Efficient Substructure RMSD Query Algorithms

*Tetsuo Shibuya
Human Genome Center, Institute of Medical Science,
University of Tokyo
4-6-1 Shirokanedai, Minato-ku, Tokyo 108-8639, JAPAN
E-mail: tshibuya@hgc.jp

Keywords: algorithm, computational complexity,
protein 3-D structure comparison

Abstract

Protein structure analysis is a very important research topic in the molecular biology of the post-genomic era. The RMSD (root mean square deviation) is the most frequently used measure for comparing two protein 3-D structures. In this paper, we deal with two fundamental problems related to the RMSD. We first deal with a problem called the 'range RMSD query' problem. Given an aligned pair of structures, the problem is to compute the RMSD between two aligned substructures of them without gaps. This problem has many applications in protein structure analysis. We propose a linear-time preprocessing algorithm that enables constant-time RMSD computation. Next, we consider a problem called the 'substructure RMSD query' problem, which is a generalization of the above range RMSD query problem. It is a problem to compute the RMSD between any substructures of two unaligned structures without gaps. Based on the algorithm for the range RMSD problem, we propose an $O(nm)$ preprocessing algorithm that enables constant-time RMSD computation, where n and m are the lengths of the given structures. Moreover, we propose $O(nm \log r/r)$-time and $O(nm/r)$-space preprocessing algorithm that enables $O(r)$ query, where r is an arbitrary integer such that $1 \leq r \leq \min(n, m)$. We also show that our strategy also works for another measure called the URMSD (unit-vector root mean square deviation), which is a variant of the RMSD.

1 Introduction

In the post-genomic molecular biology, analysis of protein 3-D structures becomes more and more important. Recently, enormous amount of protein 3-D structures are solved and the number of them are still increasing, as seen in the PDB database [Berman et al. 2000]. As a result, faster algorithms to deal with such structures are seriously needed. Two proteins are said to have similar functions if their 3-D structures are similar to each other, and structure comparison is one of the keys to the protein function analysis. There are many structure comparison algorithms for proteins [Eidhammer et al. 2000]. Many of these algorithms are dedicated to align the structures by inserting gaps so as to minimize some score, as in the cases of ordinary sequence alignment algorithms. But the alignment is not the goal of the protein analysis. For example, we must search for specific domains like conserved substructures or hinge-mediated domains from the aligned result.

The most famous similarity measure for protein structures is the RMSD (root mean square deviation) [Arun et al. 1987, Eggert et al. 1997, Kabsch 1976, Kabsch 1978, Schawrtz et al. 1987]. There are several variations of the RMSD measure. The URMSD (unit-vector root mean square deviation) [Chew et al. 1999, Kedem et al. 1999] is an example. To analyze the aligned protein structures, we often need to compute local RMSDs (or URMSDs) of aligned substructures. For example, in [Huang et al. 1993], they compute all the RMSDs between all the possible aligned substructures in the aligned pair of structures to find hinge-mediated domains. Similar techniques are used also in many others [Nigham et al. 2007, Shibuya 2007]. Hence, we consider the following problem in this paper.

Problem 1 (Range RMSD Query Problem) *Given a pair of aligned*[1] *protein structures represented by lists of points:* $\mathbf{P} = \mathbf{P}[1..n]$ *and* $\mathbf{Q} = \mathbf{Q}[1..n]$*, the problem is to compute the RMSD between* $\mathbf{P}[i..j]$ *and* $\mathbf{Q}[i..j]$ *for a query pair of integers* i *and* j $(1 \leq i < j \leq n)$*.*

Without any preprocessing, we need $O(k)$ time to compute it, where $k = j - i$ (see section 2 for details). We propose an algorithm that answers the above problem in constant time, with linear-time preprocessing. Note that naive preprocessing algorithm would require $O(n^2)$ time or more, as there are $O(n^2)$ possible aligned substructures. Moreover, we will show that the same strategy works for the URMSD.

We next consider the following problem, which is a generalization of the above range RMSD query problem.

Problem 2 (Substructure RMSD Query Problem) *Given two protein structures* $\mathbf{P} = \mathbf{P}[1..n]$ *and* $\mathbf{Q} = \mathbf{Q}[1..m]$*, the problem is to compute the RMSD (without considering gaps) between* $\mathbf{P}[i..i+\ell-1]$ *and* $\mathbf{Q}[j..j+\ell-1]$ *for a query set of integers* i*,* j *and* ℓ $(1 \leq i \leq i+\ell-1 \leq n,\ 1 \leq j \leq j+\ell-1 \leq m)$*.*

By straightforwardly using our range RMSD query algorithm, we can solve this problem in constant time after $O(nm)$-time preprocessing. Note that we would require $O(nm \min(n, m))$ time or more for preprocessing if we do it naively. For this problem, we also propose $O(nm \log r/r)$-time and $O(nm/r)$-space preprocessing algorithm that enables $O(r)$ query, where r is an arbitrary integer such that $1 \leq r \leq \min(n, m)$.

The substructure RMSD query problem is more general than the range RMSD query problem, and must have many applications in protein structure analysis. This problem appears everywhere, *e.g.*, as a subroutine for alignment algorithms. Most protein alignment algorithms take very large time and is sometimes requires more than $O(nm)$ time. Thus we assume our algorithm can be very valuable for designing such alignment algorithms.

The organization of this paper is as follows. In section 2, we present the definitions of the RMSD and the URMSD. Then, we present an algorithm for the range RMSD query problem in section 3, and we present algorithms for the substructure RMSD query problem in section 4. Finally in section 5, we conclude our results.

2 Preliminaries

2.1 RMSD: The Root Mean Square Deviation

A protein is a chain of amino acids. Each amino acid has one unique carbon atom named C_α, and the coordinates of the C_α atom is often used as the representative position of the amino acid. The set of C_α atom positions in a protein is called the backbone of the protein. The backbone structures are topologically linear, which can be represented by just a list of points in 3-D space, regardless of how complex the structures are. In this paper, we deal with this coordinates list, as in the most structural research on proteins.

The similarity between two backbone structures is often measured by the RMSD (root mean square deviation) [Arun et al. 1987, Eggert et al. 1997, Kabsch 1976, Kabsch 1978, Schawrtz et al. 1987].[2] 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 coordinates of the i-th C_α atoms of $\mathbf{P}$ and $\mathbf{Q}$. We here consider $\vec{p}_i$ corresponds to $\vec{q}_i$ for

[1]We mean by 'aligned' that we do not consider gaps, *i.e.*, $\mathbf{P}[i]$ corresponds to $\mathbf{Q}[i]$.

[2]To do so, we need to know the correspondence between atoms of the two structures. There are many cases that we know it, such as the cases of structures formed by the same flexible protein. If we do not know the correspondence, we must use some structural alignment algorithms to find the correspondence, and remove the atoms with no corresponding atoms. In this paper, we do not deal with alignment algorithms.

each i. Then the RMSD between $\mathbf{P}$ and $\mathbf{Q}$ is defined as the minimum value of

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

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

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. Thus, the following equation holds:

$$\hat{\vec{v}}(\mathbf{P},\mathbf{Q}) = \frac{1}{n}\sum_{i=1}^{n}(\vec{p}_i - \hat{R}(\mathbf{P},\mathbf{Q})\cdot\vec{q}_i). \tag{2}$$

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 the following value:

$$f_R(\mathbf{P},\mathbf{Q}) = \sum_{i=1}^{n}\|\vec{p}_i - R\cdot\vec{q}_i\|^2. \tag{3}$$

We can find $\hat{R}(\mathbf{P},\mathbf{Q})$ in linear time by using the singular value decomposition (SVD) [Arun et al. 1987, Kabsch 1976, Kabsch 1978] as follows. Let $H = \sum_{i=1}^{n}\vec{p}_i\cdot\vec{q}_i^{\,t}$, where $\vec{v}^t$ means the transpose of $\vec{v}$. Clearly, H can be computed in $O(n)$ time. Then $f_R(\mathbf{P},\mathbf{Q})$ can be described as

$$f_R(\mathbf{P},\mathbf{Q}) = \sum_{i=1}^{n}(\vec{p}_i^{\,t}\vec{p}_i + \vec{q}_i^{\,t}\vec{q}_i) - trace(R\cdot H), \tag{4}$$

and $trace(RH)$ is maximized when $R = VU^T$, where $U\Lambda V$ is the 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 [Golub et al. 1996]. Note that there are rare degenerate cases where $det(VU^T) = -1$, which means that VU^T is a reflection matrix. We ignore the degenerate cases in this paper. 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.

2.2 URMSD: The Unit-Vector Root Mean Square Deviation

The URMSD (unit-vector root mean square deviation) [Chew et al. 1999, Kedem et al. 1999] is a variation of the RMSD. The RMSD is sometimes influenced badly by very distant pairs of points, and the URMSD is designed to avoid such influence. Let $\vec{p}_i' = (\vec{p}_{i+1} - \vec{p}_i)/\|\vec{p}_{i+1} - \vec{p}_i\|$ and $\vec{q}_i' = (\vec{q}_{i+1} - \vec{q}_i)/\|\vec{q}_{i+1} - \vec{q}_i\|$. Then the URMSD is the minimum value of

$$d_R'(\mathbf{P},\mathbf{Q}) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n-1}\|\vec{p}_i' - R\cdot\vec{q}_i'\|^2} \tag{5}$$

over possible rotation matrices R. Let $\check{R}(\mathbf{P},\mathbf{Q})$ denote R that minimizes the value. The URMSD can also be computed in the same way as the RMSD, which can be easily imagined from the fact that the equation of $d_R'(\mathbf{P},\mathbf{Q})$ resembles that of $f_R(\mathbf{P},\mathbf{Q})$ very much.

Let $H' = \sum_{i=1}^{n-1}\vec{p}_i'\cdot(\vec{q}_i')^t$. Then $d_R'(\mathbf{P},\mathbf{Q})$ can be described as $(f_R'(\mathbf{P},\mathbf{Q})/(n-1))^{1/2}$, where $f_R'(\mathbf{P},\mathbf{Q}) = \sum_{i=1}^{n-1}(\vec{p}_i'^{\,t}\vec{p}_i' + \vec{q}_i'^{\,t}\vec{q}_i') - trace(R\cdot H')$. Thus, $d_R'(\mathbf{P},\mathbf{Q})$ is minimized when $R = V'U'^T$ where $U'\Lambda V'$ is the SVD of H'.[3] It is obvious that we can compute the URMSD value from $\check{R}(\mathbf{P},\mathbf{Q})$ in linear time. Hence the total computation time for the URMSD is also linear.

[3]Like in the case of the RMSD computation, there are rare degenerate cases where $V'U'^T$ becomes a reflection matrix, but we do not deal with these cases in this paper.

3 Preprocessing for the Range RMSD Query Problem

In this section, we show an algorithm for the range RMSD query problem introduced in section 1. Let the two given aligned structures be $\mathbf{P} = \{\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_n\}$ and $\mathbf{Q} = \{\vec{q}_1, \vec{q}_2, \ldots, \vec{q}_n\}$. Let $\mathbf{P}[i..j]$ denote the substructure $\{\vec{p}_i, \vec{p}_{i+1}, \ldots \vec{p}_j\}$, and $\mathbf{Q}[i..j]$ denote the substructure $\{\vec{q}_i, \vec{q}_{i+1}, \ldots \vec{q}_j\}$, Then the range RMSD query problem is a problem to compute the RMSD between $\mathbf{P}[i..j]$ and $\mathbf{Q}[i..j]$ for any i and j ($i \leq j$), without considering gaps.

Let $\vec{c}(\mathbf{X})$ denote the centroid of structure $\mathbf{X}$. Then, $\vec{c}(\mathbf{P}[i,j]) = \sum_{k=i}^{j} \vec{p}_k/(j-i+1)$, and $\vec{c}(\mathbf{Q}[i,j]) = \sum_{k=i}^{j} \vec{q}_k/(j-i+1)$. Let $\bar{\vec{p}}_k = \vec{p}_k - \vec{c}(\mathbf{P}[i..j])$, and $\bar{\vec{q}}_k = \vec{q}_k - \vec{c}(\mathbf{Q}[i..j])$. Note that $\bar{\vec{p}}_k$ is the coordinates of the atom $\vec{p}_k$ translated so that the centroid of $\mathbf{P}[i..j]$ is at the origin of the coordinates, and so is the $\bar{\vec{q}}_k$. According to the equations of $f_R(\mathbf{P}, \mathbf{Q})$ in section 2.1, the RMSD between $\mathbf{P}[i..j]$ and $\mathbf{Q}[i..j]$ can be obtained in constant time if we are given the value $g(i,j) = \sum_{k=i}^{j} (\bar{\vec{p}}_k{}^t \bar{\vec{p}}_k + \bar{\vec{q}}_k{}^t \bar{\vec{q}}_k)$ and the 3×3 matrix $\bar{H}(i,j) = \sum_{k=i}^{j} \bar{\vec{p}}_k \cdot \bar{\vec{q}}_k{}^t$.

Let $\vec{h}_{\mathbf{P}}(i,j) = \sum_{k=i}^{j} \vec{p}_k$, and $h^2_{\mathbf{P}}(i,j) = \sum_{k=i}^{j} \vec{p}_k{}^t \vec{p}_k$. Similarly, let $\vec{h}_{\mathbf{Q}}(i,j) = \sum_{k=i}^{j} \vec{q}_k$, and $h^2_{\mathbf{Q}}(i,j) = \sum_{k=i}^{j} \vec{q}_k{}^t \vec{q}_k$. Add to them, let $G(i,j) = \sum_{k=i}^{j} \vec{p}_k \vec{q}_k{}^t$. Note that $\vec{h}_{\mathbf{X}}(i,j)$ is a vector of length 3, $h^2_{\mathbf{X}}(i,j)$ is a numerical value, and $G(i,j)$ is a 3×3 matrix. Then, the following two equations hold:

$$g(i,j) = h^2_{\mathbf{P}}(i,j) + h^2_{\mathbf{Q}}(i,j) - \frac{\vec{h}_{\mathbf{P}}(i,j)^t \vec{h}_{\mathbf{P}}(i,j)}{j-i+1} - \frac{\vec{h}_{\mathbf{Q}}(i,j)^t \vec{h}_{\mathbf{Q}}(i,j)}{j-i+1} \tag{6}$$

$$\bar{H}(i,j) = G(i,j) - \frac{\vec{h}_{\mathbf{P}}(i,j) \vec{h}_{\mathbf{Q}}(i,j)^t}{j-i+1} \tag{7}$$

These equations suggest that the RMSD value can be obtained in constant time if we are given $\vec{h}_{\mathbf{P}}(i,j)$, $\vec{h}_{\mathbf{Q}}(i,j)$, $h^2_{\mathbf{P}}(i,j)$, $h^2_{\mathbf{Q}}(i,j)$, and $G(i,j)$. Note that similar techniques are used in [Shibuya 2006] for incremental RMSD computation.

$\vec{h}_{\mathbf{X}}(i,j)$ can be computed in constant time, if we are given $\vec{h}_{\mathbf{X}}(1,k)$ for all k ($1 \leq k \leq n$), as $\vec{h}_{\mathbf{X}}(i,j) = \vec{h}_{\mathbf{X}}(1,j) - \vec{h}_{\mathbf{X}}(1,i-1)$ for either case of $\mathbf{X} = \mathbf{P}$ or $\mathbf{X} = \mathbf{Q}$.[4] It is obvious that $\vec{h}_{\mathbf{X}}(1,k)$ can be computed for all k in linear time. $h^2_{\mathbf{X}}(i,j)$ can also be computed in constant time, if we are given $h^2_{\mathbf{X}}(1,k)$ for all k ($1 \leq k \leq n$), as $h^2_{\mathbf{X}}(i,j) = h^2_{\mathbf{X}}(1,j) - h^2_{\mathbf{X}}(1,i-1)$. Here, it is also obvious that $\vec{h}_{\mathbf{X}}(1,k)$ can be computed for all k in linear time. The same technique also works for $G(i,j)$. $G(i,j)$ can be computed in constant time, if we are given $G(1,k)$ for all k ($1 \leq k \leq n$), as $G(i,j) = G(1,j) - G(1,i-1)$. $G(1,j)$ can also be computed for all k in linear time.

Hence we can compute the RMSD between $\mathbf{P}[i..j]$ and $\mathbf{Q}[i..j]$ in constant time, if we are given $\vec{h}_{\mathbf{P}}(1,k)$, $\vec{h}_{\mathbf{Q}}(1,k)$, $h^2_{\mathbf{P}}(1,k)$, $h^2_{\mathbf{Q}}(1,k)$, and $G(1,k)$ for all k ($1 \leq k \leq n$), all of which can be computed in linear time before the query. Note that the corresponding optimal rotation matrix and the translation vector can be obtained at the same time.

The same strategy also works for the URMSD. This case is much easier because we do not have to consider the translation vector. According to the equation of $f'_R(\mathbf{P}, \mathbf{Q})$, we can compute the URMSD between $\mathbf{P}[i..j]$ and $\mathbf{Q}[i..j]$ in constant time, if we are given $H'(i.j) = \sum_{k=i}^{j-1} \vec{p}'_k \cdot (\vec{q}'_k)^t$ and $\sum_{k=i}^{j-1} (\vec{p}'_k{}^t \vec{p}'_k + \vec{q}'_k{}^t \vec{q}'_k)$. Furthermore, we can compute them if we are given $H'(1,k)$, $h'^2_{\mathbf{P}}(1,k)$ and $h'^2_{\mathbf{Q}}(1,k)$, letting $h'^2_{\mathbf{P}}(i,j) = \sum_{k=i}^{j} \vec{p}'_k{}^t \vec{p}'_k$, and $h'^2_{\mathbf{Q}}(i,j) = \sum_{k=i}^{j} \vec{q}'_k{}^t \vec{q}'_k$. As $\vec{p}'_k$ and $\vec{q}'_k$ can be easily computed from $\vec{p}_k$, $\vec{p}_{k+1}$, $\vec{q}_k$, and $\vec{q}_{k+1}$ in constant time, we conclude that the URMSD can also be computed in constant time after linear-time preprocessing.

[4]To ease discussion, we let $\vec{h}_{\mathbf{X}}(i,j) = \vec{0}$ (zero vector) if $i > j$. Similarly, we let $h^2_{\mathbf{X}}(i,j) = 0$ and $G(i,j) = O$ (zero matrix) if $i > j$.

4 Preprocessing for the Substructure RMSD Query Problem

In this section, we propose algorithms for the substructure RMSD query problem introduced in section 1. Given two structures $\mathbf{P} = \{\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_n\}$ and $\mathbf{Q} = \{\vec{q}_1, \vec{q}_2, \ldots, \vec{q}_m\}$,[5] the range RMSD query problem is a problem to compute the RMSD between $\mathbf{P}[i..i{+}\ell{-}1]$ and $\mathbf{Q}[j..j{+}\ell{-}1]$ for any i, j and ℓ ($0 \le i \le i{+}\ell{-}1 \le n$, $0 \le j \le j + \ell - 1 \le m$). From now on, we let $i \le j$ without loss of generality.[6] Then, by preprocessing (as discussed in section 3) straightforwardly for all the range RMSD query problems between $\mathbf{P}[1..n']$ and $\mathbf{Q}[k..k + n' - 1]$ for all k ($1 \le k \le m$) where $n' = \min(n, m - k + 1)$, we can solve the substructure RMSD query problem in constant time. This straightforward preprocessing algorithm requires $O(nm)$ time and space in total. From now on, we discuss how to reduce the preprocessing time and space, permitting larger query time, as the $O(nm)$ time/space can sometimes be too large.

According to the discussion in section 3, we can compute the RMSD between $\mathbf{P}[i..i{+}\ell{-}1]$ and $\mathbf{Q}[j..j{+}\ell{-}1]$ in constant time, if we are given the followings:

$$\vec{h}_{\mathbf{P}}(i, i + \ell - 1) = \sum_{k=i}^{i+\ell-1} \vec{p}_k \tag{8}$$

$$\vec{h}_{\mathbf{Q}}(j, j + \ell - 1) = \sum_{k=i}^{j+\ell-1} \vec{q}_k \tag{9}$$

$$h^2_{\mathbf{P}}(i, i + \ell - 1) = \sum_{k=i}^{i+\ell-1} \vec{p}_k^{\,t} \cdot \vec{p}_k \tag{10}$$

$$h^2_{\mathbf{Q}}(j, j + \ell - 1) = \sum_{k=j}^{j+\ell-1} \vec{q}_k^{\,t} \cdot \vec{q}_k \tag{11}$$

$$E(i, j, \ell) = \sum_{k=1}^{\ell} \vec{p}_{i+k-1} \cdot \vec{q}_{j+k-1}^{\,t} \tag{12}$$

Note that $E(i, j, \ell)$ is the $G(i, i + \ell - 1)$ for the range RMSD query problem between $\mathbf{P}[1..n'']$ and $\mathbf{Q}[j - i + 1..n'' + j - i]$, where $n'' = \min(n, m - j + i)$. As for $\vec{h}_{\mathbf{P}}(i, i + \ell - 1)$, $\vec{h}_{\mathbf{Q}}(j, j + \ell - 1)$, $h^2_{\mathbf{P}}(i, i + \ell - 1)$ and $h^2_{\mathbf{Q}}(j, j{+}\ell{-}1)$, they can be computed in constant time after linear time preprocessing by the same technique discussed in section 3, *i.e.*, we need to compute $\vec{h}_{\mathbf{P}}(1, k)$, $\vec{h}_{\mathbf{Q}}(1, k)$, $h^2_{\mathbf{P}}(1, k)$ and $h^2_{\mathbf{Q}}(1, k)$ for all k before the query. As for $E(i, j, \ell)$, if we compute $E(1, j, \ell)$ for all possible j and ℓ beforehand, we can compute it in constant time, as $E(i, j, \ell) = E(1, j, \ell) - E(1, i - 1, \ell)$ if $i > 1$. This preprocessing requires $O(nm)$ time and space. But if we permit $O(r)$ time ($1 \le r \le \min(n, m)$) to compute the $E(i, j, \ell)$, we can reduce them to $O(nm \log r/r)$ time and $O(nm/r)$ space as follows.

Let r be an integer such that $1 \le r \le \min(n, m)$. To ease discussion, we consider that both n and m are multiples of r. Then we can divide $\mathbf{P}$ into n/r substructures of length r. Let them be $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_{n/r}$. Note that $\mathbf{P}_p = \mathbf{P}[(r \cdot (p - 1) + 1)..r \cdot p]$. Likewise, $\mathbf{Q}$ can be divided into m/r substructures of length r, and

[5]In this section, the length of $\mathbf{Q}$ can be different from that of $\mathbf{P}$.
[6]In the case of $i > j$, we can do the same discussion by changing $\mathbf{P}$ and $\mathbf{Q}$.

let them be $\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_{m/r}$. For $\mathbf{P}_p$, $\mathbf{Q}_q$, and k such that $-r < k < r$, consider the following value:[7]

$$E_{p,q}(k) = \begin{cases} \sum_{s=1}^{r-k} \mathbf{P}_p[s] \cdot \mathbf{Q}_q[s+k]^t & (k \geq 0) \\ \sum_{s=1-k}^{r} \mathbf{P}_p[s] \cdot \mathbf{Q}_q[s+k]^t & (k < 0) \end{cases} \tag{13}$$

For each pair of p and q, we can compute $E_{p,q}(k)$ for all k in $O(r \log r)$ time, by the convolution technique based on the FFT (fast Fourier transform) (see [Cooley et al. 1965] for details). Thus we need $O(nm \log r/r)$ time to compute them for all the $O(nm/r^2)$ pairs of p and q. Next, we consider the following $F_{p,q}(k)$ for all the set of p, q and k ($1 \leq p \leq n/r$, $1 \leq q \leq m/r$, $p \leq q$, $-r < k < r$).

$$F_{p,q}(k) = \sum_{t=1}^{p} E_{t,t-p+q}(k) \tag{14}$$

Obviously, $F_{p,q}(k)$ can be computed in $O(nm/r)$ time for all the $O(nm/r)$ sets of p, q and k, if we are given all the $E_{p,q}(k)$ values.

Recall that $E(i,j,\ell) = E(1,j,\ell) - E(1,i-1,\ell)$ $(i > 1)$. Thus, if we can compute $E(1,j,\ell)$ in $O(r)$ time, we can also compute $E(i,j,\ell)$ in $O(r)$ time for any i, j and ℓ. Thus, hereinafter, we discuss how to compute $E(1,j,\ell)$ in $O(r)$ time by using the $F_{p,q}(k)$ values. Let $a = \lfloor \ell/r \rfloor$ and $b = \lceil j/r \rceil$. To ease discussion, we let $F_{p,q}(k) = 0$ if $p = 0$ or $k \geq r$, and $E(i,j,\ell) = 0$ if $\ell \leq 0$. Then, $E(1,j,\ell)$ can be described as follows:

$$\begin{aligned} E(1,j,\ell) = \; & F_{a,a+b-1}(j + r - b \cdot r - 1) \\ & + F_{a,a+b}(b \cdot r - j + 1) \\ & + E(a \cdot r + 1, j + a \cdot r, \ell - a \cdot r) \end{aligned} \tag{15}$$

Notice that $0 \leq \ell - a \cdot r < r$, and $E(a \cdot r + 1, j + a \cdot r, \ell - a \cdot r)$ can be computed in $O(r)$ time. Thus, we conclude that $E(1,j,\ell)$ can be computed in $O(r)$ time, after $O(nm \log r/r)$ time preprocessing time in total. The space required for the preprocess is $O(nm/r)$.

Consequently, we conclude that we can compute the RMSD in $O(r)$ time after $O(nm \log r/r)$ time and $O(nm/r)$ space preprocessing. It is trivial that the same techniques can be applied to the substructure URMSD query problem, and it can also be computed in $O(r)$ time after $O(nm \log r/r)$ time and $O(nm/r)$ space preprocessing.

5 Conclusions

In this paper, we dealt with two fundamental problems called the 'range RMSD query problem' and the 'substructure RMSD query problem', both of which are very important for protein structure analysis. For the range RMSD query problem, we proposed a constant-time query algorithm after linear-time preprocessing. Based on the algorithm, we showed that we can solve the substructure RMSD query problem in constant time after $O(nm)$ time preprocessing, where n and m are the lengths of the structures to be compared. Moreover, for the latter problem, we also developed an $O(r)$ query algorithm after $O(nm \log r/r)$ time and (nm/r) space preprocessing, where r is an arbitrary integer such that $1 \leq r \leq \min(n,m)$.

Our algorithms can be used in various scenes in protein structure research. Actually, we have developed a hinge detection algorithm for flexible protein 3-D structures based on our algorithm [Shibuya 2007]. Development of faster query algorithms, or more flexible query algorithms remains as future work.

[7] $\mathbf{P}_p[s]$ is same as $\vec{p}_{r(p-1)+s}$. Similarly, $\mathbf{Q}_q[s+k]$ is same as $\vec{q}_{r(q-1)+s+k}$

References

[Arun et al. 1987] 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,

[Berman et al. 2000] 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.

[Chew et al. 1999] Chew, L. P., Huttenlocher, D., Kedem K., and Kleinberg. J. 1999. Fast detection of common geometric substructure in proteins. *J. Comput. Biol.*, 6(3), 313-325.

[Cooley et al. 1965] Cooley, J. W., and Tukey, J. W. 1965. An algorithm for the machine calculation of complex Fourier series, *Math. Comput.*, 19, 297-301.

[Eggert et al. 1997] 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.

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

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

[Huang et al. 1993] 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.

[Kabsch 1976] Kabsch, W. 1976. A solution for the best rotation to relate two sets of vectors. *Acta Cryst.*, A32, 922-923.

[Kabsch 1978] Kabsch, W. 1978. A discussion of the solution for the best rotation to relate two sets of vectors. 1978. *Acta Cryst.*, A34, 827-828,

[Kedem et al. 1999] Kedem, K., Chew, P., and Elber, R. 1999. Unit-vector RMS (URMS) as a tool to analyze molecular dynamics trajectories. *Proteins: Struct. Funct. Genet.*, 38, 1-12.

[Nigham et al. 2007] Nigham, A., and Hsu, D. 2007. Protein Conformational Flexibility Analysis with Noisy Data, *International Conference on Research in Computational Molecular Biology (RECOMB).*

[Schawrtz et al. 1987] 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.

[Shibuya 2006] Shibuya, T. 2006. Geometric Suffix Tree: A New Index Structure for Protein 3-D Structures, *Combinatorial Pattern Matching 2006 (CPM 2006), LNCS 4009*, 84-93,

[Shibuya 2007] Shibuya, T. 2007. RMSDh: A New Measure for Comparing Flexible Protein Structures with Hinges, submitted to *J. Comput. Biol.*.