アルゴリズム 41－1
（1994．9．21）

タンパク質立体構造に対する部分構造検索および アラインメント・アルゴリズム

阿久津 達也
群馬大学 工学部 情報工学科

本稿ではタンパク質立体構造に対する実用的な部分構造検索アルゴリズムおよびアラインメント・アルゴリズムを示す。どちらの場合でも立体構造は3次元空間上の点列として扱われる。部分構造検索アルゴリズムでは固定したサイズの部分構造に対して「構造が似ていればベクトル間の距離も近い」という性質を持つハッシュ・ベクトルを計算して高速な検索を行う。アラインメント・アルゴリズムでは二つの構造間の距離が小さくなるような点間の対応を計算する。どちらのアルゴリズムに対しても、出力される結果について理論的な保証を与え、かつ、実データに対しての有効性を計算機実験により示す。

Substructure Search and Alignment Algorithms for Three-Dimensional Protein Structures

Tatsuya AKUTSU

Department of Computer Science, Gunma University,
1-5-1 Tenjin, Kiryu, Gunma 376, Japan.
e-mail: akutsu@cs.gunma-u.ac.jp

This paper presents two practical algorithms for pattern matching of 3D protein structures: a hashing technique for quick substructure search and an alignment algorithm for 3D structures. In both algorithms, protein structures are treated as point sequences. In the hashing technique, for each fixed-length sequence, a hash vector is computed, where the distance between two hash vectors is small if two sequences are similar. In the alignment algorithm, a correspondence of points between two sequences is computed. In each algorithm, a theoretical proof for the quality of outputs is given. Moreover, experimental results show that both algorithms are effective.

1 Introduction

The comparison of three-dimensional (3D, in short) protein structures plays a very important role in the study of protein structures [5]. Thus, it is important to study pattern matching algorithms for 3D protein structures. Recently, geometric pattern matching algorithms have been studied extensively in computational geometry [3, 6, 7, 8]. However, most of them do not seem to be practical since they are too complicated to be implemented and the time complexities are not small. On the other hand, a large number of practical pattern matching algorithms for 3D protein structures have been proposed in molecular biology [2, 11, 12, 14, 15, 19, 20, 21]. However, they do not seem to be sufficient from the viewpoint of the computation time and the robustness against insertions or deletions of sequences. Moreover, there are no theoretical proofs for the qualities of the outputs of them.

This paper presents two pattern matching algorithms for 3D protein structures, which are practical and have theoretical proofs for the qualities of outputs. One is developed for the *substructure search* problem, and the other is developed for the *alignment* problem. In the substructure search problem, given a small fragment (pattern) of 3D structure, all structures which have substructures similar to the fragment are enumerated. It is important for searching 3D protein structure databases since the number of proteins, for which 3D structures are known, exceeds 1000 and it grows year by year. In the alignment problem, given two (or more) 3D structures, structurally equivalent atoms are identified. It is important to get good matchings between large protein structures since insertions and deletions of sequences must be considered. The *root mean square deviation* is used as a measure for the quality of outputs, which is a common measure for comparing two 3D structures in molecular biology.

2 Problems

In this section, we define the substructure search problem and the alignment problem formally. First we consider representation of 3D protein structures. As we are only interested in representing an outline of 3D structure, we follow the common procedure of ignoring side chains and consider only the carbon and nitrogen atoms (or Cα atoms) in the main chain, which are treated as points in 3D space. Only the geometry of protein structures is considered and details such as the identity of specific atoms are ignored. Thus, each protein structure is treated as a sequence of points. If we consider Cα atoms only, a typical size of a protein sequence is at most 600. For each structure $P = (\boldsymbol{p}^1, \cdots, \boldsymbol{p}^n)$, $P_{i,j}$ denotes the fragment $(\boldsymbol{p}^i, \boldsymbol{p}^{i+1}, \cdots, \boldsymbol{p}^j)$ of P.

2.1 Root Mean Square Deviation

Here, we briefly review the root mean square deviation (*rms distance* or *rmsd*, in short), which is used as a common measure for comparing two protein structures in molecular biology. The *rmsd* fitting is a kind of least-squares fitting method for two sequences of points, and was developed by several persons independently [9, 13, 17].

Let $P = (\boldsymbol{p}^1, \cdots, \boldsymbol{p}^n)$ and $Q = (\boldsymbol{q}^1, \cdots, \boldsymbol{q}^n)$ be two sequences of points. We assume that P is translated so that its centroid $(\frac{1}{n}\sum_{k=1}^{n} \boldsymbol{p}^k)$ is at the origin. For each point (resp. vector) $\boldsymbol{s}$, $\boldsymbol{s}_i$ $(i = 1, 2, 3)$ denotes the i-th coordinate value of $\boldsymbol{s}$. Let

$$d(P, Q, R, \boldsymbol{a}) = \sqrt{\frac{1}{n}\sum_{k=1}^{n} |R\boldsymbol{p}^k + \boldsymbol{a} - \boldsymbol{q}^k|^2} \quad ,$$

where R is a rotation matrix and $\boldsymbol{a}$ is a translation vector. Then, the *rmsd* value $d(P, Q)$ between P and Q is defined by $d(P, Q) = \min_{R, \boldsymbol{a}} d(P, Q, R, \boldsymbol{a})$.

$d(P, Q)$ is obtained by com-

puting $\boldsymbol{a} = \frac{1}{n}\sum_{k=1}^{n} \boldsymbol{q}^k$ and $R = (A^tA)^{1/2}A^{-1}$ which minimize $d(P, Q, R, \boldsymbol{a})$, where the matrix $A = (A_{ij})$ $(i, j = 1, 2, 3)$ is given by $A_{ij} = \sum_{k=1}^{n} \boldsymbol{p}_i^k \boldsymbol{q}_j^k$ [17]. Note that $d(P, Q)$, R and $\boldsymbol{a}$ can be computed in $O(n)$ time. Also note that reflections can be included without increasing the order of the time complexity although they are not considered in this paper.

2.2 Substructure Search Problem

Using *rmsd*, we define the substructure search problem as follows.

Input: A fragment $P = (\boldsymbol{p}^1, \cdots, \boldsymbol{p}^m)$, a real number $\delta > 0$ and a set of proteins $QS = \{Q^1, \cdots, Q^N\}$,

Output: All structures Q^j each of which contains at least a fragment $Q^j_{i,i+m-1}$ such that $d(P, Q^j_{i,i+m-1}) \leq \delta$.

The substructure search problem can be solved by a naive algorithm which computes *rmsd* for all $Q^j_{i,i+m-1}$'s. However, it takes $O(Nmn)$ time, where we assume that the length of each Q^j is $O(n)$. In fact, experimental results described in Section 5 show that it takes about a minute. It is too long time for interactive uses of protein structure database systems. If we use an FFT-based algorithm by Schwartz and Sharir [17], the time complexity is reduced to $O(Nn \log m)$. But, the constant factor is too large. Indeed, experimental results show that it is faster than the naive algorithm only if $m > 200 \sim 300$ [18]. If $m > 60 \sim 100$, insertions and deletions of sequences can not be ignored and the substructure search problem should be defined in another way. So, the FFT-based algorithm is not practical.

2.3 Alignment Problem

The alignment problem for 3D protein structures is defined in a similar way as in the case of strings.

For two point sequences P and Q, M is called an alignment between P and Q if M is a subset of $P \times Q$ and $(\forall(\boldsymbol{p}^{i_1}, \boldsymbol{q}^{j_1}) \in M)(\forall(\boldsymbol{p}^{i_2}, \boldsymbol{q}^{j_2}) \in M)(i_1 = i_2 \leftrightarrow j_1 = j_2)$ and $(\forall(\boldsymbol{p}^{i_1}, \boldsymbol{q}^{j_1}) \in M)(\forall(\boldsymbol{p}^{i_2}, \boldsymbol{q}^{j_2}) \in M)(i_1 < i_2 \leftrightarrow j_1 < j_2)$ hold. Let M_P (resp. M_Q) be the subsequence of P (resp. Q) such that each element appears in M. Then, the alignment problem is defined as follows (see Fig. 1).

Input: Two sequences $P = (\boldsymbol{p}^1, \cdots, \boldsymbol{p}^m)$ and $Q = (\boldsymbol{q}^1, \cdots, \boldsymbol{q}^n)$, a positive integer K and a positive real δ,

Output: An alignment M between P and Q such that $|M| \geq L$ and $d(M_P, M_Q) \leq \delta$.

In the alignment problem, we assume without loss of generality that $m \leq n$ holds. Alignment of protein structures is important for finding common structural patterns of proteins, since insertions and deletions of sequences should be ignored to apply the *rmsd* fitting effectively. In the case of Fig. 1, $\{ (\boldsymbol{p}^1, \boldsymbol{q}^1), \cdots, (\boldsymbol{p}^4, \boldsymbol{q}^4), (\boldsymbol{p}^7, \boldsymbol{q}^5), \cdots, (\boldsymbol{p}^{13}, \boldsymbol{q}^{11}), (\boldsymbol{p}^{14}, \boldsymbol{q}^{13}), (\boldsymbol{p}^{15}, \boldsymbol{q}^{16}), (\boldsymbol{p}^{16}, \boldsymbol{q}^{17}) \}$ is an example of a good alignment.

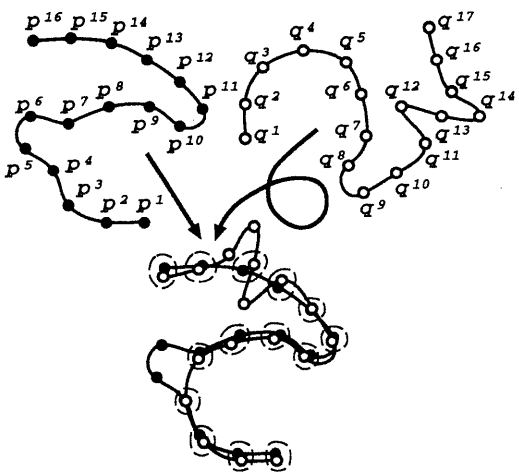

Figure 1: Alignment problem.

3 Hashing Technique for Substructure Search

For quick substructure search, we use a kind of hashing technique. Several hashing techniques for geometric objects have been proposed, and they are called as *geometric hashing*. However, none of them has a theoretical proof for hash function.

In our hashing technique, a vector of reals is associated for each fragment of fixed length. For each fragment $P = (\boldsymbol{p}^1, \cdots, \boldsymbol{p}^H)$ of length H, a hash vector $\boldsymbol{hs}(P)$ is associated. Then, the following conditions should be satisfied by $\boldsymbol{hs}(P)$:

(A) $\boldsymbol{hs}(P)$ is invariant with any isometric transformation for P,

(B) $\boldsymbol{hs}(P)$ is close to $\boldsymbol{hs}(Q)$ if $d(P,Q)$ is small.

Although condition (A) may be implied by condition (B), we describe them separately to make the presentation clear. Note that once such a vector is given, $d(P,Q)$ is required to be computed only when $\boldsymbol{hs}(P)$ is close to $\boldsymbol{hs}(Q)$. Moreover, elaborate data structures for orthogonal range query [22] might be used effectively.

Here, we describe our hash vectors. All vectors are very simple and computed in a similar way. First, we describe a basic one, denoted by HASH(A).

HASH(A):
$\boldsymbol{hs}(P) = (c_1(P), s_1(P), \cdots, c_D(P), s_D(P))$, where

$$c_i(P) = \alpha \sum_{k=1}^{H} \|\boldsymbol{p}^k - \boldsymbol{c}\| \, (\sin(\frac{2\pi i(k-1)}{H}) + \beta),$$

$$s_i(P) = \alpha \sum_{k=1}^{H} \|\boldsymbol{p}^k - \boldsymbol{c}\| \, (\cos(\frac{2\pi i(k-1)}{H}) + \beta).$$

Note that $\boldsymbol{c}$ denotes the centroid of P (i.e., $\boldsymbol{c} = \sum_{k=1}^{H} p^k$), and $\|\boldsymbol{p}\|$ denotes the length of a vector $\boldsymbol{p}$. Also note that α ($\alpha > 0$) and β ($\beta \geq 0$) are fixed reals and D is a fixed integer, to be determined later. $\boldsymbol{hs}(P)$ is similar to (a low frequency part of) the Fourier spectrum of the distances between the points and the centroid.

For $\boldsymbol{hs}(P)$, condition (A) is trivially satisfied since $\boldsymbol{hs}(P)$ is computed only from the distances between the points and the centroid. To show that condition (B) is satisfied, we first prove the following theorem.

Theorem 1: Assume that $P = (\boldsymbol{p}^1, \cdots, \boldsymbol{p}^H)$ and $Q = (\boldsymbol{q}^1, \cdots, \boldsymbol{q}^H)$ are translated so that the centroids are at the origin. Then, $|\sum_{i=1}^{H} \|\boldsymbol{p}^i\| - \sum_{i=1}^{H} \|\boldsymbol{q}^i\|| \leq H\, d(P,Q)$ holds.

Proof: Let $\hat{Q} = (\hat{\boldsymbol{q}}^1, \cdots, \hat{\boldsymbol{q}}^H)$ denotes the rotated sequence of Q such that $d(P,\hat{Q},I,\mathbf{o}) = d(P,Q)$, where $\mathbf{o}$ denotes the origin and I denotes the identity matrix.

Then, the following inequality holds:

$$|\sum_{i=1}^{H} \|\boldsymbol{p}^i\| - \sum_{i=1}^{H} \|\boldsymbol{q}^i\|| = |\sum_{i=1}^{H} \|\boldsymbol{p}^i\| - \sum_{i=1}^{H} \|\hat{\boldsymbol{q}}^i\||$$
$$\leq \sum_{i=1}^{H} |\,\|\boldsymbol{p}^i\| - \|\hat{\boldsymbol{q}}^i\|\,| \leq \sum_{i=1}^{H} \|\boldsymbol{p}^i - \hat{\boldsymbol{q}}^i\|,$$

where the last inequality comes from the triangular inequality. Since $t_1 + \cdots + t_H \leq \sqrt{H}\sqrt{t_1^2 + \cdots + t_H^2}$ holds for all $t_1 \geq 0, \cdots, t_H \geq 0$,

$$\sum_{i=1}^{H} \|\boldsymbol{p}^i - \hat{\boldsymbol{q}}^i\| \leq \sqrt{H}\sqrt{\sum_{i=1}^{H} \|\boldsymbol{p}^i - \hat{\boldsymbol{q}}^i\|^2} = H\, d(P,Q)$$

holds and the theorem follows. □

From Theorem 1, the following corollary is immediately proved, which shows that HASH(A) satisfies condition (B).

Corollary 1: For all i, $|c_i(P) - c_i(Q)| \leq H\alpha(1+\beta)d(P,Q)$ and $|s_i(P) - s_i(Q)| \leq H\alpha(1+\beta)d(P,Q)$ hold. □

From Corollary 1, if $|c_i(P) - c_i(Q)| > H\alpha(1+\beta)\delta$ or $|s_i(P) - s_i(Q)| > H\alpha(1+\beta)\delta$ holds for some i, then $d(P,Q) > \delta$ holds. Note that $\boldsymbol{hs}(P)$ can be computed in $O(H)$ time, and whether or not $(|s_i(P) - s_i(Q)| \leq \gamma \wedge |c_i(P) - c_i(Q)| \leq \gamma)$ holds for all i can be tested

in constant time since we assume that D is a fixed integer. Let $HS(P,Q,\gamma)$ denote this condition ($(|s_i(P)-s_i(Q)| \leq \gamma \wedge |c_i(P)-c_i(Q)| \leq \gamma)$ holds for all i).

Next, we describe several variants of HASH(A). HASH(B) and HASH(B') are obtained by replacing $\boldsymbol{c}$ of HASH(A) with $\boldsymbol{d} = \sum_{k=1}^{L} \boldsymbol{p}^k$ and $\boldsymbol{e} = \sum_{k=N-L+1}^{N} \boldsymbol{p}^k$, respectively. HASH(A+B) is a combination of HASH(A) and HASH(B), and HASH(A+B+B') is a combination of HASH(A), HASH(B) and HASH(B'). It is not difficult to see that similar properties as Corollary 1 hold for all vectors.

4 Approximate Alignment

The alignment problem might be solved exactly using similar approach as in Refs. [3, 8]. However, it would be too complicated and the time complexity would become very large.

Thus, we have developed an approximation algorithm using an idea introduced by Heffernan and Schirra [6, 7, 16]. Moreover, ideas in Refs. [11, 14] are combined. In this paper, we overview the algorithm, and details and proofs are omitted.

4.1 Algorithm

The algorithm in a basic form is very simple as shown below, where α, β and γ denote constants.

Procedure *ApproAlign*(P,Q,K,δ)
$M \leftarrow \emptyset$;
for all triplets $PP = (p^{i_1}, p^{i_2}, p^{i_3})$ of P **do**
  for all triplets $QQ = (q^{j_1}, q^{j_2}, q^{j_3})$ of Q **do**
    if $D(PP,QQ) \leq \gamma\delta$ **then** #
      Compute a matching M between P and Q; #
      if $|M'| \geq (1-1/\alpha)K$ and M' is *better* than M #
      then $M \leftarrow M'$; #
Output M;

Note that $D(PP,QQ)$ denotes the minimax distance defined in the next subsection. To compute a matching M, we construct a weighted bipartite graph $G(P,Q;E)$, such that $(\boldsymbol{p}^i, \boldsymbol{q}^j) \in E$ if $\|\boldsymbol{p}^i - T(\boldsymbol{q}^j)\| \leq \alpha\beta\delta$, and $w(\boldsymbol{p}^i, \boldsymbol{q}^j) = \|\boldsymbol{p}^i - T(\boldsymbol{q}^j)\|$. Then, we compute a minimum weight maximum matching M (such that the orders of sequences are preserved) using a dynamic programming procedure. We say that M' is better than M if $M = \emptyset$ or $|M'| \geq |M| \wedge d(M'_P, M'_Q) \leq d(M_P, M_Q)$ holds.

4.2 Analysis

First we analyze the time complexity of the procedure *ApproAlign*(P,Q,K,δ). Since there are $O(n^3)$ triplets PP (resp. QQ) for P (resp. Q), part (#) is executed $O(n^6)$ times. The most time consuming part in (#) is the computation of a matching M. It can be done in $O(n^2)$ time. Thus, the total time complexity is $O(n^8)$. Of course, it is too long time. However, the average case computation time can be reduced considerably using several techniques such as random sampling, sparse dynamic programming and binary search.

Next we consider the approximation ratio. Here we consider the minimax distance $D(P,Q)$ defined by:

$$D(P,Q) = \min_{T} \max_{1 \leq i \leq n} \|\boldsymbol{p}^i - T(\boldsymbol{q}^i)\| \ ,$$

where T is an isometric transformation, and $|P| = |Q| = n$. Then it is trivial that $d(P,Q) \leq D(P,Q)$. Moreover, the following lemma holds.

Lemma 1: If there exists an alignment M between P and Q such that $d(M_P, M_Q) \leq \delta$ and $|M| \geq K$, then there exists an alignment M' between P and Q such that $D(M'_P, M'_Q) \leq \alpha\delta$ and $|M'| \geq (1-1/\alpha)K$ for any $\alpha > 1$. □

From $d(P,Q) \leq D(P,Q)$ and this lemma, it is seen that a good approximation under the *rms* distance can be computed if a good approximation under the minimax distance can be computed. Thus we consider the approximation ratio under the minimax distance.

The following lemma states that there is a pair of triplets from which we can find an isometric transformation approximating the minimax distance.

Lemma 2: Let M be an alignment between

P and Q such that $D(M_P, M_Q) \leq \delta$. Let $\{(\boldsymbol{p}^{i_1}, \boldsymbol{q}^{j_1}), (\boldsymbol{p}^{i_2}, \boldsymbol{q}^{j_2})\}$ the subset of M such that $\|\boldsymbol{p}^{i_1} - \boldsymbol{p}^{i_2}\|$ is maximum. Let $(\boldsymbol{p}^{i_3}, \boldsymbol{q}^{j_3})$ be the element of M such that the distance between the point $\boldsymbol{p}^{i_3}$ and the line $\overline{\boldsymbol{p}^{i_1}\boldsymbol{p}^{i_2}}$ is maximum. Then, $\|\boldsymbol{p}^i - T(\boldsymbol{q}^j)\| \leq \beta\delta$ holds for any pair $(\boldsymbol{p}^i, \boldsymbol{q}^j) \in M$ and for any isometric transformation T such that $(1 \leq \forall k \leq 3)(\|\boldsymbol{p}^{i_k} - T(\boldsymbol{q}^{j_k})\| \leq \gamma\delta)$, where γ is any constant such that $\gamma > 1$, and β is a constant dependent on γ only. □

From Lemma 2, we obtain the following theorem.

Theorem 2: If there is an alignment M such that $|M| \geq (1 - 1/\alpha)K$ and $D(M_P, M_Q) \leq \alpha\delta$, then $ApproAlign(P, Q, K, \delta)$ computes an alignment M' such that $|M'| \geq (1 - 1/\alpha)K$ and $D(M'_P, M'_Q) \leq \alpha\beta\delta$, where $\alpha > 1$ is any constant and β is some fixed constant. □

Using the technique described in Ref. [16], the constant β can be made arbitrary small (where $\beta > 1$) with increasing the computation time by only a constant factor. From Lemma 1 and Theorem 2, we obtain the following.

Corollary 2: If there is an alignment M such that $|M| \geq K$ and $d(M_P, M_Q) \leq \delta$, then $ApproAlign(P, Q, K, \delta)$ computes an alignment M' such that $|M'| \geq (1 - 1/\alpha)K$ and $d(M'_P, M'_Q) \leq \alpha\beta\delta$, where $\alpha > 1$ is any constant and β is some fixed constant. □

Note that the constant β can be made arbitrary small too.

5 Experimental Results

Experiments have been made using PDB (Protein Data Bank) data [4]. Although PDB data contain various information, only positions of Cα atoms are used. All algorithms are implemented in C-language on a SUN SPARC STATION-10 (a UNIX workstation).

5.1 Results for Substructure Search

The new hashing methods are compared with the previous and the naive ones in Table 1. NV denotes the naive algorithm described in Subsection 2.2. LS denotes the least-squares hashing method proposed in Ref. [2]. A, B, A+B and A+B+B' denote HASH(A), HASH(B), HASH(A+B) and HASH(A+B+B'), respectively, where $D = 4$ and $L = \frac{H}{4}$ is used in each case. In general, it is expected that search time is reduced if larger D is used. However, search time was hardly reduced even if $D = 8$ was used. Thus, $D = 4$ is used.

Table 1: Comparison of hashing methods.

DATA	4HHB (50-94) (A)	7LZM (35-80)	1R69 (5-55)	3ADK (115-170)	8LDH (150-194)
NV	63.0 12.6 %	64.0 25.5 %	67.5 25.5 %	70.9 4.4 %	63.2 8.0 %
LS	12.1 6.7 %	23.7 12.7 %	24.8 9.1 %	5.1 8.7 %	8.1 1.2 %
A	4.9 6.6 %	8.8 12.5 %	6.7 9.0 %	6.8 8.7 %	1.2 1.1 %
B	4.0 5.0 %	5.1 6.5 %	8.6 11.2 %	4.9 5.7 %	1.1 0.7 %
A+B	2.5 3.0 %	2.3 2.5 %	3.8 4.5 %	3.4 3.8 %	0.6 0.3 %
A+B +B'	1.5 1.3 %	1.3 0.9 %	1.5 1.1 %	1.8 1.6 %	0.5 0.1 %

Each item in DATA denotes a filename (denoting a protein structure) of PDB data, where chain A is used in the case of 4HHB. Each pair of numbers in parentheses denotes the range of positions of a fragment P. For each method and each pattern fragment, search time (CPU time (sec)) amongst all structures in a database is shown, where 811 structures are used and all structural data are stored on main memory of the workstation. A percentage of indices for which $rmsd$'s are computed is described too. In each hashing method, preprocessing (i.e., computation of hash vectors) for all structures was completed in a few minutes, so that it can be neglected.

The following parameters were used: $H = 40$, $\alpha = 20.0$, $\beta = 0.5$, $\delta = 4.0(\mathring{A})$ and $\gamma = 1200.0$. Although $\gamma \ll H\alpha(1 + \beta)\delta$ was used, each method except LS could find all structures each of which contained a fragment $Q^j_{i,i+m-1}$ such that $d(P, Q^j_{i,i+m-1}) \leq \delta$. LS failed to find 3 structures in the case of 3ADK. Moreover, LS took longer time than the

other hashing methods in most cases. Thus, it is proved that the new hashing methods are much more useful than the least-squares hashing method.

From Table 1, it is seen that the following relation holds approximately:

$$\text{A+B+B'} \succ \text{A+B} \succ \text{A} \approx \text{B} \succ \text{LS} \succ \text{NV},$$

where $x \succ y$ denotes that x is faster than y, and $x \approx y$ denotes that x is as fast as y. Thus, it can be said that we had better combine different types of hash vectors. In each new hashing method, it can be seen that the search time was reduced considerably compared with the naive algorithm. Especially, it is seen that HASH(A+B+B') is $30 \sim 100$ times faster than the naive algorithm.

5.2 Results for Alignment

The new alignment algorithm is compared with the previous and the naive ones in Table 2. NV denotes the naive algorithm described in Subsection 2.2. DP denotes the dynamic matching algorithm proposed in Ref. [2], where it is based on the dynamic programming technique.

Each item in DATA denotes a filename of PDB data, where chain A is used in the cases of 4HHB and 4MDH. For each algorithm and each pair of structures, *rmsd* (Å) and the length of the obtained alignment and CPU time (sec) are described in this order.

From Table 2, it is seen that *rmsd*'s obtained by DP and NEW are much smaller than those obtained by NV, and the lengths of the alignments obtained by DP and NEW are not so much shorter than those obtained by NV (i.e., the lengths of shorter sequences). Since the typical distance between the adjacent Cα atoms is 4Å, it can be said that good alignments are obtained by DP and NEW.

It is seen that *rmsd*'s obtained by NEW are slightly better than those by DP in general. However, NEW takes much more computation time than DP. Thus, NEW is not necessarily better than DP for practical use.

Table 2: Comparison of alignment algorithms.

DATA	5MBN 4HHB(A)	2ILA 4I1B	3ICB 5CPV	8LDH 4MDH(A)
NV	5.35Å 140 0.01sec	11.1Å 144 0.01sec	7.79Å 74 0.02sec	15.09Å 328 0.10sec
DP	1.25Å 120 1.62sec	2.58Å 100 7.98sec	2.14Å 40 0.24sec	3.59Å 300 54.3sec
NEW	1.33Å 134 75.6sec	1.56Å 105 16.7sec	1.78Å 58 69.6sec	2.08Å 273 83.7sec

6 Conclusion

In this paper, two practical algorithms for 3D protein structures have been described: a hashing technique for quick substructure search, and an alignment algorithm for 3D structures. There is a theoretical proof for the quality of outputs in each algorithm. Moreover, experimental results show that proposed algorithms are effective. Especially, it is seen that the proposed hashing technique is much better than the previous one. This hashing technique is already included in PROTEIX [1], which is a database management system for 3D protein structures, being developed by us.

On the other hand, the proposed alignment algorithm is not so much better than the previous one. Thus, better alignment algorithms should be developed. Moreover, an alignment algorithm for multiple structures should be developed because the proposed algorithm can not be applied for more than two structures. In the hashing method, only small pattern structures can be treated. Thus, quick substructure search methods which can be applied for large structures should be developed.

Acknowledgement

This research was partially supported by the Grant-in-Aid for Scientific Research on Priority Areas, "Genome Informatics", of the Ministry of Education, Science and Culture of Japan.

References

[1] T. Akutsu, "PROTEIX: an interactive database system for three dimensional protein structures," *Proc. Genome Informatics Workshop IV*, pp.430-433, 1993.

[2] T. Akutsu, "Efficient and robust three-dimensional pattern matching algorithms using hashing and dynamic programming techniques," *Proc. 27th Hawaii International Conference on System Sciences*, Vol. 5, pp. 225–234. 1994.

[3] H. Alt, K. Melhorn, H. Wagener and E. Welzl. "Congruence, similarity, and symmetries of geometric objects," *Discrete and Computational Geometry*, Vol. 3, pp. 237-256, 1988.

[4] F. C. Bernstein, T. F. Koetzle, G. J. B. Williams, E. F. Meyer Jr., M. D. Brice, J. R. Rodgers, O. Kennard, T. Shimanouchi and M. Tasumi, "The Protein Data Bank: A computer-based archival file for macromolecular structures," *J. Molecular Biology*, Vol. 112, pp. 535-542, 1976.

[5] C. Branden and J. Tooze, *Introduction to Protein Structure*, Garland Publishing, 1991.

[6] P. J. Heffernan and S. Schirra, "Approximate decision algorithm for point sets congruence," *Proc. ACM Symp. Computational Geometry*, pp. 93–101, 1992.

[7] P. J. Heffernan, "Generalized approximate algorithms for point sets congruence," *Proc. Workshop on Algorithms and Data Structures*, pp. 373–384, 1993.

[8] K. Imai, S. Sumino and H. Imai, "Minimax geometric fitting of two corresponding sets of points," *Proc. ACM Symp. Computational Geometry*, pp. 266-275, 1989.

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

[10] E. Kishon and H. Wolfson, "3-D curve matching," *Proc. AAAI Workshop on Spatial Reasoning and Multi-Sensor Fusion*, pp. 250-261, 1987.

[11] R. Nussinov and H. J. Wolfson, "Efficient detection of three-dimensional structural motifs in biological macromolecules by computer vision techniques," *Proc. Natl. Acad. Sci. (USA)*, Vol. 88, pp. 10495-10499, 1991.

[12] C. A. Orengo and W. R. Taylor, "A rapid method of protein structure alignment," *J. Theoretical Biology*, Vol. 147, pp. 517-551, 1990.

[13] S. T. Rao and M. G. Rossmann, "Comparison of super-secondary structures in proteins," *J. Molecular Biology*, Vol. 76, pp. 241-256, 1973.

[14] M. G. Rossmann and P. Argos, "Exploring structural homology of proteins," *J. Molecular Biology*, Vol. 105, pp. 75-95, 1976.

[15] R. B. Russell and G. J. Barton, "Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels," *PROTEINS: Structure, Function, and Genetics*, Vol. 14, pp. 309-323, 1992.

[16] S. Schirra, "Approximate decision algorithms for approximate congruence," *Information Processing Letters*, Vol. 43, pp.29-34, 1992.

[17] J. T. Schwartz and M. Sharir, "Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves," *Int. J. Robotics Research*, Vol. 6, pp. 29-44, 1987.

[18] A. Tamura and M. Hirota, "A study on automatic methods for finding relationship between structural patterns and sequence patterns of proteins," *Bachelor Thesis* (in Japanese), Science University of Tokyo, 1994.

[19] W. R. Taylor and C. A. Orengo, "Protein structure alignment," *J. Molecular Biology*, Vol. 208, pp. 1-22, 1989.

[20] J. M. Thornton and S. P. Gardner, "Protein motifs and data-base searching," *Trends in Biochemical Science*, Vol. 14, pp. 300-304, 1989.

[21] G. Vriend and C. Sander, "Detection of common three-dimensional substructures in proteins," *PROTEINS: Structure, Function, and Genetics*, Vol. 11, pp. 52-58, 1991.

[22] D. E. Willard, "New data structures for orthogonal range queries," *SIAM J. Computing* Vol.24, pp. 232-253, 1995.