BIOINFORMATICS ORIGINAL PAPER Vol. 26 no. 18 2010, pages 2250–2258
doi:10.1093/bioinformatics/btq402

Structural bioinformatics

Advance Access publication July 7, 2010

Fast overlapping of protein contact maps by alignment of eigenvectors

Pietro Di Lena[1,*], Piero Fariselli[2], Luciano Margara[1], Marco Vassura[1] and Rita Casadio[2]

[1]Department of Computer Science, University of Bologna, Mura Anteo Zamboni 7 and [2]Biocomputing Group, University of Bologna, Via S.Giacomo 9/2, 40127 Bologna, Italy

Associate Editor: Burkhard Rost

ABSTRACT

Motivation: Searching for structural similarity is a key issue of protein functional annotation. The maximum contact map overlap (CMO) is one of the possible measures of protein structure similarity. Exact and approximate methods known to optimize the CMO are computationally expensive and this hampers their applicability to large-scale comparison of protein structures.

Results: In this article, we describe a heuristic algorithm (AI-Eigen) for finding a solution to the CMO problem. Our approach relies on the approximation of contact maps by eigendecomposition. We obtain good overlaps of two contact maps by computing the optimal global alignment of few principal eigenvectors. Our algorithm is simple, fast and its running time is independent of the amount of contacts in the map. Experimental testing indicates that the algorithm is comparable to exact CMO methods in terms of the overlap quality, to structural alignment methods in terms of structure similarity detection and it is fast enough to be suited for large-scale comparison of protein structures. Furthermore, our preliminary tests indicates that it is quite robust to noise, which makes it suitable for structural similarity detection also for noisy and incomplete contact maps.

Availability: Available at http://bioinformatics.cs.unibo.it/AI-Eigen

Contact: dilena@cs.unibo.it

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

Received on March 12, 2010; revised on June 30, 2010; accepted on July 2, 2010

1 INTRODUCTION

Measuring the similarity of protein structures is a necessary step in several problems of bioinformatics, including the determination of structure conservation through evolution across species and/or the relation among structure and function. Several similarity measures of protein structures are available albeit no general agreement on the best one has been achieved (Godzik, 1996; Oakley *et al*., 2008; Sadreyev *et al*., 2009). Routinely, each similarity measure relies on the choice of a scoring function and on the assumption that its optimum corresponds to the best possible match between two protein structures.

The most widely adopted scoring measures are based on the *root mean square deviation* (RMSD; Kabasch, 1976), *distance map similarity* (Holm and Sander, 1993), *contact map overlap* (CMO; Godzik *et al*., 1992) and, recently, *universal similarity metric* (UMS; Krasnogor and Pelta, 2004; Rahmati and Glasgow, 2009). The maximum CMO measure quantifies the level of similarity between two protein structures by measuring the maximum overlap of their contact maps and it is obtained by computing the sequence alignment that maximizes the number of corresponding contacts between pairs of aligned residues. The maximum CMO is one of the few measures for which *exact* algorithms are known. On the other hand, the maximum CMO problem is known to be NP hard (Goldman *et al*., 1999).

Although not explicitly stated in previous papers dealing with CMO algorithms, the goal of the these tools is to address the problem of protein structure prediction. In practice, given a set of predicted contacts the basic idea is to look for the best possible physical contact map among the available ones by means of a CMO program: methods for contact prediction are presently not accurate enough (Ezkurdia *et al*., 2009) to allow protein structure reconstruction (Vassura *et al*., 2008). The main contribution of CMO algorithms is therefore to provide tools for template recognition when the native protein structure is unknown and not to substitute structural alignment methods (i.e. methods based on RMSD and distance map similarity measures), which perform quite well.

The first exact algorithm for the maximum CMO problem, based on *integer programming* (IP), was developed in Lancia *et al*. (2001) and improved in Caprara and Lancia (2002). Later, several other methods based on the same approach were proposed (Andonov *et al.*, 2008; Strickland *et al.*, 2005; Xie and Sahinidis, 2007). The IP approach consists in formulating the CMO as the maximization of some integer linear function and solving it with Lagrangian Relaxation (LR) and/or Branch and Bound reduction techniques. The disadvantage of IP-based methods is that, due to the intractability of the problem, they are exponential in the worst case. For a practical usage, the running time of these algorithms is bounded and the best solution within the time limit is returned. The counter part is that the IP-based methods provide upper and lower bounds to the optimal solution and this makes it possible to evaluate the quality of the partial solution from the distance between the upper and lower bound. The best possible overlap is found when the upper and lower bounds coincide. Recently, two polynomial-time approximation schemes for the protein structure alignment problem (in particular, contact map alignment) have been developed (Agarwal *et al.*, 2007; Xu *et al.*, 2007). The approximation algorithm described in Xu *et al.* (2007) is polynomial in the protein length but it is exponential with respect to some constant parameters and its running time increases with decrease of an approximation factor. The method developed

*To whom correspondence should be addressed.

in Agarwal *et al.* (2007) is based on a decomposition procedure on the input graphs. It is a six-approximation algorithm, i.e. it returns a solution that is at least $1/6$ distant from the optimal one, and it has a polynomial running time.

Despite the strength of the underlying formalization, the CMO-based algorithms are scarcely used to compare protein structures. Indeed, the algorithmic implementations of the exact and approximation methods are on the average too slow to be used for wide-scale comparison. Furthermore and most importantly, there is no agreement on the most suitable value of contact threshold to represent a protein structure (Caprara *et al.*, 2004; Duarte *et al.*, 2010; Vassura *et al.*, 2008). High-threshold contact maps ($\geq$10 Å) are more informative; however, their comparison requires an even higher computational time and this makes the adoption of exact/approximation methods not feasible.

To our knowledge, only three heuristic methods, SADP (Jain and Lappe, 2007), MSVNS (Pelta *et al.*, 2008) and BIMAL (Jain and Obermayer, 2009), have been proposed for the CMO problem. Even if not optimal, SADP, MSVNS and BIMAL can produce acceptable solutions in a reasonable time compared with exact methods. The main limitation of available heuristic CMO methods is that their running time depends on the number of contacts making them extremely slow when the contact maps contain a huge number of contacts.

In this article, we describe a new heuristic algorithm for the CMO problem. Our approach is based on the property that a contact map can be well-approximated by few of its eigenvectors. Thus, an acceptable overlapping of two contact maps can be heuristically obtained by performing a global alignment of few eigenvectors. The exploitation of the eigenvalues and eigenvector properties of the graph adjacency matrices have been extensively investigated in the more general context of inexact graph matching. In the pioneering work by Umeyama (1988), lately improved in Zhao *et al.* (2007), the eigendecomposition is introduced to detect a near-optimal permutation matrix that maximizes the similarity between two labeled graphs of the same size. The limitation on the graph dimensions is overcome in more recent approaches, such as Luo and Hancock (2001) and Singh *et al.* (2007), where the eigendecomposition is used to detect local similarities between the neighborhood topologies of the nodes in the two graphs. A different approach that exploits eigendecomposition is based on the conversion of a graph into a string by using the ordering of the nodes as defined by the principal eigenvector of its adjacency matrix (Robles-Kelly and Hancock, 2002). By this, two graphs can be matched by computing an alignment that minimizes the edit distance between the corresponding strings. However, these approaches, differently from the CMO methods, do not require to preserve the ordering of the nodes in the adjacency matrices. Differently, in our method the eigendecomposition is used to transform the two-dimensional alignment problem between contact maps into a one-dimensional alignment problem between eigenvectors. Therefore, in our case, the ordering between the nodes in the maps is naturally imposed by the one-dimensional alignment between their eigenvectors.

Our algorithm is easily implementable and fast. Noticeably, by design, its running time does not depend on the number of contacts contained in the map and thus on the contact threshold. Experimental results show that it can compute good overlaps compared with exact CMO methods and that its performances in terms of protein structure recognition/classification are comparable with those of structural alignment methods. The running time of our implementation is comparable with the fastest structural alignment algorithms and heuristic CMO methods. Our tests confirm that contact maps computed at thresholds values $\geq$10 Å are more informative of the protein structures than those at lower thresholds and this makes our algorithm more suitable than other available CMO methods for protein structure comparison on large scale. Furthermore, in order to evaluate the robustness our algorithm, we test the effect of random noise on the accuracy of structural similarity recognition performance. To the best of our knowledge, this is the first time that a CMO algorithm is evaluated in this way. The experimental tests indicate that our method is quite robust and can tolerate high amount of noise without dramatically affecting its recognition capabilities.

2 BACKGROUND

2.1 Contact maps

A protein *contact map* is a two-dimensional *approximation* of the protein three-dimensional structure. For a given protein P, its contact map of *threshold* τ is a square binary symmetric matrix defined by

$$M_{ij}^{P} = \begin{cases} 1 & \text{if the distance between residues } i,j \text{ is } \leq \tau \text{ Å} \\ 0 & \text{otherwise} \end{cases}$$

There are several definitions of *distance* between residues in literature. The particular choice of a distance is not critical since the CMO problem is independent of the distance used to represent contacts between residues.

Following the CMO literature, we consider here the C_α *distance*, which defines the distance between residues i,j as the Euclidean distance between the coordinates of their respective C_α atoms. Typical threshold values for C_α contact maps vary from 6 Å to 16 Å. For this range of thresholds, consecutive residues are always in contact (consecutive residues share a peptide bond and the distance between their respective C_α atoms is about 3.7 Å). For low-threshold values, typically 6–9 Å, the number of *contacts* (i.e. 1s) observed in the map is sparse compared with the number of *non-contacts* (i.e. 0s). Moreover, these threshold values are the ones which minimize the distance between C_α contact maps and *physical* contact maps (Bartoli *et al.*, 2007). On the contrary, high-threshold contact maps (10–16 Å) have a higher number of contacts and are more informative about the protein structure. At low-threshold values, several different three-dimensional structures can be consistent with the same contact map: the ambiguity can be minimized by increasing the threshold of the contact map (Duarte *et al.*, 2010; Vassura *et al.*, 2008). The threshold problem was also noticed in Caprara *et al.* (2004), the authors report that a threshold value smaller than 7 Å is not suitable to represent the protein structures in their benchmark set.

2.2 The maximum CMO problem

Given two proteins P_1, P_2, whose (ordered) sets of residues are denoted, respectively, by $R_1 = \{1,\ldots,n\}$ and $R_2 = \{1,\ldots,m\}$, an *alignment* between P_1 and P_2 is a mapping $f : R_1 \to R_2$ that respects the following two conditions:

(1) f is an injective partial function; and

(2) for each pair of residues $i,j \in R_1$ in the domain of f (i.e. $f(i) \neq \emptyset \neq f(j)$) we have that

$$i < j \text{ if and only if } f(i) < f(j).$$

Condition 1 imposes that a residue in the first/second protein can be aligned at most with one (possibly none) residue in the second/first protein. The non-aligned residues are assumed to be matched with *gaps*. Biologically, the introduction of a gap reflects an insertion/deletion event during the evolution of protein sequences. The number of gaps in an alignment f is defined by

$$\text{gap}_f = |\{i \mid i \in R_1, f(i) = \emptyset\}| + |\{i \mid i \in R_2, f^{-1}(i) = \emptyset\}|.$$

Condition 2 imposes the ordering of the residues to be preserved in the alignment.

The maximum CMO for proteins P_1, P_2, is an alignment that maximizes the overlap between their respective contact maps M^{P_1}, M^{P_2}. More formally, the maximum CMO problem is defined as the problem of computing the alignment f that maximizes the quantity

$$O(M^{P_1}, M^{P_2}) = \sum_{\substack{f(i) \neq \emptyset \neq f(j) \\ j > i+1, f(j) > f(i)+1}} M^{P_1}_{ij} \cdot M^{P_2}_{f(i)f(j)} \qquad (1)$$

Noticeably, since contacts between consecutive amino acids are always present, they are not counted in (1). Moreover, a match between a contact and a non-contact is not penalized in (1).

The CMO can be used as a measure of the similarity between two proteins structures: the higher the overlap between two contact maps, the higher the probability that the two related protein structures are similar. The CMO measure is quite robust to perturbations and does not greatly penalize the insertion of gaps and deletions. The CMO as similarity measure was introduced in Godzik *et al.* (1992). The problem of computing the maximum CMO was proven to be NP hard in Goldman *et al.* (1999). To quantify the level of similarity of two overlapped contact maps, we use the most widely adopted scoring function, originally proposed in Xie and Sahinidis (2007):

$$\frac{2 \cdot O(M^{P_1}, M^{P_2})}{C(M^{P_1}) + C(M^{P_2})} \qquad (2)$$

where $C(M) = \sum_{j > i+1} M_{ij}$ denotes the number of contacts in the contact map M.

3 MATERIALS AND METHODS

3.1 Datasets

In order to compare our results with those published before, we use the protein sets previously described. In practice, we use five different datasets (see Table 1 and the Supplementary Material) for three different tests: performance comparison with exact and heuristic CMO methods (Section 4.1), recognition and classification performance comparison with structural alignment methods (Section 4.2) and error tolerance test (Section 4.3).

For the comparison with exact CMO methods, we refer to the results published in Andonov *et al.* (2008) on the Skolnick dataset. The contact maps used in Andonov *et al.* (2008) have been computed at 7.5 Å threshold and are available as Supplementary Material.

For the comparison with heuristic CMO methods, we refer to the results published in Jain and Obermayer (2009) on the Sokol, Lancia, Skolnick and Fischer datasets. These datasets are usually adopted as standard benchmark to evaluate overlap quality of CMO algorithms. The maps for the Skolnick, Lancia and Sokol datasets used in Jain and Obermayer (2009) are available as Supplementary Material of Xie and Sahinidis (2007). The contact threshold used to compute these maps is not reported but it seems to be lower than 7.5 Å. The maps for the Fischer dataset are not publicly available; thus, we computed our maps at threshold 7.5 Å for this test.

Table 1. Dataset statistics and references

Dataset	Proteins/Domains	Residues		
		Min	Avg ± SD	Max
Sokol	9	12	39±14	51
Lancia	269	44	57±4	68
Skolnick	40	97	160±61	256
Fischer	68	62	183±105	534
Proteus300	300	64	193±97	465

References: Sokol (Strickland *et al.*, 2005), Lancia (Lancia *et al.*, 2001), Skolnick (Lancia *et al.*, 2001), Fischer (Fischer *et al.*, 1996), Proteus300 (Andonov *et al.*, 2008).

For the comparison with structural alignment methods and for testing the error tolerance of our method, we use the Fischer and Proteus300 datasets, which contain medium–large protein domains and are more varying with respect to the SCOP (Andreeva *et al.*, 2008) classification than the other benchmarks considered. The 68 domains in the Fischer dataset are distributed in 56 distinct SCOP families, 44 distinct super families and 40 distinct folds. Among them, 22 domains belong to a non-unique family (i.e. for each one of them there is in the dataset at least one representative in the same family class), five have one representative in the same fold class but not in the same family/s.family classes and 22 domains belong to a unique fold class. The 300 domains in the Proteus300 dataset are distributed in 30 distinct SCOP families (10 domains per family), 27 distinct super families and 24 folds. For the recognition/classification experiments, we test several different threshold values. For all the other experiments, we use contact maps at threshold 7.5 Å (or lower).

3.2 Al-eigen implementation

The maximum CMO problem involves the alignment of two-dimensional objects. While there are optimal polynomial-time algorithms for the alignment in one-dimensional space, the problem is not tractable in two dimensions. Here, we describe a heuristic method to obtain a two-dimensional alignment of contact maps by means of a one-dimensional alignment of their eigenvectors. Our approach uses standard techniques such as the canonical eigendecomposition of symmetric matrices (Strang, 2003) and the Needleman–Wunsch (NW) alignment algorithm (Needleman and Wunsch, 1970).

The *spectral theory* provides conditions under which a matrix can be decomposed into a canonical form in terms of *eigenvalues* and *eigenvectors*. This canonical decomposition is usually called *eigendecomposition* or *spectral decomposition*. By the spectral theorem, every real $n \times n$ symmetric matrix M can be eigendecomposed as

$$M = \sum_{i=1}^{n} \lambda_i (\mathbf{v}_i \otimes \mathbf{v}_i) \qquad (3)$$

where λ_i represents the i-th eigenvalue, $\mathbf{v}_i$ the corresponding eigenvector and $\otimes$ denotes the outer product between vectors. The ordering of the eigenvalues is not important provided that the eigenvectors are permuted accordingly; thus, we can always assume that the eigenvalues are sorted in decreasing order, i.e. $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. Equation (3) defines matrix M as the sum of $n \times n$ matrices $\mathbf{v}_i \otimes \mathbf{v}_i$, weighted by the corresponding eigenvalues λ_i.

The eigendecomposition allows the detection of the most relevant information contained in a contact map. For example, in Porto *et al.* (2004),

the authors show that a contact map can be perfectly reconstructed starting from the only knowledge of its principal eigenvector. In practice, a contact map can be *approximated* by considering only few of its eigenvectors/eigenvalues. For instance, for $1 \leq t \leq n$, the approximation of order t of M can be defined as

$$\tilde{M} = \sum_{i=1}^{t} \lambda_i (\mathbf{v}_i \otimes \mathbf{v}_i) \tag{4}$$

This way of approximating a contact map is effective because the smaller is eigenvalue λ_i the smaller is the contribution of matrix $\mathbf{v}_i \otimes \mathbf{v}_i$ in Equation (3). This is actually one of the approaches used for image data compression (Andrews and Patterson, 1976).

Consider now two proteins P_1, P_2 with contact maps $M^{P_1} \in \{0,1\}^{n \times n}, M^{P_2} \in \{0,1\}^{m \times m}$, respectively. For some given $1 \leq t \leq \min\{n,m\}$, we can heuristically compute an overlap between M^{P_1} and M^{P_2} by computing an alignment that maximizes an opportune scoring function defined on their respective t eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_t$ and $\mathbf{v}_1, \ldots, \mathbf{v}_t$. Since the scoring function (1) does not penalize the eventual match of a contact with a non-contact, a *global alignment* of eigenvectors is preferred to a *local alignment*.

The NW algorithm computes in polynomial-time the optimal global alignment of two sequences with respect to some *scoring matrix* $S \in \mathbb{R}^{n \times m}$ and (constant) *gap penalty* $G \in \mathbb{R}$. The entry S_{ij} of the scoring matrix denotes the level of similarity between the i-th residue of P_1 and the j-th residue of P_2. The constant value G defines the cost for the introduction of a gap in the alignment. The NW algorithm can be easily modified in order to encode non-constant gap penalties. In this work, we consider only constant gap penalties. Formally, the NW algorithm computes the alignment $f : R_1 \to R_2$ that maximizes the objective function

$$\sum_{\substack{i=1 \\ f(i) \neq \emptyset}}^{n} S_{if(i)} + G \cdot \mathrm{gap}_f$$

In the following, we describe the scoring function (a) and the constant gap penalty (b) of our NW implementation.

(a) By Equation (3), for residues i,j of protein P_1 the quantity

$$\lambda_1 (\mathbf{v}_1)_i (\mathbf{v}_1)_j + \cdots + \lambda_t (\mathbf{v}_t)_i (\mathbf{v}_t)_j \tag{5}$$

will tend to 1 with increasing of t if i,j are in contact and to 0 otherwise. In the quantity (5), the contribution of each product $(\mathbf{v}_k)_i (\mathbf{v}_k)_j$ is weighted by the corresponding eigenvalue λ_k. Moreover, when λ_k is positive, such a contribute is positive if and only if $(\mathbf{v}_k)_i$ and $(\mathbf{v}_k)_j$ agree in sign. We describe the i-th residue of P_1 by the i-th entries of eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_t$ weighted by the square root of the corresponding eigenvalue λ_k:

$$[(\mathbf{v}'_1)_i, \ldots, (\mathbf{v}'_t)_i] = [\sqrt{|\lambda_1|}(\mathbf{v}_1)_i, \ldots, \sqrt{|\lambda_t|}(\mathbf{v}_t)_i] \tag{6}$$

According to this representation, the i-th residue of P_1, described by $[(\mathbf{u}'_1)_i, \ldots, (\mathbf{u}'_t)_i]$, should be matched with the j-th residue of P_2, described by $[(\mathbf{v}'_1)_i, \ldots, (\mathbf{v}'_t)_i]$, when the pairwise entries of the these vectors highly agree both in sign and relative magnitude. The vectors do not need to be equal to obtain a high score; for this reason, a scoring scheme based on the Euclidean distance is not appropriate. Experimentally, we found that the scoring function that provides the best performances is

$$S_{ij} = \sum_{k=1}^{t} (\mathbf{u}'_k)_i (\mathbf{v}'_k)_j \tag{7}$$

The scoring function (7) assigns high scores when the corresponding entries of the vectors describing residues i and j have the same sign. Moreover, in S_{ij} the contribution of each product $(\mathbf{u}'_k)_i (\mathbf{v}'_k)_j$ is weighted by the square root of the corresponding eigenvalues.

(b) The gap penalty was evaluated experimentally. We found that in nearly all cases a good choice is

$$G = \min\{0, \min\{S_{ij} \mid 1 \leq i \leq n, 1 \leq j \leq m\}\} \tag{8}$$

Note that the sign of the eigenvectors has no influence in Equation (3), i.e. $\mathbf{v}_i \otimes \mathbf{v}_i = -\mathbf{v}_i \otimes -\mathbf{v}_i$. In fact, there is no way to standardize the sign of the eigenvectors in the eigendecomposition. This implies that, when aligning two sets of t eigenvectors, we are forced to try all possible combinations of their signs. By definition of (7), to consider all possible combinations, it is sufficient to try all possible sign combinations for just one set of t eigenvectors, thus producing 2^t different alignments. Moreover, in some cases, we found that, with increasing number of eigenvectors, the quality of the alignment (in terms of the number of overlapping contacts recovered) slightly decreases. Thus, when aligning t eigenvectors, our algorithm proceeds as follows. First, it computes all alignments with one eigenvector for each pair, then with two and so on up to t. This procedure evaluates a total of $\sum_{k=1}^{t} 2^k = 2^{t+1} - 2$ alignments. The best alignment in terms of overlapping contacts is chosen. One execution of the NW algorithm costs $\Theta(nm)$, so our algorithm costs $\Theta(2^{t+1} \cdot nm)$, i.e. it is exponential in the order of approximation t. Anyway, the running time is not dependent on the contact map threshold and for values of t up to 7 the running time is small enough to assure a fast computation, as shown below.

3.3 Evaluation of the running time

The experiments were run on an Intel Pentium machine with a 2.80 GHz CPU and with 1 GB RAM. In our case, the time necessary to perform the eigendecomposition is not included for the comparison with other methods, since the eigenvectors can be computed (and stored) in a preprocessing phase. In general, the eigendecomposition time depends on the map sizes but the it is very fast compared with the time needed to perform the alignments. Nonetheless, for sake of clarity, we include here the time necessary to perform the eigendecomposition of the maps in our datasets described in Section 3.1. The eigendecomposition of the Sokol dataset requires <1 ms in total. The Lancia, Skolnick, Fischer and Proteus300 datasets require approximatively 3, 4, 14 and 55 s, respectively. The eigendecomposition of the most large map in our datasets (534 residues) requires 2.3 s.

4 RESULTS AND DISCUSSION

4.1 Comparison with exact/heuristic CMO methods

In the first test, the Al-Eigen algorithm is compared on the Skolnick set with two exact IP-based methods, LAGR (Caprara and Lancia, 2002) and A_Purva (Andonov *et al.*, 2008), and with the heuristic MSVNS (Pelta *et al.*, 2008). In Table 2, the total number of recovered overlapping contacts (Total overlap) and the required computational time (Time) are listed. For the sake of comparison, we use MSVNS version 3 (as suggested by the authors) with number of restarts equal to 5, 10, 30, 50, 70. All the runs are on the same contact maps with threshold value set to 7.5 Å.

The implementations of LAGR and A_purva are not publicly available and we use the data previously published as Supplementary Material (Andonov *et al.*, 2008). Another exact method, CMOS (Xie and Sahinidis, 2007), available online as a web server, has limitations on the size of the submitted problems and it was not included in our benchmark.

As previously described (Andonov *et al.*, 2008), the computational time of the exact algorithms LAGR and A_purva has been limited to a maximum of 30 min per pair of contact maps. Even under this constraint, the two exact methods require a large computational time (from 7 to 13 days) to compute all the 780 alignments of the Skolnick set. It has been reported (Andonov *et al.*, 2008) that A_purva in this test returns an upper bound to the total overlap equal to 218 316 contacts and a total lower bound of 216 372 contacts (listed in Table 2). This indicates that the quality of the overlaps returned by A_purva within the time limitation is very good, since it is quite

Table 2. Comparison with exact methods on the Skolnick dataset

Method[a]	Total overlap[b]	% wrt A_purva[c]	Time[d]
A_purva	216 372	100%	7 day
LAGR	210 395	97.2%	13 day
MSVNS v3 r70	199 270	92.1%	15 h
MSVNS v3 r50	197 777	91.4%	10 h
MSVNS v3 r30	195 007	90.1%	6 h
MSVNS v3 r10	186 776	86.3%	2 h
MSVNS v3 r5	178 757	82.6%	1 h
Al-Eigen$_{14}$	198 124	91.6%	18 h
Al-Eigen$_{13}$	197 386	91.2%	9 h
Al-Eigen$_{12}$	196 512	90.8%	4 h 30 min
Al-Eigen$_{11}$	195 640	90.4%	2 h
Al-Eigen$_{10}$	194 654	90.0%	1 h
Al-Eigen$_{9}$	193 571	89.5%	25 min
Al-Eigen$_{8}$	192 177	88.8%	12 min
Al-Eigen$_{7}$	190 923	88.2%	6 min

[a]A_purva (Andonov *et al*., 2008) and LAGR (Caprara and Lancia, 2002) are exact CMO methods. MSVNS (Pelta *et al*., 2008) and Al-Eigen$_k$ (our method) are heuristic. MSVN has been run with 5, 10, 30, 50, 70 restarts. Al-Eigen$_k$ has been run by taking $k=7,\ldots,14$ principal eigenvectors.
[b]Total number of recovered overlapping contacts.
[c]Percentage of total overlapping contacts recovered with respect to the best performing method (A_purva).
[d]Running time needed to compute the entire set of 780 pairwise alignments in the Skolnick set.

close to the optimum solution (the distance between the total upper bound and the total lower bound of A_purva is very short). With A_purva, the solution is in fact <1% distant from the optimal one. The other exact algorithm LAGR similarly returns a solution <4% distant from the optimal one. The total overlap returned by A_purva can, therefore, be adopted as the best approximation of the optimal solution for the Skolnick set. Therefore, all the other solutions listed in Table 1 are normalized to that of A_purva (%wrt A_purva).

In Table 3, the performance of Al-Eigen is compared with the two heuristic methods MSVNS and BIMAL (Jain and Obermayer, 2009), which is not publicly available, with respect to the results published in Jain and Obermayer (2009). The maps for Lancia and Sokol datasets include also some contacts between contiguous residues which are not taken into account by Al-Eigen (thus, its performance is slightly lowered). It was not possible to compare the performance of Al-Eigen with BIMAL on the Fischer dataset, since the original maps used in Jain and Obermayer (2009) are not available. In Table 4, the performance on the Fischer dataset (7.5 Å contact maps) is compared with MSVNS only. The running times of BIMAL and MSVNS in Table 3 are reduced with respect to that published in Jain and Obermayer (2009) in order to take into account the different clock frequencies: 1.75 GHz in Jain and Obermayer (2009) and 2.8 GHz for our machine, respectively. In detail, the running times of BIMAL and MSVNS have been scaled down by a factor equal to 0.8, which is the minimum ratio we obtained by comparing the running time of MSVN on our machine (on the same maps used for the tests in Table 3) and the running time reported in Jain and Obermayer (2009).

Summing up, heuristic algorithms, when compared with exact algorithms, can provide good solutions in a much smaller computational time. Interestingly enough, when the elapsed time

Table 3. Comparison with heuristic methods on the Skolnick, Lancia and Sokol datasets

Method[a]	Skolnick[b]	Time[c]	Lancia[b]	Time[c]	Sokol[b]	Time[c]
MSVNS v1	155 308	40 min	945 813	2 h 30 min	833	5 s
MSVNS v2	179 382	50 min	1 010 559	3 h	885	4 s
MSVNS v3	185 753	1 h	1 038 186	3 h 20 min	862	5 s
BIMAL 1	187 049	1 min	818 357	4 min	740	0.1 s
BIMAL 2	189 498	2 min	867 691	8 min	793	0.1 s
BIMAL r1	192 961	20 min	936 996	1 h	814	1 s
BIMAL r2	194 499	45 min	984 481	2 h	828	3 s
Al-Eigen$_7$	187 003	6 min	731 865	40 min	679	2 s
Al-Eigen$_8$	188 246	12 min	737 718	1 h	694	5 s
Al-Eigen$_9$	189 308	25 min	744 095	2 h	702	7 s
Al-Eigen$_{10}$	190 324	1 h	750 888	4 h	706	13 s

[a]The results for MSVNS and BIMAL have been taken from Jain and Obermayer (2009). MSVN has been run with 10 restarts.
[b]Total number of recovered overlapping contacts. The total overlap includes also the overlap of a map with itself.
[c]Running time needed to compute the entire set of pairwise alignments.

Table 4. Comparison with MSVNS method on the Fischer dataset

Method	Fischer[a]	Time[b]
MSVNS v1 r10	445 887	4 h
MSVNS v2 r10	465 892	4 h
MSVNS v3 r10	476 469	6 h
Al-Eigen$_7$	476 249	20 min
Al-Eigen$_8$	479 745	40 min
Al-Eigen$_9$	482 898	1 h 20 min
Al-Eigen$_{10}$	485 388	3 h

[a]Total number of recovered overlapping contacts on the Fischer dataset. The total overlap includes also the overlap of a map with itself.
[b]Running time needed to compute the entire set of 2346 pairwise alignments.

is 6 min, Al-Eigen$_7$ returns 88.2% of the best overlap (Table 2). The comparison in Tables 3 and 4 shows that Al-Eigen has worst performance than other heuristic methods, MSVNS and BIMAL, on maps containing a small number of contacts. On the other hand, on maps containing a higher number of contacts the performance of Al-Eigen improves both in terms of quality of the overlap (compare the results on the Lancia/Sokol datasets with those on the Skolnick dataset) and computational time (compare the computational times of MSVNS and Al-Eigen on the Skolinck dataset in Tables 2 and 3).

4.2 Comparison with structural alignment methods

In this section, we evaluate the accuracy of our algorithm as a *classifier* (the ability to recognize protein structural similarities at the SCOP family/s.family/fold level) on the Proteus300 and Fischer dataset. These two datasets are more computationally demanding than the Skolnick set (they require 44 850 and 2278 alignments, respectively) and contain non-trivial superfamilies and folds.

We compare the effectiveness of contact map alignment methods, including MSVNS, for the structural classification task with three structural alignment methods such as CE (Shindyalov and Bourne, 1998), TM-align (Zhang and Skolnick, 2005) and DaliLite (Holm and Park, 2000). On the Proteus300 dataset, we consider also

Table 5. Scoring the family/s.family/fold recognition and binary classification performance on the Proteus300 dataset

Method[a]	Contact th.[b]	Fam. rec.[c]	S.Fam. rec.[c]	Fold rec.[c]	AUC Fam.[d]	AUC S.Fam.[d]	AUC Fold[d]	Time[e]
BLAST	N/A	274/300	274/300	275/300	0.82	0.78	0.71	2 min
CE	N/A	297/300	299/300	**300/300**	0.99	0.98	0.97	40 h
DaliLite	N/A	299/300	**300/300**	**300/300**	0.99	0.97	0.97	9 h 30 min
TM-align	N/A	**300/300**	**300/300**	**300/300**	0.99	0.98	0.98	5 h
A_purva+sse	7.5 Å	**300/300**	**300/300**	**300/300**	0.99	0.97	0.97	23 h
MSVNS.v3 r10	7.5 Å	297/300	298/300	299/300	0.97	0.93	0.93	94 h
Al-Eigen$_7$	7.5 Å	294/300	294/300	296/300	0.98	0.94	0.94	6 h
Al-Eigen$_7$	8 Å	297/300	297/300	298/300	0.98	0.94	0.94	6 h
Al-Eigen$_7$	9 Å	299/300	299/300	**300/300**	0.98	0.94	0.94	6 h
Al-Eigen$_7$	10 Å	**300/300**	**300/300**	**300/300**	0.99	0.94	0.95	6 h
Al-Eigen$_7$	11 Å	**300/300**	**300/300**	**300/300**	0.99	0.95	0.96	6 h
Al-Eigen$_7$	12 Å	**300/300**	**300/300**	**300/300**	0.99	0.95	0.96	6 h
Al-Eigen$_7$	13 Å	**300/300**	**300/300**	**300/300**	0.99	0.95	0.96	6 h

[a]BLAST (Altschul *et al*., 1997) is a local sequence alignment tool. CE (Shindyalov and Bourne, 1998), DaliLite (Holm and Park, 2000) and TM-align (Zhang and Skolnick, 2005) are structural alignment methods. A_purva+sse (Andonov *et al*., 2008) is an exact CMO method that uses secondary structure information. MSVNS (Pelta *et al*., 2008) and Al-Eigen$_7$ (our method) are the only *pure* CMO methods.
[b]Contact map threshold used for CMO methods. N/A: not applicable.
[c]Family/S.Family/Fold recognition results on the 300 proteins in the Proteus300 dataset. When CE returns more than one alignment for a pair, the one with the best *Z*-score is chosen. The TM-scores are normalized with respect to the shortest sequence. The best results are highlighted with bold fonts.
[d]AUC values of the ROC curves at the SCOP Family/S.Family/Fold level.
[e]Running time needed to compute the entire set of 44 850 pairwise alignments in the Proteus300 dataset.

the performance of A_purva+sse, a variant of A_purva that encodes secondary structure constraints (two residues can be aligned only if they belong to the same secondary structure class). The results shown for A_purva+sse have been taken from the Supplementary Material of Andonov *et al*. (2008). The scoring function for the CMO methods is provided by Equation (2). CE, DaliLite and TM-align have their own scoring functions: *Z*-score (for CE and DaliLite) and TM-score (for TM-align). The only *pure* CMO methods considered for these tests are MSVNS and our algorithm. We use different contact map thresholds to test the performance of our method. Due to the demanding computational time required by MSVNS on high-threshold contact maps, it is not possible to test it for thresholds >7.5 Å. In order to highlight the level of difficulty of our benchmarks, we also include the performance of BLAST (Altschul *et al*., 1997), so that it is easier to detect the amount of information contained in the dataset at the protein sequence level. The performance of BLAST is evaluated with respect to its bit-score. In detail, we perform the following two tests.

4.2.1 Structural class recognition We check the ability to detect the correct protein family/super family/fold with a leave-one-out test (Tables 5 and 6). For every query-protein, we select the model-protein in the set that obtains the best similarity score and we measure the fraction of query-proteins for which the chosen model-protein belongs to the same family/s.family/fold. The results on the Proteus300 and Fischer datasets are shown in Tables 5 and 6, respectively. On the Fischer dataset we test the performance only at the fold level since almost 1/3 of the proteins in the dataset (22) belong to a unique fold. In particular, the fold recognition test is performed only on the 46 proteins that belong to a non-unique fold in the dataset.

On the Proteus300 dataset, all the methods have similar performance on the task of recognition at the family/s.family/fold level. Notably, most of the errors reported in Table 5 are related to query-proteins that belong to non-trivial folds or non-trivial super families (see the Supplementary Material for detailed results). For example, as can we notice in Table 5, CE and DaliLite for 3 and 1 queries, respectively, can detect the correct superfamily

Table 6. Scoring the fold recognition and binary classification performance on the Fischer dataset

Method[a]	Contact th[b]	Fold rec[c]	AUC Fold[d]	Time[e]
BLAST	N/A	19/46	0.67	6 s
CE	N/A	40/46	0.98	1 h 20 min
DaliLite	N/A	**45/46**	0.99	30 min
TM-align	N/A	39/46	0.99	10 min
MSVNS v3 r10	7.5 Å	23/46	0.90	6 h
Al-Eigen$_7$	7.5 Å	24/46	0.88	20 min
Al-Eigen$_7$	8 Å	23/46	0.90	20 min
Al-Eigen$_7$	9 Å	22/46	0.89	20 min
Al-Eigen$_7$	10 Å	29/46	0.91	20 min
Al-Eigen$_7$	11 Å	30/46	0.93	20 min
Al-Eigen$_7$	12 Å	33/46	0.94	20 min
Al-Eigen$_7$	13 Å	32/46	0.93	20 min

[a]BLAST (Altschul *et al*., 1997) is a local sequence alignment tool. CE (Shindyalov and Bourne, 1998), DaliLite (Holm and Park, 2000) and TM-align (Zhang and Skolnick, 2005) are structural alignment methods. MSVNS (Pelta *et al*., 2008) and Al-Eigen$_7$ (our method) are CMO methods.
[b]Contact map threshold used for CMO methods. N/A: not applicable
[c]Fold recognition results on Fischer dataset. Among the 68 proteins in the dataset, 22 belong to a unique fold. Thus, the fold recognition performance are evaluated only on the remaining 46 proteins. When CE returns more than one alignment for a pair, the one with the best *Z*-score is chosen. The TM-scores are normalized with respect to the shortest sequence. The best results are highlighted with bold fonts.
[d]AUC values of the ROC curves at the SCOP Fold level.
[e]Running time needed to compute the entire set of 2278 pairwise alignments in the Fischer dataset.

but not the correct family. MSVNS and Al-Eigen (on 7.5 Å and 8 Å maps) in few cases fail to detect also the correct fold. It is worth noticing how the family recognition performance of our method varies at increasing values of the contact map threshold: all queries are correctly recognized only at thresholds $\geq$10 Å. The good performance of the CMO method A_purva+sse on 7.5 Å contact maps can be related to the use of secondary structure constraints, which can be of help in discriminating structure similarity, at least at the SCOP class level.

As shown in Table 6, on the Fischer dataset, structural alignment methods have better performance than CMO methods. In this case none of the methods is able to obtain full accuracy. The differences with respect to the results obtained on the Proteus300 dataset are justified by the low level of similarity between the targets in the Fischer dataset, as highlighted also by the performance of BLAST. Most of the errors reported in Table 6 are related to queries that have a representative only in the same superfamily but not in the same family (Supplementary Material). In particular, the only query misclassified by DaliLite has only one representative in the same fold. Except for this case, the other four queries with representatives only in the same fold class are classified correctly by all the other methods (except BLAST). It is worth noticing that, also in this case, the performance in fold recognition of our method increases with increasing values of the contact threshold.

In terms of computational time, in both tests, TM-align and Al-Eigen are the fastest.

4.2.2 Binary classification We measure the area under curve (AUC) values of the receiver operating characteristic (ROC) curves, when the alignment scores are used as a binary classifier system at the SCOP family/s.family/fold level. As in the previous test, the performance on the Fischer dataset are evaluated only at the fold level (Tables 5 and 6).

The comparison based on AUC values shows that all the methods perform quite well as binary classifiers on both Proteus300 and Fischer datasets. In particular, the performance gap between structural alignment and CMO methods on the Fischer dataset is much smaller in terms of AUC values. This result indicates that, even if almost half of the queries are misclassified by CMO comparison, the most similar structures in the dataset are top-ranked. This test also confirms that high-threshold contact maps are more suitable for structure similarity detection.

4.3 Error tolerance test

In this section, we evaluate the error tolerance of Al-Eigen and MSVNS on the Proteus300 and Fischer datasets.

In this preliminary test, we adopt two different random error models. In particular, we apply some random perturbation to the original contact maps and evaluate the performance in terms of structural class recognition accuracy and AUC values, as performed in Section 4.2. In this case, we compare a perturbed map with all the other non-perturbed maps in the dataset, except itself. For the sake of comparison, we use only 7.5 Å contact maps. This threshold is also very close to the standard distance (8 Å) adopted for contact prediction (Ezkurdia *et al*., 2009). In detail, we consider the following two models for random error.

- **Model 1:** an amount of $\%x$ errors means that exactly n random contacts are set to 0 and exactly n random non-contacts are set to 1, where $n = \text{round}(\frac{x \cdot C}{100})$, and C is the number of contacts in the map. Only contacts between non-consecutive residues are considered.
- **Model 2:** an amount of $\%x$ errors means that exactly n random entries of the map are flipped, where $n = \text{round}(\frac{x}{100}\frac{(L-1)(L-2)}{2})$ and L is the length of the protein. Also in this case only the entries of the map related to non-consecutive residues are considered.

Error model 1 preserves the total number of contacts in the map. On the contrary, model 2 can eventually introduce more false positive contacts than the original number of contacts in the map. We test error model 1 with percentage of errors equal to 70, 80 and 90% and error model 2 with 10, 20 and 30%. In the latter case, the perturbed contact maps contain on the average 4, 8 and 11 times more contacts than the original ones, respectively. The results obtained on the Proteus 300 and Fischer datasets are shown in Tables 7 and 8, respectively.

The AUC values in Tables 7 and 8 show that the CMO comparison approach has some discriminative power in distinguishing similar structures also in presence of highly noisy contact maps. In particular, the approach seems to be more robust when the total number of contacts in noisy maps does not differ too much from the native amount of contacts. In terms of recognition accuracy, by comparing the results obtained on the two different datasets it is evident that, in presence of noise (not surprisingly) the approach is much robust in detecting similarities at the family level more than at the fold level. In particular, on the Proteus300 dataset, the performance of Al-Eigen are almost unaffected also for 70% errors of type 1 (compare Tables 5 and 7).

The better performance of Al-Eigen with respect to those of MSVNS can be related to a higher tolerance of noise due to the eigendecomposition. In particular, we tested (data not shown) how the two algorithms align a perturbed map with its native version. In almost all cases the alignment computed by Al-Eigen is the identity, i.e. the i-th residue in the perturbed map is aligned with the i- residue in the original map. In detail, the alignments of Al-Eigen, on the average, have $>$94% of identity (the identity abruptly drops from 94% at 90% error to 3% at 100% error, with respect to model 1). On the contrary, the same test on MSVN reports an average identity $<$10%.

5 CONCLUSIONS

In this article, we described a heuristic algorithm suited to address the CMO problem. Our algorithm computes an overlap of two contact maps by performing a global alignment based on few eigenvectors. The approach is effective since contact maps can be well approximated by just a fraction of their eigenvectors. Our algorithm is reasonably simple and, by design, its computing time does not depend on the number of contacts in the map and then on the contact threshold.

There are two main differences between our approach and the other methods developed so far for the CMO problem: (i) our algorithm is completely heuristic and it has no way to detect if the overlap found in some point of the computation is the best possible. In contrast, exact IP-based methods can stop the computation when the lower and upper bounds to the optimal solution coincide. (ii) The computing time of our algorithm depends uniquely on the protein

Table 7. Error tolerance test on the Proteus300 dataset (7.5 Å contact maps)

Method	Error model[a]	%Err[b]	Fam. rec.[c]	S.Fam. rec[c]	Fold rec.[c]	AUC Fam.[d]	AUC S.Fam.[d]	AUC Fold[d]
MSVNS.v3 r10	1	70	114/300	119/300	138/300	0.73	0.71	0.73
Al-Eigen$_7$			283/300	285/300	289/300	0.91	0.86	0.86
MSVNS.v3 r10		80	22/300	27/300	38/300	0.66	0.65	0.68
Al-Eigen$_7$			253/300	256/300	268/300	0.85	0.81	0.82
MSVNS.v3 r10		90	8/300	9/300	22/300	0.60	0.60	0.64
Al-Eigen$_7$			53/300	54/300	64/300	0.71	0.69	0.72
MSVNS.v3 r10	2	10	229/300	231/300	240/300	0.68	0.67	0.70
Al-Eigen$_7$			291/300	292/300	293/300	0.80	0.77	0.79
MSVNS.v3 r10		20	181/300	185/300	192/300	0.63	0.62	0.66
Al-Eigen$_7$			273/300	277/300	277/300	0.73	0.70	0.73
MSVNS.v3 r10		30	86/300	92/300	107/300	0.57	0.57	0.61
Al-Eigen$_7$			241/300	243/300	245/300	0.67	0.65	0.69

[a]Error models described in Section 4.3.
[b]Percentage of random error introduced in the maps.
[c] Family/S.Family/Fold recognition results on the 300 proteins in the Proteus300 dataset. Every perturbed map is compared against every non-perturbed map in the dataset except itself.
[d]AUC values of the ROC curves at the SCOP Family/S.Family/Fold level.

Table 8. Error tolerance test on the Fischer dataset (7.5 Å contact maps)

Method	Error model[a]	%Err[b]	Fold rec.[c]	AUC Fold[d]
MSVNS.v3 r10	1	70	6/46	0.74
Al-Eigen$_7$			16/46	0.81
MSVNS.v3 r10		80	3/46	0.74
Al-Eigen$_7$			11/46	0.78
MSVNS.v3 r10		90	2/46	0.71
Al-Eigen$_7$			6/46	0.73
MSVNS.v3 r10	2	10	16/46	0.78
Al-Eigen$_7$			19/46	0.81
MSVNS.v3 r10		20	15/46	0.72
Al-Eigen$_7$			19/46	0.76
MSVNS.v3 r10		30	7/46	0.67
Al-Eigen$_7$			12/46	0.73

[a]Error models described in Section 4.3.
[b]Percentage of random error introduced in the maps.
[c] Fold recognition results on Fischer dataset. Among the 68 proteins in the dataset, 22 belong to a unique fold. Thus, the fold recognition performance are evaluated only on the remaining 46 proteins. Every perturbed map is compared against every non-perturbed map in the dataset except itself.
[d]AUC values of the ROC curves at the SCOP Fold level.

length and on the number of eigenvectors considered and it is not affected by the threshold value of the contact map. Differently, for all the other methods developed so far (exact, approximate and heuristic), the computing time increases with increasing number of contacts in the map (namely, with increasing threshold value of the contact map): more contacts result into more constraints to be taken into account and in a longer running time.

We experimentally validated the performance of our algorithm by comparison with exact CMO methods (LAGR and A_purva), with heuristic CMO approaches (MSVNS and BIMAL) and with three structural alignment methods (CE, TM-align and DaliLite). Our algorithm is fast, it has good performance in terms of quality of the overlap when compared with exact/heuristic CMO methods and it is also competitive with widely used structural alignment methods in the task of protein structure comparison. In our tests, we found that contact maps computed at thresholds values $\geq$10 Å are more informative of the protein structures than those at lower thresholds. The computational time of our method is independent of the threshold value of the contact map and this makes it more suitable than other CMO methods for large-scale detection of protein structure similarities.

Finally, in this article for the first time, to the best of our knowledge, we tested the effect of random noise on CMO alignments. According to these tests our algorithm is quite robust to noise and performs well in the task of structural similarity detection also with highly perturbed contact maps, at least at the SCOP family level.

Conflict of Interest: none declared.

REFERENCES

Agarwal,P.K. *et al.* (2007) Fast molecular shape matching using contact maps. *J. Comput. Biol.*, **14**, 131–143.

Altschul,S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Andonov,R. *et al.* (2008) An efficient Lagrangian relaxation for the contact map overlap problem. *Lect. Notes Bioinform.*, **5251**, 162–173. Supplementary material available at http://www.irisa.fr/symbiose/old/softwares/resources/proteus300

Andreeva,A. *et al.* (2008) Data growth and its impact on the SCOP database: new developments. *Nucleic Acids Res.*, **36**, 419–425.

Andrews,H.C. and Patterson,C.L. (1976) Singular value decomposition (SVD) image coding. *IEEE Trans. Commun.*, **24**, 425–432.

Bartoli,L. *et al.* (2007) The effect of backbone on the small-world properties of protein contact maps. *Phys. Biol.*, **4**, 1–5.

Caprara,A. and Lancia,G. (2002) Structural alignment of large-size proteins via Lagrangian relaxation. In *Proceedings of the Annual International Conference on Computational Molecular Biology (RECOMB 2002)*, Washington, DC, USA, pp. 100–108.

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

Duarte,J.M. *et al.* (2010) Optimal contact definition for reconstruction of contact maps. *BMC Bioinformatics*, **11**, 283.

Ezkurdia,I. *et al.* (2009) Assessment of domain boundary predictions and the prediction of intramolecular contacts in CASP8. *Proteins*, **77**, 96–209.

Fischer,D. *et al.* (1996) Assessing the performance of fold recognition methods by means of a comprehensive benchmark. In *Proceedings of the Pacific Symposium on Biocomputing 1996*, The Big Island of Hawaii, USA, pp. 300–318.

Godzik,A. (1996) The structural alignment between two proteins: is there a unique answer? *Protein Sci.*, **5**, 1325–1338.

Godzik,A. *et al.* (1992) Topology fingerprint approach to the inverse protein folding problem. *J. Mol. Biol.*, **227**, 227–238.

Goldman,D. *et al.* (1999) Algorithmic aspects of protein structure similarity. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, New York City, NY, USA, pp. 512–521.

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

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

Jain,B.J. and Lappe,M. (2007) Joining softassign and dynamic programming for the contact map overlap problem. *Lect. Notes Bioinform.*, **4414**, 410–423.

Jain,B.J. and Obermayer,K (2009) BIMAL: Bipartite matching alignment for the contact map overlap problem. In *Proceedings of the International Joint Conference on Neural Networks*, Atlanta, Georgia, USA, pp. 1394–1400.

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

Krasnogor,N. and Pelta, D.A. (2004) Measuring the similarity of protein structures by means of the universal similarity metric. *Bioinformatics*, **20**, 1015–1021.

Lancia,G. *et al.* (2001) 101 Optimal PDB structure alignments: a branch-and-cut algorithm for the maximum contact map overlap problem. In *Proceedings of the Annual International Conference on Computational Molecular Biology, RECOMB 2001*, Montreal, Quebéc, Canada, pp. 193–202.

Luo,B. and Hancock,E. (2001) Structural graph matching using the EM algorithm and singular value decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.*, **23**, 1120–1136.

Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.

Oakley,M.T. *et al.* (2008) Search strategies in structural bioinformatics. *Curr. Protein Pept. Sci.*, **9**, 260–274.

Pelta,D.A. *et al.* (2008) A simple and fast heuristic for protein structure comparison. *BMC Bioinformatics*, **9**, 161.

Porto,M. *et al.* (2004) Reconstruction of protein structures from a vectorial representation. *Phys. Rev. Lett.*, **92**, 218101.

Rahmati,S. and Glasgow,J.I. (2009) Comparing protein contact maps via Universal Similarity Metric: an improvement in the noise-tolerance. *Int. J. Comput. Biol. Drug Des.*, **2**, 149–167.

Robles-Kelly,A. and Hancock A.R. (2002) String edit distance, random walks and graph matching. *Lect. Notes Comput. Sci.*, **2396**, 104–112.

Sadreyev,R.I. *et al.* (2009) Discrete-continuous duality of protein structure space. *Curr. Opin. Struct. Biol.*, **19**, 321–328.

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.

Singh,R. *et al.* (2007) Pairwise global alignment of protein interaction networks by matching neighborhood topology. *Lect. Notes Comput. Sci.*, **4453**, 16–31.

Strang,G. (2003) *Introduction to Linear Algebra*. Wellesley-Cambridge Press, Wellesley, MA.

Strickland,D.M. *et al.* (2005) Optimal protein structure alignment using maximum cliques. *Oper. Res.*, **53**, 389–402.

Umeyama,S. (1988) Eigendecomposition approach to weighted graph matching problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, **10**, 695–703.

Vassura,M. *et al.* (2008) Reconstruction of 3D structures from protein contact maps. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **5**, 357–366.

Vassura,M. *et al.* (2008) FT-COMAR: fault tolerant three-dimensional structure reconstruction from protein contact maps. *Bioinformatics*, **24**, 1313–1315.

Xie,W. and Sahinidis,N.V. (2007) A reduction-based exact algorithm for the contact map overlap problem. *J. Comput. Biol.*, **14**, 637–654.

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

Zhang,Y. and Skolnick,J. (2005) TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.*, **22**, 2302–2309.

Zhao,G. *et al.* (2007) Using eigen-decomposition method for weighted graph matching. *Lect. Notes Comput. Sci.*, **4453**, 1283–1294.