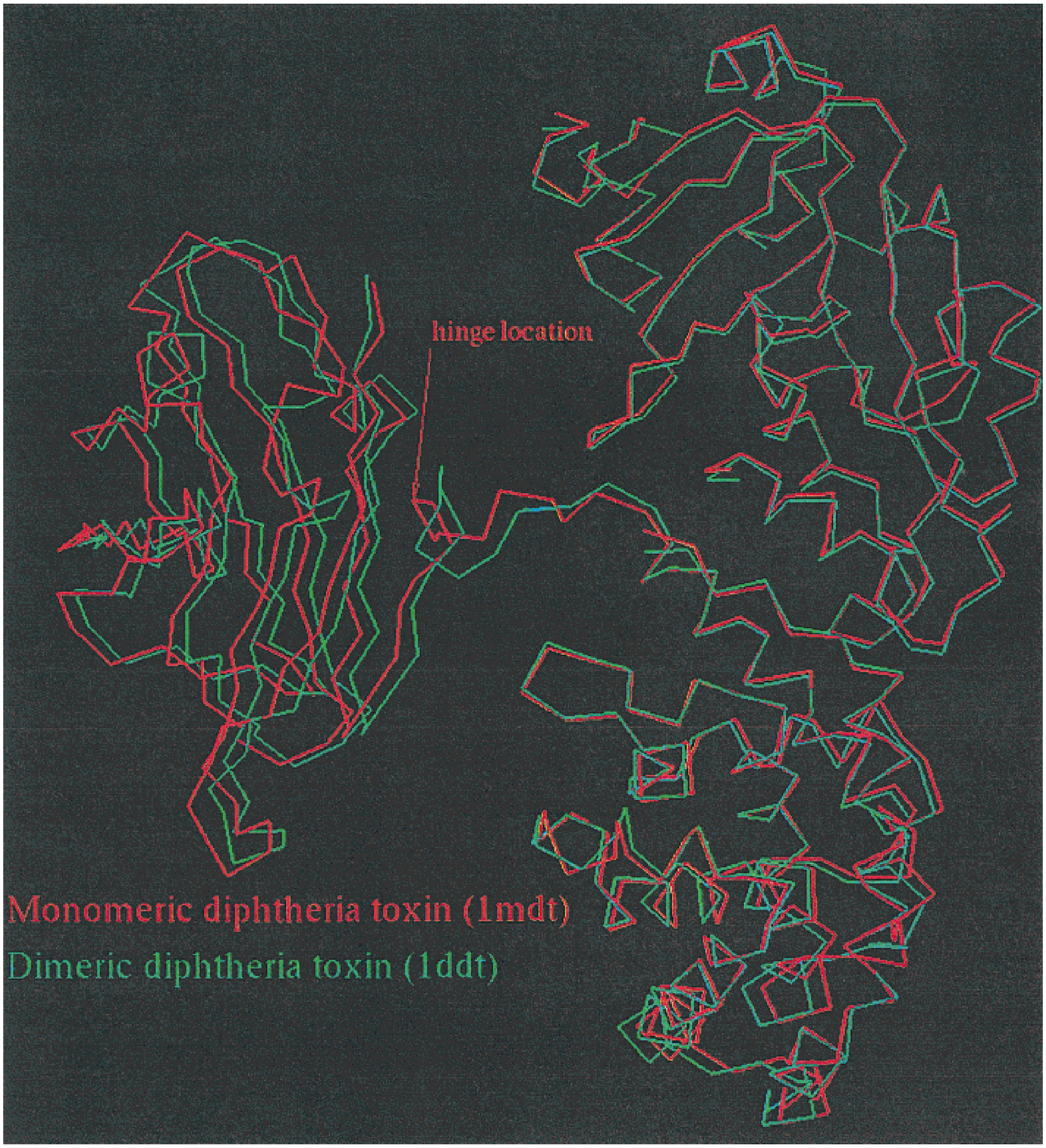
hinge location
Monomeric diphtheria toxin (1mdt)
Dimeric diphtheria toxin (1ddt)

Fig. 8. Monomeric diphtheria toxin (**1mdt**) versus dimeric diphtheria toxin (**1ddt**). To match to **1ddt**, the two parts of **1mdt** have been rotated with respect to each other by 179.27 deg.

comparison of a model to a database of molecules, in searches for a recurring, hinge-bent motif. There is no need to predefine the motif, nor the angular rotation. Both of these would be found automatically, *after* the transformations have been computed. Hence, we avoid the extremely time-consuming conformational searches through 3-D space, making it an especially attractive tool.

ACKNOWLEDGEMENTS

We thank Dr. J.V. Maizel for discussions, and for encouragement. The research of R. Nussinov has been funded in part with Federal funds from the National Cancer Institute, National Institutes of Health, under Contract Number NO1-CO 56000. The content of this publication does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products, or organizations imply endorsement by the U.S. Government. The research of R. Nussinov and H. Wolfson in Israel has been supported in part by grant No. 95-00208 from the BSF, Israel, by a grant from the *Israel Science Foundation* administered by the *Israel Academy of Sciences*, by the Israel Ministry of Science, by the Magnet fund, administered by the Israel Ministry of Industry and Trade, by the Tel Aviv University

Basic Research and Adams Brain Center grants. The research of H. Wolfson is partially supported by the Hermann Minkowski–Minerva Center for Geometry at Tel Aviv University.

APPENDIX A

Parameters Definition

The program is written in C++.

Parameters

Here we provide a detailed description of each of the parameters in the program.

neighborhood_radius During the recognition stage we visit not only the R-Table entry with the computed address, but also its neighbors provided **neighborhood_radius** is greater than 0. This is done since the matching cannot be expected to recur precisely. Setting **neighborhood_radius** to value greater than 1, affects the complexity of the algorithm.

index_difference To reduce the complexity of the algorithm we impose additional constraints on the triple of C_α atoms comprising the frame-invariant. One of these constraints requires that the difference between C_α atoms indices be less than or equal to an **index_difference.**

research_radius In some cases it may be useful to limit the set of C_α atoms used in the frame-invariant definition by considering only those which are fairly close to the chosen hinge. This is controlled by the **research_radius** parameter.

max_coordinate This parameter, together with the next one, defines the 3-D space of the program. The 3-D space is a set of all points (x, y, z) in 3-D space which meets the following conditions

$$\textbf{min\_coordinate} \le x, y, z \le \textbf{max\_coordinate}$$

min_coordinate This parameter, together with the previous one, defines the 3-D space for the program.

max_triangle_side_length One of the conditions for a triple of C_α atoms to be a frame-invariant is that the edges' lengths of the triangle they form are less than or equal to **max_triangle_side_length.**

min_triangle_side_length One of the conditions for a triple of C_α atoms to be a frame-invariant is that the edges' lengths of the triangle they form are greater or equal to **min_triangle_side_length.**

matching_parameter One of the conditions in the definition of a matching pair of C_α atoms where one belongs to the transformed model molecule and the second to the target molecule, is that the distance between them is less than or equal to a **matching_parameter.**

epsilon Two coordinate frames are defined to be identical if the distance between their x-axes, and between their y-axes is less than **epsilon.**

upper_coefficient To reduce the number of candidates, during the computation of the high scoring candidate reference center locations, we reject all candidate reference center locations whose votes score is less than the best votes score multiplied by **upper_coefficient.**

lower_coefficient When the high scoring candidate reference center locations are being selected, to reduce the complexity of the algorithm we keep track of M, the maximal votes score among all the candidate reference center locations which have been encountered so far. We insert the candidate reference center location in the heap of the best candidate reference center locations only if its votes score is greater than M multiplied by **lower_coefficient.**

candidate_joint_location_list_size Defines the size of the heap where the high-scoring candidate reference center locations are stored during the high-scoring candidate reference center locations selection.

best_matching_values_coefficient During verification of the high-scoring candidate reference center location, we apply appropriate coordinate transformation on the corresponding parts of the model molecule and compare the resulting transformed molecule with the target one. We also keep track of the maximal number of matching pairs, for the high scoring candidate reference center locations. If for a high-scoring candidate reference center location the number of matching pairs is greater than this maximal number multiplied by **best_matching_values_coefficient,** we shift the transformed model molecule in different directions in 3-D space in an attempt to improve the result. There may be several high-scoring candidate reference center locations which have been verified. We reject those which obtain relatively few matching pairs. Namely, if the number of matching pairs for some model molecule is less than the maximal number of matching pairs for this molecule multiplied by **matching_value_coefficient,** then this candidate reference center location is considered as inadequate in the verification stage.

cluster_radius During the clustering of high-scoring candidate reference frame locations, we go through the list of high-scoring candidate reference center locations. For each one, we compute the average of the origins and axes vectors. Only high-scoring candidate reference center locations with origins and axes vectors sufficiently close to the considered candidate reference center location are taken into account. The proximity of the origins is controlled by **cluster_radius.**

cluster_of_clusters_radius Following the clustering of high scoring candidate reference frame locations, we go through the clusters and reject those having another high-scoring candidate reference frame location with a higher votes score, and with an origin within the radius **cluster_of_clusters_radius.**

store_results By default, the program outputs results on the screen. If a user sets **store_results** parameter, the results will also be written to the file ⟨target_molecule_name⟩.drs in the directory specified by **outputdir.**

store_detailed_results If a user set **store_results** parameter, the more detailed version of results will be written to the file ⟨target_molecule_name⟩.drs in the directory specified by **outputdir.** The difference between detailed and concise results is that in the first a user receives also the coordinates of C_α atoms of the transformed model

Parameter	Default value	Note
filename_length	80	
db	db	
scene	scene	
pdbdir	pdb/	Slash at the end is mandatory.
outputdir	res/	Slash at the end is mandatory. The directory has to be created before the program invocation.
neighborhood_radius	1	It is not recommended to change this value.
index_difference	6	
research_radius	50	
max_coordinate	100	
min_coordinate	−50	
max_triangle_side_length	15	
min_triangle_side_length	3.9	Heavily affects the performance of the program.
matching_parameter	3	
epsilon	0.3	
upper_coefficient	0.5	
lower_coefficient	0.001	
candidate_joint_location_list_size	1500	
best_matching_values_coefficient	0.4	
matching_value_coefficient	0.3	
cluster_radius	1.5	
cluster_of_clusters_radius	0.2	
store_results	1	
store_detailed_results	0	
max_number_of_interest_features	1200	
resolution	2	
self_identification	1	
CreateTransfPDBFile	0	

molecules and their matching C_α atoms of the target molecule.

max_number_of_interest_features Defines the maximal size of the molecules (number of C_α atoms).

resolution Used to control the size of a bin in the SpaceNet hash table.

self_identification If some molecule is in both *database file* and *scene file*, the user can either take into account or ignore the molecule in the database when this molecule is run as a target molecule against the database. This is done by setting this parameter on or off.

Parameters' Default Values

Here we present the list of all the program's parameters' default values. These parameters enable to customize the program to meet the different requirements which may arise by consideration of different objects. (See Appendix table.)

REFERENCES

1. Murthy MRN. A fast method of comparing protein structures. FEBS Lett 1984;168:151–166.
2. Matthews BW, Rossman MG. Meth Enzymol 1985;115:397–420.
3. Abagyan RA, Maiorov VN. A simple qualitative representation of polypeptide chain folds: Comparison of protein tertiary structures. J Biomol Struct Dyn 1988;5:1267–1279.
4. Richards FM, Kundrot CE. Identification of structural motifs from protein coordinate data: Secondary structure and first-level supersecondary structure. Proteins 1988;3:71–81.
5. Zuker M, Somorjai RL. The alignment of protein structures in three dimensions. Bull Math Biol 1989;51:55–78.
6. Taylor WR, Orengo CA. Protein structure alignment. J Mol Biol 1989;208:1–22.
7. Sali TL, Blundell A. Definition of general topological equivalence in protein structures, a procedure involving comparison of properties and relationships through simulated annealing and dynamic programming. J Mol Biol 1990;212:403–428.
8. Mitchell EM, Artymiuk PJ, Rice DW, Willet P. Techniques derived from graph theory to compare secondary structure motifs in proteins. J Mol Biol 1990;212:151–166.
9. Vriend G, Sander C. Detection of common three-dimensional substructures in proteins. Proteins 1991;11:52–58.
10. Alexandrove NN, Takahashi K, Go N. Common spatial arrangements of backbone fragments in homologous and non-homologous proteins. J Mol Biol 1992;225:5–9.
11. Koch I, Kaden F, Selbig J. Analysis of protein sheet topologies by graph theoretical methods. Proteins 1992;12:314–323.
12. Grindley HM, Artymiuk PJ, Rice W, Willett P. Identification of tertiary structure resemblance in proteins using a maximal common subgraph isomorphism algorithm. J Mol Biol 1993;229: 707–721.
13. Artymiuk PJ, Poirrette AR, Grindley HM, Rice DW, Willet P. A graph-theoretic approach to the identification of three-dimensional patterns of amino acid side-chains in protein structures. J Mol Biol 1994;243:327–344.
14. Fischer D, Wolfson H, Lin SL, Nussinov R. Three-dimensional, sequence order-independent structural comparison of a serine protease against the crystallographic database reveals active site similarities: Potential implications to evolution and to protein folding. Protein Sci 1994;3:769–778.
15. Tsai C-J, Lin SL, Wolfson H, Nussinov R. A dataset of protein-protein interfaces generated with a sequence-order-independent comparison technique. J Mol Biol 1996;260:604–620.
16. Alesker V, Nussinov R, Wolfson H. Detection of non-topological motifs in protein structures. Protein Eng 1996;9:1103–1119.
17. Faber HR, Matthews BW. A mutant T4 lysozyme displays five different crystal conformations. Nature 1990;348:263–266.

18. Dixon MM, Nicholson H, Shewchuk L, Baase WA, Matthews BW. Structure of a Hinge-Bending Bacteriophage T4 Lysozyme Mutant, Ile3Pro. J Mol Biol 1992;227:917–933.
19. Gerstein M, Lesk AM, Chothia C. Structural mechanism for domain movements in proteins. Biochemistry 1994;33:6739–6749.
20. Bennett MJ, Choe S, Eisenberg D. Domain swapping: Entangling alliances between proteins. Proc Natl Acad Sci USA 1994;91:3127–3131.
21. Nussinov R, Wolfson H. Efficient detection of motifs in biological macromolecules by computer vision techniques. Proc Natl Acad Sci USA 1991;88:10495–10499.
22. Wolfson HJ. Generalizing the generalized Hough transform. Pattern Recog Lett 1991;12:565–573.
23. Ballard DH. Generalizing the Hough transform to detect arbitrary shapes. Pattern Recognition 1981;13:111–122.
24. Lamdan Y, Wolfson HJ. Geometric hashing: A general and efficient model-based recognition scheme. In: Bajcy, Ullman, editors. Proceedings of the IEEE International Conference on Computer Vision. Washington, D.C.: IEEE Computer Society Press; 1988. p 238–249.
25. Bernstein FC, Koetzle TF, Williams GJB, et al. The protein data bank: A computer-based archival file for macromolecular structures. J Mol Biol 1977;112:535–542.
26. Sandak B, Nussinov R, Wolfson HJ. An Automated Computer Vision and Robotics-Based Technique. Comput Appl Biosci (CABIOS) 1995;11:87–99.
27. Sandak B, Wolfson HJ, Nussinov R. Flexible docking allowing induced fit in proteins: Insights from an open to closed conformational isomers. Proteins 1998;32:159–174.
28. Wriggers W, Schulten K. Protein domain movements: Detection of rigid domains and visualization of hinges in comparisons of atomic coordinates. Proteins 1997;29:1–14.
29. Cormen TH, Leiserson CE, Rivest RL. Introduction to Algorithms. Cambridge, MA: MIT Press; 1990. 1028 p.
30. Bachar O, Fischer D, Nussinov R, Wolfson HJ. A computer vision based technique for 3-D sequence independent structural comparison of proteins. Protein Eng 1993;6:279–288.
31. Ikura M, Clore GM, Gronenborn AM, Zhu G, Klee CB, Bax A. Solution structure of a calmodulin-target peptide complex by multidimensional NMR. Science 1992;256:632–644.