BIOINFORMATICS ORIGINAL PAPER

Vol. 30 no. 9 2014, pages 1259–1265
doi:10.1093/bioinformatics/btu020

Structural bioinformatics

Advance Access publication January 17, 2014

GR-Align: fast and flexible alignment of protein 3D structures using graphlet degree similarity

Noël Malod-Dognin* and Nataša Pržulj

Department of Computing, Imperial College London, SW7 2AZ, UK

Associate Editor: Burkhard Rost

ABSTRACT

Motivation: Protein structure alignment is key for transferring information from well-studied proteins to less studied ones. Structural alignment identifies the most precise mapping of equivalent residues, as structures are more conserved during evolution than sequences. Among the methods for aligning protein structures, maximum Contact Map Overlap (CMO) has received sustained attention during the past decade. Yet, known algorithms exhibit modest performance and are not applicable for large-scale comparison.

Results: Graphlets are small induced subgraphs that are used to design sensitive topological similarity measures between nodes and networks. By generalizing graphlets to ordered graphs, we introduce GR-Align, a CMO heuristic that is suited for database searches. On the Proteus_300 set (44 850 protein domain pairs), GR-Align is several orders of magnitude faster than the state-of-the-art CMO solvers Apurva, MSVNS and AlEigen7, and its similarity score is in better agreement with the structural classification of proteins. On a large-scale experiment on the Gold-standard benchmark dataset (3 207 270 protein domain pairs), GR-Align is several orders of magnitude faster than the state-of-the-art protein structure comparison tools TM-Align, DaliLite, MATT and Yakusa, while achieving similar classification performances. Finally, we illustrate the difference between GR-Align's flexible alignments and the traditional ones by querying a flexible protein in the Astral-40 database (11 154 protein domains). In this experiment, GR-Align's top scoring alignments are not only in better agreement with structural classification of proteins, but also that they allow transferring more information across proteins.

Availability and implementation: GR-Align is coded in C++. software and supplementary material are available at: http://bio-nets.doc.ic.ac.uk/home/software/gralign/.

Contact: n.malod-dognin@imperial.ac.uk

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

Received on June 5, 2013; revised on December 16, 2013; accepted on January 10, 2014

1 INTRODUCTION

1.1 Aligning protein structures

Protein alignment is key for answering biological questions that involve the transfer of information from well-studied proteins to less studied ones. Earlier methods align proteins according to their sequences (Altschul *et al*., 1990; Needleman and Wunsch, 1970; Smith and Waterman, 1981). Because protein 3D structures are more conserved during evolution than sequences, structural alignments allow the most precise mappings of equivalent residues, especially when the proteins have low sequence identities. This is notably important for (i) detecting and investigating structural motifs (i.e.small rigid substructures) and functional sites (i.e. substructures of enzymes where the catalytic activities take place) and (ii) measuring the similarity between proteins and bringing them in evolutionary relationship, e.g. by classification.

Many protein structure comparison methods have been proposed and there is no consensus which one is the best (Godzik, 1996; Hasegawa and Holm, 2009), as the alignments that maximize different scoring schemes can differ considerably (Mayr *et al*., 2007). The alignment methods can be divided into two categories. The rigid-body approaches consider the proteins as rigid objects and aim to find alignments that have the maximum number of mapped residues and the minimum deviations between the mapped structures [often expressed in terms of Root Mean Square Deviation (*RMSD*)]. These two objectives are contradictory, as *RMSD* has a trivial minimum when no residue is aligned and then tends to increase with the number of mapped residues. The rigid-body approaches mainly differ in how they combine these two objectives (Gibrat *et al*., 1996; Holm and Sander, 1993; Malod-Dognin *et al*., 2010; Zhang and Skolnick, 2005). However, proteins are flexible molecules and can undergo large conformational changes that are not captured by the rigid-body approaches. Flexible alignment methods overgo this limitation by either allowing twists between rigidly aligned fragments (Menke *et al*., 2008; Ye and Godzik, 2003) or by only maximizing local (in terms of Euclidean distance) similarities (Godzik and Skolnick, 1994; Wohlers *et al*., 2009).

In this article, we focus on maximum Contact Map Overlap (CMO), a flexible protein structure alignment method defined by Godzik and Skolnick, 1994. CMO is interesting because it is translation/rotation invariant, and its corresponding scoring scheme is in good agreement with the structural classification of proteins (SCOP, Murzin *et al*., 1995).

1.2 Maximum contact map overlap

In CMO, the 3D structure of a protein is represented by its contact map. A *contact map* is an *ordered graph*, $CM = (V, E)$, where nodes, V, and edges, E, are defined as follows. Each node in V represents an amino acid of a protein. Nodes in a contact map are labeled with the sequence position of the amino acids from which they originated. This labeling leads to a *strict total ordering* of the nodes: for each two distinct nodes u and v, either

*To whom correspondence should be addressed.

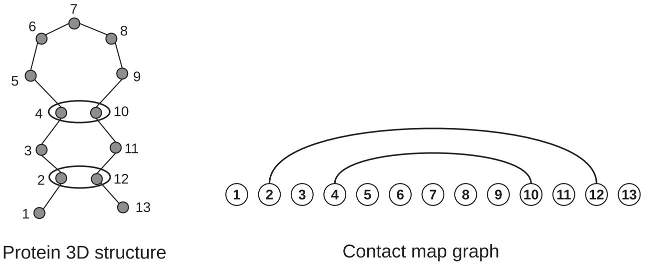

Fig. 1. A protein structure and its corresponding contact map. Left: schematic representation of a protein backbone. Amino acid 2 is in contact with 12 and 4 is in contact with 10 (the corresponding Euclidean distances are smaller than a given threshold). Right: in the corresponding contact map graph, edges connect node 2 with 12 and 4 with 10

$u<v$ if u is before v in the protein sequence or $u>v$ otherwise. Two nodes u and v are connected by an edge $(u,v)\in E$ if and only if the Euclidean distance between the α-carbons of the corresponding amino acids is less than a given threshold ϵ. Such amino acids are said to be in contact, even if they do not share hydrogen bonds. This is illustrated in Figure 1.

Between two contact maps $CM_1=(V_1,E_1)$ and $CM_2=(V_2,E_2)$, the CMO's objective is to find a one-to-one mapping between a subset of V_1 and a subset of V_2 that both preserves the total ordering of the mapped residues and maximizes the number of common edges (NCE), which is the number of times two nodes connected by an edge in E_1 are mapped to two nodes connected by an edge in E_2. Such a mapping is represented by a sequence of mapped pairs $(a_1\leftrightarrow b_1, a_2\leftrightarrow b_2, \ldots, a_n\leftrightarrow b_n)$, where the i^{th} mapped pair $a_i\leftrightarrow b_i$ means that node a_i from the first contact map is mapped to node b_i from the second contact map. The similarity between two proteins is then measured by the edge correctness of their alignment (EC), which is the ratio between NCE and their total number of contact edges:

$$EC(CM_1, CM_2) = 2\times NCE/(|E_1|+|E_2|) \qquad (1)$$

An example of an optimal CMO alignment is presented in Figure 2.

CMO, which is a special case of the maximum common edge-induced subgraph problem, is both NP-hard (Goldman *et al.*, 1999) and hard to approximate (Crescenzi and Kann, 1998). Thus, researches on CMO took three directions. First, the exact algorithms that aim at finding the optimal solution at the expense of high running times, with best approaches relying on integer programming formulations of CMO, coupled with branch and bound strategies (Andonov *et al.*, 2008, 2011; Caprara and Lancia, 2002; Caprara *et al.*, 2004; Carr *et al.*, 2000; Strickland *et al.*, 2005; Xie and Sahinidis, 2007). Second, the heuristic algorithms that aim at smaller running times at the expense of the quality of the solutions, with best approaches relying on eigen-decomposition of the contact maps (Di Lena *et al.*, 2010; Jain and Lappe, 2007; Pelta *et al.*, 2008; Shibberu and Holder, 2011). Finally, the polynomial-time approximation algorithms that aim at finding a solution within a given error threshold τ, with time complexity that is polynomial with respect to the input data length, but exponential with respect to τ (Agarwal *et al.*, 2007; Xu *et al.*, 2007).

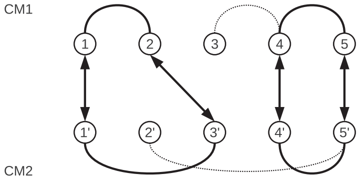

Fig. 2. An optimal CMO alignment. The arrows show the order-preserving mapping between the nodes of the two contact maps ($1\leftrightarrow 1', 2\leftrightarrow 3', 4\leftrightarrow 4', 5\leftrightarrow 5'$). This alignment yields two common contacts (the bold edges), which is optimal. The corresponding edge-correctness is 2/3

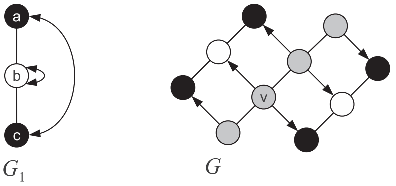

Fig. 3. Graphlets and automorphism orbits. Left: graphlet G_1 contains three nodes, a, b and c. The only automorphism that is not the identity is the one that maps node a to c and b to itself (represented by the arrows). It defines two automorphism orbits: {a,c} (in black) and {b} (in white). Right: graphlet G_1 can be mapped onto graph G (in gray) in two topologically distinct ways: in the first case, node v is mapped to orbit {a,c} (black nodes) and plays in G_1 the role of a node having degree equal to 1, whereas in the second case, v is mapped to orbit {b} (white node) and plays in G_1 the role of a node having degree equal to 2

CMO has desirable properties. First, earlier approaches suggest that CMOs are in good agreement with structural classifications of proteins (Andonov *et al.*, 2011; Caprara *et al.*, 2004). Second, contact map-based approaches are known to be resilient to noise (Di Lena *et al.*, 2010). However, none of the proposed algorithms is applicable for large-scale studies.

1.3 Graphlets and graphlet degrees

Graphlets are small, connected, non-isomorphic and induced subgraphs of a larger graph $G=(V,E)$ having $\mathrm{n}\geq 2$ nodes (Pržulj *et al.*, 2004). Within each graphlet, some nodes are topologically identical to each other: such identical nodes are said to belong to the same *automorphism orbit* (see Fig. 3: left) (Pržulj, 2007). The automorphism orbits represent the topologically different ways that a graphlet can touch a node in V (as illustrated in Fig. 3: right).

Graphlets are used to generalize the notion of node degree: the *graphlet degree* of node n, denoted by d_n^i, is the number of times a graphlet touches node n at orbit i (Pržulj, 2007). In System Biology, graphlet degrees are successfully used to designed network distance measures, such as the *graphlet degree distribution agreement*. In *graphlet degree distribution agreement*, using all 2–5 node graphlets and their corresponding 73 automorphism orbits, the degree distribution is extended to 73 *graphlet degree distributions* (the first of the 73 graphlet degree distributions is the degree distribution). The similarity between two networks is measured by the similarity between their 73 graphlet degree distributions (Pržulj, 2007). Graphlet degrees are also the basis of network alignment tools such as Matching-based Integrative GRAph ALigner (MI-GRAAL), where the topological similarity between the to-be-mapped nodes is measured by the similarity between their 73 graphlet degrees (Kuchaiev and Pržulj, 2011).

Graphlet degree-based network alignment methods are known to be robust to noisy interaction/edges (Milenković *et al.*, 2010).

1.4 Our contribution

By extending the notion of graphlet degree to the case of ordered graphs, we introduce GR-Align, a novel heuristic for comparing protein structures based on their CMO. First, on a small benchmark from the literature (Proteus_300, 44 850 protein domain pairs), we show that GR-Align is up to 1247 times faster than the state-of-the-art CMO solvers, A_purva (Andonov *et al.*, 2011), MSVNS (Pelta *et al.*, 2008) and AlEigen7 (Di Lena *et al.*, 2010), whereas it produces similarity scores that are in better agreements with the gold standard protein structure classification SCOP. Second, on a large dataset (the Gold-standard benchmark dataset, 3 207 270 domain pairs), we show that GR-Align is particularly suited for database searches. It is up to 1658 times faster than the state-of-the-art protein structure comparison methods, DaliLite (Holm and Park, 2000), TM-Align (Zhang and Skolnick, 2005) and MATT (Menke *et al.*, 2008), whereas it achieves comparable classification performances. Finally, for flexible proteins, we show that GR-Align's alignments are more informative than traditional ones, as the top scoring alignments are in better agreements with the SCOP classification and have better coverage in terms of mapped residues.

2 MATERIALS AND METHODS

2.1 GR-Align principle

Ordered graphlets and orbits. Because the nodes of contact maps are ordered, we first extend the notion of graphlets and graphlet degrees to the case of ordered graphs. *Ordered graphlets* are small, connected, non-isomorphic and induced subgraphs of an ordered graph, whose nodes inherit the strict total ordering of their originating graph. Because the only order-preserving automorphism is the identity, each node in an ordered graphlet defines a different automorphism orbit. The i^{th} *ordered graphlet degree* of node v, denoted by d_v^i, is the number of times an ordered graphlet touches the node v at orbit i. Because we aim at low computational times, we focused on the five 2-and 3-node ordered graphlets that are presented in Figure 4, which define 14 orbits. Each node v of a contact map is thus described by a 14-dimensional vector $(d_v^1, d_v^2, ..., d_v^{14})$. For a given contact map $CM = (V, E)$, computing all 2-and 3-node ordered graphlet degrees is done in a preprocessing step in $\mathrm{O}(|V| \times d^2)$ time, where $|V|$ is the number of nodes and d is the maximum degree of a node. In the general case, d could be at most $|V| - 1$, which leads to a cubic worst time complexity, but in a contact map, the degree of a node is limited by the number of residues that can fit in a sphere of radius ϵ. In practice, using a distance threshold ϵ of 7.5 Å, $d \leq 20$, leads to a linear worst time complexity.

Ordered graphlet degree similarity. Between two nodes u and v, we define the ordered graphlet degree similarity (S) as follows:

$$S(u, v) = \left(\frac{1}{14} \sum_{i=1}^{14} \frac{\min(d_u^i, d_v^i) + 1}{\max(d_u^i, d_v^i) + 1} \right)^2 \tag{2}$$

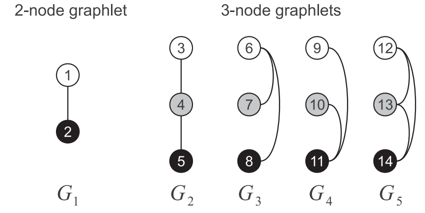

Fig. 4. The five 2-and 3-node ordered graphlets and the corresponding 14 automorphism orbits. The ordering of the graphlet nodes within each graphlet $G_i, i \in \{1, \ldots, 5\}$ is represented by their colors: white nodes < gray nodes < black nodes

which is a similarity score in [0,1], that rewards mapping nodes having similar local topologies.

Alignment algorithm. The alignment between two contact maps having, respectively, n_1 and n_2 nodes is computed using the Needleman–Wunsch dynamic programming algorithm (Needleman and Wunsch, 1970), where the score of mapping two nodes is their ordered graphlet degree similarity, and where the gap cost is zero, as in the original CMO. It corresponds to the following dynamic programming recurrence:

$$\begin{array}{rcl} T[u,0] & = & 0, \\ T[0,v] & = & 0, \\ T[u,v] & = & \max\begin{pmatrix} T[u-1][v-1] + S(u,v), \\ T[u-1][v], T[u][v-1] \end{pmatrix} \end{array} \tag{3}$$

Finding the optimal alignment score requires computing $T[n_1, n_2]$, which is done in $O(n_1 \times n_2)$ time. Retrieving the corresponding alignment is done in $O(n_1 + n_2)$ time.

2.2 Experimental settings

2.2.1 Contact map generation We generated our contact maps in the following way. The atoms' coordinates are taken from the PDB files coming either from the SCOP classification of proteins v1.75b for the Skolnick set and the Proteus_300 set, or from the Astral compendium v1.75b. Only the first entries are used when a structure is described by several models and when an atom is assigned multiple coordinates. Two amino acids u and v are in contact if the Euclidean distance between their α-carbons is smaller than a distance threshold ϵ. As in previous studies, we do not create contact edges between nodes corresponding to residues that are consecutive in the protein sequence. The effect of the distance threshold on the classification performance of GR-Align is discussed in Section 3.2.

2.2.2 Datasets ***The Proteus_300 set***. First introduced in Andonov *et al.* (2008) and later used in CMO-related papers (Di Lena *et al.*, 2010; Shibberu and Holder, 2011), the Proteus_300 set contains 300 domains, with a number of residues varying from 64 to 456. It is a subset of Astral_40 (i.e. SCOP domains having <40% sequence identity), with many pairs of structurally similar domains having <20% sequence identity. The 300 domains are classified by SCOP into 24 folds, 27 superfamilies and 30 families. As shown in Section 3, aligning the $\binom{300}{2} = 44\,850$ pairs of domains from Proteus_300 set challenges most of the exact and heuristic CMO solvers, both in terms of running time and of structural identification of the domains.

The Gold-standard benchmark dataset. Introduced in Csaba *et al.* (2009), the Gold-standard benchmark dataset contains protein domains that are consistently defined in both SCOP v1.75 and CATH v3.2.0 (i.e.with domain overlap >80%) and that share <50% of sequence identity. Then, the Gold-standard benchmark only considers the domain pairs that are consistently classified across the SCOP fold classification and the CATH topology classification. We additionally removed the six multiple chain domains to avoid residue ordering ambiguities, leading to 6747 domains having from 21 to 1147 residues, and to 3 207 270 domain pairs.

The Astral_40 database. Astral_40 (Brenner *et al.*, 2000) is a derivative of SCOP database, so that it only contains protein domains having <40% sequence identity. It contains 11 211 domains, from which we remove the 51 multiple chain domains to avoid residue ordering ambiguities. The 11 160 remaining domains have 21–1419 residues.

2.2.3 Evaluation ***Retrieved edge rate analysis.*** Let NCE_{heur} be the number of common contact edges found in a heuristic alignment, and NCE_{opt} be the optimal number of common contacts. The *retrieved edge rate* $RER = NCE_{heur}/NCE_{opt}$ measures how close the heuristic solution is to the optimal one in terms of the number of common contact edges. Because computing NCE_{opt} is NP-hard, we use the following property of the exact solver A_purva: as it is a branch and bound algorithm, when a computation ends, it returns both a lower-bound (LB) and an upper-bound (UB) on NCE_{opt}. If $LB = UB$, then A_purva's alignment is optimal and $RER = NCE_{heur}/LB$, otherwise RER is approximated by averaging its lower and upper bounds:

$$RER = \frac{1}{2}\left(\frac{NCE_{heur}}{UB} + \frac{NCE_{heur}}{LB}\right) \quad (4)$$

Agreement with reference classifications. Here, we assess the classification performance of our new distance measure against reference classifications.

First, clustering algorithms rely on the early retrieved pairs (i.e. the ones at the shortest distances) to build their clusters. The *nearest neighbor identification rate* (NNI) is a simple way of assessing the quality of the early retrieved pairs, by counting the number of times a protein structure p_1 and its nearest neighbor p_2 are from the same class in the reference classification.

Then, the overall agreement between a protein structure distance and a protein structure classification is evaluated using precision-recall curves analyses (Davis and Goadrich, 2006). Using a given threshold δ on the structural distance between the proteins, four values are computed: the true positives, TP_δ, the number of structure pairs coming from the same class and having pairwise distances smaller than δ; the true negatives, TN_δ, the number of structure pairs coming from different classes and having pairwise distances greater than or equal to δ; the false negatives, FN_δ, the number of structure pairs coming from the same class and having pairwise distances greater than or equal to δ; and the false positives, FP_δ, the number of structure pairs coming from different classes and having pairwise distances smaller than δ. The precision-recall curve plots the precision, $TP_\delta/(TP_\delta + FP_\delta)$, against the recall, $TP_\delta/(TP_\delta + FN_\delta)$, when δ is varied from the minimum to the maximum distance. The *area under the precision-recall curve* (AUPRC) measures the average precision of the protein structure distance. Thus, the closer the AUPRC is to one, the better is the agreement between considered distance and the considered classification.

Another standard method is the Receiver-Operating Characteristic (ROC) curve analysis (Fawcett, 2006), where the true-positive rates, $TP_\delta/(TP_\delta + FN_\delta)$, are plotted against the false-positive rates, $FP_\delta/(FP_\delta + TN_\delta)$, which are obtained when δ is varied from the minimum to the maximum distance. The *area under the ROC curve* (AUC) is equal to the probability that the considered distance score will rank a randomly chosen pair of protein structures coming from the same class before a randomly chosen pair of protein structures coming from different classes. However, ROC curves are dependent on the ratio between the number of elements coming from the same class and the number of elements coming from different classes. Precision-recall curves are known to be more resilient in this respect (Davis and Goadrich, 2006).

2.2.4 Comparison with other methods We first compare GR-Align with state-of-the-art CMO solvers: the exact solver A_purva (Andonov *et al.*, 2011) that is used here as a slow but accurate heuristic, and the two heuristics MSVNS (Pelta *et al.*, 2008) and AlEigen7 (Di Lena *et al.*, 2010). When assessing the classification ability of GR-Align, we compare it with state-of-the-art protein structure comparison methods, namely, TM-Align (Zhang and Skolnick, 2005), DaliLite (Holm and Park, 2000) and MATT (Menke *et al.*, 2008). TM-Align produces rigid alignments that maximize the TM-Score, which is a derivative of the root mean squared deviation of the superimposed coordinates ($RMSD_c$). Although DaliLite does not consider the $RMSD_c$, it also tends to produce rigid alignments, as it penalizes matching internal distances that have >20% of differences. On the opposite, MATT produces flexible alignments by combining small rigid fragments. We additionally compare Gr-Align with two fast heuristics, namely, Yakusa, which uses a blast-like algorithm with a structural alphabet based on dihedral angles, and Fast, which finds the longest alignments having a $RMSD_c$ smaller than 2 Å using a heuristic-based on five-residue long fragments.

3 RESULTS

3.1 Comparing CMO-based methods

First, we compare GR-Align with state-of-the-art CMO solvers, namely, A_purva, MSVNS and AlEigen7, in terms of running time, the quality of their heuristic solutions and the agreements with the SCOP classification. We do this on our smallest dataset, the Protein_300 set, as computing the corresponding 44 850 alignments already represents the limit of previous solvers. We limit A_purva's computations to 500 iterations without branch and bound [the setting used in (Andonov *et al.*, 2011) for obtaining faster computations], which implies that A_purva's solutions are not always optimal. AlEigen7 is run with its default settings and MSVNS is run with its recommended settings (version 3, with 10 restarts). We use the 7.5 Å contact maps from Andonov *et al.* (2008) that were also used in several CMO-related papers (Andonov *et al.*, 2011; Di Lena *et al.*, 2010; Pelta *et al.*, 2008).

Table 1 summarizes the performance comparison between the CMO solvers. First, in terms of overlap between the contact maps, we observe two tendencies. On one hand, when two protein domains come from the same SCOP family, the contact maps have large edge-correctness (68.4% on average according to A_purva), and the best approximations are found by A_purva (RER. S.F = 99.4%), followed by GR-Align (RER S.F. = 86.6%), AlEigen7 (RER S.F. = 86.0%) and MSVNS (RER S.F. = 80.9%), i.e.all CMO solvers associate high similarity scores to protein domains coming from the same family. On the other hand, when two protein domains come from different SCOP families, the contact maps have small edge correctness (31.4% on average according to A_purva), and the best approximations are found by MSVNS (RER D.F. = 88.2%), followed by AlEigen7 (RER D.F. = 87.5%), A_purva (RER D.F = 81.2%) and GR-Align (RER D.F. = 57.5%), i.e. only GR-Align associates low scores to the protein domains coming from different families. Because classification relies heavily on the highest scoring pairs for building clusters (Swamidass *et al.*, 2010), the ability of GR-Align to finely approximate the high scores of similar proteins, while pushing down the scores of dissimilar proteins, leads to its best classification performance, as GR-Align achieves the highest NNI (100%), the highest average

Table 1. Performance comparison for CMO solvers

Measure	GR-Align	AlEigen7	A_purva	MSVNS
RER S.F.	86.6%	86.0%	99.4%	80.9%
RER D.F.	57.5%	87.5%	81.2%	88.2%
NNI	100%	98.0%	100%	98.7%
AUPRC	96.6%	78.5%	93.7%	72.9%
AUC	99.6%	97.9%	99.5%	97.0%
Time	1 min 55 s	2 h 30 min	5 h 24 min	39 h 51 min

Note: It is done using the Proteus_300 dataset. GR-Align (on column 2) is compared with AlEigen7, A_purva and MSVNS (columns 3, 4 and 5, respectively). Using A_purva's bounds on the optimal NCE, row 2 presents the retrieved edge rate that is achieved by each method on the 1350 domain pairs in which both domains come from the same SCOP families (RER S.F.). Row 3 presents the retrieved edge rate that is achieved on the 43 500 pairs in which the two domains come from different SCOP families (RER D.F.). Row 4 presents the NNIs, whereas rows 5 and 6 show the AUPRC and under the ROC curve (AUC) that are obtained by the scoring scheme of each method. Finally, row 7 presents the running time needed for computing the 44 850 alignments of the Proteus_300 set.

Table 2. Classification performance on the Gold-standard benchmark dataset

Method	CATH Topology			Running time
	NNI (%)	AUPRC (%)	AUC (%)	
GR-Align 12 Å	88.6	57.6	88.5	1 h 42 min
Yakusa	72.3	19.8	60.9	14 h 11 min
FAST	4.26	3.62	50.0	3 min 30.33 s
TM-align	88.6	75.6	93.9	134 h 16 min
MATT	89.2	51.5	85.0	831 h 44 min
DaliLite	90.9	85.3	96.2	2819 h 7 min

Note: For each method (rows), column 2 presents the percentage of protein domains whose CATH Topology is correctly identified using NNI, column 3 presents the AUPRCs, column 4 presents the AUCs and column 5 presents the running times needed for computing the 3 207 270 alignments of the Gold-standard benchmark dataset. FAST returned only 2241 alignments and Yakusa only 336 341 alignments.

precision (area under the precision recall curve of 96.6%) and the highest AUC (99.6%). The corresponding precision-recall and ROC curves are presented in Supplementary Figure S1.

Finally, GR-Align is the fastest CMO heuristics, as it requires only 1 min 54.82 s for computing the 44 850 alignments of the Proteus_300 set, including 1.52 s for the preprocessing step, being ~78 times faster than AlEigen7, 169 times faster than A_purva and 1247 times faster than MSVNS.

3.2 Effect of the distance threshold

Previous CMO-related studies (Caprara *et al.*, 2004; Di Lena *et al.*, 2010) suggest that using larger distance threshold ϵ when generating contact maps may lead to higher classification performances. However, because the time complexities of the proposed CMO solvers depend on the number of edges in the contact maps, and thus, on ϵ, these studies are limited to both small datasets and to small value of ϵ (up to 10 Å). Here, on the large Gold-standard benchmark (3 207 270 protein domain pairs), we measure the classification performance of Gr-Align when the distance threshold varies from 5–20 Å, in increments of 0.5 Å.

Both the NNI and the AUPRC reach their maximum for ϵ between 11.5 and 12 Å, with NNI = 88.8% and AUPRC = 57.6%. The AUC stays >88% for all ϵ between 8 and 12.5 Å (the results are presented in details in Supplementary Fig. S2). Hence, we only consider a distance threshold of 12 Å in the remainder of the article. This large distance threshold makes the other CMO methods not applicable, as their time complexities strongly depend on the number of edges in the contact maps, and thus, on ϵ.

3.3 Large-scale comparisons

We use the Gold-standard benchmark dataset for assessing the classification performance of GR-Align versus the state-of-the-art protein structure classification methods. On this benchmark, GR-Align (using edge-correctness and 12 Å contact maps) is compared with TM-Align (using TM-score), DaliLite (using Z-score), MATT (using raw score, as using *P*-values resulted in lower classification performances of MATT), Yakusa (using Z-score) and FAST (using normalised score). The agreement between the protein structure comparison methods and the CATH Topology classification is evaluated using nearest neighbor identification, precision-recall curve analysis and ROC curve analysis.

First, GR-Align is the fastest of the tested comparison methods and requires only 1 h 42 min (including its preprocessing step) for computing the 3 207 270 protein pairs of the Gold-standard benchmark dataset (Table 2). It is ~8 times faster than Yakusa, 79 times faster than TM-Align, 489 times faster than MATT and 1658 times faster than DaliLite. Only FAST achieved faster computation, but it returned only 2241 alignments, leading to poor classification performances. This is probably because the hard-coded thresholds in FAST heuristic are too stringent for recovering the distant homology relationships of CATH Topology.

Second, the classification performance of GR-Align, measured with respect to CATH Topology, is comparable with the performances of much slower tools. The nearest neighbors associated to each query protein domain come from the same Topology in 90.9% of the cases for DaliLite, 89.2% for MATT, 88.6% for both GR-Align and TM-Align, 72.3% for Yakusa and only 4.3% for FAST. When considering all protein pairs, the average precisions are 85.3% for DaliLite, 75.6% for TM-Align, 57.5% for GR-Align, 51.5% for Matt, 19.8% for Yakusa, and only 3.62% for FAST. The corresponding precision-recall curves are presented in Figure 5, whereas the ROC curves are presented in Supplementary Figure S3. Because CATH is built using rigid body superimposition-based alignments, the performances of TM-Align (which favors low RMSD of superimposed coordinates) and of DaliLite (which favors low distance differences) are expected. Here, the real contender of GR-Align is MATT, which is the only other flexible method. GR-Align and MATT achieve similar classification performances, whereas GR-Align is 489 times faster.

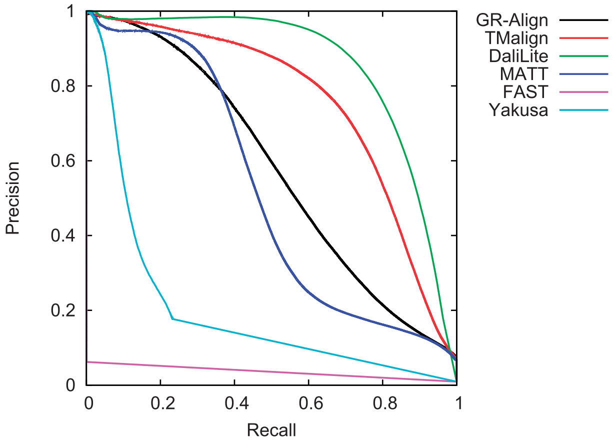

Fig. 5. Precision-recall curves of GR-Align and traditional methods, performed on the Gold-standard benchmark dataset

3.4 Aligning flexible proteins

GR-Align does not minimize the root-mean-square deviation of the alignments, and thus allows flexibility between the mapped structures. When aligning flexible proteins, we find that GR-Align's alignments are significantly different from the ones produced by traditional alignment methods such as DaliLite, MATT and TM-Align. GR-Align thus provides alternative homology relationships between protein structures. To highlight these differences in the alignments, we focus on a specific protein family called Calmodulin-like. Calmodulins are small (~145 residues) calcium-binding messenger proteins that are expressed in all eukaryotic cells, composed of two rigid calcium-binding EF-hand units that are connected by a flexible linker (an EF-hand is a small helix-loop-helix structural motif, and an 'EF-hand' unit contains two EF-hands). Calmodulins are well studied because of their flexible nature that allows them to bind a large variety of targets by wrapping around them (Vetter and Leclerc, 2003). Choosing the human Calmodulin (144 residues; SCOP_id d1clla_) as a query, we use GR-Align, DaliLite, MATT and TM-Align to find the 10 most similar protein domains in the Astral40 v1.75B database.

GR-Align outperforms all other methods for aligning flexible proteins (Table 3). In terms of running time, GR-Align requires 1 min 58 s for computing the 1 1154 alignments, followed by TM-Align (18 min17 s), MATT (1 h 43 min) and DaliLite (5 h 7 min). In terms of agreements with the SCOP family classification, GR-Align retrieves 90% of Calmodulin-like protein domains, whereas DaliLite, MATT and TM-Align retrieve 80, 40 and 40% of Calmodulin-like domains, respectively. The returned alignments are explained in details in Supplementary Tables S1 and S2. Also, when retrieving Calmodulin-like proteins, GR-Align's alignments are long, mapping on average 95.9% of d1clla_'s residues. In comparison, DaliLite, MATT and TM-Align map on average 71.9, 70.0 and 67.2% of the residues, respectively. GR-Align's alignments are thus more informative and allow transferring more information across proteins. An example is provided in Supplementary Table S3, where GR-Align's flexible alignment allows uncovering the rigid and flexible regions of d1clla_.

Table 3. Classification performance for flexible proteins

Method	Same family (%)	Cov. (%)	Time
GR-Align 12 Å	90	95.9	1 min 58 s
DaliLite	80	71.9	5 h 07 min
MATT	40	70.0	1 h 43 min
TM-Align	40	67.2	18 min 17 s

Note: For each method (given in rows), column 2 presents the percentage of the 10 nearest neighbors of the human Calmodulin (d1clla_) that come from the correct SCOP family (i.e. 'Calmodulin-like'). For the nearest neighbors in these top 10 that involve Calmodulin-like proteins, column 3 presents the length of the corresponding alignments (on average, as percentage of the number of residues in d1clla_). Finally, column 4 presents the running times that are needed to compute the 11 154 alignments.

4 CONCLUDING REMARKS

By generalizing graphlets and graphlet degrees to ordered graphs, we propose GR-Align, a novel maximum CMO heuristic that is particularly suited for database searches: it is 78–1247 times faster than the state-of-the-art CMO heuristics A_purva, AlEigen7 and MSVNS and 79–1658 times faster than the state-of-the-art alignment methods, TM-Align, MATT and DaliLite. GR-Align's similarity score is in better agreement with the SCOP classification than the similarity scores of other CMO heuristics and is on par with the similarity scores of the state-of-the-art protein structure comparison methods. Finally, for flexible proteins, GR-Align's alignments are more informative than the traditional alignments of DaliLite, MATT and TM-Align: they are in better agreement with the SCOP classification and have better coverage in terms of mapped residues, allowing to transfer more information across proteins.

In this article, we also extend previous studies about the effect of the distance threshold in the contact map generation on the classification performances, showing that the best classification performances are obtained with a threshold of 12 Å between α-carbons. Because GR-Align can be used with any contact map, it also makes possible large-scale analyses for contact maps using different contact definitions, e.g. using distances between β-carbons or the minimum distances between the atoms of the two residues.

Many parallel computation paradigms can be used to further speed-up GR-Align computations. For the preprocessing step (i.e.graphlet degree computations), each node can be processed independently, allowing a node level parallelism, leading to a potential two order of magnitude improvement of the corresponding running times (as most protein domains have between 100 and 1000 amino acids). For the alignment step, the Needleman–Wunsh and Smith–Waterman algorithm both have parallel implementations. In particular, the Single Instruction on Multiple Data implementation proposed in Farrar (2007) is shown to be up to 16 times faster than the linear implementation, and the Graphics Processing Unit (GPU) implementation proposed in Liu *et al.* (2009) is shown to be up to 30 times faster the linear one, which is faster than the NCBI Blast heuristic. Because the main difference between GR-Align and Needleman–Wunsh or Smith–Waterman is the computation of the amino acid pairwise costs, using similar parallel implementation techniques would lead to similar speed-ups.

The similarity between GR-Align and the traditional sequence alignment algorithms also paves the way to multiple flexible alignments of protein structures. This can be done by modifying the progressive multiple sequence alignment method CLUSTALW (Higgins *et al*., 1994), for which efficient parallel implementations exist (Li, 2003; Liu *et al*., 2006).

Funding: European Research Council (ERC) Starting Independent Researcher Grant (278212), the National Science Foundation (NSF) Cyber-Enabled Discovery and Innovation (CDI) (OIA-1028394), ARRS project (J1-5454) and the Serbian Ministry of Education and Science Project (III44006).

Conflict of Interest: none declared.

REFERENCES

Agarwal,P. *et al*. (2007) Fast molecular shape matching using contact maps. *J. Comput. Biol*., **14**, 131–147.

Altschul,S.F. *et al*. (1990) Basic local alignment search tool. *J. Mol. Biol*., **215**, 403–410.

Andonov,R. *et al*. (2008) An efficient lagrangian relaxation for the contact map overlap problem. In: *WABI'08: Proceedings of the 8th International Workshop on Algorithms in Bioinformatics*. Springer, Berlin, Heidelberg, pp. 162–173.

Andonov,R. *et al*. (2011) Maximum contact map overlap revisited. *J. Comput. Biol*., **18**, 27–41.

Brenner,S. *et al*. (2000) The astral compendium for sequence and structure analysis. *Nucleic Acids Res*., **28**, 254–256.

Caprara,A. and Lancia,G. (2002) Structural alignment of large—size proteins via lagrangian relaxation. In: *RECOMB'02: Proceedings of the Sixth Annual International Conference on Computational biology*. ACM, New York, NY, 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.

Carr,R. *et al*. (2000) Branch-and-cut algorithms for independent set problems: integrality gap and an application to protein structure alignment. In: *Technical report*. Sandia National Laboratories.

Crescenzi,P. and Kann,V. (1998) How to find the best approximation results – a follow-up to Garey and Johnson. *ACM SIGACT News*, **29**, 90–97.

Csaba,G. *et al*. (2009) Systematic comparison of SCOP and CATH: a new gold standard for protein structure analysis. *BMC Struct. Biol*., **9**, 23.

Davis,J. and Goadrich,M. (2006) The relationship between precision-recall and roc curves. In: *Proceedings of the 23rd International Conference on Machine learning, ICML'06*. ACM, New York, NY, pp. 233–240.

Di Lena,P. *et al*. (2010) Fast overlapping of protein contact maps by alignment of eigenvectors. *Bioinformatics*, **26**, 2250–2258.

Farrar,M. (2007) Striped smithwaterman speeds database searches six times over other SIMD implementations. *Bioinformatics*, **23**, 156–161.

Fawcett,T. (2006) An introduction to ROC analysis. *Pattern Recognit. Lett*., **27**, 861–874.

Gibrat,J.-F. *et al*. (1996) Surprising similarities in structure comparison. *Curr. Opin. Struct. Biol*., **6**, 377–385.

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

Godzik,A. and Skolnick,J. (1994) Flexible algorithm for direct multiple alignment of protein structures and seequences. *CABIOS*, **10**, 587–596.

Goldman,D. *et al*. (1999) Algorithmic aspects of protein structure similarity. In: *FOCS'99: Proceedings of the 40th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society, Washington, DC, pp. 512–521.

Hasegawa,H. and Holm,L. (2009) Advances and pitfalls of protein structural alignment. *Curr. Opin. Struct. Biol*., **19**, 341–348.

Higgins,D. *et al*. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*., **22**, 4673–4680.

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

Holm,L. and Sander,C. (1993) Protein structure comparison by alignment of distance matrices. *J. Mol. Biol*., **223**, 123–138.

Jain,B. and Lappe,M. (2007) Joining softassign and dynamic programming for the contact map overlap problem. In: Hochreiter,S. and Wagner,R. (eds) *BIRD*. Vol. 4414 *of Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, pp. 410–423.

Kuchaiev,O. and Pržulj,N. (2011) Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics*, **27**, 1390–1396.

Li,K.-B. (2003) Clustalw-mpi: clustalw analysis using distributed and parallel computing. *Bioinformatics*, **19**, 1585–1586.

Liu,W. *et al*. (2006) Gpu-clustalw: Using graphics hardware to accelerate multiple sequence alignment. In: Robert,Y. *et al*. (ed.) *High Performance Computing - HiPC 2006*. Vol. 4297 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, pp. 363–374.

Liu,Y. *et al*. (2009) Cudasw++: optimizing Smith-Waterman sequence database searches for CUDA-enabled graphics processing units. *BMC Res. Notes*, **2**, 73.

Malod-Dognin,N. *et al*. (2010) Maximum clique in protein structure comparison. In: *Proceedings of the 9th International Symposium on Experimental Algorithms, SEA 2010*. Ischia Island, Naples, Italy, pp. 106–117.

Mayr,G. *et al*. (2007) Comparative analysis of protein structure alignments. *BMC Struct. Biol*., **7**, 50.

Menke,M. *et al*. (2008) Matt: Local flexibility aids protein multiple structure alignment. *PLoS Comput. Biol*., **4**, e10.

Milenković,T. *et al*. (2010) Optimal network alignment with graphlet degree vectors. *Cancer Inform*., **9**, 121–137.

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

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.

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

Pržulj,N. (2007) Biological network comparison using graphlet degree distribution. *Bioinformatics*, **23**, 177–183.

Pržulj,N. *et al*. (2004) Modeling interactome: scale-free or geometric? *Bioinformatics*, **20**, 3508–3515.

Shibberu,Y. and Holder,A. (2011) A spectral approach to protein structure alignment. *IEEE/ACM Trans. Comput. Biol. Bioinform*., **8**, 867–875.

Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol*., **147**, 195–197.

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

Swamidass,S.J. *et al*. (2010) A croc stronger than ROC: measuring, visualizing and optimizing early retrieval. *Bioinformatics*, **26**, 1348–1356.

Vetter,S. and Leclerc,E. (2003) Novel aspects of calmodulin target recognition and activation. *Eur. J. Biochem*, **270**, 404–414.

Wohlers,I. *et al*. (2009) Paul: protein structural alignment using integer linear programming and lagrangian relaxation. *BMC Bioinformatics*, **10** (**Suppl. 13**), P2.

Xie,W. and Sahinidis,N. (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.

Ye,Y. and Godzik,A. (2003) Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics*, **19**, II246–II255.

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