BIOINFORMATICS ORIGINAL PAPER Vol. 25 no. 21 2009, pages 2751–2756
doi:10.1093/bioinformatics/btp530

*Structural bioinformatics*

# Algorithms for optimal protein structure alignment

Aleksandar Poleksic

Department of Computer Science, University of Northern Iowa, Cedar Falls, IA 50614, USA

Received on May 19, 2009; revised on August 14, 2009; accepted on September 1, 2009

Advance Access publication September 4, 2009

Associate Editor: Anna Tramontano

**ABSTRACT**

**Motivation:** Structural alignment is an important tool for understanding the evolutionary relationships between proteins. However, finding the best pairwise structural alignment is difficult, due to the infinite number of possible superpositions of two structures. Unlike the sequence alignment problem, which has a polynomial time solution, the structural alignment problem has not been even classified as solvable.

**Results:** We study one of the most widely used measures of protein structural similarity, defined as the number of pairs of residues in two proteins that can be superimposed under a predefined distance cutoff. We prove that, for any two proteins, this measure can be optimized for all but finitely many distance cutoffs. Our method leads to a series of algorithms for optimizing other structure similarity measures, including the measures commonly used in protein structure prediction experiments. We also present a polynomial time algorithm for finding a near-optimal superposition of two proteins. Aside from having a relatively low cost, the algorithm for near-optimal solution returns a superposition of provable quality. In other words, the difference between the score of the returned superposition and the score of an optimal superposition can be explicitly computed and used to determine whether the returned superposition is, in fact, the best superposition.

**Contact:** poleksic@cs.uni.edu

**Supplementary information:** Supplementary data are available at *Bioinformatics* online.

## 1 INTRODUCTION

Protein structural alignment is a valuable tool for protein fold and function classification. The success of the structural genomics initiative, which aims to experimentally determine 3D structures of thousands of representative proteins, critically depends on our ability to develop accurate tools for comparison of protein structures (Goldsmith-Fischman and Honig, 2003). However, despite its utmost importance, the problem still lacks a fast and accurate solution. While some structural similarity scoring functions can be approximated in polynomial time (Kolodny and Linial, 2003; Xu *et al.,* 2007), there has been no procedure (of any running time) to optimize any commonly used structural alignment measure (Caprara *et al.*, 2004; Goldman *et al.*, 1999; Zhang and Skolnick, 2005). In their review article on progress in the field of structure comparison, Taylor and coworkers write: 'In structure comparison, we do not even have an algorithm that guarantees an optimal answer for pairs of structures' (Eidhammer *et al.*, 2000).

There are several different, but related definitions of an optimal alignment of two proteins. Some methods define an optimal superposition as a superposition that minimizes the distances between the aligned atoms (Oldfield, 2007; Ortiz *et al.*, 2002; Russell and Barton, 1992; Shindyalov and Bourne, 1998; Subbiah *et al.*, 1993; Ye and Godzik, 2003). Other methods attempt to minimize the difference between the intra-atomic distances (Alexandrov *et al.*, 1992; Holm and Sander, 1993; Orengo and Taylor, 1996; Taylor and Orengo, 1989; Vriend and Sander, 1991).

Several methods for improved matching of protein structures have recently been introduced, including the methods based on the *phenotypic plasticity* (Csaba *et al.*, 2008) and the method for flexible alignments by a sequence of local transformations (Rocha *et al.*, 2009).

Perhaps the most intuitive and most widely used measure of similarity of two proteins is the largest number of atoms (such as alpha-carbons, *CA*) in two structures that can be superimposed under a specified distance of each other. From now on, we will denote this metric by "$CA \leq \sigma$", where $\sigma > 0$ denotes the distance threshold in ångströms.

Many structural alignment measures build upon $CA \leq \sigma$, including GDT (Zemla, 2003), AL0 (Sali and Blundell, 1993), *MaxSub* (Siew *et al.*, 2000), *CA*-atoms $<3$ Å (Ginalski *et al.*, 2005), Q-*score* (Ginalski *et al.*, 2005) and TM-*Score* (Zhang and Skolnick, 2005). The Global Distance Test (GDT) is routinely used to evaluate the quality of models in the CASP experiment (Moult *et al.*, 2007). The accuracy of a predicted model in CASP is measured by the GDT_TS score, which represents the average of GDT scores computed at several distance thresholds. More specifically,

$$\text{GDT\_TS} = \frac{(\text{GDT\_P1} + \text{GDT\_P2} + \text{GDT\_P4} + \text{GDT\_P8})}{4},$$

where GDT_P*n* denotes percent of *CA* atoms that can be structurally superimposed under *n* ångströms.

One of the main measures of model quality in *LiveBench* is '*CA*-atoms $<3$ Å' (Ginalski *et al.*, 2005) (in our notation $CA < 3$). Due to the difficulty in optimizing the scoring function itself, *LiveBench* approximates $CA < 3$ using *3deval*, a program that attempts to maximize another metric, namely 3D-*score*.

CAFASP benchmark of fully automated structure prediction servers (Fischer *et al.*, 2003) uses *MaxSub* to assess the quality of servers' predictions. *MaxSub* is defined as the weighted fraction of the residues in the model falling within 3.5 Å of the aligned residues in the experimental structure (Siew *et al.*, 2000).

Irrespective of the scoring system used, the major difficulty that any structural alignment method must overcome is the

 

infinite (uncountable) space of all possible structural superpositions. To circumvent this problem, current research in the field focuses on reducing the size of search space by enumerating a relatively small, but representative set of superpositions. However, while the solutions obtained by heuristic approaches are often accurate, they are never guaranteed to be even close to the optimal solutions.

An obvious way to address the limitations of heuristic techniques is to develop a fast and extremely accurate method for maximizing $CA \leq \sigma$. Such a method would allow for accurate computation of a range of structural alignment measures, including GDT_TS, AL0 and *MaxSub*.

In this article we present:

(1) A polynomial time algorithm that returns a superposition of any specified accuracy, and

(2) A procedure that returns an optimal superposition, for all but finitely many distance cutoffs.

The first algorithm is capable of finding a superposition that is arbitrarily close to an optimal superposition. More specifically, for any given distance cutoff $\sigma > 0$ and any $\varepsilon > 0$, the algorithm returns a superposition that fits at least as many residue pairs under the distance $\sigma + \varepsilon$ as the optimal superposition fits under the distance $\sigma$. In addition to having relatively low time complexity and being amenable to parallel implementations, our algorithm provides the 'solution quality', which is defined as the difference between the score of the returned superposition and the score of an optimal superposition. The solution quality metric can be used to determine whether the returned superposition is, in fact, an optimal superposition (or whether another and more detailed search is needed).

Our work on the algorithm for approximate solution was inspired by the excellent study of Kolodny and Linial on the polynomial-time approximation scheme for a class of continuous structural similarity measures (Kolodny and Linial, 2003). However, we emphasize that the results presented here cannot be derived from their study, since the objective function $CA \leq \sigma$ does not belong to the category of scoring functions described by Kolodny and Linial. In fact, the class of scoring functions amenable to techniques of Kolodny and Linial (namely the class of functions satisfying the Lipschitz condition) does not include GDT_TS, AL0, *MaxSub*, TM-*score*, Q-*score*, and some other protein structure similarity metrics.

Finally, we address an open problem of finding a procedure that returns an optimal superposition of two structures. We present a procedure that is guaranteed to return an optimal solution with probability one, i.e. for all but finitely many distance cutoffs. However, we emphasize that the algorithm for optimal solution is not practical. This is expected, since the problem has already been proven to be NP-*hard* (Lathrop, 1994).

## 2 METHODS AND RESULTS

### 2.1 Preliminaries and definitions

The pairwise protein structure alignment problem can be formulated as follows: 'Given two proteins $a$ and $b$ and a distance cutoff $\sigma > 0$, find a rigid transformation $t$ and a residue-residue correspondence (alignment) that maximizes the number of pairs of residues in $a$ and $t(b)$ at distance $\leq \sigma$'. We call $t$ a *$\sigma$-optimal transformation for a and b*. Note that, without loss of generality, we can assume that the protein $a$ is held fixed, while protein $b$ is transformed.

In order to precisely formulate the above problem, we need some definitions.

DEFINITION: A *protein* is a sequence of points in 3D space

$$a=(a_1,a_2,\ldots,a_n), a_i \in \mathbf{R}^3 \text{ for } i=1,\ldots,n$$

In most applications, $a_i$ represent the residues' *CA*.

DEFINITION: An *alignment* of proteins $a=(a_1,a_2,\ldots,a_n)$ and $b=(b_1,b_2,\ldots,b_m)$ is a sequence of pairs of points from $a$ and $b$:

$$S(a,b)=((a_{i_1},b_{j_1}),\ldots,(a_{i_k},b_{j_k}))$$

where $1 \leq i_1 < \ldots < i_k \leq n$ and $1 \leq j_1 < \ldots < j_k \leq m$.

DEFINITION: A *$\sigma$-optimal alignment* of proteins $a=(a_1,a_2,\ldots,a_n)$ and $b=(b_1,b_2,\ldots,b_m)$, denoted by $S(a,b;\sigma)$, is an alignment of $a$ and $b$ that maximizes the number of aligned points in $a$ and $b$ at distance $\leq \sigma$.

In the definition above, we assume that proteins $a$ and $b$ are fixed in space. $S(a,b;\sigma)$ refers to an optimal pairing of amino-acid letters, without changing the structural superposition. It is well known that, for any two, fixed in space, proteins $a$ and $b$ of length $n$ and any $\sigma > 0$, the alignment $S(a,b;\sigma)$ can be computed in $O(n^2)$ time using a standard dynamic programming algorithm (Smith and Waterman, 1981). From now on, we will use $|S(a,b;\sigma)|$ to denote the number of pairs of points in $S(a,b;\sigma)$ at distance $\leq \sigma$.

DEFINITION: A *$\sigma$-optimal transformation for a and b*, denoted by $t^\sigma$, is a rigid transformation that maximizes $|S(a,t(b);\sigma)|$ over all rigid transformations $t$. In other words

$$t^\sigma = \arg\max_t |S(a,t(b);\sigma)|.$$

It is easy to see that $CA \leq \sigma$ is precisely $|S(a,t^\sigma(b);\sigma)|$, where $t^\sigma$ is a $\sigma$-optimal transformation for $a$ and $b$.

DEFINITION: A transformation $t$ satisfying

$$|S(a,t(b);\sigma+\varepsilon)| \geq |S(a,t^\sigma(b);\sigma)|$$

is called a *$(\sigma,\varepsilon)$-optimal transformation for a and b*, denoted by $t^\sigma_\varepsilon$.

Note that $t^\sigma_\varepsilon$ is any transformation $t$ with the property that the number of pairs of points in $a$ and $t(b)$ that can be superimposed at distance $\leq \sigma+\varepsilon$ is not smaller than the number of pairs of points in $a$ and $t^\sigma(b)$ that can be superimposed at distance $\leq \sigma$.

We recall that any orientation preserving rigid transformation $t$ is a composition of a rotation and a translation $t = t_{tr} \circ t_{rot}$. Any such transformation can also be viewed as a point in 6D space

$$t=(\alpha,\beta,\gamma,u,v,w) \in \boldsymbol{R}^6$$

where $\alpha$ and $\gamma$ are the Euler angles of rotations around the $z$-axis, $\beta$ is the angle of rotation around the $x-$axis, and $u$, $v$ and $w$ are the translations along the axes $x$, $y$ and $z$, respectively.

### 2.2 Near-optimal solution

Without loss of generality, we can assume that both proteins $a$ and $b$ have center of mass at the origin. Thus, a superposition that aligns the center of mass of $a$ and the center of mass of $b$ will be the first (among many) superpositions inspected by our method. Our algorithm for finding $t^\sigma_\varepsilon$, called EPSILON-OPTIMAL, is based on the continuity of rigid transformations (Kolodny and Linial, 2003). In other words, small change in any one of the six arguments of $t=(\alpha,\beta,\gamma,u,v,w)$ results in a small change in the spatial position of the protein $b$. For example, if $b$ is rotated around the $x$ axis by a small angle $\delta > 0$, then the distance travelled by any point $p(x,y,z) \in b$ is

$$d=\sqrt{2(y^2+z^2)(1-\cos\delta)} \leqslant \sqrt{2}R_b \sin\delta \leqslant \sqrt{2}R_b\delta,$$

where $R_b$ denotes the radius of the bounding sphere for $b$.

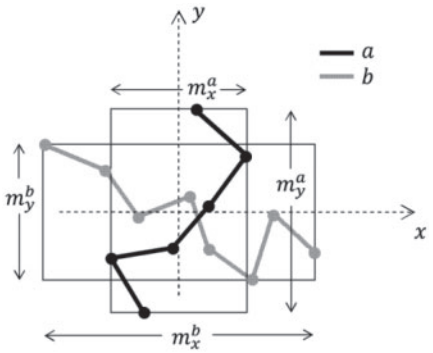


**Fig. 1.** The projection of the enclosing intervals of proteins $a$ and $b$, $B(a)$ and $B(b)$, respectively, onto the $xy$ plane.

*2.2.1 Algorithm for near optimal solution.* It is not difficult to see that the only candidates for $t_\varepsilon^\sigma$ are those transformations that do not move the protein $b$ far away from $a$, i.e. the transformations $t=(\alpha,\beta,\gamma,u,v,w)\in \boldsymbol{R}^6$ from the closed interval

$$I=[0,2\pi]\times[0,\pi]\times[0,2\pi]\times[-M_x,M_x]\times[-M_y,M_y]\times[-M_z,M_z],$$

where

$$M_x=\frac{m_x^a+m_x^b}{2},\ M_y=\frac{m_y^a+m_y^b}{2},\ M_z=\frac{m_z^a+m_z^b}{2}$$

and $m_x^a, m_y^a, m_z^a$ and $m_x^b, m_y^b, m_z^b$ are the dimensions of the smallest intervals $B(a)$ and $B(b)$ in $\boldsymbol{R}^3$ enclosing $a$ and $b$, respectively (Figure 1).

To further reduce the size of the search space, we define a finite set $NET(\varepsilon)$ of transformations from the interval $I$, obtained by partitioning $I$ into small 6D boxes. The points of $NET(\varepsilon)$ are the vertices of 6D subintervals of $I$ of dimensions

$$d_1=d_2=d_3=\varepsilon/(3\sqrt{2}R_b), \tag{1}$$

$$d_4=d_5=d_6=\varepsilon/\sqrt{3}. \tag{2}$$

Note that, in order to find $t_\varepsilon^\sigma$, it is enough to inspect the transformations from $NET(\varepsilon)$. This is because for every $t=(\alpha,\beta,\gamma,u,v,w)\in I$, there exists a transformation $\bar{t}=(\bar{\alpha},\bar{\beta},\bar{\gamma},\bar{u},\bar{v},\bar{w})\in NET(\varepsilon)$ such that

$$|\alpha-\bar{\alpha}|\le d_1/2, |\beta-\bar{\beta}|\le d_2/2, |\gamma-\bar{\gamma}|\le d_3/2, \tag{3}$$

$$|u-\bar{u}|\le d_4/2, |v-\bar{v}|\le d_5/2, |w-\bar{w}|\le d_6/2 \tag{4}$$

and hence

$$||t(b_i)-\bar{t}(b_i)||\le\varepsilon, \text{ for all } b_i\in b \tag{5}$$

where $||t(b_i)-\bar{t}(b_i)||$ denotes the Euclidean distance between $t(b_i)$ and $\bar{t}(b_i)$. In particular, if $t$ is a $\sigma$-optimal transformation $t^\sigma$ then $\bar{t}$ is a $(\sigma,\varepsilon)$-optimal transformation $t_\varepsilon^\sigma$.

A pseudo-code for EPSILON-OPTIMAL is given in the Supplementary Material.

*2.2.2 Time complexity.* For a pair of proteins of lengths $n$, the worst-case behavior of EPSILON-OPTIMAL occurs when the radius of the bounding sphere of $b$ is linear in $n$, i.e. $R_b=O(n)$. In this case, the total number of transformations inspected by EPSILON-OPTIMAL is $NET(\varepsilon)=O(n^6/\varepsilon^6)$. For every such transformation, an optimal correspondence can be computed using a $O(n^2)$ dynamic programming procedure, resulting in $O(n^8/\varepsilon^6)$ worst-case running time of EPSILON-OPTIMAL.

However, the total cost of EPSILON-OPTIMAL is usually better in practice, since the volume of a protein scales proportionally with the number of residues (Hao *et al.*, 1992). For instance, if $b$ is a globular protein, then $R_b=O(n^{1/3})$ and thus the running time of EPSILON-OPTIMAL is only $O(n^4/\varepsilon^6)$.

For comparison, the efficiency of the algorithm of Kolodny and Linial for optimizing the class of scoring functions satisfying Lipschitz condition is $O(n^{10}/\varepsilon^6)$ for globular and $O(n^{12}/\varepsilon^6)$ for non-globular proteins.

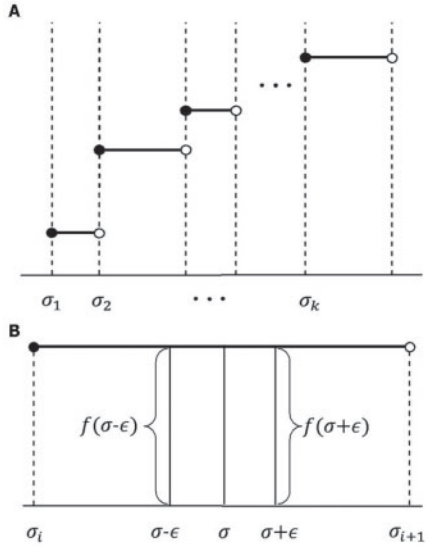


**Fig. 2.** (**A**) The graph of $f(\sigma)=|S(a,t^\sigma(b);\sigma)|$. Note that $f:R^+\to N$ is a step function, with $k$ steps $\sigma_1,\ldots,\sigma_k$, where $k$ does not exceed the length of the shorter of two proteins, $a$ and $b$. (**B**) For every $\sigma\in\{\sigma_1,\ldots,\sigma_k\}$ there exists $\varepsilon>0$, such that $f(\sigma-\varepsilon)=f(\sigma+\varepsilon)$.

*2.2.3 Solution quality.* The 'quality' of the solution $t_\varepsilon^\sigma$ provided by EPSILON-OPTIMAL is defined as the difference between the score of an optimal solution $t^\sigma$ and the score of $t_\varepsilon^\sigma$:

$$Err(t_\varepsilon^\sigma)=|S(a,t^\sigma(b);\sigma)|-|S(a,t_\varepsilon^\sigma(b);\sigma)| \tag{6}$$

While $Err(t_\varepsilon^\sigma)$ cannot be computed within the time window of EPSILON-OPTIMAL (since we do not know $t^\sigma$), the upper bound of $Err(t_\varepsilon^\sigma)$:

$$MaxErr(t_\varepsilon^\sigma)=|S(a,t_\varepsilon^\sigma(b);\sigma+\varepsilon)|-|S(a,t_\varepsilon^\sigma(b);\sigma)| \tag{7}$$

can easily be calculated with a small modification of EPSILON-OPTIMAL, without increasing its asymptotic cost (an extra call to $O(n^2)$ alignment procedure).

## 2.3 Optimal solution

Our procedure for finding optimal solution is based on the observation that $f(\sigma)=|S(a,t^\sigma(b);\sigma)|$ is a step function of $\sigma$, with finitely many steps $\sigma_1,\ldots,\sigma_k$ (Figure 2A). If $\sigma>0$ is any real number different from each $\sigma_i$, then, for some sufficiently small $\varepsilon>0, f(\sigma+\varepsilon)=f(\sigma-\varepsilon)$ (Figure 2B). Since

$$\begin{aligned} f(\sigma+\varepsilon)&=|S(a,t^{\sigma+\varepsilon}(b);\sigma+\varepsilon)|\ge|S(a,t_\varepsilon^\sigma(b);\sigma+\varepsilon)|\\ &\ge|S(a,t^\sigma(b);\sigma)|\ge|S(a,t_\varepsilon^{\sigma-\varepsilon}(b);\sigma)|\\ &\ge|S(a,t^{\sigma-\varepsilon}(b);\sigma-\varepsilon)|=f(\sigma-\varepsilon)\end{aligned}$$

it follows that, for any such $\varepsilon$, the transformation $t_\varepsilon^{\sigma-\varepsilon}$ is an optimal transformation Ï.

The algorithm for optimal solution can be summarized in a pseudo-code:

```
OPTIMAL(a,b,σ)
1   ε ← 1
2   do
3     t_ε^σ ← EPSILON-OPTIMAL(a,b,σ,ε)
4     t_ε^{σ−ε} ← EPSILON-OPTIMAL(a,b,σ−ε,ε)
5     ε ← ε/2
6   while |S(a,t_ε^σ(b);σ+ε)| − |S(a,t_ε^{σ−ε}(b);σ)| > 0
7   return t_ε^{σ−ε}
```

Note that OPTIMAL returns an optimal superposition with probability one, i.e. whenever the input distance threshold $\sigma$ is not one of $\sigma_1, \ldots, \sigma_k$. However, the number of operations performed by OPTIMAL cannot be estimated upfront, since its running time depends on the difference between $\sigma$ and the closest $\sigma_i$ (which depends on the intrinsic geometry of the input proteins $a$ and $b$).

## 2.4 Practical benefits

Aside from having relatively low cost, EPSILON-OPTIMAL is amenable to parallel computing since $NET(\varepsilon)$ can be partitioned and the search procedure carried out simultaneously on the subsets of $NET(\varepsilon)$. To assess potential benefits of parallel implementations of EPSILON-OPTIMAL, we developed and tested a faster, heuristic version of the algorithm, called MAX-PAIRS.

For efficiency, MAX-PAIRS explores only a small subset of $NET(\varepsilon)$, consisting of only those transformations from $NET(\varepsilon)$ that are close to some high-scoring, 'seed' transformations. The assumption is that an optimal transformation is not far away from a sufficiently high scoring transformation, obtained by some fast and fairly accurate heuristic method.

To compute each seed transformation, MAX-PAIRS applies a well known, iterative alignment extension algorithm (see, e.g. Siew *et al.*, 2000) to $S = S(a, t(b); \sigma)$, where $t$ is the transformation minimizing RMSD between short segments of $k$ consecutive residues in $a$ and $b$ (default is $k = 5$). In every iteration of the extension algorithm, the proteins are superimposed to minimize the RMSD between the aligned residues (Kabsch, 1976) and a new alignment is computed by dynamic programming. The whole procedure is repeated until the alignment length $|S(a, t(b); \sigma)|$ remains unchanged between two consecutive iterations.

After generating all high scoring seed transformations, MAX-PAIRS 'refines' them, one by one, by exploring the 'nearby conformations' from $NET(\varepsilon)$. More specifically (and assuming that the seed transformation has already been applied to $b$), the algorithm selects three pairs of aligned points $\{(a_{i_k}, b_{i_k})\}_{k=1}^{3}$ from $S(a, b; \sigma)$ and then searches $NET(\varepsilon)$ while keeping the points $a_{i_k}$ and $b_{i_k}$ 'in contact', i.e. at distance $\leqslant \sigma$. By examining only the transformations $\tau$ such that $||a_{i_k} - \tau(b_{i_k})|| \leq \sigma$, for all $k \in \{1, 2, 3\}$, MAX-PAIRS significantly reduces the size of the search space, resulting in increased efficiency (for technical details, we refer the reader to the Supplementary Material).

*2.4.1 Added value of optimal solution.* We tested the performance of MAX-PAIRS on a representative set of protein chains selected from the SCOP database (Murzin *et al.*, 1995). Our test set contains 195 pairs of proteins related at various levels according to SCOP structural classification: 57 family pairs, 75 superfamily pairs, and 63-fold pairs (complete test set can be found at http://bioinformatics.cs.uni.edu/opt_align.html).

The sensitivity of MAX-PAIRS is compared to sensitivity of four methods: CASP's program LGA (Zemla, 2003), TM-align (Zhang and Skolnick, 2005), MAMMOTH (Ortiz *et al.*, 2002), and MUSTANG (Konagurthu *et al.*, 2006). For efficiency, we set the accuracy parameter $\varepsilon$ of MAX-PAIRS to 1 (we recall that decreasing $\varepsilon$ yields more accurate, but less efficient procedure). The scores for MAMMOTH and MUSTANG, presented in Tables 1 and 2, are shown for reference only, since, in contrast to MAX-PAIRS and LGA, which attempt to maximize $CA \leq \sigma$, these programs seek to optimize a different objective function.

As seen in Tables 1 and 2, even at $\varepsilon = 1$, MAX-PAIRS compares favorably with the other methods across all SCOP categories and at both distance threshold parameters (3 and 5 Å).

Figure 3 shows the distribution of the added value of MAX-PAIRS, measured by the additional number of $CA \leq \sigma$ pairs detected by our algorithm. As seen in the left tails of the distributions in Figure 3, LGA and TM-align perform better than MAX-PAIRS on several pairs of test structures. This is not surprising, given that LGA, TM-align, and MAX-PAIRS each explore different sets of transformations in search for the best superposition.

Since MAX-PAIRS searches a subspace of transformations with a 'fine-tooth comb', one would also expect to see examples of significantly better performance of MAX-PAIRS compared to other methods. We show one of those examples in Figure 4. Although the cases similar to one in Figure 4 are relatively rare, they illustrate the advantage of a rigorous approach to protein structure comparison.

**Table 1.** The total number of pairs of residues in the test set that can be superimposed under 3 Å

| | MAX-PAIRS | LGA | TM-align | Mammoth | Mustang |
|---|---|---|---|---|---|
| *Family* | 4689 | 4585 | 4460 | 4264 | 4231 |
| *Superfamily* | 4378 | 4247 | 4140 | 3713 | 3319 |
| *Fold* | 2870 | 2720 | 2634 | 2100 | 1834 |

**Table 2.** The total number of pairs of residues in the test set that can be superimposed under 5 Å

| | MAX-PAIRS | LGA | TM-align | Mammoth | Mustang |
|---|---|---|---|---|---|
| *Family* | 5251 | 5130 | 5059 | 5019 | 4983 |
| *Superfamily* | 5240 | 5033 | 4928 | 4702 | 4562 |
| *Fold* | 3575 | 3409 | 3279 | 2842 | 2816 |

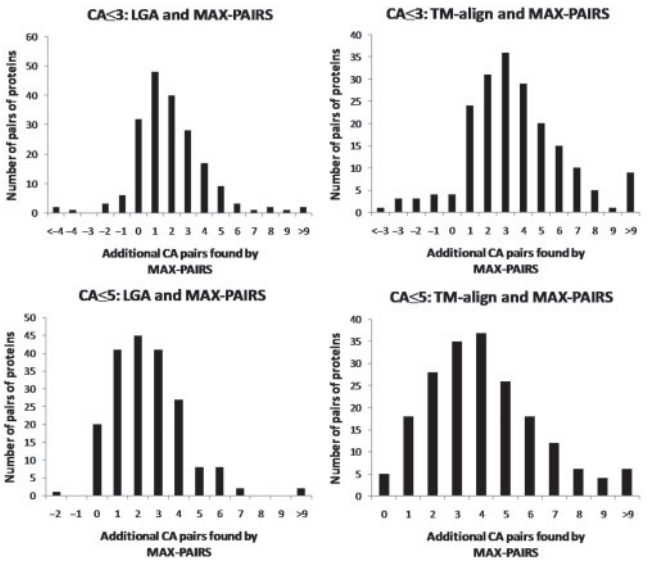


**Fig. 3.** The distribution of the difference in $CA \leq \sigma$ scores of MAX-PAIRS and LGA (TM-align) provides insight into the advantage of the algorithms for optimal solution.

Table 3 shows the efficiency of MAX-PAIRS as a function of $\varepsilon$ on the set of pairs of structures from our test set. The analysis presented in Table 3 is performed on a 2.13 GHz Intel(R) CPU computer running Linux. The results summarized in this table suggest that, although both LGA and TM-align are much more efficient programs than MAX-PAIRS, the sensitivity of MAX-PAIRS compares favorably to both of these programs, for each tested accuracy threshold $\varepsilon$.

Many pairwise protein structural alignment methods, including the methods discussed in this article, employ a key routine for computing a superposition that maximizes $CA \leq \sigma$ between a pair of protein structures. It is reasonable to expect that improving this superposition increases the


**Fig. 4.** The superpositions of 1jb0C and 1a6lA (same SCOP superfamily) reported by LGA ($CA \leq 3 = 31$) and MAX-PAIRS ($CA \leq 3 = 55$). The thin gray line shows the backbone of 1jb0C (CA atoms only). The black line represents segments of 1a6lA that can be superimposed under 3 Å with the corresponding segments in 1jb0C. In this test case, LGA completely misaligns two structures, even at higher distance thresholds (LGA's $CA \leq 5 = 36$).

**Table 3.** Speed versus accuracy of MAX-PAIRS

| $\varepsilon$ | Cumulative $CA \leq 3$ | Time per pair (s) |
|---|---|---|
| 1.0 | 11 937 | 6608 |
| 1.5 | 11 862 | 713 |
| 2.0 | 11 789 | 140 |
| 2.5 | 11 711 | 46 |
| 3.0 | 11 602 | 19 |
| 3.5 | 11 566 | 9 |

For comparison, the cumulative $CA \leq 3$ of LGA and TM-align are 11 552 (2.1 s) and 11 234 (<0.1 s), respectively.

accuracy of a protein structure alignment algorithm. We evaluated extent of this increase using the Sisyphus benchmark for the alignment accuracy (Andreeva *et al.*, 2007). The Sisyphus test set contains 125 manually created structural alignments for protein pairs with non-trivial structural relationships. These alignments can be used (as gold standards) for assessing the accuracy of a protein structure alignment method. In order to compare the alignment accuracy of the algorithms in our study with accuracy of the methods previously tested in Sisyphus benchmark, we, like Rocha *et al*., utilize only a subset of the Sisyphus test set containing 106 pairs of single chain proteins.

To test usefulness of the algorithms for optimizing $CA \leq \sigma$, we modified the TM-align method by replacing the original TM-align superpositions with the superpositions generated by the MAX-PAIRS program. The modified TM-align program, called MP-TM-align, uses the TM-align scoring function (TM-score) to compute an optimal structural alignment of proteins superimposed with the MAX-PAIRS program. As seen in Table 4, not only the MP-TM-align program outperforms the original TM-align method for each tolerance shift, but the accuracy of this simple hybrid method is comparable to the accuracy of the top performing methods tested by Rocha and coworkers (Rocha *et al*., 2009).

It is interesting to note that, according to study of Rocha *et al*., the most accurate structure alignment methods, such as Matt (Menke *et al.*, 2008), PPM (Csaba *et al*., 2008) and ProtDeform (Rocha *et al*., 2009), consider proteins as flexible objects. These methods achieve high alignment accuracy by applying a sequence of different rigid transformations at different sites, rather than a single global rigid transformation. On the other hand, the results of our study show that highly accurate methods can still be developed that rely on a single rigid transformation to assess the similarity of two protein structures.

**Table 4.** The agreement with reference alignments for six different tolerance shifts

| | Tolerance shift | | | | | |
|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 |
| FLEXPROT | 0.449 | 0.672 | 0.707 | 0.725 | 0.742 | 0.747 |
| MATRAS | 0.776 | 0.806 | 0.828 | 0.836 | 0.847 | 0.847 |
| PD | 0.791 | 0.849 | 0.858 | 0.868 | 0.881 | 0.882 |
| PPM | 0.782 | 0.813 | 0.823 | 0.833 | 0.843 | 0.844 |
| RASH | 0.688 | 0.793 | 0.812 | 0.840 | 0.854 | 0.855 |
| SSAP | 0.750 | 0.786 | 0.797 | 0.804 | 0.808 | 0.811 |
| VOROLIGN | 0.722 | 0.765 | 0.790 | 0.808 | 0.826 | 0.830 |
| DALI | 0.800 | 0.830 | 0.845 | 0.851 | 0.859 | 0.860 |
| MATT | 0.829 | 0.866 | 0.889* | 0.904* | 0.915* | 0.917* |
| LGA | 0.765 | 0.820 | 0.831 | 0.839 | 0.847 | 0.849 |
| TM-align | 0.762 | 0.815 | 0.823 | 0.834 | 0.841 | 0.844 |
| MP-TM-align | 0.809 | 0.861 | 0.875 | 0.884 | 0.896 | 0.896 |
| MP-TM-align+ | 0.830* | 0.867* | 0.881 | 0.887 | 0.897 | 0.898 |

For the tolerance shift $s$, the agreement is defined as $I_s/L_{\text{ref}}$, where $I_s$ is the number of aligned residues that are shifted by no more than $s$ positions in the reference alignment and $L_{\text{ref}}$ is the length of the reference alignment (see Rocha *et al*., 2009). The best results are denoted by asterisk.

An even further increase in the accuracy of TM-align can be achieved by utilizing the residue type information into the alignment process. Combining the distance-based measures with the residue mutation scores is a standard technique used in many structure alignment methods, such as CE (Shindyalov and Bourne, 1998). As seen in Table 4, a variant of the MP-TM-align method, called MP-TM-align+, in which the alignment scoring function is defined as the sum of the TM-score and the BLOSUM62 score (Henikoff and Henikoff, 1992) yields the most accurate alignments in the Sisyphus benchmark of the alignment accuracy.

A closer look at the results summarized in Table 4 and those obtained by Rocha *et al*. reveal a significant difference in the performance of TM-align in these two studies. This difference is due to different versions of TM-align used in two experiments. More specifically, the TM-align program tested in our benchmark is released on 14 March 2009 and is 4% more accurate than the older TM-align program evaluated by Rocha *et al.* (see http://zhang.bioinformatics.ku.edu/TM-align/).

The alignments of the proteins in the Sisyphus test set generated by LGA, TM-align, MP-TM-align and MP-TM-align+ can be downloaded from http://bioinformatics.cs.uni.edu/opt_align.html. The alignments generated by the remaining 10 methods can be found at http://dmi.uib.es/people/jairo/bio/ProtDeform.

There remains a lot of work to be done on speeding up MAX-PAIRS and making it practical for large-scale protein structure analysis. However, even in its present form, MAX-PAIRS can be useful in assessing the performance of protein 3D structure prediction methods. For example, the accuracy of MAX-PAIRS is 3.6% higher than the accuracy of the LGA program, officially used at the biannual CASP competition (http://predictioncenter.org). This is a significant advantage of our method, given that the difference in GDT_TS scores between the first and the second ranked method in CASP7, as measured by LGA, is only 2.6% (3.5% in CASP8).

## 3 CONCLUSION

The homology between two proteins is often concluded based on their structural similarity. However, due to the infinite (uncountable) space of all possible spatial configurations, finding an optimal superposition for a pair of proteins is a challenging problem.

In this article, we show that the approximate structural alignment problem is tractable for a range of commonly used structural alignment metrics, such as GDT, AL0 and *MaxSub*. Although our algorithm for near-optimal solution consumes large computational time, the running time can be easily reduced with parallel implementations. An added benefit of the algorithm for approximate solution is that it provides a measure of the solution quality, which signals whether the returned superposition is, in fact, an optimal superposition.

We also present a procedure capable of finding an optimal superposition of any two proteins, for all but finitely many distance thresholds. Although it theoretically addresses a long-standing question of whether such an algorithm exists, our procedure for finding an absolute optimum is too slow for practical applications.

## ACKNOWLEDGEMENTS

The authors thank Dr Adam Zemla for kindly providing his LGA program.

*Conflict of Interest*: none declared.

## REFERENCES

Andreeva,A. *et al*. (2007) SISYPHUS—structural alignments for proteins with non-trivial relationships. *Nucleic Acids Res.*, **35**, D253–D259.

Alexandrov,N.N. *et al*. (1992) Common spatial arrangements of backbone fragments in homologous and nonhomologous proteins. *J. Mol. Biol.*, **225**, 5–9.

Caprara,A. *et al*. (2004) 1001 optimal PDB structure alignments: integer programming methods for finding the maximumcontact map overlap. *J. Comput. Biol.*, **11**, 27–52.

Csaba,G. *et al*. (2008) Protein structure alignment considering phenotypic plasticity. *Bioinformatics*, **24**, i98–i104.

Eidhammer,I. *et al*. (2000) Structure comparison and structure patterns. *J. Comput. Biol.*, **7**, 685–716.

Fischer,D. *et al*. (2003) CAFASP3: the third critical assessment of fully automated structure prediction methods. *Proteins*, **53**(Suppl. 6), 503–516.

Ginalski,K. *et al*. (2005) Practical lessons from protein structure prediction. *Nucleic Acids Res.*, **33**, 1874–1891.

Goldman,D. *et al*. (1999) Algorithmic aspects of protein structure similarity. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, IEEE Computer Science, Washington, DC, USA, pp. 512–522.

Goldsmith-Fischman,S. and Honig,B. (2003) Structural genomics: computational methods for structure analysis. *Prot. Sci.*, **12**, 1813–1821.

Henikoff,S. and Henikoff,J.G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl Acad. Sci. USA*, **89**, 10915–10919.

Hao,M.H. *et al*. (1992) Effects of compact volume and chain stiffness on the conformations of native proteins. *Proc. Natl Acad. Sci. USA*, **89**, 6614–6618.

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

Kabsch,W. (1976) solution for the best rotation to relate two sets of vectors. *Acta Crystallographica*, **32**, 922–923.

Kolodny,R. and Linial,N. (2003) Approximate protein structural alignment in polynomial time. *Proc. Natl Acad. Sci. USA*, **101**, 12201–12206.

Konagurthu,A.S. *et al*. (2006) MUSTANG: multiple structural alignment algorithm. *Proteins*, **64**, 559–574.

Lathrop,R.H. (1994) The protein threading problem with sequence amino acid interaction preferences is NP-complete. *Protein Eng.*, **7**, 1059–1068.

Menke,M. *et al*. (2008) Matt: local flexibility aids protein multiple structure alignment. *PLOS Computat. Biol.*, **4**, 88–99.

Moult,J. *et al*. (2007) Critical assessment of methods of protein structure prediction Round VII. *Proteins*, **69**, 3–9.

Murzin,A.G. *et al*. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, **247**, 536–540.

Oldfield,T.J. (2007) CAALIGN: a program for pairwise and multiple protein structure alignment. *Acta Crystallogr. D Biol. Crystallogr.*, **63**, 514–525.

Orengo,C.A. and Taylor,W.R. (1996) SSAP: sequential structure alignment program for protein structure comparison. *Methods Enzymol.*, **266**, 617–635.

Ortiz,A.R. *et al*. (2002) MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. *Protein Sci.*, **11**, 2606–2621.

Rocha,J. *et al*. (2009) Flexible structural protein alignment by a sequence of local transformations. *Bioinformatics*. **25**, 1625–1631.

Russell,R.B. and Barton,G.J. (1992) Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels. *Proteins*, **14**, 309–323.

Sali,A. and Blundell,T.L. (1993) Comparative protein modeling by satisfaction of spatial restraints. *J. Mol. Biol.*, **234**, 779–815.

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

Siew,N. *et al*. (2000) MaxSub: an automated measure for the assessment of protein structure prediction quality. *Bioinformatics*, **16**, 776–785.

Smith,T.F. and Waterman,M.S. (1981) Identification of Common Molecular Subsequences. *J. Mol. Biol.*, **147**, 195–197.

Subbiah,S. *et al*. (1993) Structural similarity of DNA-binding domains of bacteriophage repressors and the globin core. *Curr. Biol.*, **3**, 141–148.

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

Vriend,G. and Sander,C. (1991) Detection of common three-dimensional substructures in proteins. *Proteins*, **11**, 52–58.

Xu,J. *et al*. (2007) A parameterized algorithm for protein structure alignment. *J. Comput. Biol.*, **14**, 564–577.

Ye,Y. and Godzik,A. (2003) Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics*, **19**(Suppl. 2), ii246–ii255.

Zemla,A. (2003) LGA—a method for finding 3D similarities in protein structures. *Nucleic Acids Res.*, **31**, 3370–3374.

Zhang,Y. and Skolnick,J. (2005) TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.*, **33**, 2302–2309.