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] Indian Institute of Technology Kanpur

Abstract. Identifying the common 3-D substructure between two drug or protein molecules is an important problem in synthetic drug design and molecular biology. This problem can be represented as 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 in order 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 of using randomization, and/or an approximate maximum matching scheme for bipartite graphs.

1 Introduction

Finding the largest common substructure between two drug or protein molecules has important implications in synthetic drug design, and in studying biomolecular recognition and interaction of proteins. Of late, there has been considerable effort to develop computational tools to expedite this process ([7, 9, 15] and the references therein). Towards this, a molecule is modelled as a set of points in 3-D space, each point representing the centre of an atom. Given two such point sets, the problem is to find a rigid transformation of one point set relative to the

⋆ This work was carried out when the author was at IIT Kanpur.

⋆⋆ Author to whom all correspondence should be directed.
Currently visiting University of Nebraska-Lincoln. E-mail: sbiswas@cse.unl.edu

other, so that the number of points of the transformed set which are superimposed on the points of the other set is maximized. In computational geometry parlance this problem is called the *largest common point set* problem, or, LCP [4]. However, since it is unreasonable to expect an exact match between two atom positions, two points are considered to be superimposed if the distance between them is less than a predefined constant ϵ, called the *point location error* [9]. Hence the abstract version of the problem is that of finding the LCP of two 3-D point sets with exact congruence replaced by ϵ-*congruence*.

Closely related problems, involving both exact and also ϵ-congruence have been extensively studied in computational geometry, references to which can be found in [6, 13]. There is also a large body of literature on computational chemistry which address the substructure identification problem, an overview of which can be obtained from [9] and the references therein. However, none of them are systematic, but are based on some heuristic. Moreover, they do not provide any theoretical guarantee on the size of the substructure obtained, compared to the *largest* common substructure between the two input molecules. To address this, Akutsu [3] proposed an approximation algorithm for the protein structure alignment problem, with a guaranteed approximation ratio. Akutsu's algorithm, when given two 3-D point sets A and B corresponding to the protein structures and the constant ϵ, 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, by making use of an algorithm for point set matching due to Goodrich *et al.* [11]. 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.

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 non-trivial generalizations of the notion of partial decision algorithms for solving the ϵ-congruence decision problem of two equal cardinality point sets in 2-D, due to Schirra [17]. We next suggest various modifications of the basic algorithms, resulting in an improvement of their run time. The first involves the use of an approximate graph matching due to Efrat and Itai [8], and the second is through the use of random sampling. The time complexity of our algorithms can be further reduced by a considerable amount in the case of protein molecules [6]. However, due to space limitations the details of this are skipped in this paper and only the result is mentioned.

In the next section we formally state our abstract geometric problem and outline the algorithm due to Akutsu [3] which approximates the ϵ-constraint. We then show how this algorithm, in combination with the decision algorithm due to Schirra [17], 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. Due to space restrictions proofs are not presented in this paper; we refer the interested reader to [6].

2 Formal Definition and Initial Algorithms

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. Many substructure matching algorithms for proteins additionally make use of the sequence property in protein chains [3, 10]. However, there is now widespread agreement that similarities among distantly related proteins are often preserved at the level of their 3-D structures, even when very little similarity remains at the sequence level. So we make no assumptions about the linear ordering of the atoms (or more specifically amino acids) in the protein molecule.

First we state some definitions. A map $\mathcal{I} : \Re^n \rightarrow \Re^n$ is called an isometry if $d(a,b) = d\big(\mathcal{I}(a), \mathcal{I}(b)\big)$ 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 $s \in S$, $d\big(\mathcal{I}(s), l(s)\big) \leq \epsilon$. In other words, two equal cardinality point sets S and S' are ϵ-congruent if there exists an isometry $\mathcal{I}$ for which the *bottleneck matching measure* [8] between $\mathcal{I}(S)$ and S' is at most ϵ. 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 [3], 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 $P = (p_1, p_2, p_3)$ and $Q = (q_1, q_2, q_3)$, 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 $\Big[R_1\big(T_1(p_1)\big)\text{-}R_1\big(T_1(p_2)\big)\Big]$ axis, that causes $R_1\big(T_1(p_1)\big)$, $R_1\big(T_1(p_2)\big)$, $R_1\big(T_1(p_3)\big)$ 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. $T_{PQ}(p) = R_2\Big(R_1\big(T_1(p)\big)\Big)$. 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 [11].

Lemma 1. *Let l be the bijective mapping underlying α-$LCP\big(A, B, \epsilon_{min}(\alpha)\big)$. Let $P = (p_1, p_2, p_3)$ and $Q = (q_1, q_2, q_3)$ be triplets belonging to α-$LCP\big(A, B, \epsilon_{min}(\alpha)\big)$ 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-$LCP\big(A, B, 8\epsilon_{min}(\alpha)\big)$.*

Definition For point sets A, B, isometry $\mathcal{I} : A \to B$ 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 $E = \big\{(u, v) \,\big|\, d\big(\mathcal{I}(a), b\big) \leq \epsilon\big\}$.

```
Input: Point sets A, B, real number ε > 0
α := 0;
for all triplets of points P from A
   for all triplets of points Q from B {
      α' := size of maximum matching in G(T_PQ, 8ε, A, B);
      if (α' ≥ α) then α := α'; }
return α;
```

Fig. 1. Algorithm 1.

Theorem 1. *Given point sets A, B, and a real number $\epsilon \geq 0$, Algorithm 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 $\alpha \geq \alpha_{max}(\epsilon)$.*

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

Making use of the isometry T_{PQ} stated in Lemma 1, we shall now state a *partial decision algorithm* to decide if α-$LCP(A, B, \epsilon)$ exists, when 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 has an indecision interval equal to $\left[\frac{1}{8}\epsilon_{min}(\alpha), 8\epsilon_{min}(\alpha)\right)$. 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.

```
Input: Point sets A, B, and real numbers ε > 0, 0 < α ≤ 1
for all triplets of points P from A
   for all triplets of points Q from B
      if G(T_PQ, ε, A, B) has a matching of size ≥ α min(|A|, |B|)
      then return YES;
decision := NO;
for all triplets of points P from A
   for all triplets of points Q from B
      if G(T_PQ, 8ε, A, B) has a matching of size ≥ α min(|A|, |B|)
      then decision := YES;
if (decision = NO) then return NO else return DON'T KNOW;
```

Fig. 2. Algorithm 2.

Lemma 2. *Algorithm 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 $\epsilon \in \left[\frac{1}{8}\epsilon_{min}(\alpha), 8\epsilon_{min}(\alpha)\right)$.*

Theorem 2. *Given point sets A, B and a real number $\epsilon > 0$, Algorithm 3 runs in time $O(n^{8.5})$ and returns real numbers $0 < \alpha_l \leq \alpha_u \leq 1$, such that:*

$$max\{\alpha : \epsilon > 8\epsilon_{min}(\alpha)\} \leq \alpha_l \leq \alpha_{max}(\epsilon) < \alpha_u \leq min\{\alpha : \epsilon < \frac{1}{8}\epsilon_{min}(\alpha)\}$$

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

```
Input: Point sets A, B, and a real number ε > 0
M := 0;
for all triplets of points P from A
   for all triplets of points Q from B {
      M' := size of the maximum matching in G(T_PQ, ε, A, B);
      if (M' > M) then { M := M'; T := T_PQ; } }
α_l := M/min(|A|, |B|);
M := 0;
for all triplets of points P from A
   for all triplets of points Q from B {
      M' := size of the maximum matching in G(T_PQ, 8ε, A, B);
      if (M' > M) then M := M'; }
α_u := (M + 1)/min(|A|, |B|);
return (α_l, α_u);
```

Fig. 3. Algorithm 3.

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 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 for which a_i is within the ϵ-ball around b_j. Let D_{ij} be the circular figure traced out by the point p' on the surface of S_p by rotations which cause a_i to lie within the ϵ-ball around b_j. We refer to this circular figure as the *dome* D_{ij} (see Fig. 4) and is representative of the solid angle corresponding to which a_i is within the ϵ-ball around b_j.

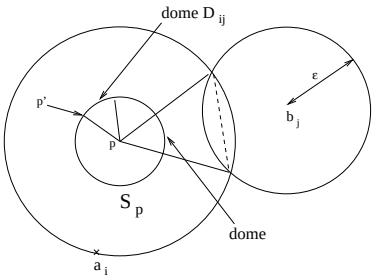

Fig. 4. Dome D_{ij} resulting from points a_i and b_j

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 $D_{i_1 j_1} \cap D_{i_2 j_2} \cap \ldots \cap D_{i_k j_k}$ 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 [16]: 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 $D_{i_1 j_1}, D_{i_2 j_2}, \ldots, D_{i_k j_k}$, 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 [12] 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, p, ϵ).

4 Algorithms with Improved Approximation Ratio

In this section shall we make use of the 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 [17]. Here, for arbitrary points a and b in space, we use t_{ab} to denote the translation that maps a to b.

Lemma 3. *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 of the point set $t_{a\mathcal{I}(a)}(A)$ about the point $\mathcal{I}(a)$, such that R and l correspond to α-$LCP\big(t_{a\mathcal{I}(a)}(A), B, \epsilon\big)$. Let b be an arbitrary point in space. There is a rotation R' of the point set $t_{ab}(A)$ about the point b, such that R' and l correspond to α-$LCP\Big(t_{ab}(A), B, \epsilon + d\big(b, \mathcal{I}(a)\big)\Big)$.*

```
Input : Point sets A, B, real number ε > 0
α := 0;
for each point a ∈ A
   for each point b ∈ B {
      α' := LCP-ROT(t_ab(A), B, b, 2ε);
      if (α' ≥ α) then α := α'; }
return α;
```

Fig. 5. Algorithm 4.

Now consider Algorithm 4. It follows from Lemma 3 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 1 which outputs a subset which is 8ϵ congruent to a subset of B. Thus we have the following theorem.

Theorem 3. *Given point sets A, B, and a real number $\epsilon > 0$, Algorithm 4 runs in time $O(n^{8.5})$ and returns 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 and $\alpha \geq \alpha_{max}(\epsilon)$.*

In exactly the same way, using algorithm LCP-ROT decreases the indecision interval of Algorithm 2 from $\big[\frac{1}{8}\epsilon_{min}(\alpha), 8\epsilon_{min}(\alpha)\big)$ to $\big[\frac{1}{2}\epsilon_{min}(\alpha), 2\epsilon_{min}(\alpha)\big)$,

which leads to the following bounds α_l and α_u, in contrast to those obtained by Algorithm 3:

$$\max\{\alpha : \epsilon > 2\epsilon_{min}(\alpha)\} \leq \alpha_l \leq \alpha_{max}(\epsilon) < \alpha_u \leq \min\{\alpha : \epsilon < \frac{1}{2}\epsilon_{min}(\alpha)\}$$

Hence it is a substantially better approximation of $\alpha_{max}(\epsilon)$. Details of the algorithm can be found in [6].

The indecision interval of $[\frac{1}{2}\epsilon_{min}(\alpha), 2\epsilon_{min}(\alpha))$ can be further reduced to any arbitrarily small interval $[\epsilon_{min}(\alpha) - \gamma, \epsilon_{min}(\alpha) + \gamma)$, by using a technique described in [17]. 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 to reduce the factor of 2 to any $\delta > 1$, by appropriately 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 [3] 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 two different modifications of the basic algorithms stated so far, which improve their running time, however, at the expense of the approximation ratio.

5.1 Using an Approximation Algorithm for Maximum Matching

In all the algorithms presented so far we use the Hopcroft and Karp's algorithm [12] 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 [8]. Now consider the graph $G(T_{PQ}, \epsilon, A, B)$ in Algorithm 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.* [5] 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 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 4. *Algorithm 2 with the Hopcroft and Karp's algorithm replaced by the approximate graph matching algorithm due to Efrat and Itai [8] 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 $\epsilon \in \left[\frac{\epsilon_{min}(\alpha)}{8(1+\delta)}, \frac{\epsilon_{min}(\alpha)}{1+\delta}\right) \cup \left[\epsilon_{min}(\alpha), 8\epsilon_{min}(\alpha)\right)$, and for $\epsilon \in \left[\frac{\epsilon_{min}(\alpha)}{(1+\delta)}, \epsilon_{min}(\alpha)\right)$ 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-$LCP\big(A,B,(1+\delta)\epsilon\big)$.*

Using this new decision algorithm to approximate $\alpha_{max}(\epsilon)$ results in the following bounds :

$$\max\big\{\alpha : \epsilon > 8\epsilon_{min}(\alpha)\big\} \;\leq\; \alpha_l \;\leq\; \alpha_{max}\big((1+\delta)\epsilon\big)$$

$$\alpha_{max}(\epsilon) \;<\; \alpha_u \;\leq\; \min\big\{\alpha : \epsilon < \frac{\epsilon_{min}(\alpha)}{8(1+\delta)}\big\}$$

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, with exactly similar results for the other algorithms.

$$\max\big\{\alpha : \epsilon \;>\; 2\epsilon_{min}(\alpha)\big\} \;\leq\; \alpha_l \;\leq\; \alpha_{max}\big((1+\delta)\epsilon\big)$$

$$\alpha_{max}(\epsilon) \;<\; \alpha_u \;\leq\; \min\big\{\alpha : \epsilon < \frac{\epsilon_{min}(\alpha)}{2(1+\delta)}\big\}$$

5.2 Improvements using Random Sampling

In this subsection we use standard random sampling techniques to reduce the time complexity of our algorithms, at the cost of a small failure probability. By now this technique has become fairly standard for this class of problems [4, 9, 14]. In our improved decision algorithm of Section 4 which has an indecision interval of $\left[\frac{1}{2}\epsilon_{min}(\alpha), 2\epsilon_{min}(\alpha)\right)$ 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 had an improved running time of $O(n^{7.5}\log n)$ compared to the $O(n^{8.5})$ obtained without random sampling, by using the approximate graph matching algorithm described in the last subsection.

Theorem 5. *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 $k \geq \left\lceil \frac{1}{\alpha} \ln \frac{1}{1-q} \right\rceil$, 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.*

Now combining the above two results, we can have a decision algorithm similar to the one stated in Theorem 4, but running in $O(n^{6.5} \log n)$ time. However, for any $\epsilon \geq 8\epsilon_{min}(\alpha)$, such an algorithm returns YES with a probability at least q, in contrast to definitely returning a YES.

All the algorithms presented so far still have time complexity which are relatively high degree polynomials of the size of the point sets. But when the point sets in question arise from protein molecules, a further improvement in running time can be achieved [6]. Using a simple geometric property of the α-carbon backbone structure of proteins along with the improvements suggested above, results in an $O(n^{2.5} \log n)$ algorithm for the common substructure identification between two protein molecules.

6 Concluding Remarks

Identifying the structural similarities between two drug or protein molecules has important applications in biology and chemistry. Towards this we have proposed a number of approximation algorithms for finding the LCP of two 3-D point sets under ϵ-congruence. These algorithms can be classified into two groups: in the first we approximate the size of the largest common point set while satisfying the constraint imposed by ϵ, whereas the second group of algorithms only approximately satisfy ϵ. We have also outlined two techniques which improve the running time of our basic algorithms, however, at the cost of the approximation ratio.

In this paper we have modelled molecules as rigid bodies, and considered only isometric transformations to superimpose the underlying point sets on one another. Although this treatment is adequate for comparing molecules with strong similarities, such a paradigm will fail to identify weak similarities between pairs of molecules. To overcome this limitation, more general transformations need to be considered in future work.

Acknowledgements

The first author is grateful to Suresh Venkatasubramanian for his helpful comments on the earlier draft of this paper.

References

1. *Proc. 12th. Annual ACM Symp. on Computational Geometry*, 1996.

2. *Proc. 3rd. Annual Intl. Conf. on Computational Molecular Biology*, April, 1999.
3. T. Akutsu. Protein structure alignment using dynamic programming and iterative improvement. *IEICE Trans. Information and Systems*, E79-D:1629–1636, 1996.
4. 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. *Discrete and Computational Geometry*, 20:307–331, 1998.
5. S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching. In *Proc. 5th. Annual ACM-SIAM Symp. on Discrete Algorithms*, pages 573–582, 1994.
6. S. Chakraborty and S. Biswas. Approximation algorithms for 3-D common substructure identification in drug and protein molecules. Technical Report TIK Report No. 69, Eidgenössische Technische Hochschule Zürich, 1999. `ftp://ftp.tik.ee.ethz.ch/pub/people/samarjit/paper/CB99a.ps.gz`.
7. L. P. Chew, K. Kedem, J. Kleinberg, and D. Huttenlocher. Fast detection of common geometric substructure in proteins. In *Proc. RECOMB'99 - 3rd. Annual International Conference on Computational Molecular Biology* [2].
8. A. Efrat and A. Itai. Improvements on bottleneck matching and related problems using geometry. In *Proc. 12th. Annual ACM Symp. on Computational Geometry* [1], pages 301–310.
9. 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*, pages 324–333, Centre Universitaire Méditerranéen, Nice, France, 1997.
10. D. Fischer, R. Nussinov, and H. J. Wolfson. 3-D substructure matching in protein molecules. In *Proc. 3rd. Annual Symposium on Combinatorial Pattern Matching*, April 1992. LNCS 644, pages 136–150.
11. 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.
12. J. Hopcroft and R. M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Computing*, 2:225–231, 1973.
13. P. Indyk, R. Motwani, and S. Venkatasubramanian. Geometric matching under noise: Combinatorial bounds and algorithms. In *Proc. 10th. Annual ACM-SIAM Symp. on Discrete Algorithms*, 1999.
14. S. Irani and P. Raghavan. Combinatorial and experimental results for randomized point matching algorithms. In *Proc. 12th. Annual ACM Symp. on Computational Geometry* [1], pages 68–77.
15. S. Lavalle, P. Finn, L. Kavraki, and J-C. Latombe. Efficient database screening for rational drug design using pharmacophore-constrained conformational search. In *Proc. RECOMB'99 - 3rd. Annual International Conference on Computational Molecular Biology* [2].
16. K. Mehlhorn. *Data Structures and Algorithms 3: Multi-dimensional Searching and Computational Geometry*. Springer Verlag, Berlin, 1984.
17. S. Schirra. Approximate decision algorithms for approximate congruence. *Information Processing Letters*, 43:29–34, 1992.