Approximation Algorithms for 3-D Common Substructure Identification in Drug and Protein Molecules*

Samarjit Chakraborty[1] and Somenath Biswas[2]

[1] Eidgenössische Technische Hochschule, Zürich.
[2] University of Nebraska-Lincoln, USA.

Abstract. Identifying the common 3-D substructure in two drug or protein molecules is an important problem in synthetic drug design and molecular biology. This problem can be represented by the following geometric pattern matching problem : given two point sets A and B in three-dimensions, and a real number $\epsilon > 0$, find the maximum cardinality subset $S \subseteq A$ for which there is an isometry $\mathcal{I}$, such that each point of $\mathcal{I}(S)$ is within ϵ distance of a distinct point of B. Since it is difficult to solve this problem exactly, in this paper we have proposed several approximation algorithms with guaranteed approximation ratio.
Our algorithms can be classified into two groups. In the first we extend the notion of partial decision algorithms for ϵ-congruence of point sets in 2-D, to approximate the size of S. All the algorithms in this class exactly satisfy the constraint imposed by ϵ. In the second class of algorithms this constraint is satisfied only approximately. In the latter case, we improve the known approximation ratio for this class of algorithms, while keeping the time complexity unchanged. For the existing approximation ratio, we propose algorithms with substantially better running times. We also suggest several improvements of our basic algorithms, all of which have a running time of $O(n^{8.5})$. These improvements consist mainly of using randomization, and/or exploiting some general structural properties of protein molecules. As a result, we finally obtain an $O(n^{2.5} \log n)$ algorithm for the protein sequence alignment problem.

1 Introduction

Determining the structural similarities between a pair of molecules is a central issue in both synthetic drug design and in studying biomolecular recognition and interaction of proteins. An underlying assumption in structure based drug design is that drug activity is obtained through the molecular recognition and binding of one molecule called the *ligand* to the pocket of another, usually larger, molecule, called the *receptor*. Thus, two drug molecules having a large common substructure might show similar drug activity because of the substructure binding or

* TIK-Report No.69, February 1999
Reduced version of a dissertation submitted at Department of Computer Science and Engineering, Indian Institute of Technology Kanpur, in June 1998, by the first author.

docking itself into the pocket of the receptor. Similarly, identifying the common substructure between two protein molecules is crucial to the understanding of how they work, since families of proteins retain a common underlying 3-D structure.

In view of these applications, this paper considers the following problem : given two molecules, find the largest common rigid sub-unit contained in both the molecules. In other words, determine the rotation and translation of one molecule, relative to the other, for which the largest *fit* or superimposition between the two occurs. It is convenient to view this problem in a geometric setting. Towards this end, we will view each molecule as a set of points in 3-D space where each point is representative of an atom of the molecule. This implies that for each atom we retain only its location information, ignoring what kind of an atom it is (e.g. Carbon, Nitrogen)[3]. The problem, therefore, is to find a rigid transformation of one point set so that the number of points of the transformed set that are superimposed on the points of the other point set, is maximized.

This problem and some other related problems have been studied in computational geometry. In computational geometry parlance the above problem is called the *largest common point set* problem, or, LCP [ATT97, AH]. An easier problem is that of determining whether two equal cardinality point sets are congruent, i.e. does there exist a rigid transformation of one set such that the other point set is superimposed on to the transformed point set [Atk87, Aku92, AMWW88]. A generalization of this, called the *congruent copy detection*, asks, given two point sets A and B, whether there exists a rigid transformation under which A becomes congruent to a subset of B [CGH$^+$93, GMO94, dRL95].

The LCP as studied in computational geometry [ATT97] does not, however, directly apply to our problem. Since atom positions are fuzzy, it is impractical to consider an exact match between two atoms. Therefore, two points p_1 and p_2 representing two atoms should be considered superimposed if the distance between p_1 and p_2 is less than some predefined constant ϵ, called the *point location error*. Hence the abstract geometric version of our problem is that of finding the LCP of two 3-D point sets with the exact congruence replaced by *ϵ-congruence*.

1.1 Previous Work

Here we shall briefly review related work. Given two 2-D point sets, an $O(n^8)$ decision algorithm for ϵ-congruence under general isometry was given in [AMWW88] (in contrast to the $O(n \log n)$ algorithms [AMWW88, Ata84] for testing exact congruence). Alt *et al.* [AMWW88] and Iwanowski [Iwa90] gave $O(n \log n)$ algorithms for some restricted decision problems and Imai *et al.* [ISI89] gave $O(n^3 \log n)$ algorithms for Euclidean metric and $O(n)$ algorithm for maximum metric. Similarly, it was shown by Arkin *et al.* [AKM$^+$92] that improved running

[3] Any implementation of our algorithms will of course retain this information also; it is an easy matter to recast our algorithms incorporating this information, without changing their time complexity.

times can be obtained if the ϵ-*balls* around each point are disjoint. A related optimization problem is that of finding the minimum tolerance value ϵ, such that the two point sets are ϵ-congruent [AMWW88, GMO94]. A similar problem that has been extensively studied is that of finding the isometric transformation that minimizes either the directed or the undirected Hausdorff distance between two point sets [CGH$^+$93, CK92, HK90, HKK92, HKS91, Rot91].

There is also a large body of literature on computational chemistry which addresses the substructure identification problem, using mostly algorithms based on clique detection [MBD$^+$93, Wil95, MARW89]. The geometric hashing technique was used by Fisher *et al.* [FBNW92, FNW92] and also by Pennec *et al.* [PA94] for matching protein structures. Randomized versions of alignment were used by Finn *et al.* [FKL$^+$97] for identifying common substructures in drug molecules. Lesk, Pascarella and Argos, and Rao and Rossmann proposed iterative improvement methods [Les91, PA92, RR73]. Vriend and Sander developed a greedy method in which small fragments of protein molecules were assembled into larger structures [VS91]. Taylor and Orengo developed a double dynamic programming technique [TO89], and Šali and Overington developed a stochastic method using probability density functions [ŠO94].

All these methods, however, have one or more limitations and most are not systematic but are based on some heuristic. Moreover, none of them give a theoretical guarantee of the quality of the output. To address this, Akutsu [Aku96] proposed an approximation algorithm for the protein structure alignment problem, with a guaranteed approximation ratio. By using an algorithm for point set matching due to Goodrich *et al.* [GMO94], Akutsu's algorithm when given two 3-D point sets A and B corresponding to the protein structures outputs a point set $S \subseteq A$ of cardinality at least as large as that of the LCP of the two sets under ϵ-congruence. The algorithm guarantees the existence of a rigid transformation under which each point of S is at most within 8ϵ distance of a distinct point of B.

1.2 Our Work

In this paper we propose algorithms which improve the approximation ratio obtained by Akutsu, without incurring any increase in running time. Next, instead of approximating the constraint imposed by ϵ, we propose algorithms which approximate the size of the largest common point set, and give upper and lower bounds on its size. Our algorithms are based on the work of Schirra [Sch92]. They are however nontrivial generalizations of Schirra's algorithms which were for solving the ϵ-congruence decision problem for two equal cardinality point sets in 2-D, making use of the centroids of the point sets. Our approximation algorithms differ significantly these because they involve an optimization aspect and cannot make use of the centroids of the point sets. We next suggest various modifications of the basic algorithms, resulting in an improvement of their run time. The first is through an approximate graph matching due to Efrat and Itai [EI96] and the second is through the use of random sampling. Finally, we show that the time complexity of our algorithms can be further reduced by a

considerable extent in the case of protein chains, by exploiting certain general structural properties of proteins.

In the next section we formally state our abstract geometric problem and outline the algorithm due to Akutsu [Aku92] which approximates the ϵ-constraint. We show then how this algorithm, in combination with the decision algorithm due to Schirra [Sch92], leads to an algorithm for approximating the size of the LCP of two point sets. Following this, we state an exact algorithm for finding the LCP of two point sets when the underlying isometry is a pure rotation. In Section 4 we make use of this exact algorithm to improve the approximation ratio of both the algorithms of Section 2, following which we describe the improvements concerning running time. Section 6 concludes the paper.

2 Formal Definition of the Problem

We mentioned in the last section that a molecule is represented as a set of points in 3-D Euclidean space, where each point corresponds to an atom of the molecule. A protein may be viewed as a sequence (linear chain) of amino acids which folds in space to generate a three-dimensional structure. Although many substructure matching algorithms for proteins make use of this sequence property, such algorithms can hardly find useful substructures whose nature is intrinsically 3-D [PA94, FNW92]. So we make no assumption about the linear ordering of amino acids in a protein molecule.

First we state some definitions. A map $\mathcal{I} : \Re^n \rightarrow \Re^n$ is called an isometry if $d(a,b) = d(\mathcal{I}(a), \mathcal{I}(b))$ for all $a, b \in \Re^n$, where $d(\cdot,\cdot)$ denotes the Euclidean metric. A point set S is ϵ-congruent to a point set S' if there exists an isometry $\mathcal{I}$ and a bijective mapping $l : S \rightarrow S'$ such that for each point $s \in S$, $d(\mathcal{I}(s), l(s)) \leq \epsilon$. For point sets A, B, and real numbers $\epsilon > 0$ and $0 < \alpha \leq 1$, α-$LCP(A,B,\epsilon)$ is a subset S of A with $|S| \geq \alpha \min(|A|,|B|)$ such that S is ϵ-congruent to a subset of B. Clearly, for any ϵ, there exists a $\alpha_{max}(\epsilon)$ such that α-$LCP(A,B,\epsilon)$ exists for all $\alpha \leq \alpha_{max}(\epsilon)$ and for any $\alpha > \alpha_{max}(\epsilon)$, α-$LCP(A,B,\epsilon)$ does not exist. Hence, our substructure identification problem is the following :

Input : 3-D point sets A, B and a real number $\epsilon \geq 0$
Output : $\alpha_{max}(\epsilon)$-$LCP(A,B,\epsilon)$

Unless otherwise mentioned, from now onwards a point set refers to a point set in 3-D, and any isometric transformation is a composition of just a rotation and a translation, not including any mirror image. This restricted definition of an isometry does not result in any loss of generality, because isometry including mirror image just increases the computation time of any of our algorithm by only a constant factor.

2.1 Approximately satisfying the ϵ-constraint

In this subsection we state the algorithm due to Akutsu [Aku92], modified to the context of our problem. Given point sets A, B, and a real number $\epsilon \geq 0$,

instead of approximating $\alpha_{max}(\epsilon)$, the algorithm outputs a subset $S \subseteq A$ of size $\alpha \min(|A|, |B|)$ which is 8ϵ-congruent to some subset of B, and $\alpha \geq \alpha_{max}(\epsilon)$. Before describing the algorithm, we define a particular transformation on which this algorithm is based.

For two triplets of points $P = (p_{i_1}, p_{i_2}, p_{i_3})$ and $Q = (q_{j_1}, q_{j_2}, q_{j_3})$, let T_1 be the translation that takes the point p_1 to q_1. Let R_1 be the rotation about the point $T_1(p_1)$ such that $T_1(p_1), T_2(p_2)$ and q_2 become collinear. Finally, let R_2 be the rotation about the $[R_1(T_1(p_1))$-$R_1(T_1(p_2))]$ axis, that causes $R_1(T_1(p_1)), R_1(T_1(p_2)), R_1(T_1(p_3))$ and q_3 to become coplanar. Now let T_{PQ} be the isometric transformation which is the composition of T_1, R_1, and R_2, i.e. $T_{PQ}(p) = R_2(R_1(T_1(p)))$. Therefore $T_{PQ}(p_1)$ and q_1 are coincident, $T_{PQ}(p_1), T_{PQ}(p_2)$ and q_2 are collinear, and $T_{PQ}(p_1), T_{PQ}(p_2), T_{PQ}(p_3)$ and q_3 are coplanar. For point sets A, B, and a real real number α, let $\epsilon_{min}(\alpha)$ denote the smallest ϵ for which α-$LCP(A, B, \epsilon)$ exists. Then the following lemma follows directly from [GMO94].

Lemma 1. *Let l be the bijective mapping underlying α-$LCP(A, B, \epsilon_{min}(\alpha))$. Let $P = (p_{i_1}, p_{i_2}, p_{i_3})$ and $Q = (q_{j_1}, q_{j_2}, q_{j_3})$ be triplets belonging to α-$LCP(A, B, \epsilon_{min}(\alpha))$ and B respectively, such that p_2 is the farthest possible point from p_1 and the perpendicular distance from p_3 to the line passing through p_1 and p_2 is maximized, and $l(p_i) = q_i$, $i = 1, 2, 3$. Then the isometry T_{PQ} and the bijective mapping l correspond to α-$LCP(A, B, 8\epsilon_{min}(\alpha))$.*

Definition For point sets A, B, isometry $\mathcal{I} : A \rightarrow B$ and a real $\epsilon > 0$, let $G(\mathcal{I}, \epsilon, A, B)$ be a bipartite graph $(U \cup V, E)$ where U and V represent the points of A and B respectively and if $u \in U$ is the node corresponding to $a \in A$ and $v \in V$ corresponds to $b \in B$, then $E = \{(u, v) \mid d(\mathcal{I}(a), b) \leq \epsilon\}$.

Algorithm 2.1
Input : *Point sets A, B, real number $\epsilon > 0$*
$\alpha = 0$;
for all triplets $P = (p_{i_1}, p_{i_2}, p_{i_3})$ from A
    for all triplets $Q = (q_{j_1}, q_{j_2}, q_{j_3})$ from B
    {
      α' = size of maximum matching in $G(T_{PQ}, 8\epsilon, A, B)$;
      if $(\alpha' \geq \alpha)$ **then** $\alpha = \alpha'$;
    }
return α;

Theorem 2. *Given point sets A, B, and a real number $\epsilon \geq 0$, Algorithm 2.1 returns a real number α in $O(n^{8.5})$ time such that there exists a subset S of A of cardinality $\alpha \min(|A|, |B|)$, which is 8ϵ-congruent to some subset of B and $\alpha \geq \alpha_{max}(\epsilon)$.*

Proof deferred to the appendix.

2.2 Approximating $\alpha_{max}(\epsilon)$

Making use of the isometry stated in Lemma 1, we shall now state a *partial decision algorithm* to decide if α-$LCP(A, B, \epsilon)$ exists, if point sets A, B, and real numbers α and ϵ are input to the algorithm. This decision algorithm is called *partial* because it is guaranteed to make a decision only for values of (α, ϵ) for which ϵ is not too close to $\epsilon_{min}(\alpha)$. When ϵ is too close to $\epsilon_{min}(\alpha)$, the algorithm might return DON'T KNOW and such values of ϵ are said to constitute the *indecision interval*. However, whenever the algorithm returns YES or NO, the answer is correct. Algorithm 2.2 has an indecision interval equal to $[\frac{1}{8}\epsilon_{min}(\alpha), 8\epsilon_{min}(\alpha))$. Using this we then construct an algorithm for approximating $\alpha_{max}(\epsilon)$ which returns real numbers α_l and α_u, such that $\alpha_l \leq \alpha_{max}(\epsilon) < \alpha_u$. Finally we analyze the approximation ratio of the algorithm. The graph $G(T_{PQ}, \epsilon, A, B)$ has the same meaning as that defined in the last subsection.

Algorithm 2.2
Input : *Point sets A, B, real numbers $\epsilon > 0$, $0 < \alpha \leq 1$.*
for all triplets $P = (p_{i_1}, p_{i_2}, p_{i_3})$ from A
  for all triplets $Q = (q_{j_1}, q_{j_2}, q_{j_3})$ from B
    if $G(T_{PQ}, \epsilon, A, B)$ has a matching of size $\geq \alpha \min(|A|, |B|)$ **then return** YES;
$decision = NO$;
for all triplets $P = (p_{i_1}, p_{i_2}, p_{i_3})$ from A
  for all triplets $Q = (q_{j_1}, q_{j_2}, q_{j_3})$ from B
    if $G(T_{PQ}, 8\epsilon, A, B)$ has a matching of size $\geq \alpha \min(|A|, |B|)$ **then** $decision = YES$;
if $(decision == NO)$ **then return** NO **else return** $DON'T\ KNOW$;

Lemma 3. *Algorithm 2.2 always returns the correct answer about the existence of α-$LCP(A, B, \epsilon)$ if $\epsilon \geq 8\epsilon_{min}(\alpha)$ or, $\epsilon < \frac{1}{8}\epsilon_{min}(\alpha)$, and it either returns the correct answer or returns $DON'T\ KNOW$ if $\epsilon \in [\frac{1}{8}\epsilon_{min}(\alpha), 8\epsilon_{min}(\alpha))$.*

Proof deferred to the appendix.

Algorithm 2.3
Input : *Point sets A, B and a real number $\epsilon > 0$.*
$M = 0$;
for all triplets $P = (p_{i_1}, p_{i_2}, p_{i_3})$ from A
  for all triplets $Q = (q_{j_1}, q_{j_2}, q_{j_3})$ from B
  {
    M' = size of the maximum matching in $G(T_{PQ}, \epsilon, A, B)$;
    if $(M' > M)$ **then** { $M = M'$; $T = T_{PQ}$; }
  }
$\alpha_l = M/\min(|A|, |B|)$;
$M = 0$;
for all triplets $P = (p_{i_1}, p_{i_2}, p_{i_3})$ from A
  for all triplets $Q = (q_{j_1}, q_{j_2}, q_{j_3})$ from B

{
$M' =$ size of the maximum matching in $G(T_{PQ}, 8\epsilon, A, B)$;
if $(M' > M)$ **then** $M = M'$;
}
$\alpha_u = (M+1)/\min(|A|, |B|)$;
return (α_l, α_u);

Theorem 4. *Given point sets A, B and a real number $\epsilon > 0$, Algorithm 2.3 runs in time $O(n^{8.5})$ and returns real numbers $0 < \alpha_l \leq \alpha_u \leq 1$, such that $max\{\alpha : \epsilon > 8\epsilon_{min}(\alpha)\} \leq \alpha_l \leq \alpha_{max}(\epsilon)$ and $\alpha_{max}(\epsilon) < \alpha_u \leq min\{\alpha : \epsilon < \frac{1}{8}\epsilon_{min}(\alpha)\}$.*

Proof deferred to the appendix.

3 An Exact Algorithm for Finding the LCP under Rotation

Now we shall describe an algorithm which on given point sets A, B, a real number $\epsilon > 0$, and a fixed point p, finds $\alpha_{max}(\epsilon)$-$LCP(A, B, \epsilon)$ where the underlying isometry consists only of pure rotation about the point p. To understand this algorithm consider ϵ-balls around each point of the set B. As the set A is rotated about the point p, points of A move into and out of the ϵ-balls of B. The problem is then essentially that of finding the rotation for which the maximum number of points of A are within distinct balls of B.

Let S_p denote a sphere centered at p, of radius less than the distance of p from the nearest point of either A or B, and let p' be a point on the surface of S_p. Now for all possible pairs of points $a_i \in A$ and $b_j \in B$, consider the rotation of the set A about the point p such that a_i is within the ϵ-ball around b_j. If the same set of rotations that are to be applied to the point a_i for it to lie within the ϵ-ball around b_j, are applied to the point p', then it traces out a circular figure called a *dome* D_{ij}, on the surface of S_p (see Figure 1). This dome is indicative of the solid angle corresponding to which the point a_i is within the ϵ-ball around the point b_j.

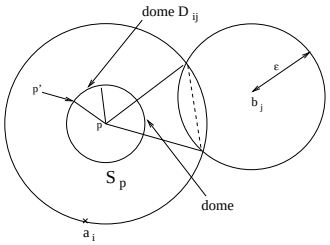

Fig. 1. Dome D_{ij} resulting from points a_i and b_j

If there is a rotation R of the set A about the point p such that $d(R(a_i), b_j) \leq \epsilon$, then obviously $D_{ij} \neq \emptyset$. Now consider every point on the surface of the sphere

S_p to be associated with a *membership vector*, which is indicative of all the domes to which the point belongs. Each dome partitions the surface of S_p into two regions, and all the domes arising out of the points of A and B define a partition of the surface of S_p into a number of distinct regions, where a *region* is defined as a set of points having the same membership vector. Therefore, for any point on the region $D_{i_1j_1} \cap D_{i_2j_2} \cap \ldots \cap D_{i_kj_k}$ there is a rotation R such that $d(R(a_{i_l}), b_{j_l}) \leq \epsilon,\ l = 1, 2, \ldots, k$. This gives rise to a bipartite graph in which the nodes correspond to the points of A and B, and the edges consist of all pairs (a_{i_l}, b_{j_l}), $l = 1, 2, \ldots, k$. The region in which this graph has the largest maximum matching is our required region, because a rotation corresponding to any point in this region finds the largest common point set between A and B. To find the largest maximum matching, it is required to traverse through all the regions and find the maximum matching in the bipartite graph arising in each region. Towards this end we use a space sweep [Meh84] : we sweep a plane $h(t) : x = t$ through the sphere S_p, starting from its leftmost end and ending in its rightmost end.

Briefly, the sweep algorithm is as follows. The *membership vector* of the sweep-plane indicating the domes which are intersected by the plane, changes only in two situations : (i) the sweep-plane just crossing the left end-point of the dome, after which this dome starts intersecting with the sweep-plane (ii) the sweep-plane just crossing the right end-point of the dome, after which this dome ceases to intersect with the sweep-plane. Our *event point schedule*, i.e. the sequence of abscissae ordered from left to right which define the halting positions of the sweep-plane, are made up of the x-coordinates of the left and right end points, and all the intersection points of all the domes lying on S_p.

When the event point is the left end-point of a dome, we update the membership vector of the sweep-plane to indicate that this dome now intersects with the sweep-plane. Next we construct the bipartite graph corresponding to this point, making use of the information in the sweep-plane membership vector, because this point can lie only on the subset of all the domes which intersect with the sweep-plane. If the event point is contained in the domes $D_{i_1j_1}, D_{i_2j_2}, \ldots, D_{i_kj_k}$, then the corresponding bipartite graph is constructed and the size of its maximum matching is found. If the event point is an intersection point of a number of domes, then we construct the bipartite graph corresponding to this point as was done in the previous case, and find the maximum matching in this graph. Finally, if the event point is the right end-point of a dome, then we just update the membership vector of the sweep-plane to indicate that from now this dome ceases to intersect with the sweep-plane. The largest maximum matching obtained over all the graphs is our required result.

If A and B are of cardinality m and n, then there are $O(mn)$ domes on the surface of S_p. Hence there are $O(mn)$ end-points and $O(m^2n^2)$ intersection points. Corresponding to each of the $O(m^2n^2)$ event points, constructing the bipartite graph takes $O(mn)$ time and finding the maximum matching using Hopcroft and Karp's algorithm [HK73] takes $O(mn\sqrt{m+n})$ time. Hence the overall time complexity of this algorithm is $O(m^3n^3\sqrt{m+n})$. In the subsequent

sections we refer to this algorithm as LCP-ROT(A, B, ϵ).

4 Algorithms with Improved Approximation Ratio

In this section shall we make use of algorithm LCP-ROT to improve the approximation ratio of the algorithms presented in Section 2. For this we first state a lemma which follows from Lemma 5 of [Sch92]. Here, for arbitrary points a and b in space, we use t_{ab} to denote the translation that maps a to b.

Lemma 5. *Let isometry $\mathcal{I}$, which is a composition of translation and rotation, and a bijective mapping l, correspond to α-$LCP(A, B, \epsilon)$. Let $a \in \alpha$-$LCP(A, B, \epsilon)$. There exists a rotation R about the point $\mathcal{I}(a)$, such that R and l correspond to α-$LCP(t_{a\mathcal{I}(a)}(A), B, \epsilon)$. Let b be an arbitrary point in space. There is a rotation R' about the point b such that R' and l correspond to α-$LCP(t_{ab}(A), B, \epsilon + d(b, \mathcal{I}(a)))$.*

Algorithm 4.1
Input : *Point sets A, B, real number $\epsilon > 0$.*
$\alpha = 0$;
for each point $a \in A$
  for each point $b \in B$
  {
    $\alpha' =$ LCP-ROT$(t_{ab}(A), B, b, 2\epsilon)$;
    if $(\alpha' \geq \alpha)$ **then** $\alpha = \alpha'$;
  }
return α;

Now consider Algorithm 4.1. It follows from Lemma 5 that it outputs a real number α such that there exists a subset S of A with cardinality $\alpha \min(|A|, |B|)$, which is 2ϵ-congruent to some subset of B. This is in contrast to Algorithm 2.1 which outputs a subset which is 8ϵ congruent to a subset of B. Thus we have the following theorem.

Theorem 6. *Given point sets A, B, and a real number $\epsilon > 0$, Algorithm 4.1 runs in time $O(n^{8.5})$ and returns a real number α such that there exists a subset S of A with cardinality α $min(|A|, |B|)$ which is 2ϵ-congruent to some subset of B and $\alpha \geq \alpha_{max}(\epsilon)$.*

Proof deferred to the appendix.

In exactly the same way, using algorithm LCP-ROT decreases the indecision interval of Algorithm 2.2 from $[\frac{1}{8}\epsilon_{min}(\alpha), 8\epsilon_{min}(\alpha))$ to $[\frac{1}{2}\epsilon_{min}(\alpha), 2\epsilon_{min}(\alpha))$, which leads to the following α_l and α_ui, in contrast to those obtained by Algorithm 2.3 : $\max\{\alpha : \epsilon > 2\epsilon_{min}(\alpha)\} \leq \alpha_l \leq \alpha_{max}(\epsilon)$ and $\alpha_{max}(\epsilon) < \alpha_u \leq \min\{\alpha : \epsilon < \frac{1}{2}\epsilon_{min}(\alpha)\}$. Hence it is a substantially better approximation of $\alpha_{max}(\epsilon)$. Details of the algorithm are given in the appendix.

The indecision interval of $[\frac{1}{2}\epsilon_{min}(\alpha), 2\epsilon_{min}(\alpha))$ can be further reduced to an arbitrarily small interval $[\epsilon_{min}(\alpha) - \gamma, \epsilon_{min}(\alpha) + \gamma)$, by using a technique described in [Sch92]. Doing this however introduces a term $(\epsilon/\gamma)^3$ in the running time of the algorithm. Here, we cover the ϵ-balls around each point of the set B with balls of radius γ. Let B' be the set of points which are the centers of these γ-balls. Now instead of testing all possible pairs of points of A and B, translations corresponding to all possible pairs of points of A and B' are tested. Since $(2\epsilon/\gamma)^3$ balls of radius γ are sufficient to cover each ϵ-ball, for each point of B there are $O((\epsilon/\gamma)^3)$ additional iterations. This improved decision algorithm clearly leads to a better approximation of $\alpha_{max}(\epsilon)$. The same technique can be applied to Algorithm 4.1 to reduce the factor of 2 to any $\delta > 1$, by approximately choosing γ. Here also an additional factor of $(\epsilon/\gamma)^3$ appears in the running time. This is however significantly better than the algorithm by Akutsu [Aku96] which obtains the same result but introduces a factor of $(\epsilon/\gamma)^9$ in the running time.

5 Algorithms with Improved Running Time

In this section we present three different modifications of the basic algorithms stated so far, which improve their running time. The first two are completely general, while the third relies on some structural properties of protein molecules.

5.1 Using an Approximation Algorithm for Maximum Matching

In all the algorithms presented so far we use the Hopcroft and Karp's algorithm [HK73] for finding the maximum matching in a bipartite graph, which runs in $O(n^{2.5})$ time. However, when the nodes of the bipartite graph are points in some d-dimensional space, and the edges are pairs of points which are within some specified distance of each other as in our case, an $O(n^{1.5}\log n)$ approximation scheme for finding the maximum matching was given by Efrat and Itai [EI96]. Now consider the graph $G(T_{PQ}, \epsilon, A, B)$ in Algorithm 2.2. The approximate graph matching algorithm finds the maximum matching in a graph G where $G(T_{PQ}, \epsilon, A, B) \subseteq G \subseteq G(T_{PQ}, (1+\delta)\epsilon, A, B)$. Here δ is a parameter of the algorithm due to Arya *et al.* [AMN+94] for answering nearest neighbor queries for a set of points in $\Re^d$, which is used by Efrat and Itai's algorithm. Replacing the Hopcroft and Karp's algorithm in Algorithm 2.2 with this new graph matching algorithm results in an improved running time of $O(n^{7.5}\log n)$, however, at the cost of an increased indecision interval which is summarized in the following theorem.

Theorem 7. *Algorithm 2.2 with the Hopcroft and Karp's algorithm replaced by the approximate graph matching algorithm due to Efrat and Itai [EI96] with parameter δ, runs in time $O(n^{7.5}\log n)$ and returns the correct answer about the existence of α-$LCP(A, B, \epsilon)$ if $\epsilon \geq 8\epsilon_{min}(\alpha)$, or, $\epsilon < \frac{\epsilon_{min}(\alpha)}{8(1+\delta)}$. It either returns the correct answer or returns DON'T KNOW for values of $\epsilon \in [\frac{\epsilon_{min}(\alpha)}{8(1+\delta)}, \frac{\epsilon_{min}(\alpha)}{1+\delta}) \cup$*

$[\epsilon_{min}(\alpha), 8\epsilon_{min}(\alpha))$, *and for* $\epsilon \in [\frac{\epsilon_{min}(\alpha)}{(1+\delta)}, \epsilon_{min}(\alpha))$ *it might return any of the three possible answers - YES, NO, DON'T KNOW. A transformation* T_{PQ}, *along with the bijective mapping* l *induced by the matching algorithm that results in the decision algorithm to return YES, correspond to* α-$LCP(A, B, (1+\delta)\epsilon)$.

Proof deferred to the appendix.

Using this new decision algorithm to approximate $\alpha_{max}(\epsilon)$ results in the following bounds : $\max\{\alpha : \epsilon > 8\epsilon_{min}(\alpha)\} \leq \alpha_l \leq \alpha_{max}((1+\delta)\epsilon)$ and $\alpha_{max}(\epsilon) < \alpha_u \leq \min\{\alpha : \epsilon < \frac{\epsilon_{min}(\alpha)}{8(1+\delta)}\}$. In Section 3 we had presented an exact algorithm for finding the LCP between two point sets when the underlying isometry is pure rotation. Replacing the Hopcroft and Karp's algorithm by the approximate matching algorithm will reduce its running time from $O(n^{6.5})$ to $O(n^{5.5}\log n)$, and thereby speedup the overall running time of all the algorithms of Section 4 which make use of it. The new bounds α_l and α_u, approximating $\alpha_{max}(\epsilon)$, however, are as follows : $\max\{\alpha : \epsilon > 2\epsilon_{min}(\alpha)\} \leq \alpha_l \leq \alpha_{max}((1+\delta)\epsilon)$ and $\alpha_{max}(\epsilon) < \alpha_u \leq \min\{\alpha : \epsilon < \frac{\epsilon_{min}(\alpha)}{2(1+\delta)}\}$. Exactly similar results are obtained for the other algorithms.

5.2 Improvements using Random Sampling

In this subsection we use standard random sampling techniques to reduce the time complexity of our algorithms by a considerable extent, at the cost of a small failure probability. By now this technique has become fairly standard for this class of problems [ATT97, FKL+97, IR96]. In our improved decision algorithm of Section 4 which has an indecision interval of $[\frac{1}{2}\epsilon_{min}(\alpha), 2\epsilon_{min}(\alpha))$ for every translation corresponding to pairs of points $a \in A$ and $b \in B$, our exact algorithm of Section 3 is invoked. Our randomized algorithms are based on the scheme of exploring all translations corresponding to randomly sampled subsets of the given point sets, instead of the original ones. The speedup obtained depends on the ratio of the size of the original sets to that of the sampled subsets. If A' is a randomly sampled subset of A and algorithm LCP-ROT is invoked for every possible pairs of points from A' and B, then we have the following theorem, assuming the use of Hopcroft and Karp's graph matching algorithm. Note that we have an improved running time of $O(n^{7.5})$ compared to the $O(n^{8.5})$ obtained without random sampling.

Theorem 8. *If point sets* A *and* B *are of cardinality* n, *and the cardinality of the randomly sampled multiset* A' *be a constant* k, *then the algorithm runs in time* $O(n^{7.5})$. *For any* $k \geq \lceil \frac{1}{\alpha} \ln \frac{1}{1-q} \rceil$, *the algorithm returns YES with probability at least* q, *for all* $\epsilon \geq 2\epsilon_{min}(\alpha)$. *For* $\epsilon < \frac{1}{2}\epsilon_{min}(\alpha)$ *the algorithm always returns NO, and for* $\frac{1}{2}\epsilon_{min}(\alpha) \leq \epsilon < \epsilon_{min}(\alpha)$ *it either returns NO or DON'T KNOW.*

Proof deferred to the appendix.

5.3 Practical Algorithms Exploiting the Structural Properties of Proteins

All the algorithms presented so far have time complexity which are relatively high degree polynomials of the size of the point sets. Recall that these 3-D point sets actually represent some drug or protein molecules, where each point is representative of an atom of the molecule. By exploiting some structural properties of proteins, it is possible to design algorithms with much smaller time complexities compared to those presented so far, however, at the cost of losing the guaranteed performance bounds. But we expect that for all practical purpose they would perform quite well.

A protein is composed of a chain of amino acids linked to each other by peptide bonds. There are three groups of atoms in each amino acid which constitute the backbone of the chain : the central atom C_α, to which are attached on each side an N-H group and a carbonyl group C'=O. An alkyl residue also bound to the C_α, characterizes the nature of the amino acid but does not take part in the backbone of the chain. The three atoms N, C_α and C' in each amino acid form a triangle which uniquely defines the position and orientation of the amino acid, and hence the entire protein structure, in space. Thus all the N, C_α and C' atoms act as a backbone or skeleton to which the alkyl residues R are attached. Since the C_α-N and C_α-C' bond lengths and the N-C_α-C' bond angle are fixed, the skeletons corresponding to two common substructures of two proteins will be exactly congruent. Since correspondence between two triplets of points is sufficient to uniquely determine a rigid transformation in space, if we know the correspondence between two amino acids belonging to the common substructures of two protein molecules then we can compute the rigid transformation that results in the two skeletons to exactly coincide.

Now consider Algorithm 2.1 where for all possible pairs of triplets form the point sets corresponding to the two molecules, the graph matching algorithm is invoked. When the two molecules in question are protein chains, we invoke the graph matching algorithm for all possible pairs of amino acids from the two proteins. Hence we hit upon a pair of amino acids a and b belonging to the two proteins, such that when the N, C_α and C' atoms of a are mapped to the corresponding atoms of b, the backbones from the two protein atoms corresponding to the common substructure exactly coincide. Clearly, any triplet of atoms belonging to the backbone satisfy the properties of the transformation of Section 2.1. Therefore, under this mapping a protein backbone corresponding to the common substructure is 8ϵ congruent with a substructure of the second protein. The only atoms which might violate this bound are the alkyl residues belonging to the substructure. Since $O(n^2)$ amino acids are tested, the overall running time is $O(n^{4.5})$.

Making use of all the improvements outlined above, the resulting algorithm for the common substructure identification for two protein molecules has a running time of $O(n^{2.5} \log n)$.

6 Concluding Remarks

In this paper we have presented several algorithms for identifying the structural similarities between two drug, or protein molecules. An abstraction of this problem is that of finding the largest common point set of two point sets in 3-D under ϵ-congruence, where the point location error ϵ is specified by the end-user. Towards this, we have presented two classes of algorithms : the first approximates the size of the largest common point set satisfying the point location error ϵ exactly, whereas the second only approximately satisfies ϵ. We have also outlined several techniques which improve the running time of our basic algorithms. We expect that for most problems, depending on the requirement, it would be possible to put together one or more of these techniques to obtain an algorithm which would work well in practice.

In this paper we have treated all molecules to be rigid bodies, and considered only rigid transformations to superimpose one molecule on the other. Although this treatment is adequate for comparing molecules with strong similarities, such a paradigm will fail to identify weak similarities between pairs of molecules. Methods to overcome this limitation need to be explored in future work.

References

[ACM96] *Proc. 12th. Annual ACM Symp. on Computational Geometry*, Philadelphia PA, USA, 1996.

[ACM97] *Proc. 13th. Annual ACM Symp. on Computational Geometry*, Centre Universitaire Méditerranéen, Nice, France, 1997.

[AH] T. Akutsu and M. M. Halldórsson. On approximation of largest common subtrees and largest common point sets. LNCS 834. 405–413.

[AKM+92] E. M. Arkin, K. Kedem, J. S. B. Mitchell, J. Sprinzak, and M. Werman. Matching points into pairwise disjoint noise regions: Combinatorial bounds and algorithms. *ORSA J. Computing*, 4(4):375–386, 1992.

[Aku92] Tatsuya Akutsu. Algorithms for determining the geometrical congruity in two and three dimensions. In *Proc. 3rd. International Symposium on Algorithms and Computation*, Nagoya, Japan, December 1992. LNCS 650, pp. 279–288.

[Aku96] Tatsuya Akutsu. Protein structure alignment using dynamic programming and iterative improvement. *IEICE Trans. Inf. & Syst.*, E78-D(0), 1996.

[AMN+94] Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Wu. An optimal algorithm for approximate nearest neighbor searching. In *Proc. 5th. Annual ACM-SIAM Symp. on Discrete Algorithms*, pages 573–582, 1994.

[AMWW88] H. Alt, K. Mehlhorn, H. Wagener, and E. Welzl. Congruence, similarity, and symmetries of geometric objects. *Discrete and Computational Geometry*, 3:237–256, 1988.

[Ata84] M. J. Atallah. Checking similarity of planar figures. *Internat. J. Comput. Inform. Sci.*, 13:279–290, 1984.

[Atk87] M. D. Atkinson. An optimal algorithm for geometrical congruence. *J. Algorithms*, 8:159–172, 1987.

[ATT97] T. Akutsu, H. Tamaki, and T. Tokuyama. Distribution of distances and triangles in a point set and algorithms for computing the largest common point sets. In *Proc. 13th. Annual ACM Symp. on Computational Geometry* [ACM97], pages 314–323.

[CGH+93] P. Chew, M. Goodrich, D. Huttenlocher, K. Kedem, J. Kleinberg, and D. Kravets. Geometric pattern matching under eucledian motion. In *Proc. 5th. Canadian Conference on Computational Geometry*, pages 151–156, Waterloo, Canada, 1993.

[CK92] L. P. Chew and K. Kedem. Improvements on geometric pattern matching. In *Proc. 3rd. Scand. Workshop on Algorithm Theory*, 1992. LNCS 621, pp. 318–325.

[dRL95] P. J. de Rezende and D. T. Lee. Point set pattern matching in d-dimensions. *Algorithmica*, 13:387–404, 1995.

[EI96] Alon Efrat and Alon Itai. Improvements on bottleneck matching and related problems using geometry. In *Proc. 12th. Annual ACM Symp. on Computational Geometry* [ACM96], pages 301–310.

[FBNW92] D. Fischer, O. Bachar, R. Nussiniv, and H. Wolfson. An efficient automated computer vision based technique for detection of three dimensional structural motifs in proteins. *Journal of Biomolecular Structures and Dynamics*, 9(4):769–789, 1992.

[FKL+97] P. W. Finn, L. E. Kavraki, J. C. Latombe, R. Motwani, C. Shelton, S. Venkatasubramanian, and A. Yao. RAPID: Randomized pharmacophore identification for drug design. In *Proc. 13th. Annual ACM Symp. on Computational Geometry* [ACM97], pages 324–333.

[FNW92] D. Fischer, R. Nussinov, and Hiam J. Wolfson. 3-d substructure matching in protein molecules. In *Proc. 3rd. Annual Symposium on Combinatorial Pattern Matching*, Tucson, Arizona, USA, April/May 1992. LNCS 644, pp. 136–150.

[GMO94] M. T. Goodrich, J. S. B. Mitchell, and M. W. Orletsky. Practical methods for approximate geometric patern matching under rigid motions. In *Proc. 10th. Annual ACM Symp. on Computational Geometry*, pages 103–112, 1994.

[HK73] J. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Computing*, 2:225–231, 1973.

[HK90] D. P. Huttenlocher and K. Kedem. Computing the minimum Hausdorff distance for point sets under translation. In *Proc. 6th. Annual ACM Symp. on Computational Geometry*, pages 340–349, 1990.

[HKK92] D. P. Huttenlocher, K. Kedem, and J. M. Kleinberg. On dynamic Voronoi diagrams and the minimum Hausdorff distance for point sets under Eucledian motion in the plane. In *Proc. 8th. Annual ACM Symp. on Computational Geometry*, pages 110–120, 1992.

[HKS91] D. P. Huttenlocher, K. Kedem, and M. Sharir. The upper envelope of Voronoi surfaces and its applications. In *Proc. 7th. Annual ACM Symp. on Computational Geometry*, pages 194–203, 1991.

[IR96] S. Irani and P. Raghavan. Combinatorial and experimental results for randomized point matching algorithms. In *Proc. 12th. Annual ACM Symp. on Computational Geometry* [ACM96], pages 68–77.

[ISI89] K. Imai, S. Sumino, and H. Imai. Minimax geometric fitting of two corresponding sets of points. In *Proc. 5th. Annual ACM Symp. on Computational Geometry*, pages 276–282, 1989.

[Iwa90] S. Iwanowski. *Approximate congruence and symmetry detection in the plane*. PhD thesis, Fachbereich Mathematik, Freie Universität, Berlin, 1990.

[Les91] A. M. Lesk. *Protein Architecture: A practical approach*. IRL Press, New York, 1991.

[MARW89] E. M. Mitchell, P. J. Artymiuk, D. W. Rice, and P. Willet. Use of techniques derived from graph theory to compare secondary structure motifs in proteins. *Journal of Molecular Biology*, 212:151–166, 1989.

[MBD+93] Y. Martin, M. Bures, E. Danaher, J. DeLazzer, and I. Lico. A fast new approach to pharmacophore mapping and its application to dopaminergic and benzodiazepine agonists. *J. of Computer Aided Molecular Design*, 7:83–102, 1993.

[Meh84] Kurt Mehlhorn. *Data Structures and Algorithms 3 : Multi-dimensional Searching and Computational Geometry*. Springer Verlag, Berlin, 1984.

[PA92] S. Pascarella and P. Argos. A data bank merging related protein structures and sequences. *Protein Engineering*, 5:121–137, 1992.

[PA94] Xavier Pennec and Nicholas Ayache. An $O(n^2)$ algorithm for 3d substructure matching for proteins. Technical Report N^o 2274, Unité de recherche INRIA Sophia Antipolis, 06902 Sophia Antipolis Cedex, France, Mai 1994.

[Rot91] G. Rote. Computing the minimum Hausdorff distance between two point sets on a line under translation. *Information Processing Letters*, 38:123–127, 1991.

[RR73] S. T. Rao and M. G. Rossmann. Comparison of super-secondary structures in proteins. *Journal of Molecular Biology*, 76:241–256, 1973.

[Sch92] Stefan Schirra. Approximate decision algorithms for approximate congruence. *Information Processing Letters*, 43:29–34, 1992.

[ŠO94] A. Šali and J. P. Overington. Derivation of rules for comparative protein modelling from a database of protein structure alignments. *Protein Science*, 3:1582–1596, 1994.

[TO89] W. R. Taylor and C. A. Orengo. Protein structure alignment. *Journal of Molecular Biology*, 208:1–22, 1989.

[VS91] G. Vriend and C. Sander. Detection of common three-dimensional structures in proteins. *PROTEINS: Structure, Function and Genetics*, 11:52–58, 1991.

[Wil95] P. Willet. Searching for pharmacophoric patterns in databases of three dimensional chemical structures. *J. of Molecular Recognition*, 8:290–303, 1995.

Appendix

A Proof of Theorem 2

The length of each edge of the graph $G(T_{PQ}, 8\epsilon, A, B)$, which has a matching of size $\alpha \min(|A|, |B|)$, is almost 8ϵ. Hence there exists a subset S of A of cardinality $\alpha \min(|A|, |B|)$ which is 8ϵ-congruent to some subset of B under the isometry T_{PQ}.

Let $P = (p_{i_1}, p_{i_2}, p_{i_3})$ be a triplet from $\alpha_{max}(\epsilon)$-$LCP(A, B, \epsilon)$ such that $p_{i_1} \in \alpha_{max}(\epsilon)$-$LCP(A, B, \epsilon)$, p_{i_2} is the point of $\alpha_{max}(\epsilon)$-$LCP(A, B, \epsilon)$ which

is farthest from p_{i_1}, and p_{i_3} is the point of $\alpha_{max}(\epsilon)$-$LCP(A, B, \epsilon)$ for which the perpendicular distance from p_{i_3} to the line passing through p_{i_1} and p_{i_2} is maximized. Let l be the bijective mapping underlying $\alpha_{max}(\epsilon)$-$LCP(A, B, \epsilon)$, and $Q = (q_{j_1}, q_{j_2}, q_{j_3})$ be the triplet from B such that $l(p_{i_l}) = q_{j_l}, l = 1, 2, 3$. It follows from Lemma 1 that the graph $G(T_{PQ}, 8\epsilon, A, B)$ has a matching of size at least $\alpha_{max}(\epsilon) \min(|A|, |B|)$. Since Algorithm 2.1 checks all possible triplets from A and B, triplets P and Q will be found.

There are $O(n^3)$ triplets from each of A and B. Constructing the bipartite graph $G(T_{PQ}, 8\epsilon, A, B)$ for pairs of triplets P and Q takes $O(n^2)$ time. Computing the maximum matching in $G(T_{PQ}, 8\epsilon, A, B)$ can be done using the Hopcroft and Karp's algorithm [HK73] in time $O(n^{2.5})$. Hence the overall running time is $O(n^{8.5})$.

B Proof of Lemma 3

If an isometry $\mathcal{I}$ and bijective mapping l correspond to α-$LCP(A, B, \epsilon)$, then $\mathcal{I}$ and l correspond to any α-$LCP(A, B, \epsilon')$ where $\epsilon' \geq \epsilon$. It follows from Lemma 1 that Algorithm 2.2 finds an isometry $\mathcal{I}$ and a mapping l which enables in finding an α-$LCP(A, B, 8\epsilon_{min}(\alpha))$. Since $\mathcal{I}$ and l also correspond to α-$LCP(A, B, \epsilon)$ for any $\epsilon \geq 8\epsilon_{min}(\alpha)$, the algorithm returns YES for all $\epsilon \geq 8\epsilon_{min}(\alpha)$.

For any ϵ, the algorithm returns YES iff $G(T_{PQ}, \epsilon, A, B)$ has a matching of size greater than or equal to $\alpha \min(|A|, |B|)$. Hence all YES answers are correct.

For any $\epsilon < \frac{1}{8}\epsilon_{min}(\alpha)$, no isometry enables in finding α-$LCP(A, B, 8\epsilon)$. Hence for any $\epsilon < \frac{1}{8}\epsilon_{min}(\alpha)$, the algorithm always returns NO.

Finally, since the algorithm finds an isometry and bijective mapping corresponding to α-$LCP(A, B, 8\epsilon_{min}(\alpha))$, it can return NO only if $\epsilon < \epsilon_{min}(\alpha)$. Hence all NO answers are also correct.

C Proof of Theorem 4

Clearly, Algorithm 2.3 returns YES for $\alpha = \alpha_l$ and NO for $\alpha = \alpha_u$ which implies $\alpha_l \leq \alpha_{max}(\epsilon)$ and $\alpha_u > \alpha_{max}(\epsilon)$. The remaining inequality follows from Lemma 3, along with the fact that if $\alpha_1 \leq \alpha_2$ then, $\epsilon_{min}(\alpha_1) \leq \epsilon_{min}(\alpha_2)$.

The running time is based on the same reasoning as that given for Theorem 2.

D Proof of Theorem 6

Let isometry $\mathcal{I}$ and bijection mapping l correspond to $\alpha_{max}(\alpha)$-$LCP(A, B, \epsilon)$. If $a \in \alpha_{max}(\epsilon)$-$LCP(A, B, \epsilon)$, then from Lemma 5 it follows that there exists a rotation R about the point $\mathcal{I}(a)$ such that $\mathcal{I} = R \circ t_{a\mathcal{I}(a)}$, and the isometry $\mathcal{I}' = R \circ t_{al(a)}$ correspond to α-$LCP(A, B, 2\epsilon)$, where $\alpha \geq \alpha_{max}(\epsilon)$. Since Algorithm 4.1 loops through all possible pairs of points of A and B, certainly Algorithm LCP-ROT gets called for a pair $a, l(a)$ where $a \in \alpha_{max}(\epsilon)$-$LCL(A, B, \epsilon)$. Hence the theorem holds.

Since LCP-ROT runs in $O(n^{6.5})$ time and Algorithm 4.1 calls it $O(n^2)$ times, its overall time complexity is $O(n^{8.5})$.

E An Algorithm with Indecision Interval of $[\frac{1}{2}\epsilon_{min}(\alpha), 2\epsilon_{min}(\alpha))$

Algorithm E.1
Input : *Point sets A, B, real numbers $\epsilon > 0$, $0 < \alpha \leq 1$.*
for each point $a \in A$
    for each point $b \in B$
    {
      $\alpha' = \mathsf{LCP\text{-}ROT}(t_{ab}(A), B, b, \epsilon)$;
      if $(\alpha' \geq \alpha)$ **then return** YES;
    }
$decision = NO$;
for each point $a \in A$
    for each point $b \in B$
    {
      $\alpha' = \mathsf{LCP\text{-}ROT}(t_{ab}(A), B, b, 2\epsilon)$;
      if $(\alpha' \geq \alpha)$ **then** $decision = YES$;
    }
if $(decision == NO)$ **then return** NO **else return** $DON'T\ KNOW$;

Theorem 9. *Algorithm E.1 always returns the correct answer about the existence of α-$LCP(A, B, \epsilon)$ if $\epsilon \geq 2\epsilon_{min}(\alpha)$, or, $\epsilon < \frac{1}{2}\epsilon_{min}(\alpha)$ and it either returns the correct answer or returns DON'T KNOW if $\frac{1}{2}\epsilon_{min}(\alpha) \leq \epsilon < 2\epsilon_{min}(\alpha)$.*

Proof: Let isometry $\mathcal{I}$ and a bijective mapping l, correspond to α-$LCP(A, B, \epsilon_{min}(\alpha))$. Let $a \in \alpha$-$LCP(A, B, \epsilon_{min}(\alpha))$. It follows from Lemma 5 that there exists a rotation R about the point $l(a)$, such that R and l correspond to α-$LCP(t_{al(a)}(A), B, \epsilon_{min}(\alpha) + d(l(a), \mathcal{I}(a)))$. Since $d(l(a), \mathcal{I}(a)) \leq \epsilon_{min}(\alpha)$, R and l correspond to α-$LCP(t_{al(a)}(A), B, 2\epsilon_{min}(\alpha))$. The rest of the proof is similar to that for Lemma 3.

F Proof of Theorem 7

It follows from Lemma 1 that Algorithm 2.2 finds a transformation T_{PQ} such that at least $\alpha \min(|A|, |B|)$ points of $T_{PQ}(A)$ are within $8\epsilon_{min}(\alpha)$ distance of distinct points of B. So the graph $G(T_{PQ}, \epsilon, A, B)$ always has a matching of size greater than or equal to $\alpha \min(|A|, |B|)$ if $\epsilon \geq 8\epsilon_{min}(\alpha)$. However, the approximate matching algorithm due to Efrat and Itai (subsequently referred to as AM), considers a graph $G \supseteq G(T_{PQ}, \epsilon, A, B)$ instead of $G(T_{PQ}, \epsilon, A, B)$. This is because some edges of G are larger than ϵ, but none are longer than $(1 + \delta)\epsilon$ if δ is the parameter of AM. AM finds a maximum matching in G

where $G(T_{PQ}, \epsilon, A, B) \subseteq G \subseteq G(T_{PQ}, (1+\delta)\epsilon, A, B)$. From [EI96] we know that any matching in $G(T_{PQ}, \epsilon, A, B)$ is also a matching in G, and if a matching in $G(T_{PQ}, \epsilon, A, B)$ can be increased by an augmenting path then for a matching of the same size there also exists an augmenting path in G. Thus if $G(T_{PQ}, \epsilon, A, B)$ has a matching of size k, then the approximate matching algorithm AM finds a matching of size $\geq k$. Hence our new decision algorithm using AM returns YES for any $\epsilon \geq 8\epsilon_{min}(\alpha)$. But since a maximum matching by AM might have some edges of length between ϵ and $(1+\delta)\epsilon$, the labelling induced by AM along with T_{PQ} correspond to α-$LCP(A, B, (1+\delta)\epsilon)$.

For any $\epsilon < \frac{1}{8}\epsilon_{min}(\alpha)$, no isometry enables in finding α-$LCP(A, B, 8\epsilon)$. But since AM considers the graph $G \subseteq G(T_{PQ}, 8(1+\delta)\epsilon, A, B)$, the indecision interval extends to $\frac{\epsilon_{min}(\alpha)}{8(1+\delta)}$ on the left. If $G(T_{PQ}, 8(1+\delta)\epsilon, A, B)$ does not contain a matching of size k, then G also does not have a matching of size k. For any $\epsilon < \frac{\epsilon_{min}(\alpha)}{8(1+\delta)}$ there does not exist any isometry T_{PQ} such that $G(T_{PQ}, 8(1+\delta)\epsilon, A, B)$ has a matching of size $\alpha \min(|A|, |B|)$. Hence for such ϵ our decision algorithm always returns NO.

For values of $\epsilon \in [\frac{\epsilon_{min}(\alpha)}{1+\delta}, \epsilon_{min}(\alpha))$, no isometry T_{PQ} enables in finding α-$LCP(A, B, \epsilon)$. But since AM considers a graph $G \subseteq G(T_{PQ}, (1+\delta)\epsilon, A, B)$, G might have a matching of size $\geq \alpha \min(|A|, |B|)$. Hence the decision algorithm might return YES. The arguments for it returning NO or DON'T KNOW are exactly similar to those given for Lemma 3.

For any $\epsilon \in [\frac{\epsilon_{min}(\alpha)}{8(1+\delta)}, \frac{\epsilon_{min}(\alpha)}{1+\delta})$, there does not exist any transformation T_{PQ} for which the graph $G(T_{PQ}, (1+\delta)\epsilon, A, B)$ has a matching of size $\geq \alpha \min(|A|, |B|)$. Since AM considers a graph $G \subseteq G(T_{PQ}, (1+\delta)\epsilon, A, B)$, it can never return a matching of this size. Hence for such ϵ the algorithm never returns YES. For $\epsilon \in [\epsilon_{min}(\alpha), 8\epsilon_{min}(\alpha))$, it follows from Lemma 1 that the algorithm finds a transformation T_{PQ}, for which $G(T_{PQ}, 8\epsilon, A, B)$ has a matching of size $\geq \alpha \min(|A|, |B|)$. For such a transformation AM also finds a matching of at least this size. Hence the algorithm never returns NO for such values of ϵ.

The time complexity follows from the fact that there are $O(n^3)$ triplets from each of the sets A and B, and the graph matching algorithm runs in time $O(n^{1.5} \log n)$.

G Proof of Theorem 8

The time complexity of $O(n^{7.5})$ follows from the fact that for $O(1)$ points of A' and $O(n)$ points of B, LCP-ROT having a running time of $O(n^{6.5})$ is invoked.

If $|A' \cap \alpha\text{-}LCP(A, B, \epsilon_{min}(\alpha))| \geq 1$, then it follows from the discussion of Section 4 that the algorithm returns YES for all $\epsilon \geq 2\epsilon_{min}(\alpha)$. We show that for $\Pr(\ |A' \cap \alpha\text{-}LCP(A, B, \epsilon_{min}(\alpha))| \geq 1\ ) \geq q$ to hold, it is sufficient that $k \geq \frac{1}{\alpha} \ln \frac{1}{1-q}$.

Let a be a randomly sampled element of A. Then $\Pr(\ a \in \alpha\text{-}LCP(A, B, \epsilon_{min}(\alpha))\ ) \geq \alpha$. Since A' is of cardinality k, $\Pr(\ A' \cap \alpha\text{-}LCP(A, B, \epsilon_{min}(\alpha) = \emptyset)\ ) < (1-\alpha)^k$. Hence for $\Pr(\ |A' \cap \alpha\text{-}LCP(A, B, \epsilon_{min}(\alpha))| \geq 1\ ) \geq q$ to hold, it is

sufficient that,

$$1 - (1 - \alpha)^k \geq q$$

Rearranging and taking logarithms of both sides,

$$k \ln (1 - \alpha) \leq \ln (1 - q)$$

Expanding the logarithmic term,

$$k(\alpha + \frac{\alpha^2}{2} + \frac{\alpha^3}{3} + \ldots) \geq \ln \frac{1}{1 - q}$$

For this inequality to hold, it is sufficient that

$$k \geq \frac{1}{\alpha} \ln \frac{1}{1 - q}$$

For any $\epsilon < \epsilon_{min}(\alpha)$, no isometry can result in finding α-$LCP(A, B, \epsilon)$. Therefore for such ϵ the algorithm never returns YES. Moreover, for $\epsilon < \frac{1}{2}\epsilon_{min}(\alpha)$ no isometry enables in finding α-$LCP(A, B, 2\epsilon)$. Thus such values of ϵ always result in the algorithm to return NO.