BIOINFORMATICS ORIGINAL PAPER Vol. 26 no. 18 2010, pages 2273–2280
doi:10.1093/bioinformatics/btq420

Structural bioinformatics Advance Access publication July 17, 2010

Towards optimal alignment of protein structure distance matrices

Inken Wohlers[1,*], Francisco S. Domingues[2] and Gunnar W. Klau[1,*]

[1]CWI, Life Sciences Group, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands and [2]Max-Planck-Institut für Informatik, Campus E1 4, 66123 Saarbrücken, Germany

Associate Editor: Anna Tramontano

ABSTRACT

Motivation: Structural alignments of proteins are important for identification of structural similarities, homology detection and functional annotation. The structural alignment problem is well studied and computationally difficult. Many different scoring schemes for structural similarity as well as many algorithms for finding high-scoring alignments have been proposed. Algorithms using contact map overlap (CMO) as scoring function are currently the only practical algorithms able to compute provably optimal alignments.

Results: We propose a new mathematical model for the alignment of inter-residue distance matrices, building upon previous work on maximum CMO. Our model includes all elements needed to emulate various scoring schemes for the alignment of protein distance matrices. The algorithm that we use to compute alignments is practical only for sparse distance matrices. Therefore, we propose a more effective scoring function, which uses a distance threshold and only positive structural scores. We show that even under these restrictions our approach is in terms of alignment accuracy competitive with state-of-the-art structural alignment algorithms, whereas it additionally either proves the optimality of an alignment or returns bounds on the optimal score. Our novel method is freely available and constitutes an important promising step towards truly provably optimal structural alignments of proteins.

Availability: An executable of our program PAUL is available at http://planet-lisa.net/

Contact: Inken.Wohlers@cwi.nl

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

Received on April 1, 2010; revised and accepted on July 12, 2010

1 INTRODUCTION

1.1 Background

Structural alignments are assignments of structurally equivalent amino acids between proteins with many applications in structural biology (Sierk and Kleywegt, 2004). Generally, the 3D structure of a protein determines its function and is often more conserved during evolution than protein sequence. Therefore, structural alignment is especially useful to detect remotely homologous proteins with low sequence identity. Structural alignment is also applied to identify common structural elements and new protein folds and is used to cluster all known protein structures into groups. Furthermore, structural similarities can be used for functional annotation (Yakunin *et al.*, 2004).

Various scoring functions and algorithms have been proposed. They can be divided into two categories: they are either based on the root mean square deviation (RMSD) of the 3D coordinates after optimal superposition of aligned residues or on matching inter-residue distances imposed by aligned residues. Finding an optimal assignment of amino acids is in both cases an *NP*-hard problem (Lathrop, 1994), since coordinate and distance matrix representation of a protein structure can be transformed into each other in polynomial time (Havel *et al.*, 1983). Therefore, many structural alignment algorithms use heuristics to find alignments. They confine the search space, e.g. by considering fragments of several residues or by applying hierarchical approaches that align secondary structure elements (SSEs) in a first step.

There are many different competitive approaches and techniques for structural alignment, either utilizing 3D coordinate superposition [MAX-PAIRS (Poleksic, 2009), PROTDEFORM (Rocha *et al.*, 2009), MATT (Menke *et al.*, 2008), CE (Shindyalov and Bourne, 1998), STRUCTAL (Subbiah *et al.*, 1993), FATCAT (Ye and Godzik, 2003), SHEBA (Jung and Lee, 2000), C-ALPHA MATCH (CA) (Bachar *et al.*, 1993), PPM (Csaba *et al.*, 2008), TMALIGN (Zhang and Skolnick, 2005), LGA (Zemla, 2003) and RASH (Standley *et al.*, 2007)], or inter-residue distances [DALI (Holm and Sander, 1993), MATRAS (Kawabata, 2003), SSAP (Taylor and Orengo, 1989) and VORORIGN (Birzele *et al.*, 2007)]. Many of these algorithms are considerably accurate when evaluated based on manually curated reference alignments, and often they are quite fast. Nonetheless, they are not capable to report whether the computed alignment is optimal according to the corresponding scoring function. As a consequence, there is no way to compare different scoring schemes for structural alignment, because weak performance can be attributed either to the scoring function that is maximized or to the algorithm that is used.

Some efforts to compute close to optimal structural alignments exist. For alignments based on 3D coordinate superposition, Kolodny and Linial (2004) give a polynomial-time algorithm for finding an alignment whose score lies within ε of the optimal score. Although their algorithm runs in polynomial time, it is too expensive to be used in practice. Poleksic (2009) suggests also a polynomial-time ε-approximation algorithm for a class of scoring functions based on 3D superposition, which works very well in practice. Finding the optimal alignment, i.e. $\varepsilon = 0$, however, is not practical with any of the two approaches.

Exact structural alignment algorithms based on inter-residue distance matrices have so far been limited to very basic

*To whom correspondence should be addressed.

scoring functions. There are, for example, algorithms that compute optimal structural alignments by finding their maximum contact map overlap (CMO) [(Caprara *et al.*, 2004), A_PURVA (Andonov *et al.*, 2008), CMOS (Xie and Sahinidis, 2007)], as well as faster heuristics for the same problem (Jain and Obermayer, 2009; Pelta *et al.*, 2008). Furthermore, other exact algorithms determine a structural alignment by identifying a maximum clique of similar inter-residue distances (Malod-Dognin *et al.*, 2010; Strickland *et al.*, 2005). These algorithms have been evaluated in terms of their capability to compute alignments to optimality and to classify rather small benchmark sets. Although optimality might be especially interesting in the case of single-case high-quality alignment, the accuracy of their alignments has not been benchmarked or compared.

1.2 Contribution

In this article, we give a general mathematical model for optimal alignment of inter-residue distance matrices. Our model is based on an integer linear programming (ILP) formulation of Caprara *et al.* (2004). We add structural scores, affine gap costs, amino acid substitution scores and a negative sequence penalty, similar to Taylor and Orengo (1989), to the model. In this way, various existing scoring functions can be emulated, e.g. those of SSAP, MATRAS and DALI. While we can model these scoring schemes, the current algorithm can handle only positive scores and is practical only for sparse distance matrices. We specifically pinpoint the two restrictions that currently prevent us from computing alignments based on the original scoring functions for complete distance matrices. These are: (i) the too large number of pairs of inter-residue distances, especially for long proteins; and (ii) the current lack of effectively considering negative structural scores. Therefore, we currently impose a distance threshold and use only positive structural scores. Our method is implemented in a software tool called PAUL. With a preliminary version of PAUL we showed, e.g. that the use of structural scores generates better alignments than the use of contact maps (Wohlers *et al.*, 2009). Now we develop a general mathematical model and a comprehensive, improved implementation, e.g. by integrating scoring schemes, gap costs, filtering and different types of inter-residue distances. We perform an extensive parameter optimization in order to determine our own customized scoring function. We do this for C_α, C_β and all-atom distances, where all-atom distances are the minimum distance between any pair of atoms of two residues. We show that the algorithm with our scoring scheme is competitive with state-of-the-art structural alignment methods in terms of alignment accuracy. Additionally, we are able to compute a provably optimal alignment in about 25% of the cases. High accuracy and optimality or near-optimality come at the price of a significantly longer run-time that is not competitive to that of heuristic methods. Therefore, the main contribution of our approach is 2-fold. First, we provide a very accurate algorithm with an interchangeable and customizable scoring scheme for high-quality single-case pairwise alignment. Second, we make an important step towards a general method for optimal alignment of distance matrices using combinatorial optimization, by formulating the corresponding ILP and pointing out the current challenges. A general method that generates optimal alignments will make it possible to evaluate and compare different distance matrix-based scoring schemes for structural alignment and help develop provably better scoring functions.

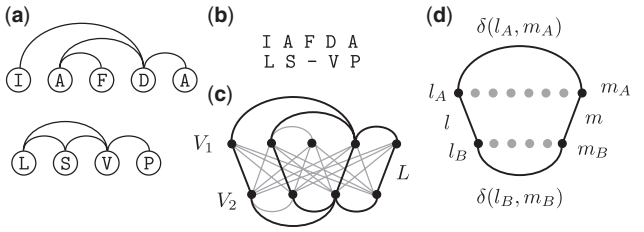

Fig. 1. Maximum CMO problem. (**a**) Two protein contact maps. (**b**) Alignment of the two proteins. (**c**) Corresponding solution in the graph problem. Alignments are characterized by non-crossing matches, or *traces*, in the complete bipartite alignment graph $(V_1 \cup V_2, L)$ (Kececioglu, 1993). Here, vertices in V_1 and V_2 denote the residues of the two proteins, respectively, and L is the complete set of alignment edges, i.e. $L = \{(i,j) \mid i \in V_1, j \in V_2\}$. The displayed trace (bold alignment edges) maximizes the CMO, in the example there are three shared contacts (also shown in bold). (**d**) An aligned pair of distances. Function $\delta(\cdot,\cdot)$ denotes the distance between two residues with respect to their 3D coordinates.

2 APPROACH

2.1 Combinatorial approach to structural alignment

Our mathematical model and algorithm are based on Caprara *et al.* (2004). Caprara *et al.* (2004) compute a pairwise alignment of two protein structures that maximizes the number of common contacts. Two residues are in contact if they are in some sort of chemical interaction, e.g. by hydrogen bonding. The model uses a simple distance criterion: whenever the distance between two residues is below a predefined distance threshold d_t, the residues are considered to be in contact. Caprara *et al.* (2004) have introduced the maximum *CMO* problem of finding the maximum number of common contacts in two proteins. They give an ILP formulation, which they propose to solve using an elegant Lagrangian relaxation approach.

Their ILP formulation relies on a reformulation of the structural alignment problem as a graph problem. Figure 1 explains the relation. In their model, Caprara *et al.* (2004) introduce two vectors $\mathbf{x}$ and $\mathbf{y}$ of binary variables. They consist of variables x_l for each alignment edge $l \in L$ aligning two residues and variables y_{lm} for each potentially shared common contact maintained by the two alignment edges l and m. The binary variables indicate the presence or absence of the corresponding objects in the solution. The authors express the set of feasible solutions using linear inequalities and integrality constraints involving $\mathbf{x}$ and $\mathbf{y}$ and find the largest set of common contacts using the objective function $\max \sum_{(l,m)\in\binom{L}{2}} y_{lm}$. For the detailed mathematical model refer to Caprara *et al.* (2004).

To align inter-residue distance matrices, we change the mathematical model of Caprara *et al.* (2004) by replacing the rigid contact definition and taking into account the 3D distances between the residues. Let (l,m) be a pair of aligned distances of two proteins A and B with $l = (l_A, l_B)$ and $m = (m_A, m_B)$, see also Figure 1d. By using a distance measure $\delta(\cdot,\cdot)$ that denotes the distance between two residues with respect to their 3D coordinates, we are now able to align inter-residue distances instead of contacts. Therefore, we replace the objective function $\max \sum_{(l,m)\in\binom{L}{2}} y_{lm}$ by

$$\max \sum_{(l,m)\in\binom{L}{2}} w_{lm} y_{lm} + S(\mathbf{x}) \ , \qquad (1)$$

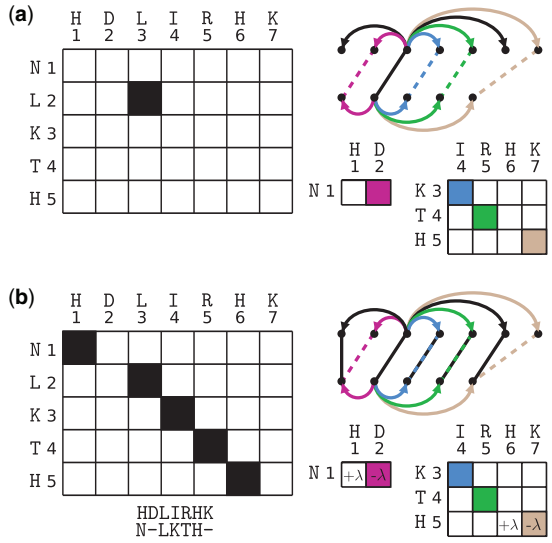

Fig. 2. Double DP. (**a**) Left, the upper-level DP matrix. Right, profit computation via two lower-level DP matrices, demonstrated for the upper-level DP cell that aligns residues 3 and 2. One profit DP matrix aligns on the left, the other on the right of alignment edge (3,2). Scores of lower-level DP cells are given by Equation (2) and each coloured cell, and its score corresponds to a pair of inter-residue distances that is realized in the profit computation. The score of an upper-level DP cell consists of this profit (computed via the two lower level DPs) minus a sequence penalty c. (**b**) An alignment, denoted by black DP cells and black alignment edges, containing two gaps with gap penalty g. Choices that have been made in the lower level DPs during profit computation do not agree with the upper-level DP and are penalized by Lagrangian multipliers λ. Steps (**a**) and (**b**) are repeated iteratively.

while keeping the constraints unchanged. In Equation (1), $S(\mathbf{x})$ is a sequence score. The structural score w_{lm} is determined by a function that assigns a value for aligning distance $\delta(l_A, m_A)$ with distance $\delta(l_B, m_B)$. Inspired by the *rigid similarity* introduced by Holm and Sander (1993) in their paper on DALI, we choose

$$w_{lm} = \begin{cases} \max\{0, \theta - \Delta_{lm}\} & \Delta_{lm} \leq \Delta_t \\ 0 & \text{otherwise} \end{cases}, \qquad (2)$$

with $\Delta_{lm} = |\delta(l_A, m_A) - \delta(l_B, m_B)|$. Here, θ and Δ_t are constant parameters.

As Caprara *et al.* (2004), we use Lagrangian relaxation to compute alignments, which leads to an iterative double dynamic programming (DP) algorithm, see Figure 2. The algorithm can handle only positive scores effectively; therefore, we have to restrict w_{lm} to be greater than zero. Pairs of inter-residue distances with a score w_{lm} smaller than or equal to zero can be omitted from the ILP.

The idea of Lagrangian relaxation is to relax constraints of an intractable problem, e.g. the structural alignment ILP, such that the relaxed problem can be solved efficiently. The relaxed constraints are moved to the objective function, penalized by so-called Lagrangian multipliers. The optimal solution of the original problem (i.e. the top-scoring alignment) is also a solution of the relaxed problem, and every optimal solution of the relaxed problem provides an upper bound on the optimal score of the original problem. In general, a solution of the relaxed problem is not a feasible structural alignment (cf. Fig. 2b). Nonetheless, from a solution of the relaxed problem a feasible structural alignment can be deduced, and the score of this alignment provides a lower bound on the optimal score of the original problem. The Lagrangian multipliers are adjusted iteratively such that the upper bound decreases gradually, and we can deduce better structural alignments and thus increase the lower bound on the optimal score. If upper and lower bound coincide, an optimal structural alignment has been found.

In each Lagrange iteration, we solve the relaxed problem for the current Lagrangian multipliers using double DP (cf. Fig. 2). In a first step, the profit of each alignment edge is computed (Fig. 2a, right). Therefore, we determine the alignment of maximum score of those residues that are within distance threshold d_t of the two residues of the respective alignment edge. These are the lower level dynamic programs, two for each pair of residues. In a second step, an alignment of the proteins based on the given profits is computed (Fig. 2b). This is the upper-level dynamic program. The resulting DP score is an upper bound. We generate a feasible structural alignment by considering all pairs of inter-residue distances that are realized by this alignment, the corresponding score constitutes a lower bound. Analogous to contact map alignment, the time and space complexity of one Lagrange iteration of our method is $O(E_1 E_2)$, where E_1 and E_2 are the numbers of considered distances in the two proteins, respectively.

2.2 Interchangeable scoring

Any positive function that scores pairs of inter-residue distances can be used to compute scores w in (1). Examples are the scoring function used by SSAP or the non-negative portion of scoring functions used by DALI or MATRAS.

The number of pairs of inter-residue distances is $\binom{n_1}{2}\binom{n_2}{2}$, where n_1 and n_2 are the lengths of the sequences. This large number currently cannot be treated explicitly even for medium-sized proteins. Therefore, algorithms that align inter-residue distance matrices resort to techniques such as aligning fragments of several residues or hierarchical alignment. In our approach, we use a distance threshold d_t; only distances smaller than the threshold are used. We hereby omit long inter-residue distances that are less significant for structural similarity than short distances. Since our scoring function assigns only positive scores, we additionally try to omit all pairs of inter-residue distances that are unlikely observed in biologically correct alignments.

In our algorithm, we use scoring function (2), which takes into account that scores must be positive and that we use a distance threshold. For the interpretation of parameters, see Supplementary Material. We determine scoring function parameters for C_α, C_β and all-atom distances. Each type of inter-residue distance places a slightly different focus on the alignment. An alignment of C_α distances is based on similar protein backbone conformations, an alignment based on C_β distances takes into account side-chain placement, and an alignment of all-atom distances highlights similar residue interactions in the two proteins. Because of the different nature and range of the three inter-residue distance types, we determine individual scoring function parameters.

We define our own scoring function that explores the characteristics of our algorithm. To make a first step towards aligning general distance matrices, we additionally emulate the positive portion of the scoring function used in the structural alignment algorithm MATRAS (Kawabata, 2003). Analogous to sequence

substitution matrices, MATRAS uses log-odds value matrices M as scores. A value $M_{\lfloor i \rfloor, \lfloor j \rfloor}$ indicates the log-likelihood that distance i is aligned to distance j. Matrices are determined for 20 different sequence separations, i.e. number of residues in the sequence over which the distance ranges. A positive log-likelihood for a given sequence separation means that the corresponding distances are aligned more likely than expected by chance. We investigate how alignment accuracy and performance change when the distance threshold is varied by computing alignments based on MATRAS scoring.

2.3 Filtering

The most effective technique to improve performance and speed of our algorithm is to keep the number of variables in our ILP low. For each variable y_{lm}, we store its score and adapt its Lagrangian multiplier. For each alignment edge l, we compute its profit, which has to be completely recomputed in each Lagrange iteration in which at least one of the multipliers of outgoing pairs of distances has been changed (cf. Fig. 2b). Accordingly, there are two ways of filtering: filtering pairs of distances affects the lower-level dynamic program, and filtering alignment edges the upper-level dynamic program. If we forbid an alignment edge l and, therefore, set its variable x_l to zero, we never have to compute its profit, i.e. the two lower-level dynamic programs.

We filter on both levels. On the upper level, we use secondary structure information. Using DSSP (Kabsch and Sander, 1983), we assign to each residue its secondary structure type. We determine experimentally that we can safely forbid to align α-helical residues (DSSP state 'H') with β-sheet residues (DSSP state 'E'). On the lower level, we filter in three different ways. Here, we take advantage of the non-negative scoring function and omit all pairs of distances to which a negative score would be assigned. We omit pairs of distances from which at least one distance exceeds the distance threshold d_t and distances that differ more than the distance difference threshold Δ_t. Additional to these filters, we use MATRAS log-odds values (Kawabata, 2003) and omit all pairs of distances with a negative log-likelihood.

2.4 Sequence scores and gap costs

Using only positive scores for pairs of distances leads to the problem that even aligning structurally dissimilar regions will increase the overall score. A poor global alignment can then yield a higher score than a shorter alignment of structurally highly conserved regions. We deal with this problem by using a global negative sequence penalty c for aligning two residues. The sequence penalty serves as a threshold on the structural score, i.e. on the profit of an alignment edge. Only if the profit exceeds this threshold, the overall score increases when the corresponding residues are aligned.

To obtain a general approach for the alignment of inter-residue distance matrices, we incorporate affine gap costs with gap open penalty g and gap extension penalty 0.25 times g into the algorithm. Practically, the gap costs simply have to be considered in the upper-level dynamic program and we do this using the Gotoh algorithm (Gotoh, 1982). Formally integrating gap costs into the ILP is more involved, and we refer to Bauer *et al.* (2007) for a comprehensive analogous extension of the model in the context of RNA alignment.

An optional sequence score can be added to the structural score of each alignment edge. This sequence score can be provided by any external amino acid substitution matrix. A scaling factor can be tuned in order to balance structural against sequence score. By default, sequence scores are not used.

2.5 Implementation

We implemented the novel structural alignment algorithm as the freely available package PAUL within the C++ software library PLANET LISA (`http://planet-lisa.net/`). PAUL supports different input formats, e.g. PDB files, lists of pre-selected distances or complete distance matrices, as well as different types of inter-residue distances, C_α, C_β and all-atom. Furthermore, also scoring function parameters, gap costs, amino acid substitution matrices and the proportion of sequence to structural score can be chosen. The use of secondary structure and log-odds for filtering is optional. By keeping the scoring adaptable, PAUL allows to incorporate additional information about the input proteins and makes it a very flexible software tool. By default, PAUL uses the optimized scoring function parameters reported in this article as well as SSE filtering. If a global alignment of two proteins is needed, we resort to TCOFFEE (Notredame *et al.*, 2000). TCOFFEE computes a sequence-based alignment for those residues for which no structural similarity could be determined.

3 METHODS

3.1 Experimental setup for parameter setting, optimization, and evaluation

To optimize parameters, we use a training set consisting of structure-based alignments from the Homologous Structure Alignment Database (HOMSTRAD, October 2008 release) (Mizuguchi *et al.*, 1998). As these alignments are manually curated by experts, we consider them as gold standard reference alignments. From HOMSTRAD, we consider only those non-corrupt 302 protein families with exactly two members from the twilight or midnight zone of sequence identities below 35%, where sequence identity denotes the percentage of identically aligned residues in comparison to the total number of aligned residues. We optimize the parameters on a training set of 200 alignments and evaluate them on a test set that consists of the remaining 102 alignments. We measure the quality of the results computed by structural alignment algorithms in terms of the achieved alignment accuracy, which is the number of correctly aligned residues divided by the number of aligned residues in the reference alignment.

We conduct an extensive parameter sweep in order to determine robust scoring function parameters that promote a high average alignment accuracy. See Supplementary Material for details. For the best overall parameter sets $(d_t, \Delta_t, \theta, c, g)$ for combinations of C_α, C_β or all-atom distances with or without filtering, we align the HOMSTRAD test set and compare PAUL performance to DALI, a widely used state-of-the-art structural alignment algorithm that is ranked very high in many benchmarks (Berbalk *et al.*, 2009; Mayr *et al.*, 2007; Poleksic, 2009; Rocha *et al.*, 2009). For this evaluation, we use the 99 test set protein pairs for which DALI returns an alignment; this subset we call the consolidated test set.

3.2 Evaluating alignment accuracy

We use the parameters optimized on the HOMSTRAD dataset to evaluate PAUL on two distinct, more challenging datasets, SISY and RIPC (Berbalk *et al.*, 2009; Mayr *et al.*, 2007) (`http://biwww.che.sbg.ac.at/RSA/`). These sets contain alignments that are difficult for various reasons and were constructed to assess and compare the accuracy of structural alignment algorithms and to investigate their limitations. We compare PAUL to the state-of-the-art structural alignment programs DALI, MATRAS, MATT, SHEBA,

FATCAT, CE and CA, which were benchmarked by Berbalk *et al.* (2009). The SISY set is assembled from SISYPHUS (Andreeva *et al.*, 2007), a manually curated database for alignments of proteins with non-trivial relationships. It consists of 130 very diverse reference alignments for which the lengths of the protein chains and the number of aligned residues in the reference alignment vary greatly. The RIPC set contains 23 reference alignments with focus on functional residues and flexible proteins. Following Berbalk *et al.* (2009), we use their consolidated set of SISY and RIPC, which are the subsets for which no structural alignment method in their study fails to produce a result. They contain 98 and 22 alignments, respectively. Note that PAUL does not fail even on the removed alignments (see also Section 5). Furthermore, two recent structural alignment methods, PROTDEFORM (Rocha *et al.*, 2009) and MAX-PAIRS (Poleksic, 2009) have been evaluated on an older version of the SISY set, which we name in the following SISY.V1. For SISY.V1, the consolidated set of alignments for which no method fails comprises 106 protein pairs. In Rocha *et al.* (2009) and Poleksic (2009), DALI, SSAP, MAX-PAIRS, PROTDEFORM, MATRAS, MATT, PPM, LGA, VOROLIGN and RASH have been benchmarked on SISY.V1. We evaluate PAUL on the consolidated SISY.V1 set and compare the results.

For aligning proteins from SISY, SISY.V1 and RIPC, we use a maximum run-time of 30 CPU minutes per alignment. Note that the actual run-time in which we observe improvements depends strongly on the length of the proteins and is usually much shorter (cf. Supplementary Fig. 3 and HOMSTRAD results). Nonetheless, to prove optimality of a solution, a conservative and rather deliberate update scheme for the Lagrangian multipliers (e.g. the one of Caprara *et al.*, 2004) is needed and thus a longer overall run-time.

4 RESULTS

4.1 Optimized scoring function and results on HOMSTRAD

For PAUL_MATRAS, which uses positive MATRAS log-odds values as structural score, a high distance threshold is beneficial, for C_α distances, $d_t = 10$ Å and for C_β distances, $d_t = 10.5$ Å (see Supplementary Table 4). Due to the high memory requirements, we currently cannot evaluate higher distance thresholds. The corresponding average alignment accuracy for C_α distances lies at 88.7% and for C_β distances at 88.9%. The respective average test set accuracies are 91.1% for C_α and 90.4% for C_β distances (cf. Supplementary Table 5).

When optimizing the PAUL parameters of scoring function (2) we find that the average alignment accuracies on HOMSTRAD vary only slightly for different parameter values (cf. Supplementary Tables 1–3). Furthermore, the parameter values after optimization are robust: slightly changed parameter settings still have comparable average alignment accuracy. Using a secondary structure filter always leads to higher accuracy. When a log-odds filter is used, higher distance thresholds improve alignment accuracy. The best cross-validation accuracy on the training set after solely optimizing structural scores is 87.6% for C_α, 87.4% for C_β and 87.8% for all-atom distances (Supplementary Tables 1–3 list all values).

In a second stage, we fix the best parameter set (d_t, Δ_t and θ) for each combination of an inter-residue type (C_α, C_β and all-atom) with a filter type and optimize the parameters without influence on structural scores, i.e. the sequence penalty c and the gap open penalty g. The final average alignment accuracies on the training set are in the range 89.0–90.2% (cf. Supplementary Table 5). The average alignment accuracies on the consolidated test set lie in the range 90.8–91.3% and the median alignment accuracies in the range 91.9–93.4%. DALI reaches on the consolidated test set an average accuracy of 89.0% and a median accuracy of 91.0%.

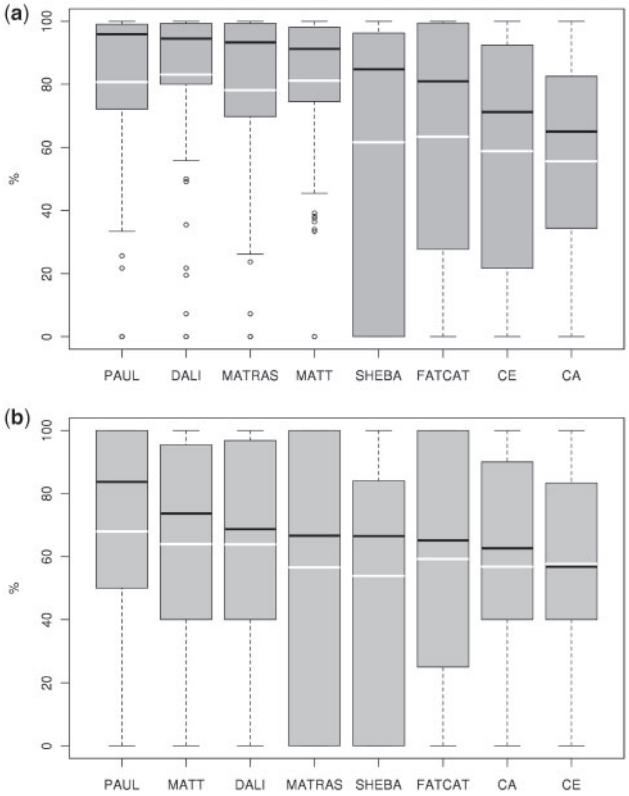

Fig. 3. Box plots display median and quartiles of the distributions of percentages of alignment accuracies for (**a**) the SISY set and (**b**) the RIPC set for PAUL, DALI, MATRAS, MATT, SHEBA, FATCAT, CE and CA. Ordering according to decreasing median. White lines denote average accuracies.

We obtain the highest average training set alignment accuracy of 90.2% for C_β distances with SSE filter and a parameter set $(d_t, \Delta_t, \theta, c, g) = (8.5, 7, 9.8, 19.6, 23)$ (refer to Supplementary Figs 1 and 2 for a histogram of aligned C_β distances and a visualization of the scoring function, respectively). We use this setting for the evaluation on the SISY set. This is also PAUL's default scoring scheme.

4.2 Results for SISY

Figure 3a displays the accuracies of PAUL, DALI, MATRAS, MATT, SHEBA, FATCAT, CE and CA. PAUL alignments reach with 95.9% the highest median, but with 80.7% a slightly lower average alignment accuracy than MATT and DALI. According to Wilcoxon signed rank tests, PAUL matches the reference alignments significantly better than SHEBA, FATCAT, CE and CA ($P < 10^{-4}$). PAUL furthermore computed >25% of alignments (26) to provable optimality. The correlation between alignment accuracies of different methods lies mainly between 0.5 and 0.6 and is thus generally low (cf. Supplementary Fig. 5).

PAUL with traditional CMO as scoring function ($d_t = 7.5$ Å) reached a median alignment accuracy of 88.5% and an average alignment accuracy of 72.1%. Because rounding can be applied for the decimal upper bound of the integer-value contact map overlap

score, even 48 of the alignments were computed to optimality, i.e. almost 50%.

We also test the performance of PAUL_MATRAS, our algorithm with emulated positive MATRAS scores. The results show that although using only positive scores and applying a distance threshold, PAUL_MATRAS achieves accuracies competitive with those achieved by MATRAS (cf. Supplementary Fig. 4), but not as high as PAUL using our customized scoring function. With C_α distances, PAUL_MATRAS reaches a higher median but a lower average alignment accuracy than MATRAS. PAUL_MATRAS with C_β distances performs slightly worse according to both median and average. The correlation between PAUL_MATRAS and MATRAS alignment accuracies is not particularly high (Pearson and Spearman's correlations 0.67 and 0.73, respectively, for C_α distances; 0.66 and 0.72 for C_β). The correlation between PAUL_MATRAS with C_α distances and PAUL_MATRAS with C_β distances is in contrast very high (Pearson's correlation 0.91, Spearman's correlation 0.87).

4.3 Results for SISY.V1

On SISY.V1, PROTDEFORM (Rocha *et al.*, 2009) and MAX-PAIRS (Poleksic, 2009) have been evaluated and compared with DALI, SSAP, MAX-PAIRS, PROTDEFORM, MATRAS, MATT, PPM, LGA, VOROLIGN and RASH. For this dataset, PAUL alignments reach with 96.3% the highest median alignment accuracy of all methods and, slightly after MATT, with 82.4% the second highest average accuracy (for MAX-PAIRS, we compare only to the version that is solely structural and does not use BLOSUM62 substitution scores). The box plots of alignment accuracies for each of the methods are provided in Supplementary Figure 8. More than 30% (32) of the alignments were computed to optimality. Many of the methods that have been tested on this dataset perform very well and should be considered competitive in terms of alignment accuracy. PAUL alignments match the reference alignments significantly better than LGA, VOROLIGN and RASH alignments ($P < 10^{-4}$). The correlation between alignment accuracies achieved by different methods is, as on SISY, generally low (cf. Supplementary Fig. 9). Average run-times of the compared structural alignment methods vary notably, between <1 s and 24 s (cf. Supplementary Table 6, which is taken from Rocha *et al.*, 2009). PAUL tries to compute the alignments to optimality and is terminated after 30 CPU minutes; nonetheless, often alignments do not change any more after much shorter time spans of 1–2 min; this is visualized in Supplementary Fig. 3.

4.4 RIPC set

PAUL reaches the highest average and median alignment accuracy of all methods tested (DALI, MATRAS, MATT, SHEBA, FATCAT, CE and CA). Box plots of alignment accuracies are given in Figure 3b. Four of the 22 PAUL alignments were computed to optimality, but only 2 of those 4 reach 100% alignment accuracy with respect to the reference alignment (cf. Supplementary Table 7).

PAUL is furthermore unsuccessful in four cases, generating alignments with 0% alignment accuracy. All these problematic cases correspond to pairs of proteins related by circular permutation where the reference alignments also include extensive indels. This result is expected as PAUL was not designed to account for circular permutation. In general, the other tested methods also perform poorly on these four pairs, with the exception of CA, which allows for non-sequential alignments. There are five cases where PAUL performs better than most of the other methods in terms of alignment accuracy (d1d5fa_/d1nd7a_, d1dlia1/d1mv8a1, d1l5ba_/d1l5ea_, d1gsaa1/d2hgsa1 and d2bbma_/d4clna_). These cases correspond to pairs of proteins with extensive structural variation resulting from flexibility or from divergent evolution, Supplementary Figures 10 and 11 provide two examples. In four of these cases PAUL alignments have 100% accuracy, with FATCAT and MATRAS also generating high-quality alignments. The fifth case (d1gsaa1/d2hgsa1) corresponds to two remote homologous proteins with considerable structural variation where the PAUL alignment is 80% accurate, whereas other methods achieve only 40% accuracy (cf. Fig. 4).

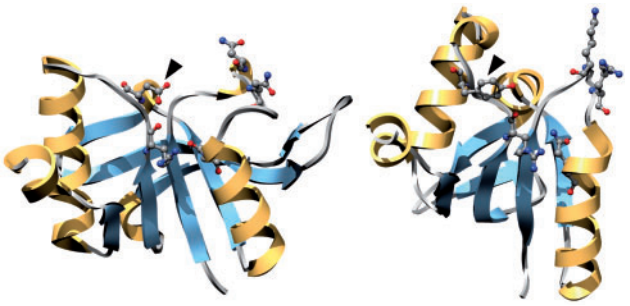

Fig. 4. An example from the RIPC dataset. The N-terminal domain of glutathione synthetase from *Escherichia coli* on the left (d1gsaa1) (Hara *et al.*, 1996), and from human on the right (d2hgsa1) (Polekhina *et al.*, 1999). β-strands are in blue, α-helices in orange. The two proteins show considerable structural variation and several indels are required to align them. The sequence identity is 16%. The five positions matched in the reference alignment are represented as ball-and-stick, the rigid body superposition of their C_α atoms has an RMSD of 2.05 Å. PAUL misaligns only one position (marked with arrow), but the shift is only one residue in the sequence. The proteins are viewed in the same orientation as given by the superposition according to the reference alignment. Image prepared with Chimera (Pettersen *et al.*, 2004).

5 DISCUSSION AND CONCLUSION

The results on HOMSTRAD show that our algorithm performs well with any type of inter-residue distance matrix, despite the qualitative difference between C_α, C_β and all-atom distances. On this test set, it reaches a higher average and median alignment accuracy than DALI with all types of inter-residue distances and any type of filter. Furthermore, our optimized scoring function is robust against slight parameter changes and always performs better than CMO scoring. Introducing additionally sequence penalty and gap costs consistently increases the alignment accuracy.

The SISY and RIPC sets give us further insight into the alignments produced by PAUL. When compared to reference alignments, the alignment accuracy is very high and often better than that of other state-of-the-art structural alignment methods. This good performance of PAUL can be attributed to several factors. First, PAUL always computes alignments on single-residue level (contrary to fragments) and without using any hierarchical approach that might introduce mistakes on the first, broad level which cannot be corrected later on. Second, structural alignment methods that are based on intramolecular distances as in PAUL, are more adequate to compare flexible proteins or proteins with considerable structural variation

than methods that rely on rigid body superposition. Twists have to be introduced in the comparison of these types of proteins by rigid body superposition methods in order to accommodate for structural variation. DALI and MATT also perform well in these datasets as they both take into account protein flexibility in comparisons. DALI relies on intramolecuar distances, and MATT is a superposition-based method that allows for different relative orientation between fragment pairs during alignment generation. When evaluating RIPC results, one should be cautious, because the dataset is very small; nonetheless, the results indicate that PAUL compares favourably to other methods in protein pairs that are challenging for structural comparison, in particular, when these proteins show considerable structural variation or flexibility.

Furthermore, in the SISY set, we find that the accuracy of PAUL_MATRAS alignments, which are computed based on positive MATRAS scores and with a distance threshold, is competitive with the accuracy obtained by MATRAS itself. This result indicates that (i) scores for some pairs of inter-residue distances might be irrelevant for a correct alignment and (ii) that we use a very accurate algorithm to compute the structural alignment.

The SISY set also contains pairwise alignments with multiple chains. These were excluded from our study because many algorithms handle multiple chains differently or cannot handle them at all. In PAUL, we add an option to concatenate several chains, which is reasonable if they correspond to one biological unit. Using this option, we are able to run PAUL also on the instances of multiple chains and obtain alignments in very good agreement with the reference alignments.

Looking closer at PAUL alignments, we find that they still share a characteristic with alignments that have been computed with CMO scoring: the exclusively positive structural scores encourage to align even remotely structurally similar regions. The alignments tend to be longer and have higher RMSD than alignments of other structural alignment methods (cf. Supplementary Fig. 6). On the other hand, a high RMSD is expected when proteins are related by considerable structural variation or flexibility. In fact, RMSD does not tend to correlate with alignment accuracy as measured by comparison to reference alignments (cf. Supplementary Fig. 7). RMSD is a useful measure of structure similarity, but does not seem to be an adequate indicator of alignment quality.

Our results give a partial answer to a question raised by Pelta *et al.* (2008): can the performance of DALI be achieved using strategies based on contact maps? On the datasets of our study and with respect to alignment accuracy, the answer is that an increased distance threshold and a more sophisticated scoring function is needed. Protein structure classification will benefit from increased alignment accuracy. For classification, the scoring function can be adjusted to obtain a better accuracy to speed ratio. Furthermore, integrating fast heuristics such as Pelta *et al.* (2008), and Jain and Obermayer (2009) into exact algorithms can help to find high-scoring alignments earlier (even if proofing optimality might still be time consuming), which can render the approach competitive also in terms of speed.

We show that the current algorithm is, in terms of alignment accuracy, competitive with state-of-the-art structural alignment methods. High accuracy and optimality or near-optimality of the computed alignment come at the price of a longer run-time in comparison to heuristic methods. PAUL is, therefore, not competitive with faster methods when performing extensive structure comparison over a database. Instead PAUL should be applied when there is a need for a high-quality pairwise structure alignment, where a run-time of minutes is acceptable.

Our study demonstrates that using combinatorial optimization and a sophisticated scoring function, we can compute very accurate and in some cases optimal alignments. Although the current approach works well, it is only a first step towards a general ILP-based approach for optimal structural alignment of complete protein inter-residue distance matrices. On the way to such a generic method, there are three major challenges. First, we have to compute more alignments to optimality. Second, an algorithm has to be developed that can effectively handle negative structural scores. Third, we need to be able to handle more or even all pairs of inter-residue distances.

With a general method for the alignment of inter-residue distance matrices, we could then align proteins using various scoring schemes; for instance, those of the widely-used structural alignment algorithms DALI, MATRAS and SSAP. Knowing always the top-scoring alignment, we could objectively compare structural alignment scoring schemes. Furthermore, these scoring schemes could be verifiably improved, for example, by designing combinations of different scoring functions.

ACKNOWLEDGEMENTS

The authors thank Lars Petzold for the implementation work on PAUL and for helpful discussions and Ingolf Sommer for valuable comments and initiation of the authors' cooperation. MATRAS log-odds matrices were kindly provided to us by Takeshi Kawabata and the SISY set and its evaluation environment by Peter Lackner.

Funding: DFG grant KL 1390/2-1; Computational experiments were sponsored by the NCF for the use of supercomputer facilities, with financial support from NWO.

Conflict of Interest: none declared.

REFERENCES

Andonov,R. *et al.* (2008) An efficient Lagrangian relaxation for the contact map overlap problem. In *Proceedings of the 8th international workshop on Algorithms in Bioinformatics*. Vol. 5251 of *Lecture Notes in Computer Science*. Springer, Berlin/Heidelberg, pp. 162–173.

Andreeva,A. *et al.* (2007) SISYPHUS—structural alignments for proteins with non-trivial relationships. *Nucleic Acids Res.*, **35**(Database issue), 253–259.

Bachar,O. *et al.* (1993) A computer vision based technique for 3-D sequence-independent structural comparison of proteins. *Protein Eng.*, **6**, 279–288.

Bauer,M. *et al.* (2007)Accurate multiple sequence-structure alignment of RNA sequences using combinatorial optimization. *BMC Bioinformatics*, **8**, 271.

Berbalk,C. *et al.* (2009) Accuracy analysis of multiple structure alignments. *Protein Sci.*, **18**, 2027–2035.

Birzele,F. *et al.* (2007) Vorolign—fast structural alignment using Voronoi contacts. *Bioinformatics*, **23**, 205–211.

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.

Csaba,G. *et al.* (2008) Protein structure alignment considering phenotypic plasticity. *Bioinformatics*, **24**, 98–104.

Gotoh,O. (1982) An improved algorithm for matching biological sequences. *J. Mol. Biol.*, **162**, 705–708.

Hara,T. *et al.* (1996) A pseudo-michaelis quaternary complex in the reverse reaction of a ligase: structure of Escherichia coli B glutathione synthetase complexed with ADP, glutathione, and sulfate at 2.0 A resolution. *Biochemistry*, **35**, 11967–11974.

Havel,T. *et al.* (1983) The theory and practice of distance geometry. *Bull. Math. Biol.*, **45**, 665–720.

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

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 (IJCNN '09)*, pp. 1394–1400.

Jung,J. and Lee,B. (2000) Protein structure alignment using environmental profiles. *Protein Eng.*, **13**, 535–543.

Kabsch,W. and Sander,C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, **22**, 2577–2637.

Kawabata,T. (2003) MATRAS: a program for protein 3D structure comparison. *Nucleic Acids Res.*, **31**, 3367–3369.

Kececioglu,J.D. (1993) The maximum weight trace problem in multiple sequence alignment. In *Proceedings of the Fourth Annual Symposium of Combinatorial Pattern Matching (CPM 93)*. Vol. 684 of *Lecture Notes in Computer Science*. Springer, pp. 106–119.

Kolodny,R. and Linial,N. (2004) Approximate protein structural alignment in polynomial time. *Proc. Natl Acad. Sci. USA*, **101**, 12201–12206.

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

Malod-Dognin,N. *et al.* (2010) Maximum cliques in protein structure comparison. In *Proceedings of the 9th International Symposium on Experimental Algorithms (SEA'10)*, Vol. 6049 of *Lecture Notes in Computer Science*. Springer, Berlin/Heidelberg, 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.

Mizuguchi,K. *et al.* (1998) HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Sci.*, **7**, 2469–2471.

Notredame,C. *et al.* (2000) T-Coffee: a novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.*, **302**, 205–217.

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

Pettersen,E.F. *et al.* (2004) UCSF Chimera—a visualization system for exploratory research and analysis. *J. Comput. Chem.*, **25**, 1605–1612.

Polekhina,G. *et al.* (1999) Molecular basis of glutathione synthetase deficiency and a rare gene permutation event. *EMBO J.*, **18**, 3204–3213.

Poleksic,A. (2009) Algorithms for optimal protein structure alignment. *Bioinformatics*, **25**, 2751–2756.

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

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.

Sierk,M.L. and Kleywegt,G.J. (2004) Déjà vu all over again: finding and analyzing protein structure similarities. *Structure*, **12**, 2103–2111.

Standley,D.M. *et al.* (2007) ASH structure alignment package: sensitivity and selectivity in domain classification. *BMC Bioinformatics*, **8**, 116.

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

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

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

Wohlers,I. *et al.* (2009) Aligning protein structures using distance matrices and combinatorial optimization. In *Proceedings of the German Conference on Bioinformatics (GCB '09)*. Vol. P-157 of *Lecture Notes in Informatics*, pp. 33–43.

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

Yakunin,A.F. *et al.* (2004) Structural proteomics: a tool for genome annotation. *Curr. Opin. Chem. Biol.*, **8**, 42–48.

Ye,Y. and Godzik,A. (2003) Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics*, **19** (Suppl. 2), 246–255.

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

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