# A Practical Solution for Aligning and Simplifying Pairs of Protein Backbones Under the Discrete Fréchet Distance

Tim Wylie[1], Jun Luo[2], and Binhai Zhu[1]

[1] Department of Computer Science, Montana State University, Bozeman, MT 59717-3880, USA. Email: timothy.wylie,bhz@cs.montana.edu.

[2] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. Email: jun.luo@siat.ac.cn.

**Abstract.** Aligning and comparing two polygonal chains in 3D space is an important problem in many areas of research, like in protein structure alignment. A lot of research has been done in the past on this problem, using RMSD as the distance measure. Recently, the discrete Fréchet distance has been applied to align and simplify protein backbones (geometrically, 3D polygonal chains) by Jiang et al., with insightful new results found. On the other hand, as a protein backbone can have as many as 500$\sim$600 vertices, even if a pair of chains are nicely aligned, as long as they are not identical, it is still difficult for humans to visualize their similarity and difference. In 2008, a problem called CPS-3F was proposed to simplify a pair of 3D chains simultaneously under the discrete Fréchet distance. However, it is still open whether CPS-3F is NP-complete or not. In this paper, we first present a new practical method to align a pair of protein backbones, improving the previous method by Jiang et al. Finally, we present a greedy-and-backtrack method, using the new alignment method as a subroutine, to handle the CPS-3F problem. We also prove two simple lemmas, giving some evidence to why our new method works well. Some preliminary empirical results using some proteins from the Protein Data Bank (PDB), with comparisons with the previous method, are presented.

## 1 Introduction

The alignment and simplification of polygonal chains are well studied problems in many fields, like computational geometry and computer vision, etc (see [1, 4, 20] and the references therein). Specifically, aligning and matching similar proteins is a central problem in structure biology. It is believed that under a lot of situations structural similarity implies functional similarity [14]. This has been verified in certain situations, e.g., in [13]. For this reason there have been quite a few famous practical systems for protein structure alignment since 1989, e.g., SSAP [19], DALI [11, 10], CATH [15], CE [17], SCOP [7], MAMMOTH [16], ProteinDBS [18] and 3D-BLAST [21]. We comment that the backbone of a protein is very much a 3D polygonal chain, with each vertex being the $\alpha$-carbon atom of a residue (amino acid).

Among many of the works done before on protein global structure alignment and protein local structure alignment, almost all use the distance measure called RMSD

(Root Mean Square Distance). Given two $m$-vectors $V_1 = \langle u_1, u_2, ..., u_m \rangle$ and $V_2 = \langle v_1, v_2, ..., v_m \rangle$,

$$RMSD(V_1, V_2) = \sqrt{\frac{\sum_i (u_i - v_i)^2}{m}}.$$

The drawback of RMSD, obviously, is its relation to $m$. So given two chains $C_1, C_2$ with $m$ vertices, if we add some vertices on $C_1$ and $C_2$ by alternatively duplicating/repeating some different vertices in $C_1$ and $C_2$ to obtain $C_1', C_2'$, then $RMSD(C_1', C_2')$ could be dramatically different from $RMSD(C_1, C_2)$, even though geometrically $C_1'$ and $C_2'$ are just as close as $C_1$ and $C_2$ are.

Recently, Jiang, Xu, and Zhu proposed to use the discrete Fréchet distance as a measure of similarity between two given protein backbones [12]. Theoretically, given two 3D chains with $m$ and $n$ vertices respectively, it takes $O(n^7 m^7 \log(n+m))$ to align them (i.e., minimize the discrete Fréchet distance between them, under both translation and rotation) [12]. (For 2D chains, or when only translation is allowed, the running times are lower but still impractical.) Because of that, Jiang et al. proposed a simple heuristic problem which is to simply align the starting, ending vertices and the centers of two 3D chains. For many protein backbones from PDB, it was reported that while this heuristic method is slower, it produces more accurate results compared with the famous ProteinDBS software, even using the traditional RMSD as a distance measure. In this paper, we will present a new method based on some theoretical evidence, which improves the results by Jiang et al., through some empirical comparisons.

On the other hand, as long as two 3D protein backbones are not identical, even after we align them it is hard for humans to see their difference and similarity. The reason is that a protein backbone has as many as 500$\sim$600 vertices and human eyes simply cannot handle a pair of 3D protein backbones with a total of around 1000 vertices (each represents an $\alpha$-carbon atom). Motivated by this, Bereg et al. proposed the following Chair Pair Simplification (CPS-3F) problem [5].

**Instance:** Given a pair of 3D chains $A$ and $B$ in 3D, each with length $O(n)$, an integer $K$, and three real numbers $\delta_1, \delta_2, \delta_3$.
**Problem:** Does there exist a pair of chains $A', B'$ each of at most $K$ vertices such that the vertices of $A', B'$ are from $A, B$ respectively, and $d_{\mathcal{F}}(A, A') \leq \delta_1, d_{\mathcal{F}}(B, B') \leq \delta_2, d_{\mathcal{F}}(A', B') \leq \delta_3$?

It is not known yet whether CPS-3F is NP-complete or not. In this paper we will present a practical solution for CPS-3F, using our new alignment method as a subroutine.

The paper is organized as follows. In Section 2 we discuss the discrete Fréchet distance and the related background. In Section 3 we present a new heuristic method to align two 3D chains under the discrete Fréchet distance. In Section 4 we propose a greedy-and-backtrack method for the problem of simplifying a pair of 3D chains simultaneously. In Section 5 we present some empirical results, together with some comparisons with the previous method. Finally in Section 6, we conclude the paper with several open problems.

## 2 Background

The Fréchet distance is a well-known distance measure defined by Maurice Fréchet in 1906 [9]. It was applied to compare the similarity of polygonal curves in the 1990s by Alt and Godau [2, 3]. Given two polygonal curves (chains or simply paths), the Fréchet distance is the shortest-length leash connecting a man and a dog, each walking forward on a curve and being able to control their speed. The discrete version, defined in 1994 by Eiter and Mannila, requires that the man and dog hop forward or stop at the vertices of the polygonal chains [8]. One of the prominent applications of the discrete Fréchet distance is on comparing the similarity of protein backbones [12, 22], in which case each vertex represents an $\alpha$-carbon atom and clearly has a biological meaning. So in this case, the discrete Fréchet distance is more meaningful. (For matching polygonal chains using the continuous Fréchet distance, interested readers are referred to [20].) We next go over the definition of the discrete Fréchet distance.

Given two paths, we define their discrete Fréchet distance as follows. (We use the graph-theoretic term "paths" instead of the geometric term "polygonal chains" here because our definition makes no assumption that the underlying space of points is geometric.) We use $d(a, b)$ to represent the Euclidean distance between two 3D points $a$ and $b$, but certainly it can be replaced with some other distance measure, depending on applications.

**Definition 1.** *Given a path $P = \{p_1, \ldots, p_n\}$ of $n$ vertices, a $t$-**walk** along $P$ partitions the path into $t$ disjoint non-empty subpaths $\{P_i\}_{i=1..t}$ such that $P_i = \{p_{n_{i-1}+1}, \ldots, p_{n_i}\}$ and $0 = n_0 < n_1 < \ldots < n_t = n$.*

*Given two paths $A = \{a_1, \ldots, a_m\}$ and $B = \{b_1, \ldots, b_n\}$, a **paired walk** along $A$ and $B$ is a $t$-walk $\{A_i\}_{i=1..t}$ along $A$ and a $t$-walk $\{B_i\}_{i=1..t}$ along $B$ for some $t$, such that, for $1 \leq i \leq t$, either $|A_i| = 1$ or $|B_i| = 1$ (that is, either $A_i$ or $B_i$ contains exactly one vertex). The **cost** of a paired walk $W = \{(A_i, B_i)\}$ along two paths $A$ and $B$ is*

$$d_{\mathcal{F}}^{W}(A, B) = \max_{i} \max_{(a,b) \in A_i \times B_i} d(a, b).$$

*The **discrete Fréchet distance** between two paths $A$ and $B$ is*

$$d_{\mathcal{F}}(A, B) = \min_{W} d_{\mathcal{F}}^{W}(A, B).$$

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

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

1. $|B_i| > |A_i| = 1$: the person stays and the dog hops forward;
2. $|A_i| > |B_i| = 1$: the person hops forward and the dog stays;
3. $|A_i| = |B_i| = 1$: both the person and the dog hop forward.

The following figure shows the relationship between discrete and continuous Fréchet distances. In Figure 1 (I), we have two chains $< a_1, a_2, a_3 >$ and $< b_1, b_2 >$, the continuous Fréchet distance between the two is the distance from $a_2$ to segment $\overline{b_1 b_2}$, i.e.,

$d(a_2, o)$. The discrete Fréchet distance is $d(a_2, b_2)$. Clearly, the discrete Fréchet distance could be arbitrarily larger compared with the continuous distance. On the other hand, if we put enough sample points on the two chains, then the resulting discrete Fréchet distance, i.e., $d(a_2, b)$ in Figure 1 (II), can closely approximate $d(a_2, o)$.

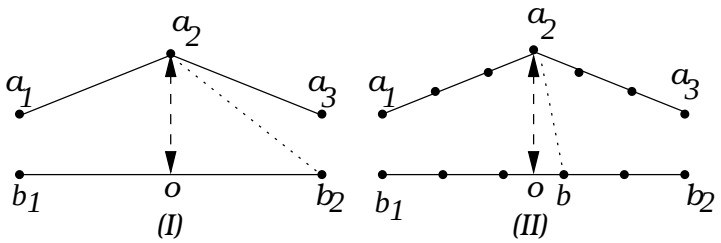


**Fig. 1**. The relationship between between discrete and continuous Fréchet distance.

With the standard dynamic programming technique, it is not hard to obtain the following theorem (which serves an important subroutine in our algorithms).

**Theorem 1.** *[8] The discrete Fréchet distance between two paths with $m$ and $n$ vertices respectively can be computed in $O(mn)$ time.*

## 3 Algorithm for Aligning 3D Polygonal Chains

The optimal (global structure-structure) alignment problem is formally defined as follows.

**Definition 2.** *Given two 3D polygonal chains $A$ and $B$, a transformation class $T$, and a distance measure $dist(-)$, find a transformation $\tau \in T$ such that $dist(A, \tau(B))$ is minimized.*

Of course, in our case $T$ contains both rotation and translation, and $dist = d_{\mathcal{F}}$.

Let $A = \langle a_1, a_2, ..., a_m \rangle$ and $B = \langle b_1, b_2, ..., b_n \rangle$. It was shown that the optimal alignment problem under the discrete Fréchet distance can be solved in $O(n^7 m^7 \log(n+m))$ time [12]. This is certainly impractical. So Jiang et al. presented a heuristic method which focuses on first aligning the center $a$ of $A$ and the center $b$ of $B$. (Given a 3D chain $C$ of $n$ vertices, the coordinates of each vertex $c_i$ of $C$ is really a vector $\boldsymbol{c}_i$, the center $c$ corresponds to $\boldsymbol{c} = \frac{\sum_i \boldsymbol{c}_i}{n}$.) Then a rotation is performed such that $\triangle a_1 a a_m$ and $\triangle b_1 b b_n$ are on the same plane. Finally, some local improvements are performed until the discrete Fréchet distance cannot be further improved. While this algorithm is still slower compared with some of the known software (like ProteinDBS [18]), it in fact can improve the accuracy in many situations [12].

We use a slightly different idea here. We can prove that if we first move $B$ such that $b_1$ is located exactly at $a_1$ and obtain subsequently an optimal solution, then this solution is a factor-2 approximation for the optimal alignment problem (when $a_1$ does

not necessarily collide with $b_1$). Of course, a factor-2 approximation is probably not good enough for biological applications. So while colliding $b_1$ at $a_1$ is our starting point, our algorithm goes way beyond that. Our complete (heuristic) algorithm is as follows.

**Algorithm Align**($A, B$):
Input: Two polygonal chains $A = \langle a_1, ..., a_m \rangle$ and $B = \langle b_1, ..., b_n \rangle$.

1. Translate $B$ so that $d(b_1, a_1) = 0$.
2. Let $\beta$ be the midpoint of $\langle a_m, b_n \rangle$. Rotate $B$ around the axis line $(a_1, \beta)$ so that $d(a_m, b_n)$ is minimized. Let $a_i \in A$ and $b_j \in B$ be the two vertices such that $d(a_i, b_j) = d_{\mathcal{F}}(A, B)$.
3. Initialize $O^*(A, B) \longleftarrow d_{\mathcal{F}}(A, B)$.
4. Loop until no improvement of $O^*(A, B)$ is made.
   (a) Rotate until no improvement of $O^*(A, B)$ is made.
      i. Let $\gamma$ be the midpoint between $a_1, b_1$. Let $\mu$ be the midpoint between $a_i, b_j$.
      ii. Rotate $B$ around the axis line $(\gamma, \mu)$ by $\theta$ such that $-180 \leq \theta \leq 180$ and $|\theta|$ is the largest angle which results in $d_{\mathcal{F}}(A, B) < O^*(A, B)$.
      iii. Update $O^*(A, B) \longleftarrow d_{\mathcal{F}}(A, B)$ and update $a_i, b_j$ accordingly.
   (b) Translate until no improvement of $O^*(A, B)$ is made.
      i. Translate $B$ along the vector $\overrightarrow{b_j a_i}$ by $\delta$ such that $\delta$ is the largest value which results in $d_{\mathcal{F}}(A, B) < O^*(A, B)$.
      ii. Update $O^*(A, B) \longleftarrow d_{\mathcal{F}}(A, B)$ and update $a_i, b_j$ accordingly.
5. Return $A, B, O^*(A, B)$.

While we are unable to prove that this algorithm is a PTAS for the optimal alignment problem, we believe that for practical data it is almost a PTAS. Some supporting empirical results will be presented in Section 5. In the following, we give some evidence that, when translating $B$ such that $b_1$ collides with $a_1$ and obtaining subsequently an optimal solution (with $b_1$ sticking at $a_1$), in fact gives us a factor-2 approximation for the optimal alignment problem.

**Lemma 1.** *Given two 3D polygonal chains $A, B$ of length $m, n$ respectively such that the optimal $d_{\mathcal{F}}(A, B) = \epsilon$, an optimal transformation $\tau$ aligning $A$ and $\tau(B)$ such that $d(a_1, \tau(b_1)) = 0$ gives a 2-approximation for the optimal alignment problem.*

*Proof.* Let $a_i \in A$, $b_j \in B$ be the two vertices defining the optimal discrete Fréchet distance $d^*_{\mathcal{F}}(A, B) = \epsilon$ for the optimal alignment of $A$ and $B$. Then, by definition $d^*_{\mathcal{F}}(A, B) = d(a_i, b_j)$; moreover, $d(a_1, b_1) \leq d^*_{\mathcal{F}}(A, B) = \epsilon$. Now run a translation $\tau'$ for this optimal alignment such that $\tau'(b_1)$ collides at $a_1$. Then obviously $d_{\mathcal{F}}(A, \tau'(B)) \leq \epsilon + d(a_1, b_1) \leq 2\epsilon$. As $\tau$ is an optimal transformation with the constraint that $d(a_1, \tau(b_1)) = 0$,

$$d_{\mathcal{F}}(A, \tau(B)) \leq d_{\mathcal{F}}(A, \tau'(B)) \leq 2\epsilon.$$

□

In our algorithm, while initially we force $b_1$ to collide at $a_1$ at Step 1-3, in the main loop, the rotations and translations will typically force them to stay away from each other.

## 4 Algorithm for Chain Pair Simplification

In this section, we cover the CPS-3F problem. Note that several versions of the (single) polygonal chain simplification problem under the discrete Fréchet distance were recently studied by Bereg et al. [5]. These problems, not surprisingly, are all polynomially solvable. Bereg et al. also considered a variation of the CPS-3F problem and proved that it is NP-complete. But, for the most interesting CPS-3F problem, it is still open whether the problem is polynomially solvable or is NP-complete.

Here we try to solve the CPS-3F problem with a practical solution. It is known that the greedy method does not always work even for simplifying a single chain under the discrete Fréchet distance, with some counterexample presented in [5]. Here, we use a greedy-and-backtrack method. Our ideas are as follows: (1) While greedy does not always work, for protein backbones we have the implicit condition that for all possible $i$ $d(a_i, a_{i+1}) \approx 3.7$ to 3.8 (angstroms), i.e., the neighboring $\alpha$-carbon atoms in a protein backbone have an almost uniform length. With this condition a lot of counterexamples do not hold anymore. (2) To mend possible holes of the algorithm, when we are stuck at a certain point (using the greedy method), we backtrack some (constant number of) steps and re-try the greedy method again. While it is not known whether this algorithm leads to an optimal solution, it works pretty well for practical protein data, some of which are to be presented in Section 5.

We first show a simple lemma which helps us to determine whether the input could lead to an infeasible solution. Certainly, this lemma also implies that having an almost optimal alignment of $A$ and $B$, which results in that $d_{\mathcal{F}}(A, B)$ is almost as small as possible, is crucial for the success of the simplification algorithm.

**Lemma 2.** *Given two 3D polygonal chains $A$ and $B$, if a solution $(A', B')$ for CPS-3F is found with $d_{\mathcal{F}}(A, A') \leq \delta_1$, $d_{\mathcal{F}}(B, B') \leq \delta_2$ and $d_{\mathcal{F}}(A', B') \leq \delta_3$, then $d_{\mathcal{F}}(A, B) \leq \delta_1 + \delta_2 + \delta_3$.*

*Proof.* As the discrete Fréchet distance satisfies the triangle inequality, the lemma follows immediately. □

Let $\mathcal{B}(b, \delta)$ be a ball centered at point $b$ with radius $\delta$. Our heuristic algorithm, which is certainly based on that $A$ and $B$ are almost optimally aligned, is as follows.

**Algorithm SIMPLIFY**($A, B, \delta_1, \delta_2, \delta_3$):
Input: Two polygonal chains $A = \langle a_1, ..., a_m \rangle$ and $B = \langle b_1, ..., b_n \rangle$, a positive integer $K$, and three positive constants $\delta_1, \delta_2, \delta_3$.
Output: Two simplified chains $A' = \langle a'_1, ..., a'_K \rangle$ and $B' = \langle b'_1, ..., b'_K \rangle$.

1. Run the algorithm ALIGN($A, B$).
2. If $d_{\mathcal{F}}(A, B) > \delta_1 + \delta_2 + \delta_3$, report 'no valid solution' and exit.
3. Initialize $a'_1 \longleftarrow a_1, b'_1 \longleftarrow b_1, i \leftarrow 1, j \leftarrow 1$.
4. Loop until $i = j = K$.
   (a) Let $\langle a_{i,1}, a_{i,2}, ..., a_{i,p} (= a_I) \rangle$ be the maximal subsequence of $A$ which is inside $\mathcal{B}(a'_i, \delta_1)$ and let $\langle b_{j,1}, b_{j,2}, ..., b_{j,q} (= b_J) \rangle$ be the maximal subsequence of $B$ which is inside $\mathcal{B}(b'_j, \delta_2)$. (Note that $a'_i = a_{i,p'}$ for some $p' \leq p$ and $b'_j = b_{j,q'}$ for some $q' \leq q$.)

(b) Let $\langle a_{I+1}, a_{I+2}, ..., a_{I+s}\rangle$ be the maximal subsequence of $A$ which is inside $\mathcal{B}(a_{I+s'}, \delta_1)$ and let $\langle b_{J+1}, b_{J+2}, ..., b_{J+t}\rangle$ be the maximal subsequence of $B$ which is inside $\mathcal{B}(b_{J+t'}, \delta_2)$, with $s' \leq s, t' \leq t$.

(c) If $d(a_{I+s'}, b_{J+t'}) \leq \delta_3$, then
   i. $I \longleftarrow I + s$, $J \longleftarrow J + t$,
   ii. $a'_{i+1} \longleftarrow a_{I+s'}$, $b'_{j+1} \longleftarrow b_{J+t'}$,
   iii. $i \leftarrow i + 1, j \leftarrow j + 1$.

(d) Else if $d(a'_i, b_{J+t'}) \leq \delta_3$, then
   i. $J \longleftarrow J + t$, $b'_{j+1} \longleftarrow b_{J+t'}$,
   ii. $j \leftarrow j + 1$.

(e) Else if $d(a'_{I+s'}, b_j) \leq \delta_3$, then
   i. $I \longleftarrow I + s$, $a'_{i+1} \longleftarrow a_{I+s'}$,
   ii. $i \leftarrow i + 1$.

(f) Else backtrack by successively letting $a'_i$ be $a_{i,p'-1}, a_{i,p'-2}, ..., a_{i,1}$ and letting $b'_j$ be
$b_{j,q'-1}, b_{j,q'-2}, ..., b_{j,1}$, and loop over Steps 4.(a)-4.(e). If neither $i$ nor $j$ can be incremented over these pairs of $a'_i$ and $b'_j$, exit with a report 'no valid solution'.

We have several observations regarding this algorithm.

(1) At Step 3 we put $a_1, b_1$ as the first vertex of $A', B'$ respectively (which is biologically more interesting). Certainly we can use the greedy idea to compute $\mathcal{B}(a_i, \delta_1)$ such that it contains a maximal subsequence starting at $a_1$ and with $i$ maximized. Then $a'_1 \leftarrow a_i$. ($b'_1$ can be computed similarly.)

(2) If we backtrack as stated in Step 4.(f), the algorithm SIMPLIFY($A, B, \delta_1, \delta_2, \delta_3$) would take an exponential time in the worst case. In the actual implementation, we make the algorithm backtrack only $O(1)$ steps. Of course, this could mean that we might miss a feasible solution and report that no valid solution exists.

(3) As the protein backbones have the property that the neighboring vertices ($\alpha$-carbon atoms) $a_i$ and $a_{i+1}$ in a backbone satisfies $d(a_i, a_{i+1}) \approx 3.7 \sim 3.8$ (angstroms), in practice we should set $\delta_1, \delta_2 \geq 4$ (otherwise, we could not simplify $A$ and $B$).

(4) While we cannot state any theoretical result regarding this algorithm, it works pretty well in terms of accuracy, while the running time still needs to be further improved. Some empirical results will be presented in the next section.

Finally, while we set $|A'| = |B'| = K$ in the algorithm, certainly we can set them as $|A'| = K_1 = O(K)$ and $|B'| = K_2 = O(K)$.

## 5 Some Empirical Results

We present some empirical results in this section. The code was written in Python and run on a regular Dell desktop. We are currently working on a software implementation which will be available to the public for general use and a website showing example alignments and comparisons. There are several obstacles we need to get over, which we will discuss is the next section. Nevertheless, we present some interesting empirical results here, with comparison with the previous work [12].

In [12], rigorous studies are performed regarding comparing protein backbone 107j.a with the other seven chains from the PDB: 1hfj.c, 1qd1.b, 1toh, 4eca.c, 1d9q.d, 4eca.b,

4eca.d. These seven chains were reported to be similar to 107j.a by the ProteinDBS software (which takes a few seconds searching the whole PDB, which contained over 30,000 protein backbones at that time). Using the discrete Fréchet distance as distance measure, while taking much longer (close to one minute for each pair), the heuristic algorithm in [12] reported that 3 of the 7 chains are in fact not really similar to 107j.a. ProteinDBS subsequently updated their webpage for this. Here, in Table 1, we simply compare our ALIGN algorithm with that of [12]. We mostly focus on accuracy. All distances are measured in angstroms.

| Protein Chain (B) | RMSD [12] | $O^*(A,B)$ [12] | $O^*(A,B)$ (this paper) |
|---|---|---|---|
| 1hfj.c | 0.27 | 1.01 | 0.95 |
| 1qd1.b | 2.81 | 22.90 | 22.65 |
| 1toh | 2.91 | 35.09 | 22.06 |
| 4eca.c | 1.10 | 6.01 | 5.55 |
| 1d9q.d | 2.88 | 22.18 | 20.87 |
| 4eca.b | 1.09 | 5.76 | 5.64 |
| 4eca.d | 1.45 | 5.92 | 5.71 |

**Table 1. Alignment with 107j.a (Chain A), all eight chains have 325 vertices.**

| Protein Chain (B) | $\delta_1$ | $\delta_2$ | $\delta_3$ | Length | Reduced Length | Ratio | $d_{\mathcal{F}}(A,B)$ | $d_{\mathcal{F}}(A',B')$ |
|---|---|---|---|---|---|---|---|---|
| 1hfj.c | 4 | 4 | 1 | 325 | 109 | 33.5% | 0.95 | 0.95 |
| 1qd1.b | 4 | 4 | 50 | 325 | 109 | 33.5% | 22.65 | 24.96 |
| 1toh | 4 | 4 | 60 | 325 | 110 | 33.8% | 22.06 | 23.39 |
| 4eca.c | 4 | 4 | 17 | 325 | 109 | 33.5% | 5.55 | 7.96 |
| 1d9q.d | 4 | 4 | 43 | 325 | 109 | 33.5% | 20.87 | 23.68 |
| 4eca.b | 4 | 4 | 17 | 325 | 109 | 33.5% | 5.64 | 7.51 |
| 4eca.d | 4 | 4 | 18 | 325 | 109 | 33.5% | 5.71 | 7.82 |

**Table 2. Run of Algorithm SIMPLIFY, with 107j.a (Chain A), and $\delta_1 = \delta_2 = 4$.**

| Protein Chain (B) | $\delta_1$ | $\delta_2$ | $\delta_3$ | Length | Reduced Length | Ratio | $d_{\mathcal{F}}(A,B)$ | $d_{\mathcal{F}}(A',B')$ |
|---|---|---|---|---|---|---|---|---|
| 1hfj.c | 12 | 12 | 4 | 325 | 26 | 8.0% | 0.95 | 3.77 |
| 1qd1.b | 15 | 15 | 33 | 325 | 17 | 5.2% | 22.65 | 22.64 |
| 1toh | 16 | 16 | 44 | 325 | 17 | 5.2% | 22.06 | 27.24 |
| 4eca.c | 12 | 12 | 12 | 325 | 28 | 8.6% | 5.55 | 11.73 |
| 1d9q.d | 15 | 15 | 34 | 325 | 21 | 6.5% | 20.87 | 23.65 |
| 4eca.b | 12 | 12 | 13 | 325 | 28 | 8.6% | 5.64 | 12.57 |
| 4eca.d | 12 | 12 | 14 | 325 | 28 | 8.6% | 5.71 | 13.65 |

**Table 3. Run of Algorithm SIMPLIFY, with 107j.a (Chain A), and various $\delta_1$ and $\delta_2$.**

Note that in Table 2 and Table 3, the approximation ratio is defined as Ratio = (Reduced Length)/Length. Of course, it is not surprising that when Ratio gets below 10%, $d_{\mathcal{F}}(A',B') \approx 4d_{\mathcal{F}}(A,B)$ (against 1hfj.c in Table 3).

## 6 Concluding Remarks

In this paper we present a practical solution for aligning two protein backbones and for simplifying a pair of protein backbones simultaneously, both under the discrete Fréchet distance. Some empirical results are also obtained, with comparisons with the only known previous result by Jiang et al. [12]. Our eventual objective is to build a web-based software, on the public domain, for the relevant protein-based applications. To achieve that, several open problems need to be solved.

(1) The running times for both the ALIGN and SIMPLIFY algorithms need to be improved. At this point, it could take over a minute to align or simplify a pair of protein backbones (each with at most 400 vertices). Of course, most of the time of SIMPLIFY is in fact spent on running ALIGN. So, ignoring the ALIGN part, SIMPLIFY is fast (practically $O(n)$ as we only backtrack a constant number of steps).

(2) More empirical testings are needed to extract the possible relations between $\delta_1, \delta_2$ and $\delta_3$, and the practical approximation ratio.

(3) Theoretically, is it possible to design a practical PTAS for the (global structure-structure) alignment problem? Is it possible to prove that CPS-3F is NP-complete (which we conjecture to be)?

## Acknowledgment

This research is partially supported by NSF under grant DMS-0901034 and by NSF of China under project 60928006.

## References

1. H. Alt, B. Behrends, and J. Blömer. Approximate matching of polygonal shapes (extended abstract). In *Proceedings of the 7th Annual Symposium on Computational Geometry (SoCG'91)*, pages 186–193, 1991.
2. H. Alt and M. Godau. Measuring the resemblance of polygonal curves. In *Proceedings of the 8th Annual Symposium on Computational Geometry (SoCG'92)*, pages 102–109, 1992.
3. H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *Internat. J. Comput. Geom. Appl.*, 5:75–91, 1995.
4. H. Alt, C. Knauer, and C. Wenk. Matching polygonal curves with respect to the Fréchet distance. In *Proceedings of the 18th Annual Symposium on Theoretical Aspects of Computer Science (STACS'01)*, pages 63–74, 2001.
5. S. Bereg, M. Jiang, W. Wang, B. Yang and B. Zhu. Simplifying 3D polygonal chains under the discrete Fréchet distance. In *Proc. 8th Latin American Theoretical Informatics Symposium (LATIN'08)*, LNCS 4957, pages 630–641, 2008.
6. R. Cole. Slowing down sorting networks to obtain faster sorting algorithms. *J. ACM*, 34:200-208, 1987.
7. L. Conte, B. Ailey, T. Hubbard, S. Brenner, A. Murzin and C. Chothia. SCOP: a structural classification of protein database. *Nucleic Acids Research*, **28**:257-259, 2000.
8. T. Eiter and H. Mannila. Computing discrete Fréchet distance. *Tech. Report CD-TR 94/64, Information Systems Department, Technical University of Vienna*, 1994.
9. M. Fréchet. Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Mathematico di Palermo*, **22**:1-74, 1906.

10. L. Holm and J. Park. DaliLite workbench for protein structure comparison. *Bioinformatics*, **16**:566-567, 2000.
11. L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.*, **233**:123-138, 1993.
12. M. Jiang, Y. Xu and B. Zhu. Protein structure-structure alignment with discrete Fréchet distance. *J. of Bioinformatics and Computational Biology*, **6**:51-64, 2008.
13. C. Mauzy and M. Hermodson. Structural homology between rbs repressor and ribose binding protein implies functional similarity, *Protein Science*, **1**:843-849, 1992.
14. S. Needleman and C. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**:443-453, 1970.
15. C. Orengo, A. Michie, S. Jones, D. Jones, M. Swindles and J. Thornton. CATH—a hierarchic classification of protein domain structures. *Structure*, **5**:1093-1108, 1997.
16. A. Oritz, C. Strauss and O. Olmea. MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. *Protein Science*, **11**:2606-2621, 2002.
17. I. Shindyalov and P. Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Engineering*, **11**:739-747, 1998.
18. Chi-Ren Shyu, Pin-Hao Chi, Grant Scott, and Dong Xu. ProteinDBS: a real-time retrieval system for protein structure comparison. *Nucleic Acids Research*, 32:W572–575, 2004.
19. W. Taylor and C. Orengo. Protein structure alignment. *J. Mol. Biol.*, **208**:1-22, 1989.
20. C. Wenk. Shape Matching in Higher Dimensions. *PhD thesis, Freie Universitaet Berlin*, 2002.
21. J.-M. Yang and C.-H. Tung. Protein structure database search and evolutionary classification. *Nucleic Acids Research*, **34**:3646-3659, 2006.
22. B. Zhu. Protein local structure alignment under the discrete Fréchet distance. *J. Computational Biology*, **14(10)**:1343-1351, 2007.