Fast Hinge Detection Algorithms for Flexible Protein Structures

Tetsuo Shibuya

***Abstract*— Analysis of conformational changes is one of the keys to the understanding of protein functions and interactions. For the analysis, we often compare two protein structures, taking flexible regions like hinge regions into consideration. The RMSD (Root Mean Square Deviation) is the most popular measure for comparing two protein structures, but it is only for rigid structures without hinge regions. In this paper, we propose a new measure called RMSDh (Root Mean Square Deviation considering hinges) and its variant RMSDh$^{(k)}$ for comparing two flexible proteins with hinge regions. We also propose novel efficient algorithms for computing them, which can detect the hinge positions at the same time. The RMSDh is suitable for cases where there is one small hinge region in each of the two target structures. The new algorithm for computing the RMSDh runs in linear time, which is the same as the time complexity for computing the RMSD and is faster than any of previous algorithms for hinge detection. The RMSDh$^{(k)}$ is designed for comparing structures with more than one hinge region. The RMSDh$^{(k)}$ measure considers at most k small hinge region, *i.e.*, the RMSDh$^{(k)}$ value should be small if the two structures are similar except for at most k hinge regions. To compute the value, we propose an $O(kn^2)$-time and $O(n)$-space algorithm based on a new dynamic programming technique. With the same computational time and space, we can enumerate the predicted hinge positions. We also test our algorithms against actual flexible protein structures, and show that the hinge positions can be correctly detected by our algorithms.**

***Index Terms*— algorithm, protein hinge detection, protein 3-D structure comparison, dynamic programming**

I. Introduction

Proteins play enormous variety of roles in living systems. The functions of the proteins are said to be determined by their 3-D structures, and consequently the analysis of protein structures is one of the most important research topics in molecular biology. The analysis of protein structures often starts with a comparison of two similar structures, and there have been proposed a tremendous number of methods to compare two protein 3-D structures [7], [19], [28]. Structure comparison algorithms can be categorized into two types: rigid structure comparison methods and flexible structure comparison methods. The former methods consider protein structures as rigid bodies. But there are many proteins whose structures change conformationally. Most of their structures can be divided into several (almost) rigid substructures separated by small flexible parts (which often consists of only one atom) called hinge regions or just 'hinges' (Figure 1).[1] They change their structures by rotating around the hinge, due to their physical conditions, relations to other proteins, or some point mutations. The latter flexible structure comparison methods take hinges into consideration when they compare structures. The hinges sometimes take very important roles for their functions [28]. Due to the importance of the roles of these flexible proteins, the number of the entries in the databases of such flexible proteins increases rapidly [5], [9], [11], [23].

There are three tasks when we compare two flexible structures. At first we have to find the correspondence between atoms. We next have to find locations of hinges and finally we have to calculate superposition for each rigid fragment. But if we have determined the corre-

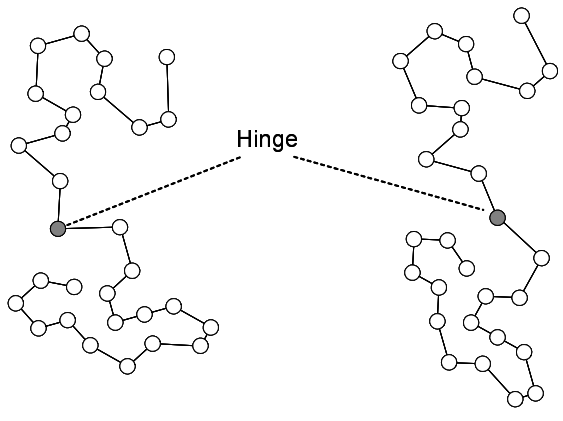


Fig. 1. Hinge bending of a protein. A protein sometimes changes its structure by rotating around an atom, which is called a hinge.

T. Shibuya is with Human Genome Center, Institute of Medical Science, University of Tokyo

[1]Notice that the 'hinge region' is different from the notion called 'hinge domain'.

spondence and the hinge positions, it is not difficult to compute the superposition. Thus, flexible structure comparison methods can be categorized into two types. One is a type of methods that does everything — they find the atom correspondence, the hinge positions, and the superposition simultaneously [4], [25], [30]. The other type of methods is dedicated to only hinge detection and calculation of superposition [3], [15], [20], [21], [29], assuming that the atom correspondence is given. The methods of the former type are more general than those of the latter, but they are definitely more difficult. Note that there are many situations in which only the latter methods are needed. For example, we always know the atom correspondence in two structures of the same, or significantly similar proteins. In case their amino acid sequences are similar, we can easily find the correspondence by using the ordinary sequence alignment. Our algorithms proposed in this paper are the methods of the latter type, and we do not deal with how to find the atom correspondence. Note that there is a third approach for hinge detection which predicts hinge positions from a single structure without comparing with other structures [8], [10], unlike the above comparison-based methods. But we do not deal with them in this paper.

When we compare two structures (by either of the two approaches), some appropriate scoring measure is desired. The measure must be mathematically clear and moreover easy to compute. The RMSD (Root Mean Square Deviation) [1], [6], [17], [18], [24] is the most commonly used measure for comparing two rigid structures (see section II for details). It is defined very clearly and can be computed very efficiently (in linear time). But it is designed only for rigid structures. There are no standard measures to be optimized for flexible structure comparison, as it seems very difficult to design a measure that can be efficiently computed.

In this paper, we propose measures for comparing flexible protein structures, and fast algorithms to compute them. With either of the algorithms, we can obtain the predicted positions of the hinges at the same time. We first propose a measure called the RMSDh (Root Mean Square Deviation considering hinges), which is a generalization of the RMSD with consideration of a single hinge region. We also propose an algorithm that computes the RMSDh in linear time, which is the same as the time complexity for computing the RMSD, even though our algorithm detects the hinge position at the same time. It is much faster than any previous hinge detection algorithms, which require at least quadratic time. We then generalize the RMSDh for proteins with at most k hinges, and call the generalized measure the RMSDh$^{(k)}$. We propose an $O(kn^2)$-time and $O(n)$-space algorithm for computing it, where n is the length of the structures to be compared. We will also show that we can detect the hinge positions with the same time and space complexity, by using a divide-and-conquer technique.

In these algorithms, we assume that each hinge region consist of only one atom (residue). Precisely, if the hinge region consists of a single atom, the angles of the rotation is limited due to physical/chemical restrictions. On the other hand, the limitation can be ignored if we consider hinge regions with several residues. In this paper, we assume that each hinge region consist of only one atom, but we ignore the limitation. By doing so, we can simplify the problem and fast algorithms can be designed as shown in later sections. Moreover, the algorithms can be used as a heuristic algorithm for proteins with non-single residue hinges, as shown in the computational experiments in section V.

Availability: FastHinge 1.0, the program developed for this research, can be downloaded from http://www.hgc.jp/˜tshibuya/softwares/. Currently, the program runs only on Windows or SunOS.

Organization of this paper: In section II, we present the definition of the RMSD and algorithms for computing it as preliminaries. Then we propose the new RMSDh measure and algorithms for it in section III. We propose the RMSDh$^{(k)}$ measure and algorithms for it in section IV. In section V, we show computational experiments. Finally in section VI, we conclude our results and discuss future work.

II. Preliminaries

A. RMSD: The Root Mean Square Deviation

A protein 3-D structure can be represented by various ways, but one popular way is to represent it by a list of 3-D coordinates of its backbone C_α atoms. The RMSD (root mean square deviation) [1], [6], [17], [18], [24] is the most common way to compare two lists of 3-D coordinates.

Let the two sets of points (*i.e.,* protein structures) to be compared be $\mathbf{P} = \{\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_n\}$ and $\mathbf{Q} = \{\vec{q}_1, \vec{q}_2, \ldots, \vec{q}_n\}$, where $\vec{p}_i$ and $\vec{q}_i$ are the coordinates of the i-th C_α atoms of $\mathbf{P}$ and $\mathbf{Q}$, respectively. Then the RMSD between $\mathbf{P}$ and $\mathbf{Q}$ is defined as the minimum value of

$$D_{R,\vec{v}}(\mathbf{P}, \mathbf{Q}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \|\vec{p}_i - (R \cdot \vec{q}_i + \vec{v})\|^2}$$

over all the possible rotation matrices R and translation vectors $\vec{v}$, where $\| \cdot \|$ denotes the norm. Let $RMSD(\mathbf{P}, \mathbf{Q})$ denote the minimum value, and let

$\hat{R}(\mathbf{P},\mathbf{Q})$ and $\hat{\vec{v}}(\mathbf{P},\mathbf{Q})$ denote the R and $\vec{v}$ that satisfy $D_{R,\vec{v}}(\mathbf{P},\mathbf{Q}) = RMSD(\mathbf{P},\mathbf{Q})$.

B. How to compute the RMSD

In this section, we briefly describe how to compute the RMSD. Let $R \cdot \mathbf{X}$ denote the structure $\mathbf{X}$ rotated by the rotation matrix R. If the rotation matrix R is fixed, $D_{R,\vec{v}}(\mathbf{P},\mathbf{Q})$ is known to be minimized when the centroid of $R \cdot \mathbf{Q}$ is translated to the centroid of $\mathbf{P}$ by the translation vector $\vec{v}$, regardless of what the rotation matrix R is. Hence, if both $\mathbf{P}$ and $\mathbf{Q}$ are translated so that their centroids are moved to the origin of the coordinates, the RMSD problem is reduced to a problem of finding R (*i.e.,* $\hat{R}(\mathbf{P},\mathbf{Q})$) that minimizes $F_R(\mathbf{P},\mathbf{Q}) = \sum_{i=1}^{n} \|\vec{p}_i - R \cdot \vec{q}_i\|^2$.

From now on, we consider that both structures have been already translated so that both centroids are moved to the origin of the coordinates. Then we can compute $\hat{R}(\mathbf{P},\mathbf{Q})$ in linear time [1], [17], [18] as follows. Let $H = \sum_{i=1}^{n} \vec{p}_i \cdot \vec{q}_i{}^t$, where $\vec{v}^t$ means the transpose of vector $\vec{v}$. Clearly, H can be computed in $O(n)$ time. Then $F_R(\mathbf{P},\mathbf{Q})$ can be described as $\sum_{i=1}^{n} (\vec{p}_i{}^t \vec{p}_i + \vec{q}_i{}^t \vec{q}_i) - trace(R \cdot H)$, and $trace(RH)$ is maximized when $R = VU^T$, where $U\Lambda V$ is the singular value decomposition (SVD) of H and A^T means the transpose of matrix A. Thus $\hat{R}(\mathbf{P},\mathbf{Q})$ can be obtained from H in constant time, as H is a 3×3 matrix and the SVD can be computed in $O(d^3)$ time for a $d \times d$ matrix [13]. Note that there are degenerate cases where $det(VU^T) = -1$, which means that VU^T is a reflection matrix. See [1], [6] for details to deal with the degenerate cases. We can compute the RMSD value in linear time once we have obtained $\hat{R}(\mathbf{P},\mathbf{Q})$. In total, we can compute the RMSD value in $O(n)$ time.

Let $\mathbf{S}[i..j]$ denote the substructure of $\mathbf{S}$ from the i-th atom to the j-th atom (*e.g.,* $\mathbf{P}[i..j] = \{\vec{p}_i, \vec{p}_{i+1}, \ldots, \vec{p}_j\}$). According to [26], the RMSD and corresponding superposition between two substructures $\mathbf{P}[i..j]$ and $\mathbf{Q}[i..j]$ can be computed in constant time for any i and j, after linear-time preprocessing, as follows: $RMSD(\mathbf{P}[i..j], \mathbf{Q}[i..j])$, $\hat{R}(\mathbf{P}[i..j], \mathbf{Q}[i..j])$ and $\hat{\vec{v}}(\mathbf{P}[i..j], \mathbf{Q}[i..j])$ can be computed in $O(1)$ time if we are given $\sum_{k=i}^{j} \vec{p}_k$, $\sum_{k=i}^{j} \vec{p}_k{}^t \vec{p}_k$, $\sum_{k=i}^{j} \vec{q}_k$, $\sum_{k=i}^{j} \vec{q}_k{}^t \vec{q}_k$, and $\sum_{k=i}^{j} \vec{p}_k \vec{q}_k{}^t$. These values can be computed also in constant time, if we compute the following values in advance: $\sum_{k=1}^{\ell} \vec{p}_k$, $\sum_{k=1}^{\ell} \vec{p}_k{}^t \vec{p}_k$, $\sum_{k=1}^{\ell} \vec{q}_k$, $\sum_{k=1}^{\ell} \vec{q}_k{}^t \vec{q}_k$, and $\sum_{k=1}^{\ell} \vec{p}_k \vec{q}_k{}^t$, for all ℓ ($1 \leq \ell \leq n$). It is easy to see that all of these values can be computed in $O(n)$ time in total. Thus we conclude that the RMSD and corresponding superposition between $\mathbf{P}[i..j]$, and $\mathbf{Q}[i..j]$ can be computed in $O(1)$ time after linear-time preprocessing.

III. RMSDh: A Linear-Time Computable Measure for Hinge Detection

A. Definition of the RMSDh

In this section, we consider a new measure to compare two flexible protein 3-D structures that are very similar except for one small hinge region. We consider that the hinge region is so small that it can be considered as only a single backbone atom.[2] Note that there are atoms other than the C_α atoms on the backbone, and the hinge can be located at any of them. Let the two structures to be compared be $\mathbf{P} = \{\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_n\}$ and $\mathbf{Q} = \{\vec{q}_1, \vec{q}_2, \ldots, \vec{q}_n\}$, and consider that the hinge is located at a backbone atom between the ℓ-th C_α atom and the $(\ell+1)$-th C_α atom, or at the ℓ-th C_α atom. Then $\mathbf{P}[1..\ell]$ and $\mathbf{Q}[1..\ell]$ should be similar to each other, and $\mathbf{P}[\ell+1..n]$ and $\mathbf{Q}[\ell+1..n]$ should also be similar to each other. Thus if the two rigid parts of $\mathbf{Q}$ are rotated and translated appropriately with different rotation matrices and translation vectors, $\mathbf{P}$ and the transformed $\mathbf{Q}$ should be very similar to each other, and consequently should have a small RMSD value. It means that

$$G_\ell(\mathbf{P},\mathbf{Q}) = \min_{R_1,R_2,\vec{v}_1,\vec{v}_2} \sqrt{\frac{1}{n}\{K_1^\ell(R_1,\vec{v}_1) + K_{\ell+1}^n(R_2,\vec{v}_2)\}}$$

must be a very small value, where

$$K_x^y(R,\vec{v}) = \sum_{i=x}^{y} \|\vec{p}_i - (R \cdot \vec{q}_i + \vec{v})\|^2.$$

Here, R_1 and R_2 are (possibly different) rotation matrices and $\vec{v}_1$ and $\vec{v}_2$ are (also possibly different) translation vectors. It can be used as the similarity measure between $\mathbf{P}$ and $\mathbf{Q}$, if the hinge is at or around the ℓ-th atom. This value is the same as the RMSD if $R_1 = R_2$ and $\vec{v}_1 = \vec{v}_2$ when they are optimized.

But we do not know the actual hinge position in most cases when we compare two structures. Hence, we consider the minimum value of $G_\ell(\mathbf{P},\mathbf{Q})$ over all the possible hinge positions ℓ, *i.e.* $\min_{1 \leq \ell < n} G_\ell(\mathbf{P},\mathbf{Q})$, as the measure to compare a pair of flexible structures with one hinge. We call it the RMSDh (Root Mean Square Deviation considering hinges), and let $RMSDh(\mathbf{P},\mathbf{Q})$ denote this value. Note that the RMSDh is always smaller than or equal to the RMSD. Let $\hat{\ell}$ denote the ℓ that minimizes $G_\ell(\mathbf{P},\mathbf{Q})$. Then we can predict that the hinge is located between the $\hat{\ell}$-th residue and the $(\hat{\ell}+1)$-th residue.

The above optimized translations and rotations does not consider any restrictions on the locations of $P[\ell]$,

[2] We do not consider longer hinge regions. But it does not mean that our algorithms cannot be applied to flexible proteins with longer hinge regions.

$Q[\ell]$, $P[\ell+1]$, and $Q[\ell+1]$, *i.e.* the distance between $P[\ell]$ and $P[\ell+1]$ and that between $Q[\ell]$ and $Q[\ell+1]$ should be fixed. But in the ideal case where the structures $P[1..\ell]$ and $Q[1..\ell]$ are all the same structure, and the structures $P[\ell+1..n]$ and $Q[\ell+1..n]$ are also all the same structure, we can ignore the restriction, as the above optimized RMSDh value becomes zero. In practice, these fragments structures should be very similar (almost the same) to each other, and thus we think that there is no problem in ignoring the restriction.

B. How to compute the RMSDh

The problem of computing $RMSDh(\mathbf{P}, \mathbf{Q})$ can be reduced to the problem computing

$$\min_{1 \leq \ell < n} L_1^{\ell}(\mathbf{P}, \mathbf{Q}) + L_{\ell+1}^{n}(\mathbf{P}, \mathbf{Q}),$$

where

$$L_x^y(\mathbf{P}, \mathbf{Q}) = \min_{R,v} \sum_{i=x}^{y} \|\vec{p}_i - (R \cdot \vec{q}_i + \vec{v})\|^2.$$

Notice that $L_i^j(\mathbf{P}, \mathbf{Q}) = n \cdot (RMSD(\mathbf{P}[i..j], \mathbf{Q}[i..j]))^2$, which means that we can compute the RMSDh value by computing $2n-2$ RMSD values, *i.e.*, $RMSD(\mathbf{P}[1..\ell], \mathbf{Q}[1..\ell])$ and $RMSD(\mathbf{P}[\ell+1..n], \mathbf{Q}[\ell+1..n])$ for all ℓ ($1 \leq \ell < n$). According to section II-B, the computation of each RMSD can be done in constant time after linear-time preprocessing. Hence, the RMSDh value can be computed in $O(n)$ time, including the preprocessing phase. Moreover, we can detect the corresponding hinge position at the same time.

IV. RMSDh$^{(k)}$: More Flexible Measures

A. Definition of the RMSDh$^{(k)}$

In the previous section, we considered only one hinge region, but many flexible protein structures are known to have more than one hinge region. In this section, we consider that the target structures have k hinges at most, which means they can be divided into $k+1$ rigid fragments. Again, let $\mathbf{P} = \{\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_n\}$ and $\mathbf{Q} = \{\vec{q}_1, \vec{q}_2, \ldots, \vec{q}_n\}$ be the flexible structures to be compared, and let the positions of the hinges be $\ell_1, \ell_2, \ldots, \ell_k$. To ease discussion, let $\ell_0 = 1$ and $\ell_{k+1} = n+1$. Then, with discussion similar to section III-A, the value

$$H_{\ell_1,\ldots,\ell_k}(\mathbf{P}, \mathbf{Q}) = \min_{R_0,\ldots,R_k,\vec{v}_0,\ldots,\vec{v}_k} \sqrt{\frac{1}{n} \sum_{j=0}^{k} K_{\ell_j}^{\ell_{j+1}-1}(R_j, \vec{v}_j)}$$

should be a very small value, where $K_x^y(R, \vec{v})$ denotes the expression defined in section III-A, $R_0, R_1, \ldots, R_k$ are possible rotation matrices, and $\vec{v}_0, \vec{v}_1, \ldots, \vec{v}_k$ are possible translation vectors.

But, as in section III-A, we do not know the actual hinge positions in most cases. Thus we propose to use the minimum value of the above expression over all the possible sets of k hinge positions $\{\ell_1, \ell_2, \ldots, \ell_k\}$, *i.e.*,

$$\min_{1 \leq \ell_1 < \ell_2 < \cdots < \ell_k \leq n} H_{\ell_1,\ldots,\ell_k}(\mathbf{P}, \mathbf{Q}),$$

as the measure for comparing such flexible proteins. We call it RMSDh$^{(k)}$, and let $RMSDh^{(k)}(\mathbf{P}, \mathbf{Q})$ denote the value. As in section III-A, the positions $\ell_1, \ell_2, \ldots, \ell_k$ that minimizes the above value can be used as the predicted locations of the hinges. Note that the RMSDh$^{(1)}$ is the same as the RMSDh. Note also that the RMSDh$^{(k)}$ is always smaller than or equal to the RMSDh$^{(k-1)}$, for any k, and that RMSDh$^{(n-1)} = 0$ no matter how different the proteins are.

B. How to Compute the RMSDh$^{(k)}$

As in the case of computing the RMSDh, the problem of computing the RMSDh$^{(k)}$ can be reduced to the problem of computing

$$\min_{1 \leq \ell_1 < \ell_2 < \cdots < \ell_k \leq n} \sum_{i=0}^{k} L_{\ell_i}^{\ell_{i+1}-1}(\mathbf{P}, \mathbf{Q}),$$

where $L_i^j(\mathbf{P}, \mathbf{Q})$ is the same expression as defined in section III-B. There are ${}_{n-1}C_k$ possible sets of k hinge positions, which means that we might need $O((k+1) \cdot {}_{n-1}C_k)$ time or more if we naively compute it. But, in the following, we propose an algorithm based on dynamic programming (DP), which compute the RMSDh$^{(k)}$ in $O(kn^2)$ time and $O(n)$ space.

To compute the RMSDh$^{(k)}$, we consider the following value:

$$I_{i,r} = \min_{1 \leq \ell_1 < \ell_2 < \cdots < \ell_r \leq i} \sum_{j=0}^{r} L_{\ell_j}^{\ell_{j+1}-1}(\mathbf{P}[1..i], \mathbf{Q}[1..i]),$$

where we let $\ell_0 = 1$ and $\ell_{r+1} = i+1$ to ease discussion. Then the RMSDh$^{(k)}$ is described as $(I_{n,k}/n)^{\frac{1}{2}}$. Note that $I_{i,r}$ is defined only when $0 \leq r < i \leq n$. In case $r = 0$, it can be easily seen from the definition that $I_{i,0} = L_1^i(\mathbf{P}, \mathbf{Q})$ for any i. In addition, the following equation holds when $r \geq 1$:

$$I_{i,r} = \min_{r \leq j < i} \{I_{j,r-1} + L_{j+1}^{i}(\mathbf{P}, \mathbf{Q})\}.$$

The above equation represents a DP algorithm for computing $I_{n,k}$ and consequently the RMSDh$^{(k)}$ (*i.e.*, $(I_{n,k}/n)^{\frac{1}{2}}$). During the DP procedure, we compute $I_{n,r}$ for all r ($1 \leq r \leq k$), from which we can immediately obtain the RMSDh$^{(r)}$ values ($1 \leq r \leq k$) too.

Recall that the $L_i^j(\mathbf{P}, \mathbf{Q})$ can be computed in constant time after linear-time ($O(n)$) preprocessing (see section III-B). Thus the values $I_{i,0}$ for all i can be computed in $O(n)$ time in total. Moreover, in case $r > 0$, the value $I_{i,r}$ can also be computed in $O(i - r)$ time by using the values of $I_{j,r-1}$ ($j < i$). It means that the overall computation time required for computing $I_{n,k}$ (and consequently the RMSDh$^{(k)}$) is $O(kn^2)$. The space required for computing the RMSDh$^{(k)}$ is only $O(n)$, because we only need the information of $I_{j,r-1}$ values (for all j such that $j < i$) to compute the $I_{i,r}$ values for any i.

To compute the positions of the corresponding hinges, we can use the ordinary tracing back technique for DP algorithms, without increasing the time complexity of the overall algorithm. But the space requirement increases to $O(nk)$ space, if we do it naively by using a table of $O(nk)$ size for tracing back. It can be reduced to $O(n)$ space by using a divide-and-conquer technique similar to the Hirschberg algorithm for sequence alignment [14], as follows.

Let

$$J_{i,r} = \min_{i \le \ell_1 < \ell_2 < \cdots < \ell_r \le n} \sum_{j=0}^{r} L_{\ell_j}^{\ell_{j+1}-1}(\mathbf{P}[i..n], \mathbf{Q}[i..n]),$$

where we let $\ell_0 = i$ and $\ell_{r+1} = n+1$ to ease discussion. $J_{i,r}$ can also be computed by DP, as the following equation holds:

$$J_{i,r} = \min_{i \le j < n} \{L_i^j(\mathbf{P}, \mathbf{Q}) + J_{j+1,r-1}\}.$$

Moreover the $I_{n,k}$ can be described as follows,[3] letting $k' = \lfloor k/2 \rfloor$ and $k'' = k - k' - 1$:

$$I_{n,k} = \min_{k' < i < n-k''} I_{i,k'} + J_{i+1,k''}.$$

The i that minimizes this value is the position of the $(k'+1)$-th hinge. Let the position be p. To compute it, we need $O(kn^2)$ time and $O(n)$ space.

Similarly, we can next compute the position of the $(\lfloor k'/2 \rfloor + 1)$-th hinge by computing the RMSDh$^{(k')}$ between $\mathbf{P}[1..p-1]$ and $\mathbf{Q}[1..p-1]$ in $O(k'p^2)$ time and $O(p)$ space. Moreover, we can also compute the position of the $(k'+1+\lfloor k''/2 \rfloor)$-th hinge by computing the RMSDh$^{(k'')}$ between $\mathbf{P}[p+1, n]$ and $\mathbf{Q}[p+1, n]$ in $O(k''(n-p)^2)$ time and $O(n-p)$ space. Notice that $k'p^2 + k''(n-p)^2 < kn^2/2$, and $kn^2 + kn^2/2 + kn^2/2^2 + \cdots < 2kn^2$. It means that we can compute all the hinge positions in $O(kn^2)$ time and $O(n)$ space by repeating the above until we obtain all of them.

[3]Once we get the $I_{n,k}$, we can immediately compute the RMSDh$^{(k)}$ value, as was described before, *i.e.*, $RMSDh^{(k)}(\mathbf{P}, \mathbf{Q}) = (I_{n,k}/n)^{\frac{1}{2}}$.

V. Experimental Results

We tested our algorithms against the 14 pairs of proteins shown in Table I, taken from the PDB database [2]. In the table, the RMSD column shows the RMSD between two structures in each set. The top 12 pairs of proteins in the table are flexible proteins. The adenosylcobinamide kinase [27] in the set AK is known to be a flexible protein with shearing movement. The HIV-1 protease [22] in the set HIV is the major drug target against the AIDS (acquired immunodeficiency syndrome), whose flexibility is said to affect the effectiveness of drugs. The lactate dehydrogenase (LDH) [12] is known to be a flexible protein with very dynamic movement, *i.e.*, more than 10Å. The other 9 flexible protein sets are pairs of flexible proteins listed in the Hinge Atlas database [11], which contains very accurate annotations on hinge positions. As for the sets other than the above 12 flexible proteins, the AT set consists of two independently determined structures of the same protein in the same state, and the MR set is a pair of unrelated proteins.

Tables II and III show the experimental results against the 12 flexible protein pairs. In the table, the 'Our results' rows show our results of the computation of the RMSDh$^{(k)}$ for $1 \le k \le 5$, and the corresponding hinge positions (by showing the fragments divided by the predicted hinge positions). See section V-A for the meaning of the mark '†' in these rows. The 'FlexProt' rows show the results of the FlexProt program [25] with the default parameter settings (*i.e.*, 3.0Å is set to the maximal RMSD between matched fragments, and 15 is set to the minimal size of matched fragments), which is one of the most widely used tools for detection of hinge positions. The 'RMSDh$^{(k)}$' column shows our RMSDh$^{(k)}$ values for 5 different ks ($k = 1, 2, \ldots, 5$). The 'Fragments divided by hinges' column shows the fragments divided by the predicted hinges (or annotated hinges). In the column, '$[x..y]$' denotes the fragment that starts at the x-th residue and ends at the y-th residue. '(rÅ)' written after the fragment means that the RMSD between the two fragments at the position is rÅ. Note that '(rÅ)' is omitted for the FlexProt results and the annotations. The fragments that end at (or near to) the annotated hinge positions are marked with '*' (*i.e.*, their distance to the annotated position are at most 3 residues).

The 'Annotation in ...' rows and the 'Hinge Atlas' rows show the protein pairs' known annotations in literature or the Hinge Atlas database. In the experiment for the set AK, HIV, and LDH, we show the annotations given in [27], [16] and [12], respectively. These annotations are based on manual analysis on the structures. For other

TABLE I
PROTEIN STRUCTURES TESTED IN OUR EXPERIMENTS.

Sets	Proteins	PDB IDs	#residues	RMSD (Å)
AK	Adenosylcobinamide kinase	1CBU(B),1C9K(B)	180	3.1092
HIV	HIV-1 protease	3HVP, 4HVP(A)	97	1.2450
LDH	Lactate dehydrogenase (LDH)	1LDM, 6LDH	329	1.7886
BTL	Bacteriophage T4 Lysozyme	149L, 1L53	164	1.8707
DPB	DNA Polymerase beta	1BPD, 2BPG	324	10.3347
EPA	Elastase of Pseudomonas Aeruginosa	1EZM, 1U4G	298	1.2194
ENL	Enolase	3ENL(A), 1EBG	436	1.4662
GB	Glutamine Binding Protein	1GGG(A), 1WDN (residues: 2–221)	220	5.3380
LF	Lactoferrin	1LFG, 1LFH	691	6.4285
LB	LAO Binding Protein	2LAO, 1LST	238	4.6985
RB	Ribose Binding Protein	1URP(A), 2DRI	271	4.0624
TC	Troponin C	4TNC(A, residues: 3–157), 2TN4	155	3.7263
AT	Asparatate transcarbamoylase	1RAB(A), 1RAC(A)	310	0.1714
MR	Myoglobin / Rhodopsin	101M, 1AYR(A) (residues:1–154)	154	19.3841

datasets, we show the annotations in the Hinge Atlas database. The Hinge Atlas annotations are made also by manual annotation, but utilizing many state-of-the-art hinge detection methods, such as the FlexProt.

A. Estimation of the Numbers of Hinges

In this section, we discuss how to estimate the number of hinges in the given pair of protein structures, and the accuracy of the method. In the tables II and III, we show the RMSD value for each pair of fragments divided by the predicted hinges.[4] As these fragments should be 'rigid' fragments, these RMSD values should be very small. Thus we can predict the number of hinges by finding the smallest k ($k \geq 1$) such that all the fragments have RMSDs smaller than some threshold. In the experiments, we used 1.5Å as the threshold. In tables II and III, we marked the predicted number of hinges with '†' in the 'Methods' column of the 'Our results' rows. In all the experiments except for the set ENL, the predicted numbers of hinges were at most 3. Note that it was 7 in the experiment for the ENL set, which is not shown in the table.

With the above method, we succeeded in predicting the number of hinges correctly (*i.e.* same as the annotations) for 9 of the 12 data sets, *i.e.*, HIV, LDH, DPB, EPA, GB, LF, LB, RB, and TC. On the other hand, the FlexProt program with the default parameter settings could predict them correctly only for 4 of the 12 sets, *i.e.*, AK, GB, LF and LB.

[4]Computation of these fragments' RMSDs requires only additional $O(k)$ time, by using the technique described in section II-B.

TABLE IV
RMSD/RMSDH/RMSDH$^{(k)}$ OF SETS AT AND MR (IN Å).

Sets	RMSD	RMSDh	RMSDh$^{(2)}$
AT	0.1714	0.1672	0.1555
MR	19.3841	15.9145	13.4793
Sets	RMSDh$^{(3)}$	RMSDh$^{(4)}$	RMSDh$^{(5)}$
AT	0.1508	0.1430	0.1380
MR	10.7877	9.3194	8.3340

B. Correctness of the Detected Hinge Positions

As for the hinge positions, we succeeded in predicting all the hinge positions correctly for 6 of the 12 sets, that is, LDH, EPA, GB, LF, LB, and RB, by using the estimated number of hinges described in the previous section. Moreover, we could succeeded in predicting some of the hinge positions correctly for other 3 sets (HIV, BTL, and TC). Our prediction was different from the annotations for the other 3 sets, *i.e.*, AK, HIV, and ENL. But even for these unsuccessful 3 sets, we can find some of the hinge positions correctly, if we set some different k.

On the other hand, the FlexProt (with the default parameters) predicted all the hinge positions correctly only for 1 of the 12 sets (*i.e.*, AK). The FlexProt predicted some of the hinge positions correctly for 4 sets (*i.e.*, DPB, GB, RB and TC). For the other 7 sets, the FlexProt could not predict any correct hinge positions with the default parameter settings.

C. Other experiments

We also computed the RMSD, RMSDh, and RMSDh$^{(k)}$ values for two independently determined structures of the same protein in the same state (*i.e.*,

TABLE V
TIME (SEC) FOR COMPUTING RMSDH$^{(k)}$ FOR ALL k ($1 \leq k < n$).

Sets	AK	HIV	LDH	BTL	DPB
Time (sec)	0.203	0.047	0.640	0.156	0.640
Sets	EPA	ENL	GB	LF	LB
Time (sec)	0.531	1.234	0.313	3.516	0.344
Sets	RB	TC	AT	MR	
Time (sec)	0.437	0.172	0.594	0.141	

the AT set), which are shown in Table IV. Note that the same set is used in [20] for a test on a pair of almost-the-same proteins. Of course all the values are very small, including the RMSD value. We consider that we do not have to compute RMSDh/RMSDh$^{(k)}$ values if the RMSD is very small like in this case.

In contrary, we compared two totally different structures in the set MR (*i.e.*, a myoglobin and a rhodopsin is compared). In this case, all the values are far larger than those in any other experiments. These experiments show that our measures are effective for discriminating flexible protein pairs from other normal rigid protein pairs, unless the two flexible proteins have a very similar structure.

D. Computation time

Table V shows the time for computing the RMSDh$^{(k)}$ values and corresponding hinge positions for all k ($1 \leq k < n$), using a single 3.2 GHz Pentium D processor with 2 GB memory. It shows that the computation time is very reasonable (most are less than a second), which is faster than any other known hinge detection algorithms. Note that the computation time is much smaller in case k is smaller.

VI. CONCLUSIONS AND FUTURE WORK

We proposed two new measures for comparing two flexible proteins, which can be very efficiently computed. The first measure, RMSDh, can be computed in linear time, while the other measure, RMSDh$^{(k)}$ can be computed in $O(kn^2)$ time, where k is the number of hinges, and n is the length of the structures. Moreover, we can detect the hinge positions while we compute the measures. Both of the measures are tested on actual flexible proteins to demonstrate the accuracy and the efficiency of our algorithms.

Some flexible proteins have large flexible regions. Our measure considers only small flexible regions, and some more flexible measure might be desired for structures with larger flexible regions. Moreover, our algorithms suppose that we know the correspondence between residues of two proteins in advance, but we do not in many cases. Thus, we should develop as future work a flexible structure alignment algorithm that finds the residue correspondence minimizing the RMSDh or the RMSDh$^{(k)}$.

Not so many 3-D structures of flexible proteins are solved today, but the number is now increasing. In the future, there should be more sample structures for every flexible protein, which means we should develop multiple alignment algorithms for flexible proteins to compare them simultaneously.

ACKNOWLEDGEMENT

This work was partially supported by the Grant-in-Aid for Young Scientist (B) No. 20700264 from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

REFERENCES

[1] Arun, K. S., Huang, T. S., and Blostein, S. D. 1987. Least-squares fitting of two 3-D point sets. *IEEE Trans Pattern Anal. Machine Intell.*, 9, 698–700.

[2] Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. 2000. The protein data bank. *Nucl. Acids Res.*, 28, 235–242.

[3] Boutonnet, N. S., Rooman, M. J., and Wodak, S. J. 1995. Automatic analysis of protein conformational changes by multiple linkage clustering, *J. Mol. Biol.*, 253, 633–647.

[4] Damm, K. L., and Carlson, H. A. 2006. Gaussian-weighted RMSD superposition of proteins: a structural comparison for flexible proteins and predicted protein structures, *Biophys. J.*, 90, 4558–4573.

[5] Echols, N., Milburn, D., and Gerstein, M. 2003. MolMovDB: analysis and visualization of conformational change and structural flexibility, *Nucleic Acids Res.*, 31(1), 478–482.

[6] Eggert, D. W., Lorusso A., and Fisher, R. B. 1997. Estimating 3-D rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications*, 9, 272–290.

[7] Eidhammer, I., Jonassen, I., and Taylor, W. R. 2000. Structure Comparison and Structure Patterns, *J. Comput. Biol.*, 7(5), 685–716.

[8] Emekil, U., Schneidman-Duhovny, D., Wolfson, H. J., Nussinov, R., and Haliloglu, T. 2008. HingeProt: Automated prediction of hinges in protein structures, *Proteins*, 79, 1219–1227.

[9] Flores, S., Echols, N., Milburn, D., Hespenheide, B., Keating, K., Lu, J., Wells, S., Yu, E. Z., Thorpe, M., and Gerstein, M., 2006. The database of macromolecular motions: new features added at the decade mark, *Nucleic Acids Res.*, 34, D296–D301.

[10] Flores, S. C., and Gerstein, M. B. 2007. FlexOracle: predicting flexible hinges by identification of stable domains, *BMC Bioinformatics*, 8, 215.

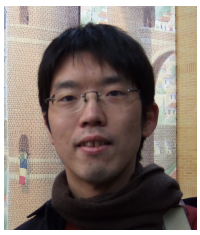
[11] Flores, S. C., Lu, L. J., Yang, J., Carriero, N., and Gerstein, M. B. 2007. Hinge Atlas: relating protein sequence to sites of structural flexibility, *BMC Bioinformatics*, 8, 167.

[12] Gerstein M., and Chothia, C. 1991. Analysis of protein loop closure, two types of hinges produce one motion in lactate dehydrogenase, *J. Mol. Biol.*, 220, 133–149.

[13] Golub, G. H., and Van Loan, C. F. 1996. *Matrix Computation*. 3rd eds., John Hopkins University Press,

[14] Hirschberg, D. S. 1975. A linear space algorithm for computing maximal common subsequences, *Commun. ACM*, 18, 341–343.

[15] Huang, E. S., Rock, E. P., and Subbiah, S. 1993. Automatic and accurate method for analysis of proteins that undergo hinge-mediated domain and loop movements, *Curr. Biol.*, 3(11), 740–748.
[16] Jacobs, D. J., Rader, A. J., Kuhn, L. A., and Thorpe, M. F. 2001. Protein flexibility predictions using graph theory, *Proteins: Structure, Function, and Genetics*, 44. 150–165.
[17] Kabsch, W. 1976. A solution for the best rotation to relate two sets of vectors. *Acta Cryst.*, A32, 922–923.
[18] Kabsch, W. 1978. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Cryst.*, A34, 827–828,
[19] Lemmen, C., and Lengauer, T. 2000. Computational methods for the structural alignment of molecules, *J. Computer-Aided Molecular Design*, 14, 215–232.
[20] Nigham A., and Hsu, D. 2007. Protein Conformational Flexibility Analysis with Noisy Data, *International Conference on Research in Computational Molecular Biology (RECOMB)*,
[21] Ochagavia, M. E., Richeele, J., and Wodak, S. J. 2002. Advanced pairwise structure alignments of proteins and analysis of conformational changes, *Bioinformatics*, 18(4), 637–640.
[22] Perryman, A. L., Lin, J., and McCammon. A. 2004. Hiv-1 protease molecular dynamics of a wild-type and of the v82f/i84v mutant: Possible contributions to drug resistance and a potential new target site for drugs, *Protein Science*, 13, 1108–1123.
[23] Qi, G., Lee, R., and Hayward, S., 2005. A comprehensive and non-redundant database of protein domain movements, *Bioinformatics*, 21(12), 2832-2838.
[24] Schwartz J. T., and Sharir, M. 1987. Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves. *Intl. J. of Robotics Res.*, 6, 29–44.
[25] Shatsky, M., Nussinov, R., and Wolfson, H. J. 2004. FlexProt: Alignment of Flexible Protein Structures without a Predefinition of Hinge Regions, *J. Comput. Biol.*, 11(1), 83–106.
[26] Shibuya, T. 2007. Efficient Substructure RMSD Query Algorithms, *J. Comput. Biol.*, 14(9), 1201–1207.
[27] Thompson, T. B., Thomas, M. G., Escalante-Semerena, J. C., and Rayment, I. 1998. Three-dimensional structure of adenosylcobinamide kinase/adenosylcobinamide phosphate guanylyltransferase from salmonella typhimurium determined to 2.3 a resolution, *Biochemistry*, 37(21), 7686–7695.
[28] Wolfson, H. J., Shatsky, M., Scheneidman-Duhovny, D., Dror, O., Shulman-Peleg, A., Ma, B., and Nussinov, R. 2005. From Structure to Function: Methods and Applications, *Current Protein and Peptide Science*, 6, 171–183.
[29] Wriggers, W., and Schulten, K. 1997. Protein domain movements: detection of rigid domains and visualization of hinges in comparisons of atomic coordinates, *Proteins: Structure, Function, and Genetics*, 29, 1–14.
[30] Ye, Y., and Godzik, A. 2003. Flexible structure alignment by chaining aligned fragment pairs allowing twists, *Bioinformatics*, 19, Suppl. 2, ii246–ii255.



Tetsuo Shibuya, Ph.D. received the Bachelor of Science, the Master of Science, and the Ph.D. of Science from University of Tokyo, in 1995, 1997, and 2002 respectively. His research interest is on algorithms in computational biology. He was a researcher at IBM Tokyo Research Laboratory from 1997 to 2004. He was also a visiting researcher at RIKEN from 2001 to 2008. He is an assistant professor at University of Tokyo from 2004. He is a board member of Japanese Society for Bioinformatics, SIG Bioinformatics of Information Processing Society of Japan, and many others. He has also been in program committees of many conferences, including Genome Informatics Workshop, VLDB bioinformatics, Workshop of Data Mining in Bioinformatics, and Workshop on Algorithms and Computation. He is a member of the JSBi (Japanese Society for Bioinformatics), the IPSJ (Information Processing Society of Japan), and the IEICE (Institute of Electronics, Information and Communication Engineers).

TABLE II

RESULTS OF HINGE DETECTION (1/2). FRAGMENTS WHICH END NEAR AT THE ANNOTATED HINGE POSITIONS ARE MARKED WITH '*'.

Sets	Methods		RMSDh$^{(k)}$	Fragments divided by hinges
AK	Our results	$k=1$	2.4417 Å	[1..52](4.47Å), [53..180](0.51Å)
		$k=2$	0.9773 Å	[1..34](1.00Å)*, [35..51](1.85Å), [52..180](0.79)Å)
		†$k=3$	0.7467 Å	[1..34](1.00Å)*, [35..47](1.44Å), [48..52](1.19Å), [53..180](0.51Å)
		$k=4$	0.5386 Å	[1..32](0.40Å)*, [33..35](0.33Å), [36..47](0.77Å), [48..52](1.19Å), [53..180](0.51Å)
		$k=5$	0.4755 Å	[1..32](0.40Å)*, [33..35](0.33Å)*, [36..45](0.44Å), [46..50](0.55Å), [51..53](0.58Å), [54..180](0.49Å)
	FlexProt		–	[1..36]*, [39..180]
	Annotation in [27]		–	[1..34], [35..180]
HIV	Our results	$k=1$	1.1064 Å	[1..33](0.64Å)*, [34..97](1.28Å)
		†$k=2$	0.7267 Å	[1..44](0.78Å)*, [45..56](0.71Å)*, [57..97](0.67Å)
		$k=3$	0.6483 Å	[1..23](0.66Å)*, [24..44](0.54Å)*, [45..56](0.71Å)*, [57..97](0.67Å)
		$k=4$	0.5795 Å	[1..23](0.66Å)*, [24..38](0.35Å)*, [39..53](0.73Å)*, [54..80](0.57Å), [81..97](0.48Å)
		$k=5$	0.5359 Å	[1..8](0.26Å), [9..23](0.58Å)*, [24..38](0.35Å)*, [39..53](0.73Å)*, [54..80](0.57Å), [81..97](0.48Å)
	FlexProt		–	— No hinge detected —
	Annotation in [16]		–	[1..15], [25..34], [55..97] (*i.e.*, [16..24] and [35..54] are flexible regions)
LDH	Our results	$k=1$	1.6160 Å	[1..116](2.05Å), [117..329](1.32Å)
		†$k=2$	1.1436 Å	[1..97](0.42Å)*, [98..109](1.07Å)*, [110..329](1.35Å)
		$k=3$	0.8902 Å	[1..97](0.42Å)*, [98..109](1.07Å)*, [110..324](1.01Å), [325..329](1.51Å)
		$k=4$	0.7234 Å	[1..97](0.42Å)*, [98..109](1.07Å)*, [110..305](0.80Å), [306..324](0.45Å), [325..329](1.51Å)
		$k=5$	0.6496 Å	[1..96](0.406Å)*, [97..109](0.92Å)*, [110..121](0.45Å), [122..305](0.72Å), [306..324](0.45Å), [325..329](1.51Å)
	FlexProt		–	— No hinge detected —
	Annotation in [12]		–	[1..97], [98..109], [110..329]
BTL	Our results	†$k=1$	0.8614 Å	[1..74](1.08Å), [75..164](0.62Å)
		$k=2$	0.5334 Å	[1..11](0.48Å)*, [12..75](0.41Å), [76..164](0.61Å)
		$k=3$	0.4684 Å	[1..11](0.48Å)*, [12..73](0.39Å), [74..93](0.37Å), [94..164](0.55Å)
		$k=4$	0.4266 Å	[1..11](0.48Å)*, [12..73](0.39Å), [74..93](0.37Å), [94..161](0.47Å), [162..164](0.32Å)
		$k=5$	0.3820 Å	[1..11](0.48Å)*, [12..73](0.39Å), [74..93](0.37Å), [94..130](0.32Å), [131..161](0.41Å), [162..164](0.32Å)
	FlexProt		–	— No hinge detected —
	Hinge Atlas		–	[1..12], [13..80], [81..167]
DPB	Our results	$k=1$	1.8047 Å	[1..83](0.81Å), [84..324](2.04Å)
		†$k=2$	1.0460 Å	[1..83](0.81Å), [84..252](1.15Å), [253..324](1.02Å)
		$k=3$	0.8625 Å	[1..82](0.79Å), [83..139](0.80Å), [140..252](0.83Å), [253..324](1.02Å)
		$k=4$	0.8368 Å	[1..82](0.79Å), [83..139](0.80Å), [140..252](0.83Å), [253..299](1.03Å), [300..324](0.66Å)
		$k=5$	0.8011 Å	[1..82](0.79Å), [83..139](0.80Å), [140..252](0.83Å), [253..291](0.72Å), [292..296](0.88Å), [297..324](0.80Å)
	FlexProt		–	[9-88]*, [89-324]
	Hinge Atlas		–	[1..88], [89..263], [264..324]
EPA	Our results	†$k=1$	0.6392 Å	[1..133](0.80Å)*, [134..298](0.47Å)
		$k=2$	0.5612 Å	[1..83](0.39Å), [84..133](0.96Å)*, [134..298](0.47Å)
		$k=3$	0.5187 Å	[1..81](0.38Å), [82..95](0.36Å), [96..133](0.90Å)*, [134..298](0.47Å)
		$k=4$	0.4867 Å	[1..78](0.38Å), [79..98](0.57Å), [99..115](0.31Å), [116..132](0.89Å)*, [133..298](0.48Å)
		$k=5$	0.4325 Å	[1..78](0.38Å), [79..98](0.57Å), [99..114](0.24Å), [115..119](0.16Å), [120..135](0.34Å)*, [136..298](0.46Å)
	FlexProt		–	— No hinge detected —
	Hinge Atlas		–	[1..134], [135..298]

TABLE III

RESULTS OF HINGE DETECTION (2/2). FRAGMENTS WHICH END NEAR AT THE ANNOTATED HINGE POSITIONS ARE MARKED WITH '*'.

Sets	Methods		RMSDh$^{(k)}$	Fragments divided by hinges
ENL	Our results	$k=1$	1.1942 Å	[1..136](1.52Å), [137..436](1.01Å)
		$k=2$	1.0680 Å	[1..41](2.03Å), [42..141](0.47Å), [142..436](1.02Å)
		$k=3$	0.9053 Å	[1..36](0.29Å), [37..41](2.27Å), [42..141](0.47Å), [142..436](1.02Å)
		$k=4$	0.7724 Å	[1..36](0.29Å), [37..41](2.27Å), [42..152](0.47Å)*, [153..329](1.03Å), [330..436](0.40Å)
		$k=5$	0.6612 Å	[1..36](0.29Å), [37..41](2.27Å), [42..152](0.47Å)*, [153..220](1.18Å), [221..340](0.49Å), [341..436](0.37Å)
	FlexProt		–	— No hinge detected —
	Hinge Atlas		–	[1..149], [150..317], [318..436]
GB	Our results	$k=1$	3.7358 Å	[1..85](0.56Å)*, [86..220](4.75Å)
		$\dagger k=2$	0.9282 Å	[1..84](0.47Å)*, [85..178](1.28Å)*, [179..220](0.62Å)
		$k=3$	0.8057 Å	[1..85](0.56Å)*, [86..105](2.01Å), [106..178](0.52Å)*, [179..220](0.62Å)
		$k=4$	0.6350 Å	[1..84](0.47Å)*, [85..97](0.50Å), [98..105](1.98Å), [106..178](0.52Å)*, [179..220](0.62Å)
		$k=5$	0.5381 Å	[1..84](0.47Å)*, [85..98](0.77Å), [99..101](0.25Å), [102..106](0.55Å), [107..178](0.51Å)*, [179..220](0.62Å)
	FlexProt		–	[5..87]*, [88..180], [181..220]
	Hinge Atlas		–	[1..85], [86..176], [177..220]
LF	Our results	$k=1$	3.8646 Å	[1..248](6.23Å), [249..691](1.26Å)
		$\dagger k=2$	1.1503 Å	[1..91](1.45Å)*, [92..250](0.52Å)*, [251..691](1.24Å)
		$k=3$	0.9290 Å	[1..91](1.45Å)*, [92..250](0.52Å)*, [251..332](0.48Å), [333..691](0.98Å)
		$k=4$	0.7880 Å	[1..3](0.315Å), [4..91](0.52Å)*, [92..250](0.52Å)*, [251..332](0.48Å), [333..691](0.98Å)
		$k=5$	0.7130 Å	[1..3](0.31Å), [4..91](0.52Å)*, [92..250](0.52Å)*, [251..417](1.07Å), [418..422](1.62Å), [423..691](0.54Å)
	FlexProt		–	[1..84], [85..244], [245..691]
	Hinge Atlas		–	[1..90], [91..250], [251..691]
LB	Our results	$k=1$	3.1264 Å	[1..91](0.39Å)*, [92..238](3.97Å)
		$\dagger k=2$	0.4734 Å	[1..90](0.32Å)*, [91..191](0.63Å)*, [192..238](0.31Å)
		$k=3$	0.4234 Å	[1..90](0.32Å)*, [91..161](0.52Å), [162..191](0.57Å)*, [192..238](0.31Å)
		$k=4$	0.3858 Å	[1..90](0.32Å)*, [91..158](0.50Å), [159..182](0.36Å), [183..191](0.35Å)*, [192..238](0.31Å)
		$k=5$	0.3469 Å	[1..90](0.32Å)*, [91..112](0.47Å), [113..158](0.35Å), [159..182](0.36Å), [183..191](0.35Å)*, [192..238](0.31Å)
	FlexProt		–	[1..83], [84..176], [177..238]
	Hinge Atlas		–	[1..90], [91..192], [193..238]
RB	Our results	$k=1$	1.9967 Å	[1..103](0.52Å)*, [104..271](2.50Å)
		$\dagger k=2$	0.5462 Å	[1..102](0.50Å)*, [103..234](0.41Å)*, [235..271](0.96Å)
		$k=3$	0.4505 Å	[1..102](0.50Å)*, [103..233](0.40Å)*, [234..262](0.42Å), [263..271](0.61Å)
		$k=4$	0.3950 Å	[1..34](0.29Å), [35..102](0.38Å)*, [103..233](0.40Å)*, [234..262](0.42Å), [263..271](0.61Å)
		$k=5$	0.3640 Å	[1..34](0.29Å), [35..102](0.38Å)*, [103..152](0.30Å), [153..234](0.36Å)*, [235..262](0.42Å), [263..271](0.61Å)
	FlexProt		–	[1..100]*, [101..271]
	Hinge Atlas		–	[1..103], [104..235], [236..265]
TC	Our results	$k=1$	3.1267 Å	[1..58](3.98Å), [59..155](2.50Å)
		$k=2$	1.6408 Å	[1..34](0.97Å)*, [35..70](1.87Å), [71..155](1.75Å)
		$\dagger k=3$	1.2040 Å	[1..36](1.22Å)*, [37..66](1.30Å), [67..107](1.10Å), [108..155](1.22Å)
		$k=4$	1.0665 Å	[1..36](1.22Å)*, [37..61](0.88Å)*, [62..69](0.69Å), [70..104](0.78Å)*, [105..155](1.23Å)
		$k=5$	0.9042 Å	[1..36](1.22Å)*, [37..61](0.88Å)*, [62..69](0.69Å), [70..104](0.78Å)*, [105..135](0.93Å), [136..155](0.27Å)
	FlexProt		–	[1..34]*, [35..66], [67..155]
	Hinge Atlas		–	[1..35], [36..62], [63..101], [102..155]