Optimal DALI protein structure alignment

Inken Wohlers, Rumen Andonov, Gunnar W. Klau

Optimal DALI protein structure alignment

Inken Wohlers[1⋆], Rumen Andonov[2], and Gunnar W. Klau[1]

[1] CWI, Life Sciences group, Amsterdam, the Netherlands
[2] INRIA Rennes - Bretagne Atlantique and University of Rennes 1, France
{inken.wohlers,gunnar.klau}@cwi.nl, randonov@inria.fr

Abstract. We present a mathematical model and exact algorithm for protein structure alignment using DALI scoring, which is an *NP*-hard problem. DALI scoring is based on comparing the inter-residue distance matrices of proteins and is the scoring model of the widely used heuristic DALI program. Our model and algorithm extend an integer linear programming approach previously used for the related contact map overlap problem. To this end, we introduce a novel type of constraint that handles negative structure scores and relax it in a Lagrangian fashion. We also review options that allow to consider less pairs of inter-residue distances explicitly, because their large number makes it difficult to optimize DALI scoring optimally. We use our exact algorithm DALIX to compute many provably score-optimal DALI alignments for the first time, using four data sets of varying structural similarity. Further, using our exact DALIX alignments, it is for the very first time possible to qualitatively benchmark the heuristic DALI program in sound mathematical terms. The results indicate that DALI often computes optimal or close to optimal alignments, but also that in cases of aligning small proteins it tends to fail generating any significant alignment although such an alignment exists.
Key words: structure alignment, inter-residue distance matrix, exact algorithm, integer linear program, Lagrangian relaxation, DALI

1 Introduction

Protein structure alignment, the assignment of structurally equivalent amino acids between protein structures, is an important problem in structural bioinformatics. Detecting and evaluating structural similarities is a standard task that lies at the basis of many applications like, for example, homology detection, fold recognition, protein classification, and functional annotation. For proteins with low sequence similarity, structure alignments improve upon sequence alignments and are successfully applied for detecting functional similarities or distinct evolutionary relationships. Due to the exponential increase in available protein structures, as for example in the protein data bank (PDB) (Berman et al., 2000), the problem will receive even more interest in the future.

⋆ Corresponding author

The structural alignment problem is *NP*-hard for biologically meaningful scoring schemes used in practice (Lathrop, 1994). As a result, almost all algorithms for structure alignment are heuristics; they aim at optimizing a given scoring scheme, but have no notion how much better their computed alignments could be with respect to that scoring scheme. DALI (distance matrix alignment) is one of the most widely used structural alignment heuristics (Holm and Sander, 1993). It is available via the EBI structural analysis tool box, a dedicated server processes about 1500 pairwise alignment user requests a month, and the first DALI paper has been cited almost 3000 times, more often than any other structure alignment program[1].

We present an exact DALI algorithm, which we call DALIX. It returns either the optimal alignment according to DALI scoring, if found within a predefined time limit, or an alignment together with an upper bound on the optimal score. Our algorithm uses techniques from combinatorial optimization. First, we cast the problem into an integer linear program (ILP) whose objective function maximizes the DALI score and whose constraints denote that the solution represents a structure alignment. We then relax a few constraints, move them to the objective function and penalize their violation by multiplying them with Lagrangian multipliers. The relaxed problem can then be solved by double dynamic programming, which is a method that is also used in other structure alignment algorithms like SSAP (Taylor and Orengo, 1989) or MATRAS (Kawabata and Nishikawa, 2000). Iteratively, multipliers are adjusted and the double dynamic programming is repeated. After a specified number of iterations, the problem is split into subproblems within a branch-and-bound algorithm.

Our mathematical model and algorithm are applicable to any distance matrix-based scoring scheme, *e.g.* those of SSAP (Taylor and Orengo, 1989), contact map overlap (CMO) (Godzik and Skolnick, 1994), MATRAS (Kawabata and Nishikawa, 2000) or PAUL (Wohlers et al., 2010); for the general framework see (Wohlers et al., 2011). Here, we focus on DALI because it is a popular structural alignment method that performs well in many benchmarks (for example, Mayr et al., 2007). Provably maximizing the DALI scoring is especially difficult, because it uses inter-residue distances between any pair of residues. Our exact algorithm thus needs to explicitly consider $O(n^4)$ distance pairs. This has great influence on performance and memory requirements.

Using our algorithm DALIX, we are able to compute many DALI alignments for the first time to provable optimality and thereby to assess the quality of the DALI heuristic. For this purpose we use (i) alignments of SCOPCath domains (Csaba et al., 2009) with lengths between 30 and 50 who share family, superfamily or fold, (ii) alignments from SKOLNICK (Caprara et al., 2004) of proteins from the same family, and alignments from (iii) the SISY and (iv) RIPC collections (Mayr et al., 2007; Berbalk et al., 2009). We find that DALI is very reliable in returning a good alignment according to DALI scoring. Although we find many cases where the DALI alignment is not optimal, the difference in score between the heuristic and the optimal alignment is often negligible. When aligning short

[1] Including closely related and follow up papers, DALI was cited more than 5000 times.

protein domains, DALI's deficiency is that it misses to detect quite a few significant similarities and wrongly reports that no such similarity exists. We also evaluate the weak points of our exact algorithm DALIX, which are large proteins and subtle structural similarities. In these cases, the (then suboptimal) alignment returned by DALIX is often worse than the heuristic alignment returned by DALI.

2 Mathematical Model and Algorithm

In this section we introduce the alignment graph representation of the structural alignment problem. Based on this representation, we formulate an integer linear program (ILP) that models the problem of finding the alignment of maximum DALI score. We devise a Lagrangian relaxation that has been used before for contact map alignment (Andonov et al., 2011) and extend it by relaxing an additional, new type of constraint that is needed for pairs of distances with negative score. We focus in the entire description of the algorithm on these novel constraints, because except for them, model and algorithm are analogous to (Andonov et al., 2011). In the last subsection we suggest algorithm engineering techniques to improve the performance of our method in practice.

2.1 Alignment Graph Representation

We denote the inter-residue distance matrix of a protein A by (A_{ij}). It is a symmetric square matrix of size $n_A \times n_A$, where n_A is the length of the protein. A matrix entry A_{ij} is the Euclidean distance between the C_α atoms of residues i and j.

We represent the structural alignment problem using an alignment graph. For two proteins of length n_A and n_B, the alignment graph is a $n_A \times n_B$ product or grid graph as displayed in Figure 1e). Rows represent the residues of protein A and columns the residues of protein B. A node $i.k$ in the alignment graph indicates the alignment of residue i from protein A with residue k from protein B. Directed edges $(i.k, j.l)$ exist between any pair of nodes for which $i < j$ and $k < l$. Edges are thus south-west to north-east bound, and we refer to nodes $i.k$ and $j.l$ as the tail and head of edge $(i.k, j.l)$, respectively. An edge denotes the matching of distance A_{ij} with distance B_{kl}, see also Figure 1c) and d). An alignment of length n is represented by a set of nodes $i_1.k_1, i_2.k_2, \ldots, i_n.k_n$ for which $i_1 < i_2 < \ldots < i_n$ and $k_1 < k_2 < \ldots < k_n$. We call such a set an increasing path. A structural alignment comprises additionally all induced edges.

2.2 Mathematical Model

We assign binary variables x_{ik} to alignment graph nodes. They indicate whether residue i is aligned with residue k, in which case $x_{ik} = 1$. An alignment graph edge between nodes $i.k$ and $j.l$ is described by a binary variable y_{ikjl} which

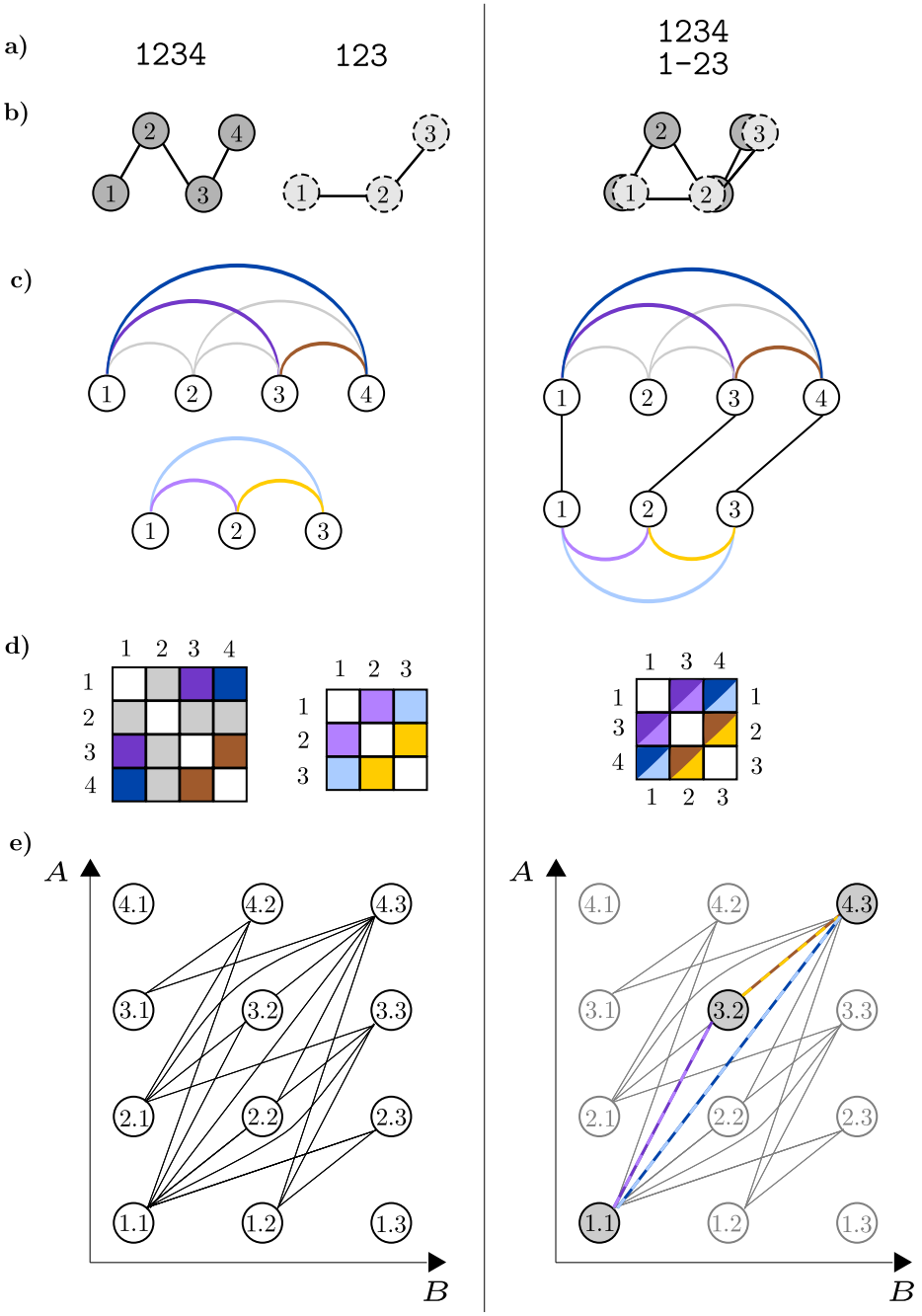

Fig. 1. Different protein and alignment representations of protein A with $n_A = 4$ residues and protein B with $n_B = 3$ residues. a) The amino acid sequence representation. Instead of the amino acid, the corresponding residue number is given. The alignment on the right denotes which residues structurally match. The second residue of protein A is unaligned. b) The corresponding superposition. Given the alignment of residue 1 with 1, 3 with 2 and 4 with 3, protein B is translated and rotated such that the superposition minimizes RMSD. c) The residues are arranged on a horizontal line and the inter-residue distances displayed. The alignment on the right highlights three pairs of aligned inter-residue distances. d) The inter-residue distance matrices. The super-imposed and collapsed distance matrices denote the alignment and highlight three pairs of aligned residues. e) The alignment graph. On the right, the activated nodes and edges that correspond to the given alignment are shown.

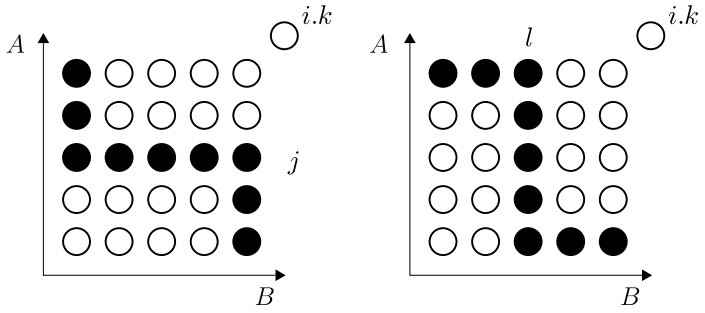

Fig. 2. The black nodes are an illustration of $\mathrm{row}_{ik}(j)$ (left) and $\mathrm{col}_{ik}(l)$ (right) in the alignment graph. In the displayed situation, $j < i$ and $l < k$, in which case the colored nodes are sets of mutually exclusive tails of contradicting edges with common head $i.k$. If $j > i$ and $l > k$ (not displayed), the colored nodes are sets of mutually exclusive heads of contradicting edges with common tail $i.k$.

denotes whether distance A_{ij} from protein A is aligned with distance B_{kl} in protein B, in which case $y_{ikjl} = 1$.

We define $\mathrm{row}_{ik}(j)$ and $\mathrm{col}_{ik}(l)$ as sets of nodes that are either tails of edges with head at $i.k$ or heads of edges with tail at $i.k$ and that mutually contradict because no two of them lie on an increasing path. There are many ways of constructing them, the one we use is introduced in (Andonov et al., 2011) and illustrated in Figure 2. The DALIX ILP is then given by

$$\max \quad \sum_{i=1}^{n_A-1} \sum_{j=i+1}^{n_A} \sum_{k=1}^{n_B-1} \sum_{l=k+1}^{n_B} 2s(A_{ij}, B_{kl}) y_{ikjl} + \sum_{i=1}^{n_A} \sum_{k=1}^{n_B} s(A_{ii}, B_{kk}) x_{ik} \tag{1}$$

$$\text{s.t.} \quad x_{ik} \geq \sum_{(r,s) \in \mathrm{row}_{ik}(j)} y_{ikrs} \qquad j \in [i+1, n_A], i \in [1, n_A - 1], k \in [1, n_B - 1] \tag{2}$$

$$x_{ik} \geq \sum_{(r,s) \in \mathrm{col}_{ik}(l)} y_{ikrs} \qquad l \in [k+1, n_B], i \in [1, n_A - 1], k \in [1, n_B - 1] \tag{3}$$

$$x_{ik} \geq \sum_{(r,s) \in \mathrm{row}_{ik}(j)} y_{rsik} \qquad j \in [1, i-1], i \in [2, n_A], k \in [2, n_B] \tag{4}$$

$$x_{ik} \geq \sum_{(r,s) \in \mathrm{col}_{ik}(l)} y_{rsik} \qquad l \in [1, k-1], i \in [2, n_A], k \in [2, n_B] \tag{5}$$

$$x_{ik} \leq \sum_{\substack{(r,s) \in \mathrm{row}_{ik}(j) \\ s(A_{ri}, B_{sk}) \leq 0}} (y_{rsik} - x_{rs}) + 1 \qquad j \in [1, i-1], i \in [2, n_A], k \in [2, n_B] \tag{6}$$

$$\sum_{l=1}^{k} x_{il} + \sum_{j=1}^{i-1} x_{jk} \leq 1 \qquad i \in [1, n_A], k \in [1, n_B] \tag{7}$$

$$\mathbf{x}, \mathbf{y} \text{ binary.} \tag{8}$$

Here, the scoring function $s(\cdot,\cdot)$ for pairs of inter-residue distances is the DALI elastic similarity function (Holm and Sander, 1993),

$$s(A_{i,j}, B_{k,l}) = \begin{cases} \left(0.2 - \frac{|A_{i,j} - B_{k,l}|}{\frac{1}{2}(A_{i,j}+B_{k,l})}\right) e^{-\left(\frac{1}{2}(A_{i,j}+B_{k,l})/20\right)^2} & i \neq j \text{ and } k \neq l \\ 0.2 & \text{otherwise .} \end{cases}$$

An alignment that maximizes the objective function of ILP (1)-(8) is thus an alignment of maximum overall DALI score. Based on this overall DALI score $S(A,B)$, the DALI z-score $Z(A,B)$ is computed according to the formula given in (Holm and Sander, 1998),

$$Z(A,B) = \frac{S(A,B) - m(L)}{0.5 \cdot m(L)}.$$

The term $m(L)$ for $L = \sqrt{n_A, n_B}$ is the approximate mean score and the denominator $0.5 \cdot m(L)$ estimates the average standard deviation. The z-score thus measures the significance of the detected structural similarity based on an experimentally determined background distribution of DALI scores.

All constraints except (6) are analogous to the constraints established by Andonov et al. (2011) for the CMO model. Constraints (2) and (3) denote that an edge can only be taken if its tail node is activated and if the heads of edges with common tail $i.k$ cannot contradict. Constraints (4) and (5) denote the reverse situation: an edge can only be taken if its head is activated and the tails of edges with common head $i.k$ do not contradict. Activated nodes have to lie on an increasing path, which is specified by constraints (7). Different from the model for CMO, the DALIX model has additional constraints (6). These describe that an edge has to be activated if its head and tail are activated. This is important since according to DALI scoring, edge scores can be negative, in which case the remaining constraints allow to omit these edges. Constraints (6) are derived from the simple constraints

$$x_{ik} + x_{jl} - y_{jlik} \leq 1$$

for all y_{jlik} with score less than or equal to zero. In these simple constraints, the term $x_{jl} - y_{jlik}$ can be lifted to

$$\sum_{(r,s) \in \mathrm{ROW}_{ik}(j)} (x_{rs} - y_{rsik}) \ ,$$

since each tail $r.s$ has, according to constraints (2) and (3), no outgoing edges with contradicting heads $i.k$.

2.3 Lagrangian Approach

Lagrangian Relaxation. We relax constraints (4), (5) and (6). This means that now an edge can be taken even if its head is not activated and even if its tail contradicts with the tail of another edge (constraints (4) and (5)) as well as that

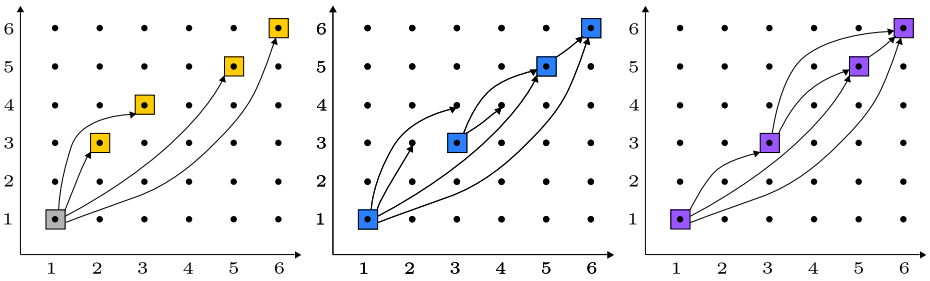

Fig. 3. Visualization of local profit computation, the solution of the relaxed problem and the feasible solution. Left: Node 1.1 picks its best set of outgoing edges, which are the edges maximizing this node's profit. The corresponding increasing path is colored yellow. Center: The solution of the relaxed problem. It is composed of the increasing path that is the solution of the global problem, colored in blue, together with the outgoing edges that these nodes picked in their respective local problem. The relaxed solution maximizes the sum of profits. For a few edges in the solution of the relaxed problem, the heads are not activated, *e.g.* for edge $(1.1, 3.2)$. Also, for nodes in the solution, the induced edge is missing, *e.g.* in the relaxed solution, there is no edge between nodes 3.3 and 6.6. Right: The feasible solution that can be deduced from the relaxed solution. It is composed of the nodes that are activated in the relaxed solution together with all induced edges.

an edge can be omitted even if its head and tail are activated (constraints (6)). Since constraints (2) and (3) are not relaxed, still any edge needs to have an activated tail, and since constraints (7) dictate that activated nodes lie on an increasing path, the tails of edges can not contradict in spite of relaxing constraints (4) and (5). The solution of the relaxed problem is the following: An increasing path of activated nodes, in which each activated node picks outgoing edges of maximum overall score. The heads of these outgoing edges are not necessarily activated. See Figure 3 for a visualization. The relaxation can then straightforwardly be strengthened by constraints

$$\sum_{l=s+1,\cdots,k} y_{rsil} + \sum_{j=r+1,\cdots,i-1} y_{rsjl} \leq 1 \tag{9}$$
$$r \in [1, n_A], s \in [1, n_B], i \in [r+1, n_A], k \in [s+1, n_B] \ ,$$

which denote that the heads of outgoing edges picked by each node must (although still not necessarily activated) form an increasing path. The objective function of the relaxed problem is given by

$$\begin{aligned}\mathrm{LR}(\lambda) = \max \sum_{i=1}^{n_A-1} \sum_{j=i+1}^{n_A} \sum_{k=1}^{n_B-1} \sum_{l=k+1}^{n_B} 2s(A_{ij}, B_{kl}) y_{ikjl} + \sum_{i=1}^{n_A} \sum_{k=1}^{n_B} s(A_{ii}, B_{kk}) x_{ik} \\ + \sum_{\substack{i,k \\ j\in[1,i-1]}} \lambda^h_{ikj} \left(x_{ik} - \sum_{(r,s)\in \mathrm{row}_{ik}(j)} y_{rsik} \right) \\ + \sum_{\substack{i,k \\ l\in[1,k-1]}} \lambda^v_{ikl} \left(x_{ik} - \sum_{(r,s)\in \mathrm{col}_{ik}(l)} y_{rsik} \right) \\ + \sum_{\substack{i,k \\ j\in[1,i-1]}} \lambda^a_{ikj} \left(1 - x_{ik} + \sum_{\substack{(r,s)\in \mathrm{row}_{ik}(j) \\ s(A_{ri},B_{sk})\le 0}} (y_{rsik} - x_{rs}) \right). \end{aligned} \tag{10}$$

All λ are greater or equal 0. Here, λ^h_{ikj} denotes the multipliers for constraints (4), λ^v_{ikl} for constraints (5) and λ^a_{ikj} for constraints (6).

Double Dynamic Programming. The relaxed problem can be solved in time $O(n_A^2 n_B^2)$, where n_A and n_B are the protein lengths. This follows from the proof in (Andonov et al., 2011), according to which the complexity is $O(|V| + |E|)$, with $|V|$ the number of alignment graph nodes and $|E|$ the number of alignment graph edges. In the case of the DALI ILP, $|V| = n_A n_B$ and $|E| = \binom{n_A}{2}\binom{n_B}{2}$, from which the stated complexity follows.

The relaxed problem can be solved by double dynamic programming, *i.e.*, dynamic programming on two levels. For this purpose we solve for each node a local problem and afterwards one global problem. In the local problems, we compute for each node $i.k$ the best set of outgoing edges with their heads on an increasing path by the use of dynamic programming. We call the sum of the corresponding edge weights the node profit p_{ik}. In the global problem, we assign to each node its profit plus a node score of 0.2 and compute the increasing path of maximum overall weight, again by the use of dynamic programming.

The edge and node scores are adjusted according to the Lagrangian multiplier coefficients of x- and y-variables in the objective function (10) of the relaxed problem. How the Lagrangian multipliers redistribute score between nodes and incident edges in the case of a violated constraint is visualized in Figure 4. In the local problems, we associate to each edge $(j.l, i.k)$ the weight

$$c_{jlik}(\lambda) = \begin{cases} 2s(A_{ij}, B_{kl}) - \lambda^h_{ikj} - \lambda^v_{ikl} + \lambda^a_{ikj} & \text{if } s(A_{ij}, B_{kl}) \le 0 \\ 2s(A_{ij}, B_{kl}) - \lambda^h_{ikj} - \lambda^v_{ikl} & \text{otherwise .} \end{cases}$$

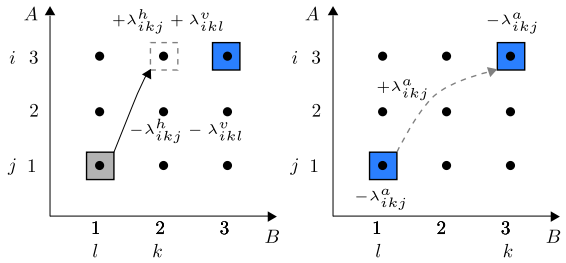

Fig. 4. An example of the redistribution of score between nodes and edges in the case of violated constraints. Left: Constraints (4) and (5) are violated, since the head of edge $(1.1, 3.2)$ is not activated and thus $x_{32} \ngeq y_{1132}$. Lagrangian multipliers will increase the weight of node 3.2 and decrease the weight of edge $(1.1, 3.2)$. Right: Constraint (6) is violated, since the edge between the activated nodes is not activated and thus $x_{11} + x_{33} - y_{1133} \nleq 1$. The multipliers for the activation constraints will decrease the weights of the nodes 1.1 and 3.3 and increase the weight of the incident edge.

Once profits p_{ik} have been computed by solving the local problem for each node $i.k$, we solve the global problem. In the global problem, the node weights are given by

$$c_{ik} = p_{ik} + \sum_{j\in[1,i-1]} \lambda^h_{ikj} + \sum_{l\in[1,k-1]} \lambda^v_{ikl} - \sum_{j\in[1,i-1]} \left(\lambda^a_{ikj} - \sum_{\substack{j:r.s\in row_{ik}(j) \\ \wedge s(A_{ri},B_{sk})\leq 0}} \lambda^a_{ikj} \right).$$

The solution of (10) subject to (2), (3) and (7) has objective function value $\mathrm{LR}(\lambda)$ which is an upper bound for the original problem. It comprises the set of nodes that solve the global problem, $\bar{x}$, together with the set of edges composing the solutions of the local problems, $\bar{y}$, for those tail nodes that are in the solution of the global problem. Nodes corresponding to $\bar{x}$ together with their induced edges represent a feasible solution for the original problem, *i.e.*, a structural alignment, with objective function value Z_{lb}, which constitutes a lower bound. This is visualized in Figure 3.

Updating Lagrangian Multipliers. Before each iteration $t + 1$, the Lagrangian multipliers are adjusted by subgradient descent,

$$\lambda^{t+1} = \max\{0, \lambda^t - \theta^t g^t\},$$

with step size

$$\theta^t = \frac{\alpha[\mathrm{LR}(\lambda^t) - Z_{lb}]}{\sum[(g^h)^t]^2 + \sum[(g^v)^t]^2 + \sum[(g^a)^t]^2} \quad .$$

For updating the multipliers, the gradients are computed as follows

$$g^h_{ikj} = \bar{x}_{ik} - \sum_{(r,s)\in \mathrm{row}_{ik}(j)} \bar{y}_{rsik} \in \{-1, 0, 1\} \tag{11}$$

$$g^v_{ikl} = \bar{x}_{ik} - \sum_{(r,s)\in \mathrm{col}_{ik}(l)} \bar{y}_{rsik} \in \{-1, 0, 1\} \tag{12}$$

$$g^a_{ikj} = 1 - \bar{x}_{ik} + \sum_{\substack{(r,s)\in \mathrm{row}_{ik}(j) \\ s(A_{ri},B_{sk})\leq 0}} (\bar{y}_{rsik} - \bar{x}_{rs}) \in \{-1, 0, 1\} \ . \tag{13}$$

There are only three situations in which gradient g^a_{ikj} for the activation constraints is non-zero. The first situation is $g^a_{ikj} = 1$, if

$$\bar{x}_{ik} = 0, \quad \sum_{\substack{(r,s)\in \mathrm{row}_{ik}(j) \\ s(A_{ri},B_{sk})\leq 0}} \bar{y}_{rsik} = 0; \quad \sum_{\substack{(r,s)\in \mathrm{row}_{ik}(j) \\ s(A_{ri},B_{sk})\leq 0}} \bar{x}_{rs} = 0.$$

In this case λ^a_{ikj} is decreased. Since $\lambda^a_{ikj} \geq 0$, we have to check for this situation only for indices i, k and j with non-zero multiplier λ^a_{ikj} in the current iteration.

The gradient g^a_{ikj} is also equal to 1, if

$$\bar{x}_{ik} = 0, \quad \sum_{\substack{(r,s)\in \mathrm{row}_{ik}(j) \\ s(A_{ri},B_{sk})\leq 0}} \bar{y}_{rsik} = 1, \quad \sum_{\substack{(r,s)\in \mathrm{row}_{ik}(j) \\ s(A_{ri},B_{sk})\leq 0}} \bar{x}_{rs} = 1.$$

This second situation is analogous to the situation detected by identifying violated constraints (4): The tail node of an edge is activated, but its head is not. Multiplier λ^a_{ikj} is decreased, as in the first situation, which is only possible if the current multiplier λ^a_{ikj} is greater than zero.

Furthermore, in the third situation $g^a_{ikj} = -1$, if

$$\bar{x}_{ik} = 1, \quad \sum_{\substack{(r,s)\in \mathrm{row}_{ik}(j) \\ s(A_{ri},B_{sk})\leq 0}} \bar{y}_{rsik} = 0, \quad \sum_{\substack{(r,s)\in \mathrm{row}_{ik}(j) \\ s(A_{ri},B_{sk})\leq 0}} \bar{x}_{rs} = 1.$$

Here, we have to identify pairs of nodes that are both activated and are connected by an edge which is not activated. Since we have to check only pairs of nodes in the solution, this can be done quickly in time $O(n^2)$.

Branch-and-Bound. After a specified number of Lagrangian iterations, the problem is split into four subproblems. The entire branch-and-bound framework is described in (Andonov et al., 2011).

2.4 Algorithm Engineering

In most cases preprocessing is beneficial. The idea is to exclude nodes from the alignment graph if they provably cannot be part of an optimal alignment.

In order to do so, a good feasible solution is needed; it can, for example, be provided by the heuristic DALI algorithm. Then, for each alignment graph node, an overestimation of the the best alignment *including* this node is computed. If this upper bound is less than the score of a known feasible solution, the node cannot be part of the optimal solution and will be discarded. The benefit of preprocessing is twofold: first, the computation time decreases if only a subset of nodes in the alignment graph needs to be considered. Second, less memory is needed, because together with a node also all its incoming and outgoing edges can be discarded. Since each alignment graph node can be handled one after another, there is no memory issue during preprocessing. As a result, successful filtering thus allows to compute alignments of protein pairs that otherwise would not fit into memory.

Another attempt to handle the memory requirements of especially large proteins is the exclusion of inter-residue distances between residues that belong to different domains. The orientation of domains to each other is thus entirely disregarded during alignment. As a result, alignments are scored on the base of different criteria, and thus the optimal alignments in both cases may differ. We evaluate empirically the influence of omitting inter-domain distances on the alignment accuracy.

3 Data Sets and Experimental Setup

The SKOLNICK data set consists of 40 proteins with length between 97 and 255 residues belonging to 5 protein families. This easy dataset has been used extensively for clustering of protein structures, see for example (Caprara et al., 2004; Xie and Sahinidis, 2007; Malod-Dognin et al., 2010, 2011; Andonov et al., 2011). We align only protein pairs from the same family, which amounts to 164 SKOLNICK instances.

SCOPCath (Csaba et al., 2009) is a benchmark containing 6759 domains that are consistently classified in SCOP (Murzin et al., 1995, version 1.75) and CATH (Greene et al., 2007, version 3.2.0) and that have a pairwise sequence similarity of less than 50%. We align all SCOPCath domains with 30 up to 50 residues which belong to the same family (386 pairs), to different families but to the same superfamily (151 pairs), and to different superfamilies but the same fold (926 pairs). We limited the length to maximally 50 residues to obtain alignments for which our algorithm can explore multiple branch-and-bound nodes within a few minutes.

SISY and RIPC (Mayr et al., 2007; Berbalk et al., 2009) are datasets of manually curated structural alignments assembled from the Sisyphus collection (Andreeva et al., 2007), which are difficult for alignment programs because of repetitions, large indels, circular permutations, conformational variability, etc. The consolidated SISY and RIPC sets consist of 98 and 22 alignments, respectively. With consolidated we denote the subsets that have been consulted for evaluation in (Berbalk et al., 2009).

For comparison with DALI we use DaliLite version 3.3. DALI computes a number of alignments and ranks them according to z-score of protein unfolding units. Therefore, we parse all alignments returned by DALI and consider the one with largest DALI score. Note that maximizing the DALI score will also maximize the z-score of the entire alignment. Nonetheless, in an attempt to report and rank interesting high local similarities, DALI also computes z-scores for parts of the alignment, the protein unfolding units, in which case a suboptimal alignment can receive the highest z-score.

We compute pairwise alignments on cluster nodes each equipped with two quad core 2.26 GHz Intel Xeon processors and 24 GB of main memory running 64 bit Linux. In each branch-and-bound node we compute 1000 Lagrange iterations. For SISY, RIPC and SKOLNICK, a maximum running time of 30 CPU hours per instance is applied and for the short SCOPCath instances a time limit of 30 CPU minutes per instance.

4 Results and Discussion

We assess the capability of our algorithm to compute optimal alignments with respect to the DALI scoring function on (i) 164 SKOLNICK alignments, (ii) 1463 SCOPCath alignments, (iii) 98 SISY alignments and (iv) 22 RIPC alignments. On the two later datasets, we also evaluate the alignment accuracy with respect to the manually curated reference alignment, which is defined as the percentage of correctly aligned residue pairs. We compare our alignments and their scores to those determined by the DALI program. The results are summarized in Table 1 and Figure 5.

SKOLNICK. 136 of the 164 SKOLNICK alignments were computed to optimality (83%). For 123 alignments (75%), the heuristic DALI solution was improved, but never more than 3%, and for 38 (23%) the heuristic solution was proven to be optimal. Only three DALI alignments were slightly better than the corresponding DALIX alignment. Results for SKOLNICK indicate that DALI computes optimal or close to optimal alignments in the case of distinct structural similarities on family level. The results also demonstrate that it is feasible to compute structural alignments to optimality in the case of considerable structural similarity.

SCOPCath. When aligning the short SCOPCath domains, for 661 (45%) neither DALI nor DALIX could compute an alignment with positive z-score, especially on fold level. It is likely, but unfortunately not proven by our upper bounds, that no such alignment exists in many cases. This situation illustrates that it is difficult to design a scoring scheme and algorithm that reliably detects structural similarities on different classification levels and discriminates them from spurious similarities. An exact algorithm maximizing a “perfect” scoring function is expected to return a significant alignment for all protein pairs investigated in

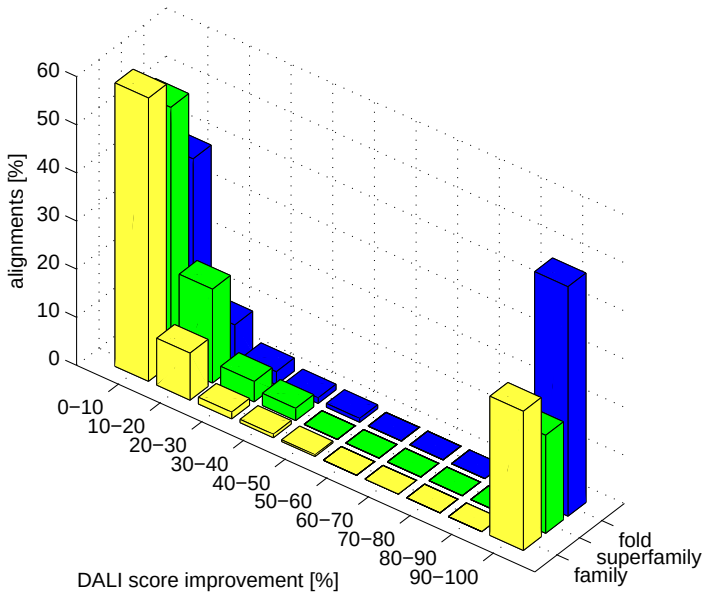

Fig. 5. The barplot bins the percentages of DALI score improvement for the cases in which the DALIX alignment has positive z-score and is better than the DALI alignment. On family level, these are 278, on superfamily level 118 and on fold level 258 alignments. The improvement is computed with respect to the DALIX alignment. The DALIX computation time limit is 30 CPU minutes. For most alignments, the score improvement is small. The large percentage of alignments with improvement between 90 and 100% traces back to the large number of protein pairs for which DALI falsely reports that there is no structural similarity.

	SKOLNICK	SCOPCath			SISY	RIPC
		Family	Superfamily	Fold		
Alignments	164	386	151	926	62	11
Positive z-score	164	359	141	302	61	11
DALIX optimal	136	143	14	31	11	2
DALI optimal	38	50	5	5	3	0
DALIX better	123	287	118	258	31	6
DALI better	3	16	14	30	27	5

Table 1. Comparison of DALIX and DALI alignments. The table lists the number alignments in each dataset in the first row and the number of alignments for which either DALIX or DALI detected an alignment with z-score greater 0 in the second row. Given that only alignments with z-score greater zero are significant and that DALI reports only significant alignments, we only consider those alignments in the following table rows. "DALIX optimal" denotes the number of DALIX alignments that have been computed to optimality and "DALI optimal" the DALI alignments thereof that are also provably optimal. "DALIX better" lists the number of DALIX alignments with higher DALI score than the DALI alignment. "DALI better" denotes the number of DALI alignments that are better than the (not yet optimal) DALIX alignment. The DALIX computation time limit for SCOPCath alignments is 30 CPU minutes and for all other data sets 30 CPU hours.

this paper. Given the DALI score and the DALI and DALIX algorithms, this is not the case. We thus exclude protein pairs from the analysis for which no algorithm returns an alignment with positive z-score.

From the 359 short SCOPCath alignments of domains from the same family, 143 (40%) were solved to optimality within the time limit of 1800 seconds, and most within a few seconds. There are differences between optimal alignments and heuristic ones computed by the DALI program, which are quantified in Figure 5. First, DALI fails to detect 83 significant alignments with z-scores up to 5 and falsely reports that there are no structural similarities. DALIX computes 143 alignments (40%) to optimality. From these cases, in which the provable top-scoring alignment is known, DALI returns 50 optimal and 93 close to optimal alignments. Altogether, our exact algorithm improves the heuristic solution in 287 cases (80%) by median 7%. In 16 cases, the solution returned by the exact algorithm after 1800 seconds is worse than the heuristic solution.

Computing exact alignments becomes more difficult when structural similarity gets less pronounced. For the 141 SCOPCath alignments of proteins that share the same superfamily, but not the same family, only 14 (10%) are computed to optimality within 1800 seconds. Nonetheless, the exact algorithm improves the heuristic alignments in 118 cases (84%), as visualized in Figure 5. In 24 of them, DALIX returns a significant alignment that is entirely missed by DALI. In 14 cases, the DALI heuristic alignment has larger DALI score than the DALIX alignment.

On fold level, only 31 of the 302 alignments were computed to optimality. 258 alignments (85%) returned by the exact algorithm are better than those returned by the heuristic algorithm, by a median of 27%, see Fig. 5. Also here, the DALI score for alignments produced by DALI or DALIX is usually very similar, but

the exact algorithm detects significant alignments with z-scores up to 2.5 that are missed by the heuristic. In 123 cases (41%), DALIX determines a significant alignment that is missed by DALI. In 30 cases the heuristic alignment is better than the one returned by our algorithm.

Figure 4 visualizes the alignment traces of three alignments, from family, superfamily or fold level, respectively. These are the instances for which the DALIX alignment improves the DALI alignment with the largest relative score difference from all alignments of the respective similarity level.

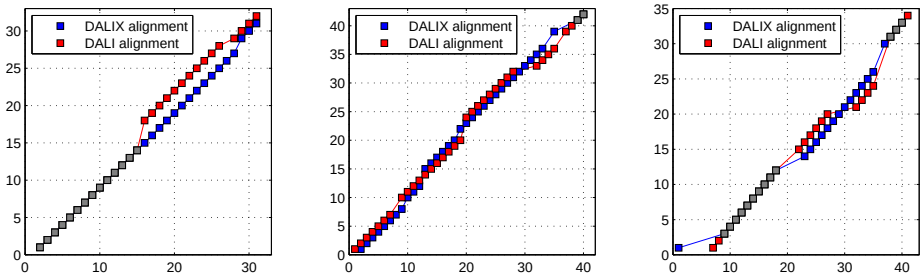

Fig. 6. Three SCOPCath examples in which the DALIX alignment improves the DALI alignment significantly. Proteins from common family, superfamily and fold are aligned (SCOPCath IDs d2glia1 vs. d2glia4, d1b9wa2 vs. d1dx5j3, and d1pfwa3 vs. d1ryta2). Each alignment has the largest percentage of score difference of alignments from the respective level of similarity (45%, 35% and 47%). Residues aligned by DALIX are colored blue, residues aligned by DALI red and residues aligned by both methods gray.

SISY and RIPC. If structural similarities are less pronounced or locally confined, determining the optimal alignment becomes inordinately more difficult. Furthermore, protein length is a problem for our algorithm, since the number of distance pairs grows quadratic with protein length. From the SISY set, whose difficulty is also confirmed by low alignment z-scores, only 62 alignments fit into memory. From these, 11 were solved to optimality in, on average, about 6000 seconds. Three of these optimal alignments were also detected by the DALI heuristic, and the remaining 8 non-optimal heuristic alignments were less than 3% worse than the optimal ones. Altogether, our exact algorithm improves the heuristic alignments in 31 cases by a median of 2%. In 27 instances, the heuristic alignment performs by a median of 19% better than the the alignment returned by our algorithm. The reason is that the instances in the SISY set are large and of subtle similarity, which is both disadvantageous for our algorithm.

From the 22 RIPC alignments, 11 fit into memory. Of those, two were computed to optimality. An improvement of median 4% over the DALI solution was found in 6 cases. The heuristic solution is better than the solution of our algo-

rithm for 5 alignments, with a large median DALI score difference of 100%. In 3 cases DALIX thus does not detect an existing alignment with positive z-score.

On SISY and RIPC we observe that, within the given time limit, our exact algorithm fails to produce good alignments for large or remotely similar protein pairs. Furthermore, we find that improving the DALI score does not necessarily implicate a higher alignment accuracy with respect to manually curated reference alignments: 13 DALIX alignments from SISY have larger DALI score than the DALI alignment, but slightly less alignment accuracy with respect to the reference alignment. Only for three alignments an increased DALI score also results in a slightly increased alignment accuracy. For the RIPC data set, in two cases an improved DALI score increased the alignment accuracy, but in two other cases alignment accuracy decreased. Inspection of a few of these instances using CSA (Wohlers et al., 2012), our web server for comparative structural alignment, shows that in DALIX alignments with larger DALI score but less alignment accuracy often short, additional gaps have been inserted rendering the alignment more scattered and less intuitive than the respective DALI alignment.

We evaluated the benefit of preprocessing as described in Section 2.4. Four more SISY and one more RIPC alignments fit into memory. Two of these SISY alignments wer computed to optimality. In most cases the alignment returned when additional preprocessing is used is better with respect to DALI score than the one obtained without preprocessing. In those instances that were solved, the overall running time including preprocessing is in most cases significantly smaller than without preprocessing, although the preprocessing itself takes, depending on protein length, up to 13 minutes. Similar observations hold for the RIPC alignments. Here, one previously unsolved protein pair is aligned optimally when preprocessing is used, in as little as 91 seconds.

Large proteins that share only little structural similarity cause memory and performance problems for our algorithm. It was thus only possible to fit 62 SISY (63%) and 11 RIPC (50%) alignments into memory. In order to reduce memory consumption while keeping alignments very close to the ones computed by the DALI program, we made another SISY and RIPC evaluation in which we excluded all inter-domain distances. Now, 92 SISY (94%) and 19 RIPC (86%) alignments fit into memory. While it allows to compute alignments for larger proteins and in shorter computing time, omitting inter-domain distances has little influence on alignment accuracy with respect to the manually curated SISY and RIPC reference alignments.

5 Conclusion

We presented the first exact general algorithm for distance matrix alignment and implemented and evaluated it for the DALI scoring function. We used our implementation DALIX to benchmark the popular DALI structural alignment method. We found that computing a DALI alignment to optimality is feasible if the proteins are not too large or if there is a clear structural similarity. In these cases we noticed that DALI alignments are often not optimal, but nonetheless almost

always very close to optimal. Further, for short protein pairs we detected that although DALI rarely returns a poor alignment, it tends to entirely miss structural similarities and wrongly does not return any alignment. These findings confirm the high quality of heuristic DALI alignments and identify possible improvements.

Acknowledgements This work was supported partly by DFG grant KL 1390/2–1 and by an INRIA internship grant. R. Andonov is supported by BioWIC ANR-08-SEGI-005 project. Computational experiments were sponsored by the NCF for the use of supercomputer facilities with financial support from NWO.

Bibliography

R. Andonov, N. Malod-Dognin, and N. Yanev. Maximum contact map overlap revisited. *J Comput Biol*, 18(1):27–41, 2011.

A. Andreeva, A. Prlić, T. J. Hubbard, and A. G. Murzin. SISYPHUS–structural alignments for proteins with non-trivial relationships. *Nucleic Acids Res*, 35 (Database issue):253–259, 2007.

C. Berbalk, C. S. Schwaiger, and P. Lackner. Accuracy analysis of multiple structure alignments. *Protein Sci*, 2009.

H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Res*, 28(1):235–242, 2000.

A. Caprara, R. Carr, S. Istrail, G. Lancia, and B. Walenz. 1001 optimal PDB structure alignments: integer programming methods for finding the maximum contact map overlap. *J Comput Biol*, 11(1):27–52, 2004.

G. Csaba, F. Birzele, and R. Zimmer. Systematic comparison of SCOP and CATH: a new gold standard for protein structure analysis. *BMC Struct Biol*, 9:23–23, 2009.

A. Godzik and J. Skolnick. Flexible algorithm for direct multiple alignment of protein structures and sequences. *Comput Appl Biosci*, 10(6):587–596, 1994.

L. H. Greene, T. E. Lewis, S. Addou, A. Cuff, T. Dallman, M. Dibley, O. Redfern, F. Pearl, R. Nambudiry, A. Reid, I. Sillitoe, C. Yeats, J. M. Thornton, and C. A. Orengo. The CATH domain structure database: new protocols and classification levels give a more comprehensive resource for exploring evolution. *Nucleic Acids Res*, 35(Database issue):291–297, 2007.

L. Holm and C. Sander. Protein structure comparison by alignment of distance matrices. *J Mol Biol*, 233(1):123–138, 1993.

L. Holm and C. Sander. Dictionary of recurrent domains in protein structures. *Proteins*, 33(1):88–96, 1998.

T. Kawabata and K. Nishikawa. Protein structure comparison using the markov transition model of evolution. *Proteins*, 41(1):108–122, 2000.

R. H. Lathrop. The protein threading problem with sequence amino acid interaction preferences is NP-complete. *Protein Eng*, 7(9):1059–1068, 1994.

N. Malod-Dognin, R. Andonov, and N. Yanev. Maximum cliques in protein structure comparison. In P. Festa, editor, *Experimental Algorithms*, volume 6049 of *LNCS*, pages 106–117. Springer–Verlag, Berlin, Heidelberg, 2010.

N. Malod-Dognin, M. L. Boudic-Jamin, P. Kamath, and R. Andonov. Using dominances for solving the protein family identification problem. In T. M. Przytycka and M.-F. Sagot, editors, *WABI*, volume 6833 of *Lecture Notes in Computer Science*, pages 201–212. Springer, 2011.

G. Mayr, F. S. Domingues, and P. Lackner. Comparative analysis of protein structure alignments. *BMC Struct Biol*, 7:50–50, 2007.

A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol*, 247(4):536–540, 1995.

W. R. Taylor and C. A. Orengo. Protein structure alignment. *Journal of Molecular Biology*, 208(1):1–22, 1989.

I. Wohlers, F. S. Domingues, and G. W. Klau. Towards optimal alignment of protein structure distance matrices. *Bioinformatics*, 26(18):2273–2280, 2010.

I. Wohlers, R. Andonov, and G. W. Klau. Algorithm engineering for optimal alignment of protein structure distance matrices. *Optimization Letters*, 5(3): 421–433, 2011.

I. Wohlers, N. Malod-Dognin, R. Andonov, and G. W. Klau. CSA: Comprehensive comparison of pairwise protein structure alignments. Research Report RR-7874, INRIA, 2012.

W. Xie and N. V. Sahinidis. A reduction-based exact algorithm for the contact map overlap problem. *J Comput Biol*, 14(5):637–654, 2007.