Linear-Time Protein 3-D Structure Searching with Insertions and Deletions

Tetsuo Shibuya[1], Jesper Jansson[2], and Kunihiko Sadakane[3]

[1] Human Genome Center, Institute of Medical Science, University of Tokyo
4-6-1 Shirokanedai, Minato-ku, Tokyo 108-8639, Japan
[2] Ochanomizu University, 2-1-1 Ohtsuka, Bunkyo-ku, Tokyo 112-8610, Japan
[3] National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
tshibuya@hgc.jp, Jesper.Jansson@ocha.ac.jp, sada@nii.ac.jp

Abstract. It becomes more and more important to search for similar structures from molecular 3-D structure databases in the structural biology of the post genomic era. Two molecules are said to be similar if the RMSD (root mean square deviation) of the two molecules is less than or equal to some given constant bound. In this paper, we consider an important, fundamental problem of finding all the similar substructures from 3-D structure databases of chain molecules (such as proteins), with consideration of indels (*i.e.*, insertions and deletions). The problem has been believed to be very difficult, but its computational difficulty has not been well known. In this paper, we first show that the same problem in arbitrary dimension is NP-hard. Moreover, we also propose a new algorithm that dramatically improves the average-case time complexity for the problem, in case the number of indels k is bounded by some constant. Our algorithm solves the above problem in average $O(N)$ time, while the time complexity of the best known algorithm was $O(Nm^{k+1})$, for a query of size m and a database of size N.

1 Introduction

Molecules with similar 3-D structures are said to have similar functions. It means that we can predict the molecular function by searching for molecules with similar structures in the databases. Thus, finding similar 3-D structures from 3-D databases is very important [2,10,12,17]. Due to recent technological evolution of molecular structure determination methods, more and more structures of biomolecules, especially proteins, are solved, as shown in the PDB (Protein Structure Data Bank) database [3]. Moreover, a huge number of molecular structures are predicted with various computational techniques recently. Hence, faster searching techniques against these molecular structure databases are seriously needed.

A protein is a chain of amino acids, and its structure is often represented by a sequence of 3-D coordinates that represents the positions of amino acids. Usually, the 3-D coordinates of the C_α atom in each amino acid is used as the representative position of that amino acid. Note that there are also other important chain molecules in living cells, such as DNAs, RNAs, and glycans. In this

S.L. Salzberg and T. Warnow (Eds.): WABI 2009, LNBI 5724, pp. 310–320, 2009.

paper, we consider a problem of searching for similar structures from a structure database of chain molecules, which consists of sequences of 3-D coordinates that represent molecular structures.

The RMSD (Root Mean Square Deviation) [1,9,14,15,17,20,21] is the most widely-used similarity measure between molecular structures, which is also used in various other fields, such as robotics and computer vision. It determines geometric similarity between two same-length sequences of 3-D coordinates. 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.) The RMSD measure corresponds to the Hamming distance in the textual pattern matching, from the viewpoint that it does not consider any indels (*i.e.*, insertions and deletions) between them. In the case of textual string comparison, especially comparison of two textual strings of bio-molecules (such as proteins and DNA), we often prefer to use the string alignment score that considers indels to compare two bio-sequences, rather than the Hamming distance. Likewise, it is also important to consider indels when we compare two molecular 3-D structures. But it is much harder than the textual string cases to compare two 3-D structures with consideration of indels, though an ordinary pair-wise alignment algorithm for textual strings requires only quadratic time.

In this paper, we consider a problem of searching for substructures of database structures whose RMSDs to a given query is within some constant, permitting indels. It is widely known that the contact map problem [13] is NP-hard and the structure alignment problems are believed to be very difficult. But the difficulty of our problem is unknown, as our problem is different from the contact map problem. We show in this paper that our problem is also NP-hard if the dimension of the problem is arbitrary. But it does not mean that our problem is always difficult. If the number of indels is at most some constant, the problem can be solved in polynomial time, though the time complexity of known algorithms is still very large. The best-known algorithm for the problem is a straight-forward algorithm that requires $O(Nm^{k+1})$ time for a database of size N and a query of size m, where k is the maximum number of indels. It is the worst-case time complexity, but the average-case time complexity of the algorithm is still all the same $O(Nm^{k+1})$. We propose in this paper a much faster algorithm that runs in average-case $O(N)$ time, assuming that the database structures can be considered as random walks. The model under this assumption is called the 'random-walk model' (It is also called the 'freely-jointed chain model' or just the 'ideal chain model'. See section 2.3 for more details.), and is very often used in molecular physics [4,8,11,18]. It is also used in the analysis of algorithms for protein structure comparison [22]. As demonstrated in [22], theoretical analyses based on the random-walk model have high consistency with the actual experimental results on the PDB database.

The organization of this paper is as follows. Section 2 describes the notations used in this paper and previous related work as preliminaries. Section 3 describes the problem that we solve. Section 4 describes the NP-hardness of our problem.

Section 5 describes our new algorithm and the computational time analysis of the algorithm. Section 6 concludes our results.

2 Preliminaries

2.1 Notations and Definitions

A chain molecule $\mathbf{S}$ whose i-th 3-D coordinates (vector) is $\boldsymbol{s}_i$ is noted as $\mathbf{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n)$. The length n of $\mathbf{S}$ is denoted by $|\mathbf{S}|$. A structure $\mathbf{S}[i..j] = (\boldsymbol{s}_i, \boldsymbol{s}_{i+1}, \ldots, \boldsymbol{s}_j)$ $(1 \leq i \leq j \leq n)$ is called a *substructure* of $\mathbf{S}$. A structure $\mathbf{S}' = (\boldsymbol{s}_{a_1}, \boldsymbol{s}_{a_2}, \ldots, \boldsymbol{s}_{a_\ell})$ $(1 \leq a_1 < a_2 < \ldots < a_\ell \leq n)$ is called a *subsequence structure* of $\mathbf{S}$. $\mathbf{S}'$ is also called a *k-reduced subsequence structure* of $\mathbf{S}$, where $k = |\mathbf{S}| - |\mathbf{S}'|$. $R \cdot \mathbf{S}$ denotes the structure $\mathbf{S}$ rotated by the rotation matrix R, *i.e.*, $R \cdot \mathbf{S} = (R\boldsymbol{s}_1, R\boldsymbol{s}_2, \ldots, R\boldsymbol{s}_n)$. $|\boldsymbol{v}|$ denotes the norm of the vector $\boldsymbol{v}$. $\mathbf{0}$ denotes the zero vector. $\langle x \rangle$ denotes the expected value of x. $Prob(\mathcal{X})$ denotes the probability of the event $\mathcal{X}$.

2.2 RMSD: Root Mean Square Deviation

The RMSD (root mean square deviation) [1,9,14,15,20,21] is the most widely-used geometric similarity measure between two sequences of 3-D coordinates. The RMSD between two 3-D coordinates sequences $\mathbf{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n)$ and $\mathbf{T} = (\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_n)$ is defined as the minimum value of $\sqrt{\frac{1}{n}\sum_{i=1}^{n} |\boldsymbol{s}_i - (R \cdot \boldsymbol{t}_i + \boldsymbol{v})|^2}$ over all the possible rotation matrices R and translation vectors $\boldsymbol{v}$. Let $RMSD(\mathbf{S}, \mathbf{T})$ denote the minimum value. $RMSD(\mathbf{S}, \mathbf{T})$ can be computed in $O(n)$ time [1,9,14,15]. Note that the RMSD can be defined in any other dimensions, by considering the above vectors and matrices are in any d-dimensions.

2.3 Random-Walk Model for Chain Molecules

The *random-walk model* (also called the *freely-jointed chain model*, or just the *ideal chain model*), is a very widely used simple model for analyzing behavior of chain molecules in molecular physics [4,8,11,18]. The model is also used for analyzing the computational time complexities of algorithms for protein structures [22]. In the model, we assume that the chain molecules can be considered as random walks. The model ignores many physical/chemical constraints, but it is known to reflect the behavior of real molecules very well. In fact, experiments in [22] showed high consistency between the experimental results obtained from the PDB database and the theoretical results deduced from the random-walk model. Consider a chain molecule $\mathbf{S} = (\boldsymbol{s}_0, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n)$ of length $n+1$, in which the distance between any two adjacent atoms is fixed to some constant r.[1] In the random-walk model, a bond between two adjacent atoms, *i.e.*, $\boldsymbol{b}_i = \boldsymbol{s}_{i+1} - \boldsymbol{s}_i$, is considered as a random vector that satisfies $|\boldsymbol{b}_i| = r$, and $\boldsymbol{b}_i$ is considered to be independent from any other bond $\boldsymbol{b}_j$ $(j \neq i)$.

[1] In the case of proteins, the distance between two adjacent C_α atoms is fixed to 3.8Å. We can let $r = 1$ by considering the distance between two adjacent atoms as the unit of distance.

2.4 Shibuya's Lower Bound of the RMSD [22]

Let $\mathbf{U}^{left}$ denote $(\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_{\lfloor \ell/2 \rfloor})$ and $\mathbf{U}^{right}$ denote $(\boldsymbol{u}_{\lfloor \ell/2 \rfloor+1}, \boldsymbol{u}_{\lfloor \ell/2 \rfloor+2}, \ldots, \boldsymbol{u}_{2\cdot\lfloor \ell/2 \rfloor})$ for a structure $\mathbf{U} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_\ell)$. Let $G(\mathbf{U})$ denote the centroid of the structure $\mathbf{U}$, *i.e.*, $G(\mathbf{U}) = \frac{1}{\ell}\sum_{i=1}^{\ell} \boldsymbol{u}_i$. 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 $|\mathbf{S}| = |\mathbf{T}|$. Shibuya proved in [22] that $D(\mathbf{S}, \mathbf{T})$ is always smaller than or equal to $RMSD(\mathbf{S}, \mathbf{T})$. In [22], he also proved the following lemma:

Lemma 1 (Shibuya [22]). *The probability* $Prob(D(\mathbf{S}, \mathbf{T}) < c)$ *is in* $O(c/\sqrt{n})$, *where* $n = |\mathbf{S}| = |\mathbf{T}|$, *under the assumption that either* $\mathbf{S}$ *or* $\mathbf{T}$ *follows the random-walk model.*

3 The *k*-Indel 3-D Substructure Search Problem

From now on, we deal with the following problem.

***k*-Indel 3-D Substructure Search Problem:** We are given a text structure $\mathbf{P}$ of size N and a query structure $\mathbf{Q}$ of size m $(1 < m \leq N)$, both of which are represented by 3-D coordinates sequences of the residues. We are also given a constant positive real c and a small constant positive integer k $(k \ll m)$. The problem is to find all the positions i $(1 \leq i \leq N - m + k + 1)$ such that the RMSD between some k'-reduced subsequence structure of $\mathbf{Q}$ and some k''-reduced subsequence structure of $\mathbf{P}[i..i - k' + k'' + m - 1]$ is at most c, for some non-negative integers k' and k'' $(k' + k'' \leq k,\ k'' - k' \leq N - m - i + 1)$.

If there exists some triple set $\{i, k', k''\}$ that satisfies the above condition, we say that $\mathbf{Q}$ matches with $\mathbf{P}[i..i - k' + k'' + m - 1]$ with threshold c and (at most) $k' + k''$ indels. Usually, c is set to a constant proportional to the distance between two adjacent residue coordinates in the molecular structures. In the case of protein structures, c is often set to 1–2Å, while the distance between two adjacent C_α atoms is 3.8Å. Usual structure databases may contain more than 1 structures, but problems against the databases with multiple structures can be reduced to the above single-text problem by just concatenating all the structures into a single long text structure and ignoring matches that cross over the boundaries of two concatenated structures.

The same problem without indels, *i.e.*, the problem in case $k = 0$, is studied very well. If we directly apply the Kabsch's algorithm [14,15] introduced in section 2.2, the problem without indels can be solved in $O(Nm)$ time. For the problem, Schwartz and Sharir [20] proposed an algorithm based on the fast Fourier transform technique that runs in $O(N \log m)$ time.[2] Recently, Shibuya [22] proposed a breakthrough average-case (expected) linear-time algorithm, assuming that the text structures follow the random-walk model. He showed that his

[2] The original algorithm runs in $O(N \log N)$ time. See [22] for the technique to improve it into $O(N \log m)$.

algorithm is much faster than other algorithms also in practice. Moreover he showed that the experimental results on the whole PDB database agrees with the theoretical analysis based on the random-walk model. But none of these algorithms considers any indels.

On the other hand, there have been almost no algorithmic study for cases $k > 0$, due to the difficulty of the problem, though the problem is very important. The difficulty of the problem is not well known, though the problem is similar to the famous contact map problem, which is known to be NP-hard [13]. In section 4, we will show that the problem is NP-hard, in case the dimension of the problem is arbitrary.

According to section 2.2, the RMSD between two structures of size m can be computed in $O(m)$ time. The possible number of subsequence structures to be compared in the k-indel 3-D substructure search problem is less than ${}_{2m+k}C_k \cdot N$, which is in $O(Nm^k)$. Thus, our problem can be computed in $O(Nm^{k+1})$ time, either in the worst-case analysis or in the average-case analysis. As far as we know, it is the best-known time complexity, and there have been known no algorithms other than the above straight-forward algorithm. But it also means that the problem can be computed in polynomial time, in case the number of indels is bounded by some constant. In section 5, we will propose the first algorithm with better average-case time complexity, *i.e.*, $O(N)$, for the above problem in case the number of the indels is at most some constant, which is a substantial improvement for the problem. Note that the worst-case time complexity of our algorithm is still the same as the above straight-forward algorithm. Note also that our analysis of the average-case time complexity is based on the assumption that the text structure follows the random-walk model,[3] like the analyses in [22].

4 An NP-Hardness Result

Consider the following variant of the k-indel 3-D substructure search problem.

k-Indel Structure Comparison Problem: We are given two structures $\mathbf{P}$ and $\mathbf{Q}$, both of whose lengths are n. Find a k-reduced subsequence structure $\mathbf{P}'$ of $\mathbf{P}$ and a k-reduced subsequence structure $\mathbf{Q}'$ of $\mathbf{Q}$, such that the RMSD between $\mathbf{P}'$ and $\mathbf{Q}'$ is at most some given threshold c.

It is trivial that the k-indel structure comparison problem is in the class NP, as the correctness of any instance can be checked in linear time. Moreover, it is also trivial that the k-indel 3-D substructure search problem is at least as difficult as the k-indel comparison problem in 3-D, and the k-indel 3-D substructure search problem is NP-hard if the k-indel structure comparison problem in 3-D is NP-complete. The two problems can be extended to the problems in any dimensional space. From now on, we show the k-indel structure comparison problem in arbitrary dimension is NP-complete, by reduction from the following

[3] We give this random-walk assumption only on the database structures, and we give no assumption on the query structures.

k-cluster problem (or the densest k-subgraph problem), whose decision problem is known to be NP-complete [6].

k-Cluster Problem (Densest k-Subgraph Problem): Given a graph $G = (V, E)$, find a size k subset of V such that the number of edges induced by the subset is the largest.

Let $V = \{v_1, v_2, \ldots, v_n\}$. Consider an arbitrary subset $V' = \{v_{g_1}, v_{g_2}, \ldots, v_{g_k}\}$ of V, where $g_1 < g_2 < \ldots < g_k$, and let x be the number of edges induced by V'.

There must exist a sequence of points $\mathbf{P} = (\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n)$ in $n-1$ dimensional space, such that $|\boldsymbol{p}_i - \boldsymbol{p}_j| = \alpha$ if $\{v_i, v_j\} \in E$ and $|\boldsymbol{p}_i - \boldsymbol{p}_j| = \beta$ if $\{v_i, v_j\} \notin E$, where α and β are any constants that satisfy $0 < \alpha < \beta < 2\alpha$. Let $\mathbf{Q}$ be a sequence of n zero vectors $(\mathbf{0}, \ldots, \mathbf{0})$ in the same $n-1$ dimensional space. Let $\mathbf{P}_{V'} = (\boldsymbol{p}_{g_1}, \boldsymbol{p}_{g_2}, \ldots, \boldsymbol{p}_{g_k})$, and $\mathbf{Q}_{V'}$ be a sequence of k zero vectors $(\mathbf{0}, \ldots, \mathbf{0})$ in the $n-1$ dimensional space.

It is well known that the translation of the two structures in 3-D is optimized when the centroids of the two structures are placed at the same position (*e.g.*, at the origin of the coordinates) [1,14], in computing the RMSD. It is also true in any dimensions d, which can be easily proved as follows. Consider two arbitrary d-dimensional structures $\mathbf{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n)$ and $\mathbf{T} = (\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_n)$, and an arbitrary d-dimensional translation vector $\boldsymbol{v}$. Then the following equation holds:

$$\sum_{i=1}^{n}(\boldsymbol{s}_i - \boldsymbol{t}_i + \boldsymbol{v})^2 = n\{\boldsymbol{v} + \frac{\sum_{i=1}^{n}(\boldsymbol{s}_i - \boldsymbol{t}_i)}{n}\}^2 + \sum_{i=1}^{n}(\boldsymbol{s}_i - \boldsymbol{t}_i)^2 - \frac{\{\sum_{i=1}^{n}(\boldsymbol{s}_i - \boldsymbol{t}_i)\}^2}{n}. \quad (1)$$

Thus the translation is optimized when $\boldsymbol{v} = -\frac{\sum_{i=1}^{n}(\boldsymbol{s}_i - \boldsymbol{t}_i)}{n}$. It means that the translation is optimized when the two structures are moved so that the centroids of the two structures are at the same position.

From now on, we consider computing the RMSD between $\mathbf{P}_{V'}$ and $\mathbf{Q}_{V'}$. It is trivial that the centroid of $\mathbf{Q}_{V'}$ is at the origin of the coordinates, and moreover $\mathbf{Q}_{V'}$ does not change its shape by any rotation, as all the vectors in $\mathbf{Q}_{V'}$ are zero vectors. Hence, we do not have to consider the optimization of the rotation for computing the RMSD between the two structures. Therfore we obtain the following equation:

$$\begin{aligned} RMSD(\mathbf{P}_{V'}, \mathbf{Q}_{V'}) &= \{\sum_{i=1}^{k}(\boldsymbol{p}_{g_i} - \frac{\sum_{j=1}^{k}\boldsymbol{p}_{g_j}}{k})^2/k\}^{1/2} \\ &= \{\sum_{i=1}^{k-1}\sum_{j=i+1}^{k}(\boldsymbol{p}_{g_i} - \boldsymbol{p}_{g_j})^2/k\}^{1/2} \\ &= \{(\alpha^2 \cdot x + \beta^2 \cdot (\frac{k(k-1)}{2} - x))/k\}^{1/2} \end{aligned} \quad (2)$$

It means that $RMSD(\mathbf{P}_{V'}, \mathbf{Q}_{V'})$ is smaller if x is larger, as $0 < \alpha < \beta$. Thus we can obtain the answer of the decision problem of the k-cluster problem by solving the $(n-k)$-indel $n-1$ dimensional structure comparison problem on

the two structures $\mathbf{P}$ and $\mathbf{Q}$. Hence the k-indel structure comparison problem in arbitrary dimensional space is NP-complete, and consequently we conclude that the k-indel substructure search problem in arbitrary dimensional space is NP-hard:

Theorem 1. *The k-indel substructure search problem in arbitrary dimensional space is NP-hard.*

5 The New Linear Expected Time Algorithm

5.1 The Algorithm

To improve the performance of the algorithms for approximate matching of ordinary textual strings, we often divide the query into several parts to improve the query performance [19]. For example, in case we want to search for textual strings with k indels, we can efficiently enumerate candidates for the matches by dividing the query into $k+1$ substrings and finding the exact matches of these substrings, as at least one of the divided substrings must exactly match somewhere in the text. Similarly, we also divide the query 3-D structure into several substructures in our algorithm.

In our algorithm, we first divide the query $\mathbf{Q}$ of size m into $3k+2$ equal-length substructures of size $m' = \lfloor m/(3k+2) \rfloor$. Note that k is the number of maximum indels defined in section 3, which is considered to be a small constant. We call each substructure a 'divided substructure'. Let $\mathbf{Q}_j$ denote the j-th divided substructure, *i.e.*, $\mathbf{Q}[(j-1)m'+1..j \cdot m']$. We ignore the remaining part (*i.e.*, $\mathbf{Q}[(3k+2)m'+1..m]$) in case m is not a multiple of $3k+2$.

Consider the case that $\mathbf{Q}$ matches with $\mathbf{P}_i = \mathbf{P}[i..i-k'+k''+m-1]$ with threshold c and (at most) $k = k'+k''$ indels. Let $\mathbf{Q}'$ and $\mathbf{P}'_i$ denote the k'-reduced subsequence structure of $\mathbf{Q}$ and the k''-reduced subsequence structure of $\mathbf{P}[i..i-k'+k''+m-1]$ respectively, such that $RMSD(\mathbf{Q}', \mathbf{P}'_i) \leq c$. Let $\mathbf{Q}'_j$ denote the largest substructure of $\mathbf{Q}'$ such that $\mathbf{Q}'_j$ is a subsequence structure of $\mathbf{Q}_j$. Let h_j denote the first index of $\mathbf{Q}'_j$ in $\mathbf{Q}'$, *i.e.*, $\mathbf{Q}'_j = \mathbf{Q}'[h_j..h_{j+1}-1]$ ($h_{3k+2} = m-k'+1$). Let $\mathbf{P}'_{i,j} = \mathbf{P}'[h_j..h_{j+1}-1]$. It is easy to see that there are at least $2k+2$ pairs of subsequence structures $\mathbf{Q}'_j$ and $\mathbf{P}'_{i,j}$ such that $\mathbf{Q}'_j = \mathbf{Q}_j$ and $\mathbf{P}'_{i,j}$ is a substructure of $\mathbf{P}_i$. We call these (at least $2k+2$ pairs of) substructures 'ungapped substructures'. Notice that the length of the ungapped substructures is m'. Let the index of an ungapped structure $\mathbf{P}'_{i,j}$ in $\mathbf{P}_i$ be p_j, *i.e.*, $\mathbf{P}'_{i,j} = \mathbf{P}_i[p_j..p_j+m'-1]$. It is easy to see that $|(j-1) \cdot m' + 1 - p_j| \leq k$, as we allow only at most k indels. Then, an inequality $RMSD(\mathbf{Q}'_j, \mathbf{P}'_{i,j}) \leq c \cdot \sqrt{m/m'}$ holds for ungapped substructures $\mathbf{Q}'_j$ and $\mathbf{P}'_{i,j}$, according to the following lemma:

Lemma 2. *Consider a pair of two structures $\mathbf{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n)$ and $\mathbf{T} = (\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_n)$, both of whose length is n. Let $\mathbf{S}' = (\boldsymbol{s}_{a_1}, \boldsymbol{s}_{a_2}, \ldots, \boldsymbol{s}_{a_{n'}})$ be some subsequence structure of $\mathbf{S}$, and let $\mathbf{T}' = (\boldsymbol{t}_{a_1}, \boldsymbol{t}_{a_2}, \ldots, \boldsymbol{t}_{a_{n'}})$. Then, $RMSD(\mathbf{S}', \mathbf{T}') \leq \sqrt{n/n'} \cdot RMSD(\mathbf{S}, \mathbf{T})$.*

Proof. According to the definition of the RMSD, the following inequality holds:

$$\begin{aligned} RMSD(\mathbf{S}', \mathbf{T}') &= \min_{R,\boldsymbol{v}} \sqrt{\frac{1}{n'} \sum_{i=1}^{n'} |\boldsymbol{s}_{a_i} - (R \cdot \boldsymbol{t}_{a_i} + \boldsymbol{v})|^2} \\ &\leq \min_{R,\boldsymbol{v}} \sqrt{\frac{1}{n'} \sum_{i=1}^{n} |\boldsymbol{s}_i - (R \cdot \boldsymbol{t}_i + \boldsymbol{v})|^2} \\ &= \sqrt{n/n'} \cdot RMSD(\mathbf{S}, \mathbf{T}). \end{aligned} \tag{3}$$

□

In summary, at least $2k+2$ divided substructures $\mathbf{Q}_j = \mathbf{Q}[(j-1)m'+1..j \cdot m']$ (among the $3k+2$ divided substructures) must satisfy the following constraint:

- There must be some substructure $\mathbf{P}[\ell..\ell + m' - 1]$ of $\mathbf{P}$ such that $RMSD(\mathbf{Q}'_j, \mathbf{P}[\ell..\ell + m' - 1]) \leq c \cdot \sqrt{m/m'}$ and $i + (j-1)m' - k \leq \ell \leq i + (j-1)m' + k$.

These $2k+2$ (or more) divided substructures must also satisfy the following weaker constraint, as an inequality $D(\mathbf{S}, \mathbf{T}) \leq RMSD(\mathbf{S}, \mathbf{T})$ holds for any pair of same-length structures $\mathbf{S}$ and $\mathbf{T}$ (see section 2.4 for the definition of $D(\mathbf{S}, \mathbf{T})$).

- There must be some substructure $\mathbf{P}[\ell..\ell + m' - 1]$ of $\mathbf{P}$ such that $D(\mathbf{Q}'_j, \mathbf{P}[\ell..\ell+m'-1]) \leq c \cdot \sqrt{m/m'}$ and $i+(j-1)m'-k \leq \ell \leq i+(j-1)m'+k$.

We call the divided substructures that satisfy the latter weaker constraint 'hit substructures' for the position i.

Based on the above discussions, we propose the following simple algorithm for the k-indel 3-D substructure problem.

Algorithm

1. We first enumerate all the positions i in $\mathbf{P}$ such that there are at least $2k+2$ hit substructures for the position i, by computing all the $D(\mathbf{Q}_j, \mathbf{P}[i..i+m'-1])$ values for all the pairs of i $(1 \leq i \leq N - m' + 1)$ and j $(1 \leq j \leq 3k+2)$.
2. For each position i found in step 1, we check the RMSDs between all the pairs of k'-reduced subsequence structure of $\mathbf{Q}$ and k''-reduced subsequence substructure of $\mathbf{P}[i..i + m - k' + k'' + m - 1]$ such that $k' + k'' \leq k$ and $k'' - k' \leq N - m - i + 1$. If any one of the checked RMSDs is smaller or equal to c, output i as the position of a substructure similar to the query $\mathbf{Q}$.

In the next section, we analyze the average-case time complexity of the algorithm.

5.2 The Average-Case Time Complexity of the Algorithm

For each $\mathbf{Q}_j$ (whether it is a hit substructure or not), we can compute $D(\mathbf{Q}_j, \mathbf{P}[i..i + m' - 1])$ for all i $(1 \leq i \leq N - m' + 1)$ in total $O(N)$ time,

as $G(\mathbf{P}[i..i+m'-1])$ (*i.e.*, the centroid of $\mathbf{P}[i..i+m'-1]$) can be computed in $O(N)$ time for all i. Thus, we can execute the step 1 of the algorithm in section 5.1 in $O(k^2 \cdot N)$ time, which is in $O(N)$ as we consider k is a small fixed constant. Let N' denote the number of candidates enumerated in step 1 of the algorithm in section 5.1. As the number of pairs to check in step 2 for each position is less than ${}_{2m+k}C_k$ (which is in $O(m^k)$), and each RMSD can be computed in $O(m)$ time, the computational complexity of step 2 is $O(N'm^{k+1})$. In total, the computational complexity of the algorithm is $O(N + N'm^{k+1})$. In the worst case, the algorithm could be as bad as the naive $O(Nm^{k+1})$-time algorithm, as N' could be N at worst.

But, in the following, we show that $\langle N' \rangle$ is only in $O(N/m^{k+1})$ and consequently the average-case (expected) time complexity of the algorithm is astonishingly $O(N)$, under the assumption that $\mathbf{P}$ follows the random-walk model. According to Lemma 1 in section 2.4, the probability that a divided substructure $\mathbf{Q}_i$ is a hit substructure for the position i is in $O(k \cdot c \cdot \sqrt{m/m'}/\sqrt{m'}) = O(c \cdot k^2/\sqrt{m})$, under the random-walk assumption. Consider that the above probability can be bounded by $a \cdot c \cdot k^2/\sqrt{m}$ if m is large enough, where a is an appropriate constant. Then, the probability that $2k+2$ of the $3k+2$ divided substructures are hit substructures is $O((a \cdot c \cdot k^2/\sqrt{m})^{2k+2} \cdot {}_{3k+2}C_{2k+2})$, which is in $O(c^{2k+2} \cdot k^{5k+4}/m^{k+1})$. Thus $\langle N' \rangle$ is in $O(N \cdot c^{2k+2} \cdot k^{5k+4}/m^{k+1})$, and the following lemma holds, considering that both c and k are small fixed constants.

Lemma 3. *$\langle N' \rangle$ is in $O(N/m^{k+1})$.*

Consequently the expected time complexity of the step 2 of the above algorithm is only in $O(N)$. (More precisely, it is $O(c^{2k+2} \cdot k^{5k+4} \cdot N)$, but we consider that both c and k are small fixed constants.) In conclusion, the total expected time complexity of the algorithm in section 5.1 is only $O(N)$, under the assumption that $\mathbf{P}$ follows the random walk model:[4]

Theorem 2. *The total expected time complexity of our algorithm is $O(N)$, under the assumption that $\mathbf{P}$ follows the random walk model.*

6 Concluding Remarks

We considered the k-indel 3-D substructure search problem, in which we search for similar 3-D substructures from molecular 3-D structure databases, with consideration of indels. We showed that the same problem in arbitrary dimensional space is NP-hard. Moreover, we proposed a linear expected time algorithm, under the assumption that the number of indels is bounded by a constant and the database structures follow the random-walk model. There are several open problems. First of all, the difficulty of our problem restricted to 3-D space is not known. As for our algorithm, its expected time complexity is $O(N)$ for a database of size N, but its coefficient, *i.e.*, $c^{2k+2} \cdot k^{5k+4}$, is very large (c is the

[4] The same discussion can be done if the query $\mathbf{Q}$ follows the random walk model, instead of $\mathbf{P}$.

threshold of the RMSD and k is the maximum number of indels, both of which we consider as constant numbers). It would be more practical if we can design algorithms with better coefficients. Another open problem is whether we can design a worst-case (deterministically) linear-time algorithm for our problem, though no worst-case linear-time algorithm is known even for the no-indel case.

Acknowledgement. This work was partially supported by the Grant-in-Aid from the Ministry of Education, Culture, Sports, Science and Technology of Japan. Jesper Jansson was supported by the Special Coordination Funds for Promoting Science and Technology.

References

1. Arun, K.S., Huang, T.S., Blostein, S.D.: Least-squares fitting of two 3-D point sets. IEEE Trans Pattern Anal. Machine Intell. 9, 698–700 (1987)
2. Aung, Z., Tan, K.-L.: Rapid retrieval of protein structures from databases. Drug Discovery Today 12, 732–739 (2007)
3. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The protein data bank. Nucl. Acids Res. 28, 235–242 (2000)
4. Boyd, R.H., Phillips, P.J.: The Science of Polymer Molecules: An Introduction Concerning the Synthesis, Structure and Properties of the Individual Molecules That Constitute Polymeric Materials. Cambridge University Press, Cambridge (1996)
5. Cooley, J.W., Tukey, J.W.: An algorithm for the machine calculation of complex Fourier series. Math. Comput. 19, 297–301 (1965)
6. Corneil, D.G., Perl, Y.: Clustering and domination in perfect graphs. Discrete Appllied Mathematics 9(1), 27–39 (1984)
7. Dayantis, J., Palierne, J.-F.: Monte Carlo precise determination of the end-to-end distribution function of self-avoiding walks on the simple-cubic lattice. J. Chem. Phys. 95, 6088–6099 (1991)
8. de Gennes, P.-G.: Scaling Concepts in Polymer Physics. Cornell University Press (1979)
9. Eggert, D.W., Lorusso, A., Fisher, R.B.: Estimating 3-D rigid body transformations: a comparison of four major algorithms. Machine Vision and Applications 9, 272–290 (1997)
10. Eidhammer, I., Jonassen, I., Taylor, W.R.: Structure comparison and structure patterns. J. Computational Biology 7(5), 685–716 (2000)
11. Flory, P.J.: Statistical Mechanics of Chain Molecules. Interscience, New York (1969)
12. Gerstein, M.: Integrative database analysis in structural genomics. Nat. Struct. Biol., Suppl., 960–963 (2000)
13. Goldman, D., Istrail, S., Papadimitriou, C.H.: Algorithmic aspects of protein structure similarity. In: Proc. 40th Annual Symposium on Foundations of Computer Science, pp. 512–522 (1999)
14. Kabsch, W.: A solution for the best rotation to relate two sets of vectors. Acta Cryst. A32, 922–923 (1976)
15. Kabsch, W.: A discussion of the solution for the best rotation to relate two sets of vectors. Acta Cryst. A34, 827–828 (1978)
16. Kallenberg, O.: Foundations of Modern Probability. Springer, Heidelberg (1997)

17. Koehl, P.: Protein structure similarities. Current Opinion in Structural Biology 11, 348–353 (2001)
18. Kramers, H.A.: The behavior of macromolecules in inhomogeneous flow. J. Chem. Phys. 14(7), 415–424 (1946)
19. Navarro, G.: A guided tour to approximate string matching. ACM Computing Surveys 33(1), 31–88 (2001)
20. Schwartz, J.T., Sharir, M.: Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves. Intl. J. of Robotics Res. 6, 29–44 (1987)
21. Shibuya, T.: Efficient substructure RMSD query algorithms. J. Comput. Biol. 14(9), 1201–1207 (2007)
22. Shibuya, T.: Searching protein 3-D structures in linear time. In: Batzoglou, S. (ed.) RECOMB 2009. LNCS (LNBI), vol. 5541, pp. 1–15. Springer, Heidelberg (2009)