Searching Protein 3-D Structures in Faster Than Linear Time

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

Abstract

Searching for similar structures from a 3-D structure database of proteins is one of the most important problems in post-genomic computational biology. To compare two structures, we ordinarily use a measure called the RMSD (root mean square deviation) as the similarity measure. We consider a very fundamental problem of finding all the substructures whose RMSDs to the query are within some given threshold, from a 3-D structure database. The problem also appears in many other fields, such as computer vision and robotics. In this paper, we propose the first algorithm that runs in faster than linear time in average. Our new algorithm runs in average-case $O(m + N/m^{1-\epsilon})$, where N is the database size, m is the query length, and ϵ is an arbitrary small constant such that $0 < \epsilon < 1$. It is a significant improvement over previous algorithms on the problem, considering that the best known worst-case time complexity of the problem is $O(N \log m)$, and the best known average-case (expected) time complexity of the problem was $O(N)$.

1 Introduction

With the aid of various state-of-the-art laboratory methods such as NMR (Nuclear Magnetic Resonance), more and more 3-D structures of bio-molecules, especially the proteins, are determined. For example, the number of structures contained in the PDB (Protein Structure Data Bank) database [3] was only around $1,000$ in 1993, but it becomes more than $58,000$ in June 2009, and it is still increasing very rapidly (about 20% per year). Moreover, tremendous number of protein structures are being predicted on super computers these days. Database searching on such databases has many applications in various fields of molecular biology [2, 12, 14, 21]. For example, we can predict the functions of some protein if its structure is similar to another protein with known functions, as proteins with similar structures tend to have similar functions. Hence, 3-D structure database searching becomes more and more important in computational biology, and faster searching techniques are seriously needed.

A protein is a chain of amino acid residues, and its structure is usually represented by a sequence of 3-D coordinates of representative atoms of residues (atoms named C_α are usually used). There are also many other important chain molecules in living cells, such as DNAs, RNAs, and glycans, and they can also be represented by a sequence of 3-D coordinates. No two molecular structures have exactly the same structure. Therefore we need to search for similar structures from the structure databases, instead of searching for exactly the same structure. In this paper, we deal with a problem of searching for similar structures from structure databases of chain molecules, each of which is represented by a sequence of 3-D coordinates.

There have been developed tremendous number of algorithms for comparing/searching protein structures, which can be categorized roughly into two types. One is a group of algorithms that compare two structures geometrically in the 3-D space, using the coordinates of structures as their inputs. They consider that the structures are rigid or near-rigid, and superimpose (substructures of) the two structures by rotating and translating one of them. The other is a group of algorithms that uses more abstract information of the structures, such as the secondary structure elements (SSEs) [23, 24]. In this paper, we focus on the first type of the algorithms, *i.e.*, we compare the sequences of coordinates without any abstraction.

Biologists ordinarily use a similarity measure called the RMSD (root mean square deviation) [1, 11, 18, 19, 21, 25, 26] to compare two structures of bio-molecules. There are also many other measures, but many of them are just variants of the RMSD [17]. It is the most fundamental measure to determine geometric similarity between two same-length sequences of 3-D coordinates, which is also used in various other fields, such as computer vision and robotics. (The problem minimizing the RMSD is called the *least squares fitting problem* in computer vision.) It is defined as the square root of the minimum value of the average squared distance between each pair of corresponding atoms, over all the possible rotations and translations. (See section 2.2 for more details.) In this paper, we consider the following problem related to the RMSD, which is one of the most fundamental computational problems in computational molecular biology.

3-D Substructure Search Problem

We are given a text structure $\mathbf{P}$ and a query structure $\mathbf{Q}$, both of which are represented by the 3-D coordinates of the residues of the structures. The problem is to find all the positions of the substructures of $\mathbf{P}$ whose RMSDs to $\mathbf{Q}$ are at most a given fixed threshold c, without considering insertions or deletions.[1]

In case the database consists of more than one structure, we can reduce the problem into the above single-text problem by concatenating all the database structures into a single text structure and ignoring substructures that cross over the boundaries of concatenated structures. In general, c is set to a fixed constant proportional to the distance between two adjacent atoms of the chain molecules. In the case of protein structures, the distance between two adjacent C_α atoms is always around 3.8Å, while structural biologists say that two protein structures are similar to each other if their RMSD is at most around 2Å.

History

In 1976, Kabsch [18, 19] proposed a sophisticated mathematical RMSD computation algorithm based on the singular value decomposition that computes the RMSD between two structures of size n in $O(n)$ time. If we directly apply the algorithm to the 3-D substructure search problem, it can be solved in $O(Nm)$ time, where N is the length of the text structure and the m is the length of the query structure. In 1987, Schwartz and Sharir [25] proposed an

[1] The problem becomes much harder if we take deletions and insertions into account. Even the simplest version of the problem is known to be NP-hard [15].

algorithm based on the fast Fourier transform technique that runs in $O(N \log m)$ time, which is the best-known worst-case time complexity.[2] More recently, Shibuya [27] proposed a linear expected-time (average-case time) algorithm, which was the best-known time complexity before this paper, though its worst-case time complexity is $O(Nm)$. To analyze the average-case time complexity, he used the freely-jointed chain model [4, 10, 13, 22] as a model of random protein structures. The model is is widely used for analyzing properties of various chain molecules (see section 2.3). The model was also used in [28] to analyze the average time complexity of an algorithm on structures of chain molecules. According to [27], there is high consistency between the theoretical results deduced from the freely-jointed chain model and the experimental results on the whole PDB database, and it is very reasonable to use the model for analyzing the average computational complexity of algorithms for molecular structures.

Shibuya [27] also proposed several preprocessing algorithms that achieves even faster expected query time complexity. One is an $O(N \log^2 N)$-time and $O(N \log N)$-space preprocessing algorithm with expected query time complexity of $O(m + N/\sqrt{m})$. Another is an $O(N \log N)$-time and $O(N)$-space preprocessing algorithm with expected query time complexity of $O(\frac{N}{\sqrt{m}} + m \log(N/m))$. But the expected query time complexity of our algorithm in this paper is theoretically better than any of these algorithms, even though our algorithm does not require any preprocessing.

Our results

The expected time complexity of the algorithm in [27] seems to be optimal, as we apparently require $\Omega(N)$ time to sequentially load all the data into memory. But in case we can randomly access the input data, there could be algorithms faster than $O(N)$. For example, the famous Boyer-Moore algorithm [5], which is one of the most widely-used searching algorithms for ordinary texts, achieves $O(m + N/\min(m, |\Sigma|))$ average time complexity, where N is the text size, m is the query size and $|\Sigma|$ is the alphabet size, by permitting random access to the characters in the text.

In this paper, we make a breakthrough by proposing the first algorithm whose expected time complexity is better than linear for the above 3-D substructure search problem. It runs in expected $O(m + N/m^{1-\epsilon})$ time, where ϵ is an arbitrary small constant such that $0 < \epsilon < 1$, under the assumption that we can randomly access any data in the database (*i.e.*, the coordinates of any atoms in the text structure). Note that this time complexity is even better than the query time complexity of the previous algorithms with preprocessing [27], though our new algorithm does not require any preprocessing. Our analysis of the expected time complexity is based on the freely-jointed chain model, which has also been used in previous work [27, 28] for the analysis of expected time complexities of previous algorithms.

The organization of this paper

In section 2, we describe the notations used in this paper and previous related work as preliminaries. In section 3, we describe our algorithm. In section 4, we analyze the expected (average-case) time complexity of our algorithm. In section 5, we conclude our results.

[2] The original algorithm runs in $O(N \log N)$ time. See [27] for the technique to improve it into $O(N \log m)$.

2 Preliminaries

2.1 Notations and Definitions

A structure of a chain molecule is represented by a notation like $\mathbf{S} = \{\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_n\}$, where $\vec{s}_i$ denotes the 3-D coordinates of the i-th atom. The length n of $\mathbf{S}$ is denoted by $|\mathbf{S}|$. A structure $\mathbf{S}[i..j] = \{\vec{s}_i, \vec{s}_{i+1}, \ldots, \vec{s}_j\}$ $(1 \leq i \leq j \leq n)$ is called a substructure of $\mathbf{S}$. For two structures $\mathbf{S} = \{\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_n\}$ and $\mathbf{T} = \{\vec{t}_1, \vec{t}_2, \ldots, \vec{t}_{n'}\}$, the concatenated structure $\{\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_n, \vec{t}_1, \vec{t}_2, \ldots, \vec{t}_{n'}\}$ is denoted by $\mathbf{S} + \mathbf{T}$. $R \cdot \mathbf{S}$ denotes the structure $\mathbf{S}$ rotated by the rotation matrix R, *i.e.*, $R \cdot \mathbf{S} = \{R\vec{s}_1, R\vec{s}_2, \ldots, R\vec{s}_n\}$.

$\vec{v}^t$ denotes the transpose of the vector $\vec{v}$ and A^T denotes the transpose of the matrix A. $trace(A)$ denotes the trace of the matrix A. $|\vec{v}|$ denotes the norm of the vector $\vec{v}$. $\vec{0}$ denotes the zero vector. $\langle x \rangle$ denotes the expected value of x. $var(x)$ denotes the variance of x. $Prob(\mathcal{X})$ denotes the probability of the event $\mathcal{X}$.

In the rest of this paper, $\mathbf{P} = \{\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_N\}$ denotes the text structure and $\mathbf{Q} = \{\vec{q}_1, \vec{q}_2, \ldots, \vec{q}_m\}$ denotes the query structure. Note that the size m of the query could be an arbitrary number such that $1 < m \leq N$. The 3-D substructure search problem is to find all the positions i such that the RMSD (see section 2.2 for its definition) between $\mathbf{P}[i..i+m-1]$ and $\mathbf{Q}$ is at most a given fixed threshold c.

2.2 RMSD: Root Mean Square Deviations

The RMSD (root mean square deviation) [1, 11, 18, 19, 25, 26] is the most widely-used geometric similarity measure between two sequences of 3-D coordinates that represent molecular structures. The RMSD between two 3-D coordinates sequences $\mathbf{S} = \{\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_n\}$ and $\mathbf{T} = \{\vec{t}_1, \vec{t}_2, \ldots, \vec{t}_n\}$ is defined as the minimum value of $E_{R,\vec{v}}(\mathbf{S}, \mathbf{T}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} |\vec{s}_i - (R \cdot \vec{t}_i + \vec{v})|^2}$ over all the possible rotation matrices R and translation vectors $\vec{v}$. Let $RMSD(\mathbf{S}, \mathbf{T})$ denote the minimum value, and let $\hat{R}(\mathbf{S}, \mathbf{T})$ and $\hat{\vec{v}}(\mathbf{S}, \mathbf{T})$ denote the rotation matrix and the translation vector that minimizes $E_{R,\vec{v}}(\mathbf{S}, \mathbf{T})$.

Kabsch [18, 19] proposed an efficient linear-time algorithm to compute $RMSD(\mathbf{S}, \mathbf{T})$, $\hat{R}(\mathbf{S}, \mathbf{T})$ and $\hat{\vec{v}}(\mathbf{S}, \mathbf{T})$ as follows. If the rotation matrix R is fixed, $E_{R,\vec{v}}(\mathbf{S}, \mathbf{T})$ is known to be minimized when the centroid (center of mass) of $R \cdot \mathbf{T}$ is translated to the centroid of $\mathbf{S}$ by the translation vector $\vec{v}$, regardless of what the rotation matrix R is. It means that $\hat{\vec{v}}(\mathbf{S}, \mathbf{T})$ can be computed in linear time if we are given $\hat{R}(\mathbf{S}, \mathbf{T})$. Moreover, it also means that the problem of computing the RMSD can be reduced to a problem of finding R (*i.e.*, $\hat{R}(\mathbf{S}, \mathbf{T})$) that minimizes $E'_R(\mathbf{S}, \mathbf{T}) = \sum_{i=1} |\vec{s}_i - R \cdot \vec{t}_i|^2$, by translating both $\mathbf{S}$ and $\mathbf{T}$ so that both of their centroids are moved to the origin of the coordinates, which can be done in linear time.

If both structures have been already translated so that both centroids are moved to the origin, we can compute $\hat{R}(\mathbf{S}, \mathbf{T})$ in linear time as follows [1, 18, 19]. Let $J = \sum_{i=1}^{n} \vec{s}_i \cdot \vec{t}_i^{\,t}$. Clearly, J can be computed in $O(n)$ time. Then $E'_R(\mathbf{S}, \mathbf{T})$ can be described as $\sum_{i=1}^{n} (\vec{s}_i^{\,t}\vec{s}_i + \vec{t}_i^{\,t}\vec{t}_i) - 2 \cdot trace(R \cdot J)$, and $trace(R \cdot J)$ is maximized when $R = VU^T$, where $U\Lambda V$ is the singular value decomposition (SVD) of J. Thus $\hat{R}(\mathbf{S}, \mathbf{T})$ can be obtained from J in constant time, as J is a 3×3 matrix and the SVD can be computed in $O(d^3)$ time for a $d \times d$ matrix [16]. Note that there are degenerate cases where $det(VU^T) = -1$, which means that VU^T is a reflection matrix. See [1, 11] for the details of the degenerate cases. Finally, we can compute the RMSD in linear time once we have obtained $\hat{R}(\mathbf{S}, \mathbf{T})$. In total, we can compute the RMSD in $O(n)$ time.

There exists an easily-computable lower bound of the RMSD [27], as in the following. Let $\mathbf{U}^{left}$ denote $\{\vec{u}_1, \vec{u}_2, \dots, \vec{u}_{\lfloor k/2 \rfloor}\}$ and $\mathbf{U}^{right}$ denote $\{\vec{u}_{\lfloor k/2 \rfloor+1}, \vec{u}_{\lfloor k/2 \rfloor+2}, \dots, \vec{u}_{2\cdot\lfloor k/2 \rfloor}\}$ for a structure $\mathbf{U} = \{\vec{u}_1, \vec{u}_2, \dots, \vec{u}_k\}$. Let $G(\mathbf{U})$ denote the centroid of the structure $\mathbf{U}$, *i.e.*, $G(\mathbf{U}) = \frac{1}{k}\sum_{i=1}^{k} \vec{u}_i$. Let $F(\mathbf{U})$ denote $|G(\mathbf{U}^{left}) - G(\mathbf{U}^{right})|/2$, and let $D(\mathbf{S}, \mathbf{T})$ denote $\sqrt{2 \cdot |\mathbf{S}^{left}|/|\mathbf{S}|} \cdot |F(\mathbf{S}) - F(\mathbf{T})|$ for two structures such that $|\mathbf{S}| = |\mathbf{T}|$. Note that $|\mathbf{S}^{left}|/|\mathbf{S}| \approx 1/2$. It is known that $D(\mathbf{S}, \mathbf{T})$ is always smaller than or equal to $RMSD(\mathbf{S}, \mathbf{T})$. Our algorithm in section 3 utilizes this lower bound.

2.3 Freely-Jointed Chain Molecular Model

To analyze the average-case (expected) time complexity of any algorithm, we need some model for random inputs. For example, we usually use just random character strings as a model of random inputs to the algorithms on textual strings in theoretical computer science.

The *freely-jointed chain model* [4, 10, 13, 22] is a simple and well-known fundamental model for average chain molecular structures, and is often used in analyses of the chain molecular behaviors in theoretical molecular physics. The model is also called the *random-walk chain model*, or just the *ideal chain model*. In the model, we assume that the chain molecules can be considered as random walks. It ignores many physical/chemical constraints, but it is known to reflect the behavior of real molecules very well. All the previous expected time complexity analyses of the algorithms on protein 3-D structures are based the model [27, 28]. According to [27], the experimental results on the PDB database consist with their theoretical result based on the freely-jointed chain model. Following them, we also assume in this paper that the structures in the database (*i.e.*, text structures) follow the freely-jointed chain model.[3]

Consider a chain molecule $\mathbf{S} = \{\vec{s}_0, \vec{s}_2, \dots, \vec{s}_n\}$ of length $n+1$, in which the distance between any two adjacent atoms is fixed to some constant r.[4] In the freely-jointed chain model, a bond between two adjacent atoms, *i.e.*, $\vec{b}_i = \vec{s}_{i+1} - \vec{s}_i$, is considered as a random vector that satisfies $|\vec{b}_i| = r$, and $\vec{b}_i$ is considered to be independent from any other bond $\vec{b}_j$ $(j \neq i)$. In this paper, we let $r = 1$ by considering the distance between two adjacent atoms as the unit of distance. If n is large enough, the distribution of the end-to-end vector $\vec{s}_n - \vec{s}_0$ is known to converge to the Gaussian distribution in 3-D space, in which $\langle \vec{s}_n - \vec{s}_0 \rangle = \vec{0}$ and $\langle |\vec{s}_n - \vec{s}_0|^2 \rangle = n$ [4, 10]. In the distribution, the probability (or probability density) that $\vec{s}_n - \vec{s}_0$ is located at some position (x, y, z) is given as follows:

$$W_n(x, y, z)dxdydz = (\frac{3}{2\pi n})^{\frac{3}{2}} e^{-3(x^2+y^2+z^2)/2n} dxdydz. \tag{1}$$

3 The Algorithm

We describe our new algorithm in this section. We first introduce in section 3.1 an efficiently-computable lower bound for the RMSD between each text substructure and the query structure. In section 3.2, we describe a new algorithm that utilizes the lower bound. The algorithm

[3] On the other hand, we give no assumption on the query structures.

[4] In the case of proteins, the distance between two adjacent C_α atoms is fixed to 3.8Å.

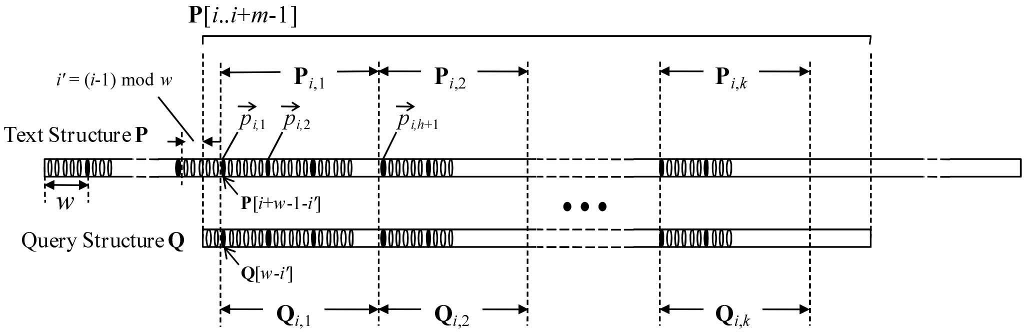

Figure 1: Division of the text and query structures for computing the lower bound B_i of $RMSD(\mathbf{P}[i..i+m-1], \mathbf{Q})$. The black ellipses represent the atoms whose coordinates are used for computing the lower bounds, while the white ellipses represent the dismissed atoms.

first searches for all the substructures with lower bounds that are at most the given threshold c, as candidates of the similar substructures. Then the algorithm checks each candidate whether its RMSD value is actually at most the threshold c. There are two keys for achieving the better expected time. One is that the lower bounds can be checked without accessing all of the coordinates of the text structure, as is described in section 3.1. The other is that not all the lower bounds are actually computed, as is described in section 3.2.

3.1 A Lower Bound of the RMSD

To achieve the time complexity better than linear time, we cannot access all the coordinates of the text structure $\mathbf{P}$.[5] Let k be some positive integer such that $k \leq m/2$, and w be some positive integer such that $w \leq \frac{m+2}{2\cdot(k+1)}$. Note that we will let in section 4 w be some integer in $\Theta(m^{1-\epsilon_1})$ and k be some constant such that $k > 4/\epsilon_1$, where ϵ_1 is an arbitrary small constant such that $0 < \epsilon_1 < 1$, to achieve the expected time complexity of $O(m + N/m^{1+\epsilon})$ (ϵ is an arbitrary small constant such that $\epsilon_1 < \epsilon < 1$). In our algorithm, we use only coordinates of atoms whose indices are multiples of w, *i.e.*, $\mathbf{P}[j \cdot w]$ for some j, to obtain lower bounds of the actual RMSD values.

From now on, we present a lower bound (B_i in the following) for $RMSD(\mathbf{P}[i..i+m-1], \mathbf{Q})$, which satisfies the above requirement, *i.e.*, it must be computable with only the text coordinates whose indices are multiples of w. Let $i' = (i-1) \bmod w$, and $h = \lfloor \frac{m}{k\cdot w} \rfloor$. Let $\vec{p}_{i,j}$ denote $\mathbf{P}[i-1+j\cdot w-i']$ and $\vec{q}_{i,j}$ denote $\mathbf{Q}[j\cdot w-i']$. Let $\mathbf{P}_{i,j} = \{\vec{p}_{i,(j-1)h+1}, \vec{p}_{i,(j-1)h+2}, \ldots, \vec{p}_{i,j\cdot h}\}$, and $\mathbf{Q}_{i,j} = \{\vec{q}_{i,(j-1)h+1}, \vec{q}_{i,(j-1)h+2}, \ldots, \vec{q}_{i,j\cdot h}\}$. Figure 1 shows the locations of these structures on $\mathbf{P}$ and $\mathbf{Q}$. In the following, we describe a lower bound for $RMSD(\mathbf{P}[i..i+m-1], \mathbf{Q})$ that can be computed by using only the coordinates in $\mathbf{P}_{i,1}+\mathbf{P}_{i,2}+\ldots+\mathbf{P}_{i,k}$ and $\mathbf{Q}_{i,1}+\mathbf{Q}_{i,2}+\ldots+\mathbf{Q}_{i,k}$.

Let B_i be $\sqrt{\frac{h}{m}\sum_{j=1}^{k}\{D(\mathbf{P}_{i,j}, \mathbf{Q}_{i,j})\}^2}$, where $D(\mathbf{S}, \mathbf{T})$ is the lower bound of $RMSD(\mathbf{S}, \mathbf{T})$ introduced in section 2.2. Let $\hat{R}_{i,j}$ and $\hat{\vec{v}}_{i,j}$ denote the rotation matrix and the translation vector that optimize $RMSD(\mathbf{P}_{i,j}, \mathbf{Q}_{i,j})$. Similarly, let $\hat{R}_i$ and $\hat{\vec{v}}_i$ denote the rotation matrix

[5] Moreover, we cannot compute all the lower bounds for all the positions of the text structure. We will handle this problem in section 3.2.

and the translation vector that optimize $RMSD(\mathbf{P}[i..i+m-1], \mathbf{Q})$. Then, we can prove that $B_i \leq RMSD(\mathbf{P}[i..i+m-1], \mathbf{Q})$ as follows:

$$\begin{aligned}
B_i &= \sqrt{\frac{h}{m}\sum_{j=1}^{k}\{D(\mathbf{P}_{i,j}, \mathbf{Q}_{i,j})\}^2} \leq \sqrt{\frac{h}{m}\sum_{j=1}^{k}\{RMSD(\mathbf{P}_{i,j}, \mathbf{Q}_{i,j})\}^2} \\
&= \sqrt{\frac{h}{m}\sum_{j=1}^{k}\{\frac{1}{h}\sum_{\ell=1}^{h}\{\vec{p}_{i,(j-1)h+\ell} - \hat{R}_{i,j}\cdot(\vec{q}_{i,(j-1)h+\ell} - \hat{\vec{v}}_{i,j})\}^2\}} \\
&\leq \sqrt{\frac{h}{m}\sum_{j=1}^{k}\{\frac{1}{h}\sum_{\ell=1}^{h}\{\vec{p}_{i,(j-1)h+\ell} - \hat{R}_{i}\cdot(\vec{q}_{i,(j-1)h+\ell} - \hat{\vec{v}}_{i})\}^2\}} \\
&= \sqrt{\frac{1}{m}\sum_{j=1}^{j=h\cdot k}\{\vec{p}_{i,j} - \hat{R}_i\cdot(\vec{q}_{i,j} - \hat{\vec{v}}_i)\}^2} \leq \sqrt{\frac{1}{m}\sum_{j=1}^{m}\{\vec{p}_{i+j-1} - \hat{R}_i\cdot(\vec{q}_{j} - \hat{\vec{v}}_i)\}^2} \\
&= RMSD(\mathbf{P}[i..i+m-1], \mathbf{Q}). \qquad (2)
\end{aligned}$$

3.2 The Search Algorithm

The strategy of our algorithm is very simple and is as follows: We first enumerate all the positions i such that $B_i \leq c$ as candidate positions of the substructures similar to the query, where c is the given threshold. Then we check whether the RMSD between the query and each candidate substructure, *i.e.*, $RMSD(\mathbf{P}[i..i+m-1], \mathbf{Q})$ for a candidate at position i, is actually less than or equal to c. But we cannot compute all the B_i values for all the positions of the text structure if we want to achieve faster algorithm than the linear time, as it apparently requires at least $\Omega(N)$ time. From now on, we discuss how to enumerate all the positions such that $B_i \leq c$ for the given threshold c without computing all the B_i values.

Let $\mathbf{P}^{\triangledown}_i$ denote $\{\vec{p}_{i\cdot w}, \vec{p}_{(i+1)\cdot w}, \ldots, \vec{p}_{(i+h-1)\cdot w}\}$ for integers i such that $1 \leq i \leq \frac{N}{w} - h + 1$, and let $\mathbf{Q}^{\triangle}_j$ denote $\{\vec{q}_j, \vec{q}_{j+w}, \ldots, \vec{q}_{j+(h-1)w}\}$ for integers j such that $1 \leq j \leq m - h\cdot w + 1$.[6] Let L_d be the set of integers i such that $\mathbf{P}_{i,1} = \mathbf{P}^{\triangledown}_d$. Explicitly, $L_d = \{d\cdot w - w + 1, d\cdot w - w + 2, \ldots, d\cdot w\}$, which means $|L_d| = w$. Let M_d denote a set of integers i such that $i \in L_d$ and $B_i \leq c$. Let $M^{all} = M_1 \cup M_2 \cup \ldots \cup M_{\lfloor N/w \rfloor - h + 1}$. Notice that $|M^{all}| = \sum_i^{\lfloor N/w \rfloor - h + 1} |M_i|$, as M_d and $M_{d'}$ are disjoint sets if $d \neq d'$. M^{all} is the set of the candidate positions that we want to obtain, *i.e.*, $B_i \leq c$ *iff* $i \in M^{all}$. From now on, we consider how to efficiently obtain the set M^{all}.

If $B_i = \sqrt{\frac{h}{m}\sum_{j=1}^{k}\{D(\mathbf{P}_{i,j}, \mathbf{Q}_{i,j})\}^2} \leq c$, it is easy to see that $D(\mathbf{P}_{i,j}, \mathbf{Q}_{i,j}) \leq \sqrt{\frac{m}{h}} \cdot c$ for each j $(1 \leq j \leq k)$. Let $K_{d,j}$ be a set of integers i such that $i \in L_d$ and $D(\mathbf{P}_{i,j}, \mathbf{Q}_{i,j}) \leq \sqrt{\frac{m}{h}} \cdot c$. Let $K_d = K_{d,1} \cap K_{d,2} \cap \ldots \cap K_{d,k}$. Notice that $M_d \subseteq K_d$ and consequently $M^{all} \subseteq K^{all}$, where $K^{all} = K_1 \cup K_2 \cup \ldots \cup K_{\lfloor N/w \rfloor - h + 1}$.[7] Thus, we consider how to compute the set K^{all} at first.

In our algorithm, we first obtain the set K_d for each d $(1 \leq d \leq \lfloor \frac{N}{w} \rfloor - h + 1)$ as follows. Recall that $D(\mathbf{S}, \mathbf{T}) = \sqrt{2 \cdot |\mathbf{S}^{left}|/|\mathbf{S}|} \cdot |F(\mathbf{S}) - F(\mathbf{T}|$. (See section 2.2 for the definition of $F(\cdot)$.)

[6] Note that $\mathbf{P}_{i,j} = \mathbf{P}^{\triangledown}_{\lfloor (i-1)/w \rfloor + (j-1)h + 1}$, and $\mathbf{Q}_{i,j} = \mathbf{Q}^{\triangle}_{(j-1)h\cdot w + w - i'}$, where $i' = (i-1) \bmod w$.

[7] If $d \neq d'$, K_d and $K_{d'}$ are disjoint sets as L_d and $L_{d'}$ are disjoint sets.

Thus $K_{d,j}$ can be obtained by finding all $i \in L_d$ such that $F(\mathbf{P}_{i,j}) - c' \leq F(\mathbf{Q}_{i,j}) \leq F(\mathbf{P}_{i,j}) + c'$, where $c' = \sqrt{\frac{m}{2\cdot\lfloor h/2 \rfloor}} \cdot c$. We can compute the values $F(\mathbf{P}_d^{\bigtriangledown})$ for all d $(1 \leq d \leq \frac{N}{w} - h + 1)$ in $O(\frac{N}{w})$ time as follows. Let $\mathbf{P}_i^{\circ}$ denote $\{\vec{p}_{i\cdot w}, \vec{p}_{(i+1)\cdot w}, \ldots, \vec{p}_{(i+h'-1)\cdot w}\}$, where $h' = \lfloor h/2 \rfloor$. As $F(\mathbf{P}_d^{\bigtriangledown}) = |G(\mathbf{P}_d^{\bigtriangledown left}) - G(\mathbf{P}_d^{\bigtriangledown right})|/2$, we can compute the value of $F(\mathbf{P}_d^{\bigtriangledown})$ in constant time if we are given all the vectors $G(\mathbf{P}_i^{\circ})$ for all i $(1 \leq i \leq \frac{N}{w} - h' + 1)$. (See section 2.2 for the definition of $G(\cdot)$.) Moreover, all these $G(\mathbf{P}_i^{\circ})$ values can be computed in $O(\frac{N}{w})$ time, as $G(\mathbf{P}_{i+1}^{\circ}) = G(\mathbf{P}_i^{\circ}) + (\vec{p}_{(i+h')\cdot w} - \vec{p}_{i\cdot w})/h'$. On the other hand, $F(\mathbf{Q}_\ell^{\bigtriangleup})$ can be computed in $O(|\mathbf{Q}_\ell^{\bigtriangleup}|) = O(h)$ time for each ℓ. Thus, we can obtain the values $F(\mathbf{Q}_\ell^{\bigtriangleup})$ for all $\ell = (j-1)h \cdot w + w - i'$ such that $0 \leq i' < w$ and $1 \leq j \leq k$, in $O(h \cdot w \cdot k) = O(m)$ time.

We can enumerate all the integers in $K_d = K_{d,1} \cap K_{d,2} \cap \ldots \cap K_{d,k}$ by utilizing the k-dimensional range search [6, 7] from a set of w points in the k-dimensional space. We can find all the $i \in K_d$ in $O(|K_d| + \log^{k-1} w)$ time after an $O(w \log^{k-1} w)$-time preprocessing on the query structure [6, 7]. Thus K^{all} can be obtained in $O(m + w \log^{k-1} w + \frac{N}{w} \log^{k-1} w + \sum_{d=1}^{\frac{N}{w}-h+1} |K_d|) = O(m + (w + \frac{N}{w}) \log^{k-1} w + |K^{all}|)$ time in total.

Then we can obtain M^{all}, *i.e.*, the set of all the positions i such that $i \in K^{all}$ and $B_i \leq c$, by checking whether the B_i value is actually at most the given threshold c for all $i \in K^{all}$. We can check each B_i value only in $O(k)$ time, as we have already computed all the necessary values of $F(\mathbf{P}_d^{\bigtriangledown})$ and $F(\mathbf{Q}_j^{\bigtriangleup})$. It takes $O(k \cdot |K^{all}|)$ time in total.

Finally, we can obtain the positions with the RMSDs actually smaller than or equal to c, by checking whether $RMSD(\mathbf{P}[i..i+m-1], \mathbf{Q})$ is actually at most the threshold c for all $i \in M^{all}$, which takes $O(m)$ time for each with Kabsch's algorithm described in section 2.2. It takes $O(m \cdot |M^{all}|)$ time in total.

Thus, the total time complexity of our algorithm is $O(m + (w + \frac{N}{w}) \log^{k-1} w + m \cdot |K^{all}|)$, as $|M^{all}| \leq |K^{all}|$. It could be $O(Nm)$ at worst, as $|K^{all}|$ could be at most $O(N)$. But we will show in the next section that such bad cases must be very rare under the assumption that the text structure $\mathbf{P}$ follows the freely-jointed chain model.[8] We will show that the expected time complexity of our algorithm is only $O(m + N/m^{1-\epsilon})$, where ϵ is an arbitrary small positive constant such that $0 < \epsilon < 1$, by setting w and k appropriately.

4 The Expected Time Complexity of the Algorithm

Consider a structure $\mathbf{S} = \{\vec{s}_1, \vec{s}_2, \ldots, \vec{s}_{1+(h-1)\cdot w}\}$ of length $1 + (h-1) \cdot w$, which follows the random walk model. Let $\vec{b}_i = \vec{s}_{i+1} - \vec{s}_i$. b_i is a random vector that satisfies $|\vec{b}_i| = 1$, and $\vec{b}_i$ and $\vec{b}_j$ are independent if $i \neq j$. $\vec{s}_i$ can be represented as $\vec{s}_1 + \sum_{j=1}^{i-1} \vec{b}_j$. Let $\mathbf{S}^{\bigtriangledown}$ denote $\{\vec{s}_1, \vec{s}_{1+w}, \vec{s}_{1+2w}, \ldots, \vec{s}_{1+(h-1)\cdot w}\}$, and $h' = \lfloor h/2 \rfloor$. Let $H(\mathbf{U}) = (G(\mathbf{U}^{left}) - G(\mathbf{U}^{right}))/2$ for some structure $\mathbf{U}$.[9] Then the following equation holds:

$$H(\mathbf{S}^{\bigtriangledown}) = \frac{1}{2}\{G(\mathbf{S}^{\bigtriangledown left}) - G(\mathbf{S}^{\bigtriangledown right})\}$$

[8] The query structure $\mathbf{Q}$ does not have to follow any model. But the same can be said if the query structure $\mathbf{Q}$ follows the freely-jointed chain model, instead of the text structure $\mathbf{P}$.

[9] Notice that $F(\mathbf{U}) = |H(\mathbf{U})|$.

$$
\begin{aligned}
&= \frac{1}{2\cdot h'}\{\sum_{i=1}^{h'}(\vec{s}_1 + \sum_{j=1}^{(i-1)\cdot w}\vec{b}_j) - \sum_{i=h'+1}^{2h'}(\vec{s}_1 + \sum_{j=1}^{(i-1)\cdot w}\vec{b}_j)\} \\
&= -\frac{1}{2\cdot h'}\{\sum_{i=1}^{h'} i\cdot(\sum_{j=1+(i-1)w}^{i\cdot w}\vec{b}_j) + \sum_{i=h'+1}^{2h'-1}(2h'-i)\cdot(\sum_{j=1+(i-1)w}^{i\cdot w}\vec{b}_j)\}. \qquad (3)
\end{aligned}
$$

Let $\vec{b}'_i$ denote $\{(1+\lfloor\frac{i-1}{w}\rfloor)\cdot\vec{b}_i\}/2h'$ if $i \le h'\cdot w$, and $\{(2h'-1-\lfloor\frac{i-1}{w}\rfloor)\cdot\vec{b}_i\}/2h'$ if $i > h'\cdot w$. Then $H(\mathbf{S}^{\bigtriangledown})$ can be described as $\sum_{i=1}^{2h'w}\vec{b}'_i$. Let z_i denote the z-coordinate of $\vec{b}_i$, and z'_i denote the z-coordinate of $\vec{b}'_i$. It is easy to see that $\langle z_i\rangle = 0$ and $var(z_i) = 1/3$, as $\vec{b}_i$ is a random vector that satisfies $|\vec{b}_i| = 1$. Then, consider the value $A_{h,w} = \sum_{i=1}^{2h'\cdot w}\langle|z'_i - \langle z'_i\rangle|\rangle^{2+\delta}/\sqrt{\sum_{i=1}^{2h'\cdot w} var(z'_i)}^{2+\delta}$, where δ is some positive constant. According to Lyapunov's central limit theorem [20], the distribution of $H(\mathbf{S}^{\bigtriangledown}) = \sum_{i=1}^{2h'w}\vec{b}'_i$ converges to the Gaussian distribution if the above $A_{h,w}$ converges to 0 as $2h'\cdot w$ grows up to infinity, for some δ ($\delta > 0$). It can be proved as follows:

$$
\begin{aligned}
A_{h,w} &= \frac{1}{\sqrt{\sum_{i=1}^{2h'\cdot w}\langle|z'_i|\rangle^2}^{2+\delta}}\sum_{i=1}^{2h'\cdot w}\langle|z'_i|\rangle^{2+\delta} \le \frac{1}{\sqrt{\sum_{i=1}^{2h'\cdot w}\langle|z'_i|\rangle^2}^{2+\delta}}\sum_{i=1}^{2h'\cdot w}\langle|z'_i|\rangle^2 \\
&= \{\sum_{i=1}^{2h'\cdot w}\langle|z'_i|\rangle^2\}^{-\delta/2} = \{\frac{w}{12h'^2}\cdot\{\sum_{i=1}^{h'} i^2 + \sum_{i=h'+1}^{2h'-1}(2h'-i)^2\}\}^{-\delta/2} \\
&= \{\frac{1}{18}h'\cdot w + \frac{w}{36h'}\}^{-\delta/2} \;\rightarrow\; 0 \quad (2h'\cdot w \rightarrow \infty). \qquad (4)
\end{aligned}
$$

The same discussion can be done for the other two axes (x and y), which are independent to each other. Thus we conclude that $H(\mathbf{S}^{\bigtriangledown})$ converges to a Gaussian distribution in 3-D space. It is easy to see that $\langle H(\mathbf{S}^{\bigtriangledown})\rangle = 0$. The variance of $H(\mathbf{S}^{\bigtriangledown})$ can be computed as follows:

$$
\begin{aligned}
var(H(\mathbf{S}^{\bigtriangledown})) &= \langle|H(\mathbf{S}^{\bigtriangledown})|^2\rangle - \langle|H(\mathbf{S}^{\bigtriangledown})|\rangle^2 = \langle|H(\mathbf{S}^{\bigtriangledown})|^2\rangle \\
&= w\cdot\langle|\sum_{i=1}^{2h'}\frac{i}{2h'}\cdot\vec{b}_i + \sum_{i=h'+1}^{2h'-1}\frac{2h'-i}{h'}\cdot\vec{b}_i|^2\rangle \\
&= \frac{w}{4h'^2}\{\sum_{i=1}^{h'} i^2 + \sum_{i=h'+1}^{2h'-1}(2h'-i)^2\} = \frac{1}{6}h'\cdot w + \frac{w}{12h'} \approx \frac{1}{6}h'\cdot w, \qquad (5)
\end{aligned}
$$

It means that the distribution of $H(\mathbf{S}^{\bigtriangledown})$ is the same as that of the end-to end vectors of random walks of length $h'\cdot w/6$. Thus, the probability distribution of $H(\mathbf{S}^{\bigtriangledown})$ is:

$$
Z_{h,w}(x,y,z)dxdydz = (\frac{9}{\pi h'w})^{\frac{3}{2}}e^{-9(x^2+y^2+z^2)/h'w}dxdydz. \qquad (6)
$$

Consequently, the probability (probability density) that $|H(\mathbf{S}^{\bigtriangledown})| = r$ is:

$$
Z_{h,w}(r)dr = 4\pi r^2(\frac{9}{\pi h'w})^{\frac{3}{2}}e^{-9r^2/h'w}dr. \qquad (7)
$$

Integrating $Z_{h,w}(r)dr$, we obtain $Prob(x \le F(\mathbf{S}^{\bigtriangledown}) \le y) = Prob(x \le |H(\mathbf{S}^{\bigtriangledown})| \le y) = \int_{r=x}^{y} Z_{h,w}(r)dr$. $Z_{h,w}(r)$ takes the maximum value at $r_{max} = \frac{1}{3}\sqrt{h'w}$, and $Z_{h,w}(r_{max}) =$

$12e^{-1}/\sqrt{\pi h' w}$. Thus $Prob(x \leq F(\mathbf{S}^{\bigtriangledown}) \leq y)$ is at most $(y-x) \cdot Z_{h,w}(r_{max}) = 12e^{-1}(y-x)/\sqrt{\pi h' w}$ for any x and y such that $x < y$.

To evaluate the expected time complexity of the algorithm proposed in the previous section, we have to investigate the expected size of K_d under the assumption that the text structure $\mathbf{P}$ follows the freely-jointed chain model.[10] Recall that $K_{d,j}$ is a set of integers i such that $i \in L_d$ and $D(\mathbf{P}_{i,j}, \mathbf{Q}_{i,j}) \leq \sqrt{\frac{m}{h}} \cdot c$. Thus $\langle |K_{d,j}| \rangle = p \cdot |L_d| = p \cdot w$, where $p = Prob(D(\mathbf{P}_{i,j}, \mathbf{Q}_{i,j}) \leq \sqrt{\frac{m}{h}} \cdot c)$. As $D(\mathbf{S}, \mathbf{T}) = |F(\mathbf{S}) - F(\mathbf{T})|$, p equals $Prob(F(\mathbf{Q}_{i,j}) - \sqrt{\frac{m}{h}} \cdot c \leq F(\mathbf{P}_{i,j}) \leq F(\mathbf{Q}_{i,j}) + \sqrt{\frac{m}{h}} \cdot c)$. According to the above discussion, p is at most $p_{max} = Z_{h,w}(r_{max}) \cdot 2 \cdot \sqrt{\frac{m}{h}} \cdot c = 24c \cdot e^{-1} \cdot \sqrt{m/(\pi \cdot h \cdot h' \cdot w)}$, for any i, j and query $\mathbf{Q}$. The value p_{max} is in $O(k\sqrt{w/m})$, as $h = \lfloor m/(k \cdot w) \rfloor$, $h' = \lfloor h/2 \rfloor$ and c is a fixed constant. As the text structure $\mathbf{P}$ follows the freely-jointed chain model, $\mathbf{P}_{i,j}$ and $\mathbf{P}_{i,j'}$ $(j \neq j')$ are built by independent random walks. It means that $K_{d,j}$ and $K_{d,j'}$ $(j \neq j')$ are independently chosen from L_d. Thus the expected size of $K_d = K_{d,1} \cap K_{d,2} \cap \ldots \cap K_{d,k}$ is at most $w \cdot p_{max}^k$, which is in $O((c' \cdot k)^k \cdot w \cdot (w/m)^{k/2})$, where c' is some constant. Thus $\langle |K^{all}| \rangle$ is in $O((c' \cdot k)^k \cdot N \cdot (w/m)^{k/2})$.

Thus, the expected time complexity of the algorithm described in the last section is $O(m + (w + \frac{N}{w}) \log^{k-1} w + (c' \cdot k)^k \cdot N \cdot w^{k/2} \cdot m^{1-k/2})$. Let w be some integer in $\Theta(m^{1-\epsilon_1})$ and k be some constant such that $k > 4/\epsilon_1$, where ϵ_1 is an arbitrary small constant such that $0 < \epsilon_1 < 1$. Then the above expected time complexity becomes $O(m + N \cdot \frac{\log^{k-1} m}{m^{1-\epsilon_1}})$. The '$\log^{k-1} m$' term can be replaced by m^{ϵ_2} where ϵ_2 is an arbitrary small constant such that $0 < \epsilon_2 < 1 - \epsilon_1$. Let $\epsilon = \epsilon_1 + \epsilon_2$. Then, the above time complexity can be simplified to $O(m + N/m^{1-\epsilon})$. Note that we can let ϵ be an arbitrary small constant such that $0 < \epsilon < 1$.

5 Concluding Remarks

We proposed a new algorithm for the 3-D substructure search problem, which is the problem to find all the similar substructures from a 3-D structure database of chain molecules. The algorithm runs in average-case $O(m + N/m^{1-\epsilon})$ time where N is the database size, m is the query size and ϵ is an arbitrary small constant such that $0 < \epsilon < 1$, while the best-known expected time complexity of the problem was $O(N)$. Our algorithm is the first algorithm that runs in faster than linear expected time.

We improved the expected time complexity, but the coefficient before the complexity is large in case ϵ is small. It is an open problem whether we can reduce it or not. It is also an open problem whether we can improve the worst-case time complexity. Our algorithm can also be applied to similar problems in different dimensions. It would be very interesting to apply it to many other problems in higher dimensions.

Acknowledgement

The author would like to thank Jesper Jansson, Gregory Kucherov, and Kunihiko Sadakane for invaluable discussions. This work was partially supported by the Grant-in-Aid from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

[10] The same discussion can be done if the query structure $\mathbf{Q}$ follows the freely-jointed chain model instead of the text structure $\mathbf{P}$.

References

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

[2] Z. Aung and K.-L. Tan. Rapid retrieval of protein structures from databases. *Drug Discovery Today*, Vol. 12, pp. 732–739, 2007.

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

[4] R. H. Boyd and P. J. Phillips. *The Science of Polymer Molecules: An Introduction Concerning the Synthesis, Structure and Properties of the Individual Molecules That Constitute Polymeric Materials*, Cambridge University Press, 1996.

[5] R. S. Boyer and J. S. Moore. A fast string searching algorithm, *Commun. ACM*, Vol. 20, pp. 762–772, 1977.

[6] B. Chazelle. Filtering search: a new approach to query-answering, *SIAM J. Comput.*, Vol. 15, No. 3, pp.703–724, 1986.

[7] B. Chazelle. A functional approach to data structures and its use in multidimensional searching, *SIAM J. Comput.*, Vol. 17, No. 3, pp. 427–453, 1988.

[8] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series, *Math. Comput.*, Vol. 19, pp. 297–301, 1965.

[9] J. Dayantis and J.-F. Palierne. Monte Carlo precise determination of the end-to-end distribution function of self-avoiding walks on the simple-cubic lattice. *J. Chem. Phys.*, Vol. 95, pp. 6088–6099, 1991.

[10] P.-G. de Gennes. *Scaling Concepts in Polymer Physics*, Cornell University Press, 1979.

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

[12] I. Eidhammer, I. Jonassen, and W. R. Taylor. Structure comparison and structure patterns. *J. Computational Biology*, Vol. 7, No. 5, pp. 685–716, 2000.

[13] P. J. Flory. *Statistical Mechanics of Chain Molecules*, Interscience, New York, 1969.

[14] M. Gerstein. Integrative database analysis in structural genomics. *Nat. Struct. Biol.*, Suppl., pp. 960–963, 2000.

[15] D. Goldman, S. Istrail and C. H. Papadimitriou. Algorithmic aspects of protein structure similarity. *Proc. 40th Annual Symposium on Foundations of Computer Science*, pp. 512–522, 1999.

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

[17] H. Hasegawa and L. Holm. Advances and pitfalls of protein structural alignment. *Current Opinion in Structural Biology*, Vol. 19, pp. 341–348, 2009.

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

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

[20] O. Kallenberg. *Foundations of Modern Probability,* Springer-Verlag, 1997.

[21] P. Koehl. Protein structure similarities. *Current Opinion in Structural Biology*, Vol. 11, pp. 348–353, 2001.

[22] H. A. Kramers. The behavior of macromolecules in inhomogeneous flow. *J. Chem. Phys.*, Vol. 14, No. 7, pp. 415–424, 1946.

[23] E. Krissinel and K. Henrick. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. *Acta Cryst. Sect.*, Vol. D60, pp. 2256–2268, 2004.

[24] A. C. R. Martin. The ups and downs of protein topology: rapid comparison of protein structure. *Protein Engineering*, Vol. 13, pp. 829–837, 2000.

[25] J. T. Schwartz and M. Sharir. Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves. *Intl. J. of Robotics Res.*, Vol. 6, pp. 29–44, 1987.

[26] T. Shibuya. Efficient substructure RMSD query algorithms. *J. Comput. Biol.*, Vol. 14, No. 9, pp. 1201–1207, 2007.

[27] T. Shibuya, Searching protein 3-D structures in linear time. *Proc. Conference on Research in Computational Molecular Biology* (RECOMB '09), LNBI 5541, pp. 1–15, 2009.

[28] T. Shibuya, J. Jansson, and K. Sadakane, Linear-Time Protein 3-D Structure Searching with Insertions and Deletions, *Proc. 9th Workshop on Algorithms in Bioinformatics*, 2009, to appear.