JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 14, Number 10, 2007
© Mary Ann Liebert, Inc.
Pp. 1343–1351
DOI: 10.1089/cmb.2007.0156

Downloaded by Shandong University from www.liebertonline.com at 12/14/07. For personal use only.

Protein Local Structure Alignment Under the Discrete Fréchet Distance

BINHAI ZHU

ABSTRACT

Protein structure alignment is a fundamental problem in computational and structural biology. While there has been lots of experimental/heuristic methods and empirical results, very few results are known regarding the algorithmic/complexity aspects of the problem, especially on protein local structure alignment. A well-known measure to characterize the similarity of two polygonal chains is the famous Fréchet distance, and with the application of protein-related research, a related discrete Fréchet distance has been used recently. In this paper, following the recent work of Jiang et al. we investigate the protein local structural alignment problem using bounded discrete Fréchet distance. Given m proteins (or protein backbones, which are 3D polygonal chains), each of length $O(n)$, our main results are summarized as follows:

- **If the number of proteins, m, is not part of the input, then the problem is NP-complete; moreover, under bounded discrete Fréchet distance it is NP-hard to approximate the maximum size common local structure within a factor of $n^{1-\epsilon}$. These results hold both when all the proteins are static and when translation/rotation are allowed.**
- **If the number of proteins, m, is a constant, then there is a polynomial time solution for the problem.**

Key words: approximation, discrete Fréchet distance, Fréchet distance, NP-hardness, protein structure alignment.

1. INTRODUCTION

As a famous distance measure in the field of abstract spaces, Fréchet distance was first defined by Maurice Fréchet (1906) a century ago. Alt and Godau (1992) first used it in measuring the similarity of polygonal chains in 1992. It is well known that the Fréchet distance between two two-dimensional (2D) polygonal chains (polylines) can be computed in polynomial time (Alt and Godau, 1992, 1995), and even under translation or rotation (though the running time is much higher) (Alt et al., 2001). In three-dimensional space (3D), Wenk (2002) showed that, given two chains with sum of length N, the minimum

Department of Computer Science, Montana State University, Bozeman, Montana.

Downloaded by Shandong University from www.liebertonline.com at 12/14/07. For personal use only.

Fréchet distance between them can be computed in $O(N^{3f+2}\log N)$ time, where f is the degree of freedom for moving the chains. So with translation alone, this minimum Fréchet distance can be computed in $O(N^{11}\log N)$ time, and when both translation and rotation are allowed, the corresponding minimum Fréchet distance can be computed in $O(N^{20}\log N)$ time. These results can be generalized to any fixed dimensions (Wenk, 2002). While computing (approximating) Fréchet distance for surfaces is in general NP-hard (Godau, 1998; Hui and Shaefer, 2004), it is polynomially solvable for restricted surfaces (Buchin et al., 2006).

Eiter and Mannila (1994) defined the *discrete Fréchet distance* between two polygonal chains A and B (in any fixed dimensions), and it turns out that this simplified distance is always realized by two vertices in A and B. They also showed that, with dynamic programming, the discrete Fréchet distance between them can be computed in $O(|A||B|)$ time.

Recently, Jiang et al. (2007) applied the discrete Fréchet distance in (globally) aligning the backbones of proteins (which is called the *protein structure-structure alignment* or more generally, the *protein global alignment* problem). In fact, in this application, the discrete Fréchet distance makes more sense, as the backbone of a protein is simply a polygonal chain in 3D, with each vertex being the alpha-carbon atom of a residue. So if the (continuous) Fréchet distance is realized by an alpha-carbon atom and some other point which does not represent an atom, it is not meaningful biologically. Jiang et al. (2007) showed that, given two 2D (or 3D) polygonal chains, the minimum discrete Fréchet distance between them, under both translation and rotation, can be computed in polynomial time. They also applied some ideas therein to design an efficient heuristic for the original protein structure-structure alignment problem in 3D, and the empirical results showed that their alignment is more accurate compared with some previously known solutions.

In essence, the result of Jiang et al. (2007) implies that the protein global alignment problem, which is to find all proteins in a given set $\mathcal{P}$ similar to a query protein or some protein in $\mathcal{P}$ (under translation and rotation), is polynomially solvable. However, very few algorithmic/complexity results are known regarding the protein local structure alignment problem until very recently. Shatsky et al. (2005) showed that, under the *bottleneck* metric (Akutsu, 1996; Efrat et al., 2001), the problem is NP-complete and the problem does admit a polynomial time approximation. Most recently, Qian et al. (2007) showed that, under the RMSD distance, the problem is NP-complete and the problem admits a PTAS. On the other hand, there have been lots of experimental/heuristic methods with practical systems since 1989, for example, SSAP (Taylor and Orengo, 1989), DALI (Holm and Sander, 1993; Holm and Park, 2000), CATH (Orengo et al., 1997), CE (Shindyalov and Bourne, 1998), SCOP (Conte et al., 2000), MAMMOTH (Oritz et al., 2002), and TALI (Miao et al., 2008). In this paper, we show that if many proteins are given then the local structure alignment problem, under the discrete Fréchet distance, is very hard; on the other hand, if only a small number of proteins are given, then there is a polynomial time solution for the problem.

The paper is organized as follows. In Section 2, we introduce some basic definitions regarding Fréchet distance and review some known results. In Section 3, we show the hardness result for the protein local structure alignment problem. In Section 4, we show how to solve the problem when m is a constant. In Section 5, we conclude the paper with several open problems.

2. PRELIMINARIES

Given two 3D polygonal chains A, B with $|A| = k$ and $|B| = l$ vertices, respectively, we aim at measuring the similarity of A and B (possibly under translation and rotation) such that their distance is minimized under certain measure. Among the various distance measures, the Hausdorff distance is known to be better suited for matching two point sets than for matching two polygonal chains; the (continuous) Fréchet distance is a superior measure for matching two polygonal chains, but it is not quite easy to compute (Alt and Godau, 1992).

Let X be the Euclidean space $\mathbb{R}^3$; let $d(a,b)$ denote the Euclidean distance between two points $a, b \in X$. The (continuous) Fréchet distance between two parametric curves $f : [0,1] \to X$ and $g : [0,1] \to X$ is

$$\delta_{\mathcal{F}}(f,g) = \inf_{\alpha,\beta} \max_{s\in[0,1]} d(f(\alpha(s)), g(\beta(s))),$$

Downloaded by Shandong University from www.liebertonline.com at 12/14/07. For personal use only.

where α and β range over all continuous non-decreasing real functions with $\alpha(0) = \beta(0) = 0$ and $\alpha(1) = \beta(1) = 1$.[1]

Imagine that a person and a dog walk along two different paths while connected by a leash; moreover, they always move forward, possibly at different paces. Intuitively, the minimum possible length of the leash is the Fréchet distance between the two paths. To compute the Fréchet distance between two polygonal curves A and B (in the Euclidean plane) of $|A|$ and $|B|$ vertices, respectively, Alt and Godau (1992) presented an $O(|A||B|\log^2(|A||B|))$ time algorithm. Later, this bound was reduced to $O(|A||B|\log(|A||B|))$ time (Alt and Godau, 1995).

We now define the discrete Fréchet distance following (Eiter and Mannila, 1994).

Definition 2.1. *Given a polygonal chain (polyline) in 3D, $P = \langle p_1, \ldots, p_k\rangle$ of k vertices, a* ***q-walk*** *along P partitions the path into q disjoint non-empty subchains $\{\mathcal{P}_i\}_{i=1..q}$ such that $\mathcal{P}_i = \langle p_{k_{i-1}+1}, \ldots, p_{k_i}\rangle$ and $0 = k_0 < k_1 < \cdots < k_q = k$.*

Given two 3D polylines $A = \langle a_1, \ldots, a_k\rangle$ and $B = \langle b_1, \ldots, b_l\rangle$, a ***paired walk*** *along A and B is a q-walk $\{\mathcal{A}_i\}_{i=1..q}$ along A and a q-walk $\{\mathcal{B}_i\}_{i=1..q}$ along B for some q, such that, for $1 \le i \le q$, either $|\mathcal{A}_i| = 1$ or $|\mathcal{B}_i| = 1$ (that is, $\mathcal{A}_i$ or $\mathcal{B}_i$ contains exactly one vertex). The* ***cost*** *of a paired walk $W = \{(\mathcal{A}_i, \mathcal{B}_i)\}$ along two paths A and B is*

$$d_F^W(A, B) = \max_i \max_{(a,b)\in\mathcal{A}_i\times\mathcal{B}_i} d(a, b).$$

The ***discrete Fréchet distance*** *between two polylines A and B is*

$$d_F(A, B) = \min_W d_F^W(A, B).$$

The paired walk that achieves the discrete Fréchet distance between two paths A and B is also called the ***Fréchet alignment*** *of A and B.*

Consider the scenario in which the person walks (jumps) along A and the dog along B. Intuitively, the definition of the paired walk is based on three cases:

1. $|\mathcal{B}_i| > |\mathcal{A}_i| = 1$: the person stays and the dog moves (jumps) forward;
2. $|\mathcal{A}_i| > |\mathcal{B}_i| = 1$: the person moves (jumps) forward and the dog stays;
3. $|\mathcal{A}_i| = |\mathcal{B}_i| = 1$: both the person and the dog move (jump) forward.

Eiter and Mannila (1994) presented a simple dynamic programming algorithm to compute $d_F(A, B)$ in $O(|A||B|) = O(kl)$ time. Recently, Jiang et al. (2007) showed that the minimum discrete Fréchet distance between two chains in 2D, A and B, under translation can be computed in $O(k^3l^3\log(k+l))$ time, and under both translation and rotation it can be computed in $O(k^4l^4\log(k+l))$ time. For 3D chains these bounds are $O(k^4l^4\log(k+l))$ and $O(k^7l^7\log(k+l))$ respectively (Jiang et al., 2007). They are significantly faster than the corresponding bounds for the continuous Fréchet distance (certainly due to a simpler distance structure), which are $O((k+l)^{11}\log(k+l))$ and $O((k+l)^{20}\log(k+l))$ respectively for 3D chains (Wenk, 2002).

We comment that, while the discrete Fréchet distance could be arbitrarily larger than the corresponding continuous Fréchet distance (e.g., in Fig. 1—I, they are $d(a_2, b_2)$ and $d(a_2, o)$ respectively), by adding sample points on the polylines, one can easily obtain a close approximation of the continuous Fréchet distance using the discrete Fréchet distance (e.g., one can use $d(a_2, b)$ in Fig. 1—II to approximate $d(a_2, o)$). This fact was pointed to before in Eiter and Mannila (1994) and Indyk (2002) and is supported by the fact that the segments in protein backbones are mostly of similar lengths. Moreover, the discrete Fréchet distance is a more natural measure for matching the geometric shapes of biological sequences such as proteins. As we mentioned in the Introduction, in such an application, continuous Fréchet does not make much sense to biologists.

[1] This definition holds in any fixed dimensions.

Downloaded by Shandong University from www.liebertonline.com at 12/14/07. For personal use only.

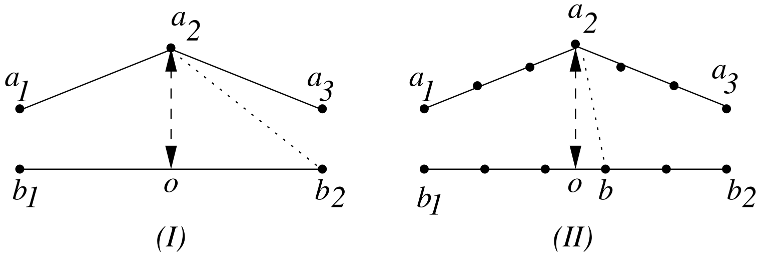

FIG. 1. Relationship between discrete and continuous Fréchet distances.

In the remaining part of this paper, for the first time, we investigate the locally aligning a set of polygonal chains (proteins or protein backbones) in 3D, under the discrete Fréchet distance. We assume that the readers are familiar with the standard terminologies on algorithms, NP-completeness and approximation algorithms which can be found in a textbook on algorithms (Cormen et al., 2001).

3. PROTEIN LOCAL STRUCTURE ALIGNMENT IS HARD

Given a set of proteins modeled as simple 3D polygonal chains, the protein local structure alignment (PLSA) problem is defined as follows:

Instance: Given a set m of proteins $P_1, P_2, \ldots, P_m$ in 3D, each with length $O(n)$, and a real number D.
Problem: Does there exist a chain C of k vertices such that the vertices of C are from P_i's, and C and a subsequence of P_i $(1 \leq i \leq m)$ has discrete Fréchet distance at most D (under translation and rotation)?

If no translation and rotation is allowed, we call the corresponding problem *static* PLSA. For the optimization version of the problem, we wish to maximize k when D is given. The (polynomial-time) approximation solution will also be referred to as approximating the optimal solution value k^* when it is hard to compute exactly. We will see that it is also hard to approximate k^* even for static PLSA. We first prove the following theorem.

Theorem 3.1. *Given $D = \delta$, the static PLSA problem does not admit any approximation of factor $n^{1-\epsilon}$ unless P=NP.*

Proof. It is easy to see that PLSA belongs to NP. We use a reduction from Independent Set to the Protein Local Structure Alignment Problem. Independent Set is a well-known NP-complete problem, which cannot be approximated within a factor of $n^{1-\epsilon}$ (Håstad, 1999). The general idea is similar to that of the longest common subsequence problem for multiple sequences (Jiang and Li, 1995), but our details are more involved due to the geometric properties of the problem.

Given a graph $G = (V, E), V = \{v_1, v_2, \ldots, v_N\}, E = \{e_1, e_2, \ldots, e_M\}$, we construct $M + 1$ 3D chains $P_0, P_1, P_2, \ldots, P_M$ as follows. (We assume that the vertices and edges in G are sorted by their corresponding indices.)

The overall reduction is as follows: $\mathcal{P} = \{P_0, P_1, P_2, \ldots, P_M\}$, and

$$P_0 = \langle v'_1, v'_2, \ldots, v'_n \rangle,$$

where $v'_i = (i, i^2, 0)$ is a 3D point for $i = 1, \ldots, n$.

For each $e_r = (v_i, v_j)$ in G, we have a corresponding sequence (3D chain)

$$P_r = \langle v'_1, v'_2, \ldots, v'_{i-1}, v'_{i+1}, \ldots, v'_n, v''_1, v''_2, \ldots, v''_{j-1}, v''_{j+1}, \ldots, v''_n \rangle,$$

where $v'_i = (i, i^2, 0)$ and $v''_i = (i, i^2, \delta)$ are 3D points for $i = 1, \ldots, n$ and δ is an arbitrarily small positive real number less than 0.1.

Downloaded by Shandong University from www.liebertonline.com at 12/14/07. For personal use only.

We claim that G has an independent set of size k if and only if there is a chain C of k vertices such that the discrete Fréchet distance between C and a subsequence of P_r, S_r, is at most δ (i.e., $d_F(C, S_r) \le \delta$). The following claims are made with the detailed proofs omitted.

Claim A. P_r is a simple polygonal chain in 3D.

Claim B. S_r is a simple polygonal chain in 3D with $|S_r| = k$.

If G has an independent set of size k, then the chain C can be constructed as follows. Let the independent set of G be ordered as $I = \langle v_{i_1}, v_{i_2}, \dots, v_{i_k} \rangle$ with $i_1 < i_2 < \cdots < i_k$. For $r = 0, 1, \cdots, M$, we scan P_r in a greedy fashion to obtain the first v'_j or v''_j such that the first component of its coordinate is i_1. Repeat this process to obtain S_r. Then let any S_r be C. Obviously, C has k vertices and $|S_r| = k$ for $r = 0, 1 \dots, M$.

If there is a chain C of k vertices such that the discrete Fréchet distance between C and a subsequence of P_r, S_r, is at most δ (i.e., $d_F(C, S_r) \le \delta$), then we can see the following:

Property a. Let $P_r = \langle v'_1, v'_2, \dots, v'_{i-1}, v'_{i+1}, \dots, v'_n, v''_1, v''_2, \dots, v''_{j-1}, v''_{j+1}, \dots, v''_n \rangle$, then $d(v'_p, v''_q) > 3$ for all $p \neq q$.

Property b. Let $P_r = \langle v'_1, v'_2, \dots, v'_{i-1}, v'_{i+1}, \dots, v'_n, v''_1, v''_2, \dots, v''_{j-1}, v''_{j+1}, \dots, v''_n \rangle$, then $d(v'_p, v''_p) \le \delta$ for all $p \neq i, p \neq j$.

Property c. Let $P_r = \langle u_1, u_2, \dots, u_{O(n)} \rangle$, then $|d(u_p, u_q) - d(u_{p'}, u_{q'})| >> \delta$ as long as the first components of the 4 coordinates of $u_p, u_q, u_{p'}, u_{q'}$ are all different.

As δ is very small, when $d_F(C, S_r) \le \delta$, the vertices of C and S_r must be matched orderly in a one-to-one fashion. (In other words, the man walking on C and the dog walking on S_r must move/jump together at each vertex; otherwise, $d_F(C, S_r) > 3 >> \delta$.) We now claim that the (ordered) vertices of C correspond to an independent set I of G; moreover, if $C = \langle C_1, C_2, \dots, C_k \rangle$ and $C_p = (x_p, y_p, z_p)$, then $v_{x_p} \in I$. Suppose that $C_p = (x_p, y_p, z_p)$, $C_q = (x_q, y_q, z_q)$ and $v_{x_p}, v_{x_q} \in I$ but there is an edge $e_t = (v_{x_p}, v_{x_q}) \in E$. By our construction of P_t (from e_t), v'_{x_p} and v''_{x_q} are not included in P_t and v'_{x_q} precedes v''_{x_p} in P_t. This is a contradiction.

To conclude the proof of this theorem, notice that the reduction take $O(MN)$ time. ■

In the example shown in Figure 2, we have

$$P_1 = \langle v'_1, v'_3, v'_4, v'_5, v''_1, v''_2, v''_4, v''_5 \rangle,$$

$$P_2 = \langle v'_1, v'_3, v'_4, v'_5, v''_1, v''_2, v''_3, v''_5 \rangle,$$

$$P_3 = \langle v'_2, v'_3, v'_4, v'_5, v''_1, v''_3, v''_4, v''_5 \rangle,$$

$$P_4 = \langle v'_2, v'_3, v'_4, v'_5, v''_1, v''_2, v''_3, v''_5 \rangle,$$

$$P_5 = \langle v'_1, v'_2, v'_4, v'_5, v''_1, v''_2, v''_3, v''_5 \rangle, \quad \text{and}$$

$$P_6 = \langle v'_1, v'_2, v'_3, v'_5, v''_1, v''_2, v''_3, v''_4 \rangle.$$

An example of P_3 is shown in Figure 1 as well, in which case black nodes are on the $Z = 0$ plane and white nodes are on the $Z = \delta$ plane (apparently for the visualization reason, the XY-plane is slanted). The solid segments are on the $Z = 0$ plane, the dotted segments are on the $Z = \delta$ plane and the only dashed segment connects two points on different planes. Corresponding to the optimal independent set $\{v_1, v_3, v_5\}$ in G, the optimal local alignment $C = \langle v'_1, v'_3, v'_5 \rangle$ matches P_3 at its subsequence $S_3 = \langle v''_1, v''_3, v''_5 \rangle$.

Downloaded by Shandong University from www.liebertonline.com at 12/14/07. For personal use only.

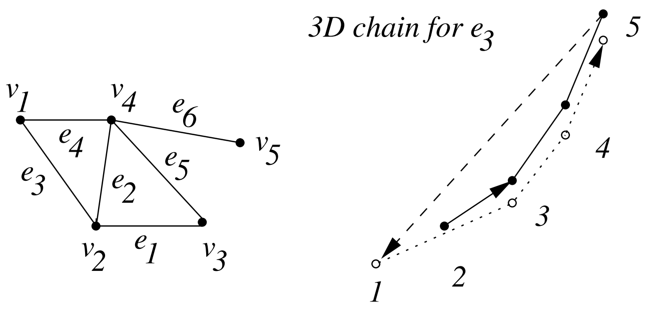

FIG. 2. Illustration of a simple graph for the reduction.

Corollary 3.1. *Given $D = \delta$ and when both translation and rotation are allowed, the (maximization version of) PLSA problem does not admit any approximation of factor $n^{1-\epsilon}$ unless P=NP.*

Proof. Due to Property a, b, and c, translation/rotation will not be able to generate another C' which is topologically different from C. ■

Notice that in our proof all the adjacent vertices in C could be non-adjacent in P_i, for $i = 0, 1, \ldots, m$. Biologically, this might be a problem as one residue alone sometimes cannot carry out any biological function. Define a *c-substring* or a *c-subchain* of P_i as a continuous subchain of P_i with at least c vertices. Unfortunately, even if we introduce this condition by forcing that C is composed of k ordered c-substrings of each P_i, for some constant c, the above proof can be modified to maintain a valid reduction from Independent Set. Call this corresponding problem Protein c-Local Structure Alignment (PcLSA), in which C must be composed of k ordered c-subchains of each P_i. We have the following corollary.

Corollary 3.2. *The maximization version of PcLSA does not admit any approximation of factor $n^{1-\epsilon}$ unless P=NP.*

4. POLYNOMIAL TIME SOLUTIONS FOR PLSA WHEN *m* IS SMALL

In this section, we present a polynomial time solution for the PLSA problem when m is a constant. We first show a dynamic programming solution for the static PLSA and then we show how to use that as a subroutine for the general PLSA problem, when m is small.

4.1. A dynamic programming solution for the static PLSA when m is small

In this subsection, we present a dynamic programming solution for the static PLSA problem when m is small. Such a solution can be used as a subroutine for the general PLSA problem. We first consider the case when $m = 2$. Besides C, we try to maximize the length of the aligned subsequences in $P_1 = A$ and $P_2 = B$ with $|A| = n_1, |B| = n_2$. For ease of description, we only show how to obtain these lengths that are stored in $D[-,-,-,-]$ and $M[-,-,-,-]$, respectively. It is easy to reconstruct C from these arrays.

Let $A[i_1, i_2]$ be a subchain of A starting from the index i_1 and ending at the index i_2. Let $B[j_1, j_2]$ be a subchain of B starting from the index j_1 and ending at the index j_2. $D[i_1, i_2, j_1, j_2]$ stores the length of the aligned subsequences of $A[i_1, i_2]$ as a consequence of the alignment of C and $A[i_1, i_2]$, and C and $B[j_1, j_2]$. $M[i_1, i_2, j_1, j_2]$ is defined symmetrically.

Intuitively $D[-,-,-,-]$ stores the length of aligned subsequences from chain A (dog's route) and $M[-,-,-,-]$ stores the length of aligned subsequences from chain B (man's route). Define $T_F(i_1, i_2, j_1, j_2)$ as the sum of aligned subsequences in both $A[i_1, i_2]$ and $B[j_1, j_2]$. Writing $A[i]$ as a_i and $B[j]$ as b_j, we have the dynamic programming solution as follows.

$$T_F(i_1, i_2, j_1, j_2) = D(i_1, i_2, j_1, j_2) + M(i_1, i_2, j_1, j_2),$$

Downloaded by Shandong University from www.liebertonline.com at 12/14/07. For personal use only.

where

$$D(i_1, i_2, j_1, j_2) = \max \begin{cases} \max\limits_{i_1 \le k_1 < i_2} \{D(i_1, k_1, j_1, j_2) + 1\} & \text{if } d(a_{i_2}, b_{j_2}) \le \delta, \backslash\backslash \text{ dog moves} \\ \max\limits_{i_1 \le k_1 < i_2, j_1 \le k_2 < j_2} \{D(i_1, k_1, j_1, k_2) + 1\} & \text{if } d(a_{i_2}, b_{j_2}) \le \delta, \backslash\backslash \text{ both move} \\ \max\limits_{j_1 \le k_2 < j_2} \{D(i_1, i_2, j_1, k_2)\} & \text{if } d(a_{i_2}, b_{j_2}) \le \delta, \backslash\backslash \text{ dog stays} \end{cases} \tag{1}$$

and

$$M(i_1, i_2, j_1, j_2) = \max \begin{cases} \max\limits_{i_1 \le k_1 < i_2} \{M(i_1, k_1, j_1, j_2)\} & \text{if } d(a_{i_2}, b_{j_2}) \le \delta, \backslash\backslash \text{ man stays} \\ \max\limits_{i_1 \le k_1 < i_2, j_1 \le k_2 < j_2} \{M(i_1, k_1, j_1, k_2) + 1\} & \text{if } d(a_{i_2}, b_{j_2}) \le \delta, \backslash\backslash \text{ both move} \\ \max\limits_{j_1 \le k_2 < j_2} \{M(i_1, i_2, j_1, k_2) + 1\} & \text{if } d(a_{i_2}, b_{j_2}) \le \delta, \backslash\backslash \text{ man moves} \end{cases} \tag{2}$$

The boundary cases are handled as follows.

$$D(i_1, i_1, j_1, j_1) = M(i_1, i_1, j_1, j_1) = \begin{cases} 1 & \text{if } d(a_{i_1}, b_{j_1}) \le \delta, \\ 0 & \text{if } d(a_{i_1}, b_{j_1}) > \delta. \end{cases} \tag{3}$$

The final solution value is stored in $T_F[1, n_1, 1, n_2]$. We have the following theorem.

Theorem 4.1. *When $m = 2$, the static PLSA problem can be solved in $O(n^4)$ time and space.*

It is easy to generalize this algorithm to the more general case when m is some constant. We thus have the following corollary.

Corollary 4.1. *When m is a constant, the static PLSA problem can be solved in $O(m^3 n^{2m})$ time and $O(mn^{2m})$ space.*

4.2. A polynomial time solution for PLSA when m is small

Apparently, for any solution for PLSA, we should allow translation and rotation. When $m = 2$ and when both translation and rotation are allowed, we can use a method similar to that in Jiang et al. (2007) to compute the optimal local alignment with fixed δ. The idea is as follows. Without loss of generality, we assume that A is static and we translate/rotate B and let $\tau(B)$ be the copy of B after some translation/rotation. Let $|A| = n_1, |B| = n_2$ and let f be the degree of freedom for moving B. As we are in 3D and both translation and rotation are allowed, we have $f = 6$. We can enumerate all possible configurations for A and $\tau(B)$ to realize a discrete Fréchet distance of δ. There are $O((n_1 n_2)^f) = O(n^{12})$ number of such configurations, following an argument similar to Wenk (2002) and Jiang et al. (2007). Then for each configuration, we can use the above Theorem 4.1 to obtain the optimal local alignment for each configuration and finally we simply return the overall optimal solution.

Corollary 4.2. *When $m = 2$ and when both translation and rotation are allowed, the PLSA problem can be solved in $O(n^{16})$ time and $O(n^4)$ space.*

Notice that, for a similar problem in 3D, namely, computing the largest common point sets under the bottleneck metric between two sets with at most n points, the running time is $O(n^{32.5})$ (Ambuhl et al., 2000). So the high running time of our solution is not really surprising.

We comment that, when m is larger, but still a constant, the above idea can be carried over so that we will still be able to solve PLSA in polynomial time. It follows from Wenk (2002) and Jiang et al. (2007)

Downloaded by Shandong University from www.liebertonline.com at 12/14/07. For personal use only.

that we have $O(n^{mf}) = O(n^{6m})$ number of configurations between the m chains. Then we can again use Corollary 4.1 to obtain the optimal local alignment for each configuration. The overall complexity would be $O(n^{6m} \times m^3 n^{2m}) = O(m^3 n^{8m})$ time and $O(mn^{2m})$ space. Certainly, such an algorithm is only meaningful in theory.

Corollary 4.3. *When m is a constant and when both translation and rotation are allowed, the PLSA problem can be solved in $O(m^3 n^{8m})$ time and $O(mn^{2m})$ space.*

5. CONCLUSION

In this paper, for the first time, we study the complexity/algorithmic aspects of the famous protein local structure alignment problem under the discrete Fréchet distance. We show that the general problem is NP-complete; in fact, it is even NP-hard to approximate within a factor of $n^{1-\epsilon}$. On the other hand, when a constant number of proteins are given then the problem can be solved in polynomial time. It would be interesting to see the empirical comparisons of protein local structure alignment under the discrete Fréchet distance (for a small number of protein backbones and possibly with some heuristic components, as done in Jiang et al. [2007]) with the existing methods. Another open problem, obviously, is whether it is possible to improve the running time of the dynamic programming algorithms in Section 4. We feel that when suitable scoring functions are given, as in Shatsky et al. (2005), it might be possible to speed up the dynamic programming algorithm.

REFERENCES

Akutsu. T. 1996. Protein structure alignment using dynamic programming and iterative improvement. *IEICE Trans. Inform. Syst.* E79-D, 1629–1636.

Ambuhl, C., Chakraborty, S., and Gartner, B. 2000. Computing largest common point sets under approximate congruence. *Lect. Notes Comput. Sci.* 1879, 52–63.

Alt, H., and Godau, M. 1992. Measuring the resemblance of polygonal curves. *Proc. 8th Annu. Symp. Comput. Geom.* 102–109.

Alt, H., and Godau, M. 1995. Computing the Fréchet distance between two polygonal curves. *Int. J. Comput. Geometry Appl.* 5, 75–91.

Alt, H., Knauer, C., and Wenk, C. 2001. Matching polygonal curves with respect to the Fréchet distance. *Proc. 18th Annu. Symp. Theoret. Aspects Comput. Sci.* 63–74.

Buchin, K., Buchin, M., and Wenk, C. 2006. Computing the Fréchet distance between simple polygons in polynomial time. *Proc. 22nd Annu. Symp. Comput. Geom.* 80–87.

Conte, L., Ailey, B., Hubbard, T., et al. 2000. SCOP: a structural classification of protein database. *Nucleic Acids Res.* 28, 257–259.

Cormen, T., Leiserson, C., Rivest, R., et al. 2001. *Introduction to Algorithms*, 2nd ed., MIT Press, Cambridge, MA.

Efrat, A., Itai, A., and Katz, M. 2001. Geometry helps in bottleneck matching and related problems. *Algorithmica* 31, 1–28.

Eiter, T., and Mannila, H. 1994. Computing discrete Fréchet distance. [Technical report CD-TR 94/64]. Technical University of Vienna, Vienna, Austria.

Fréchet, M. 1906. Sur quelques points du calcul fonctionnel. *Rendic. Circolo Math. Palermo* 22, 1–74.

Godau, M. 1998. On the complexity of measuring the similarity between geometric objects in higher dimensions [Ph.D. dissertation]. Freie Universitaet, Berlin, Germany.

Hästad, J. 1999. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Math.* 182, 105–142.

Holm, L., and Park, J. 2000. DaliLite workbench for protein structure comparison. *Bioinformatics* 16, 566–567.

Holm, L., and Sander, C. 1993. Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.* 233, 123–138.

Hui, P., and Shaefer, M. 2004. Paired pointset traversal. *Proc. 15th Annu. Symp. Algorithms Comput.* 534–544.

Indyk, P. 2002. Approximate nearest neighbor algorithms for Fréchet distance via product metrics. *Proc. 18th Annu. Symp. Comput. Geom.* 102–106.

Jiang, M., Xu, Y., and Zhu, B. 2007. Protein structure-structure alignment with discrete Fréchet distance. *Proc. 5th Asia-Pacific Bioinform. Conf.* 131–141.

Jiang, T., and Li, M. 1995. On the approximation of shortest common supersequences and longest common subsequences. *SIAM J. Comput.* 24, 1122–1139.

Miao, X., Waddell, P., and Valafar, H. 2008. TALI: local alignment of protein structures using backbone torsion angles. *J. Bioinform. Comput. Biol.* (in press).

Orengo, C., Michie, A., Jones, S., et al. 1997. CATH—a hierarchic classification of protein domain structures. *Structure* 5, 1093–1108.

Oritz, A., Strauss, C., and Olmea, O. 2002. MMAMOTH (matching molecular models obtained from theory): an automated method for model comparison. *Protein Sci.* 11, 2606–2621.

Qian, J., Li, S., Bu, D., et al. 2007. Finding compact structural motifs. *Proc. 18th Annu. Symp. Combin. Pattern Matching* 142–149.

Shindyalov, I., and Bourne, P. 1998. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng.* 11, 739–747.

Shatsky, M., Shulman-Peleg, A., Nussinov, R., et al. 2005. Recognition of binding patterns common to a set of protein structures. *Lect. Notes Comput. Sci.* 3500, 440–455.

Taylor, W., and Orengo, C. 1989. Protein structure alignment. *J. Mol. Biol.* 208, 1–22.

Wenk, C. 2002. Shape matching in higher dimensions [Ph.D. dissertation]. Freie Universitaet, Berlin, Germany.

Address reprint requests to:
Dr. Binhai Zhu
Department of Computer Science
Montana State University
Bozeman, MT 59717-3880

E-mail: bhz@cs.montana.edu

Downloaded by Shandong University from www.liebertonline.com at 12/14/07. For personal use only.