Aligning Protein Structures Using Distance Matrices and Combinatorial Optimization

Inken Wohlers* Lars Petzold† Francisco S. Domingues‡ Gunnar W. Klau*

Abstract: Structural alignments of proteins are used to identify structural similarities. These similarities can indicate homology or a common or similar function. Several, mostly heuristic methods are available to compute structural alignments.

In this paper, we present a novel algorithm that uses methods from combinatorial optimization to compute provably optimal structural alignments of sparse protein distance matrices. Our algorithm extends an elegant integer linear programming approach proposed by Caprara *et al.* for the alignment of protein contact maps. We consider two different types of distance matrices with distances either between C_α atoms or between the two closest atoms of each residue. Via a comprehensive parameter optimization on HOMSTRAD alignments, we determine a scoring function for aligned pairs of distances. We introduce a negative score for non-structural, purely sequence-based parts of the alignment as a means to adjust the locality of the resulting structural alignments.

Our approach is implemented in a freely available software tool named PAUL (Protein structural Alignment Using Lagrangian relaxation). On the challenging SISY data set of 130 reference alignments we compare PAUL to six state-of-the-art structural alignment algorithms, DALI, MATRAS, FATCAT, SHEBA, CA, and CE. Here, PAUL reaches the highest average and median alignment accuracies of all methods and is the most accurate method for more than 30% of the alignments. PAUL is thus a competitive tool for pairwise high-quality structural alignment.

1 Introduction

Background. Structural alignments of proteins help identify structural similarities. They are used to detect homologous proteins, to identify common structural elements, and to determine protein function. Frequently, the function of a protein is defined by its three-dimensional structure, and protein structure is often more conserved during evolution than protein sequence. Therefore, structural alignment is especially useful to detect remotely homologous proteins with low sequence identity, which lie either in the twilight zone [Doo86] of 20% to 35% sequence identity or in the midnight zone [Ros97] of less than 20% sequence identity. Furthermore, structural alignment is applied to identify new protein folds or to map protein structures to already established folds. Detected structural similarities are used effectively for functional annotation [YYS+04].

There are two established approaches to compute protein structural alignments: minimiz-

*CWI, P.O. Box 94079, 1090 GB Amsterdam, Netherlands, {inken.wohlers,gunnar.klau}@cwi.nl
†Freie Universität Berlin, 14195 Berlin, Germany
‡Max Planck Institute for Informatics, 66123 Saarbrücken, Germany

ing the root mean square deviation (RMSD) of rigid body superposition and maximizing the score for an assignment of distance matrix rows and columns. A popular heuristic algorithm of the second type is DALI [HS93]. DALI scans in a first step protein distance matrices for similar distance patterns by computing a similarity score for aligning fragments of six residues. In a second step combinations of non-overlapping fragments are repeatedly chosen in a random fashion. Each set of fragments makes up an alignment and is evaluated using a scoring function. Finally the alignment with highest score is reported. Several other algorithms also aim at finding good combinations of aligned fragment pairs, *e.g.*, CE [SB98] and FATCAT (flexible structure alignment by chaining aligned fragment pairs allowing twists) [YG03]. Other methods like MATRAS (Markovian transition of structure evolution) [Kaw03] match in a first step secondary structure elements and compute an alignment on atomic level in a second step. A sequence-order independent approach to compute alignments is geometric hashing, which is applied by CA [BFNW93]. The method SHEBA (structural homology by environment-based alignment) [JL00] compares in a first step lists of primary, secondary and tertiary structure characteristics and then improves the initial alignment using weighted RMSD. Further state-of-the-art approaches are SSAP [TO89], which is based on double dynamic programming, PPM [CBZ08], a method that minimizes the cost of morphing one structure into the other, TM-ALIGN [ZS05] that maximizes the TM-score, and PROTDEFORM [RSWD09] and MATT [MBC08], which align proteins in a flexible fashion. Furthermore, structural alignments have also been computed by aligning protein contact maps [CCI+04].

Contribution. In this paper, we present a structural alignment approach based on combinatorial optimization. In our approach, which builds upon an algorithm for the alignment of protein contact maps by Caprara *et al.* [CCI+04], we align sparse distance matrices. We compute an alignment by maximizing a function that scores aligned distances. We tailor our method specifically towards high-quality pairwise alignments. In order to efficiently use the elegant integer linear programming approach of [CCI+04] we determine a suitable distance threshold and scoring function, decrease the number of variables in the integer linear program and add a parameter that scores non-structural, purely sequence-based parts of the alignment in order to balance global against local alignment. We optimize our method for C_α distance matrices as well as for all-atom distance matrices that contain the minimum distance between any pair of atoms of two residues. In this study we investigate which distances should be included in the integer linear program in order to increase the accuracy of pairwise alignments. We did not optimize for speed—this issue will be dealt with in future work. Our approach is implemented in the freely available software tool PAUL (protein structural alignment using Lagrangian relaxation). We evaluate PAUL on the challenging SISY data set [MDL07] comparing it to six state-of-the-art structural alignment tools. PAUL reaches higher average and median alignment accuracies than any of the other methods.

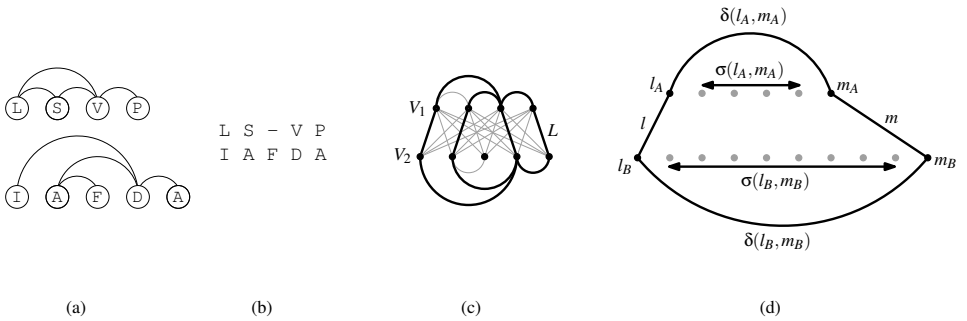


Figure 1: Maximum contact map overlap problem. (a) Two protein contact maps. (b) Alignment of the two proteins. (c) Corresponding solution in the graph problem. Alignments are characterized by non-crossing matches, or *traces*, in the complete bipartite alignment graph $(V_1 \cup V_2, L)$ [Kec93]. Here, vertices in V_1 and V_2 denote the residues of the two proteins, resp., 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 contact map overlap, in the example there are three shared contacts (also shown in bold). (d) A pair of aligned distances. Functions $\delta(\cdot, \cdot)$ and $\sigma(\cdot, \cdot)$ denote the distance between two residues with respect to their three-dimensional coordinates and sequential position, respectively.

2 Methods

Combinatorial approach to structural alignment. In [CCI+04], the authors give an algorithm to compute pairwise alignments of two protein structures that maximize the number of common contacts. Two residues of a protein are in contact if they are in some sort of chemical interaction, *e.g.*, by hydrogen bonding. However, a simple distance criterion is used: whenever the distance between two residues is below a predefined threshold, the residues are considered to be in contact. Caprara *et al.* have introduced the *maximum contact map overlap* problem and have given an integer linear programming (ILP) formulation. They propose to solve the ILP using an elegant Lagrangian relaxation approach.

The underlying ILP formulation relies on a reformulation of the structural alignment problem as a graph problem. Figure 1 explains the relation. In their ILP approach, Caprara *et al.* introduce binary variables x_l for each alignment edge $l \in L$ and binary variables y_{lm} for each potentially shared common contact represented 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 x and y and find the largest set of common contacts using the objective function $\max \sum_{(l,m)\in\binom{L}{2}} y_{lm}$. For a detailed description, refer to [CCI+04].

We extend the approach by Caprara *et al.* by replacing the rigid contact definition and taking into account the three-dimensional and sequential distances between the residues in order to align inter-residue distances. 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 Fig. 1(d). We use two distance measures $\delta(\cdot, \cdot)$ and $\sigma(\cdot, \cdot)$ that denote the distance between two residues with respect to their three-dimensional coordinates and sequential position, resp., and are

now able to align inter-residue distances instead of contacts. To this end, we replace the objective function $\max \sum_{(l,m)\in\binom{L}{2}} y_{lm}$ by

$$\max \sum_{(l,m)\in\binom{L}{2}} w_{lm} y_{lm} + \sum_{l\in L} c\, x_l \quad \text{, where} \tag{1}$$

$$w_{lm} = \begin{cases} \max\{0, \theta_R - \Delta_{lm}\} & \Delta_{lm} \leq \Delta_t \text{ and } \Gamma_{lm} \leq \Gamma_t \\ 0 & \text{otherwise} \quad , \end{cases} \tag{2}$$

with $\Delta_{lm} = |\delta(l_A, m_A) - \delta(l_B, m_B)|$ and $\Gamma_{lm} = |\sigma(l_A, m_A) - \sigma(l_B, m_B)|$. Here, θ_R, Δ_t, Γ_t, and c are constant parameters. Our choices of (1) and (2) are motivated by the following considerations.

1. A scoring function for pairs of inter-residue distances from two proteins A and B should be symmetric with respect to the order of A and B. This is achieved by taking absolute values.
2. Aligning similar distances is more preferable than aligning dissimilar ones, *i.e.*, the contribution of a pair of aligned distances to the objective function should decrease with increasing difference Δ_{lm}. Inspired by the *rigid similarity* measure introduced by Holm and Sander in their paper [HS93] on DALI, we use the term $\theta_R - \Delta_{lm}$ to score pairs of distances. Parameter θ_R modulates the score such that in one extreme example slight differences in distance cause great differences in score and in the other extreme example all combinations of distances have the same score—this second scoring is identical to the contact map overlap. See also Fig. 2(b).
3. Analogous to contact map alignment, the overall time and space complexity of our method is $O(|E_A|, |E_B|)$, where E_A and E_B are the numbers of distances in the ILP from protein A and B respectively. The major restriction of solving protein structural alignment to provable optimality is therefore the high demand of computational resources. In principle, each pair of distances has to be considered explicitly in the ILP, leading to $\binom{n_A}{2}\binom{n_B}{2}$ y-variables, where n_A and n_B are the number of residues in proteins A and B, resp. The Lagrangian relaxation approach is highly sensitive to the number of y-variables, making it practically infeasible to include all pairs of distances. Therefore, we consider only such distance combinations that are likely to denote structural similarity between the two proteins and derive a distance threshold d_t and a threshold for distance differences Δ_t. We find that sequential distance differences Γ_{lm} of aligned distances are typically low, but are aware that by applying a threshold Γ_t we neglect distances between different secondary structure elements that are divided by an insertion or deletion greater than Γ_t. Therefore we do not apply a threshold Γ_t in this study. Applying thresholds leads to a large number of variables y_{lm} with $w_{lm} = 0$. Due to the nature of the ILP formulation, we can safely omit variables with zero coefficients.
4. Due to the structure of the ILP we do not have the possibility to penalize aligned distances by using negative scores w_{lm}. Therefore we penalize parts of the alignment without structural conservation by giving each alignment edge a negative score c. Thus, alignment edges will only be chosen if they contribute significantly to multiple pairs of aligned distances with large weight. This prevents the alignment of residues that do not

indicate sufficient structural similarity. If we decrease this penalty, we can tune PAUL towards local alignment by concentrating on structurally extremely similar parts while neglecting less similar parts. On the other side we can tune PAUL also towards rigorous global alignment by increasing the penalty or setting it to zero.

Implementation. We implemented the novel structural alignment algorithm as the freely available package PAUL within the C++ software library PLANET LISA [K$^+$]. PAUL supports different input formats, *e.g.*, PDB files, lists of pre-selected distances or complete distance matrices. Distance matrix representations are currently built internally in either of two modes: based on the distances between the C_α atoms of residues or based on the minimum distance between each pair of atoms of residues. While the alignment of C_α distances aims at finding equal or similar protein backbone conformations, the alignment of all-atom distances shows similar residue interactions in the two proteins. Beside the type of distances, scoring function parameters can also be chosen. In this way additional information about the pair of proteins that are aligned can be incorporated. For instance, the penalty c can be adjusted to favour global or local alignment. By default, the optimized scoring function parameters for C_α and all-atom distances reported in this paper are used.

Experimental setup for parameter setting, optimization, and evaluation. To determine good and robust parameters for the distance difference threshold Δ_t, the steepness θ_R, and the penalty c we use structure-based alignments from the Homologous Structure Alignment Database (HOMSTRAD, Oct 2008 release) [MDBO98]. As these alignments are manually curated by experts, we consider them as gold standard reference alignments. From HOMSTRAD we consider only protein families with exactly two members from the twilight or midnight zone of sequence identities below 35%. Hereby we define sequence identity as the number of identically aligned residues divided by 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.

In a preprocessing step we compute histograms of aligned distances over the training set alignments. Fig. 2(a) displays the results for C_α distances. For all-atom distances the distribution is similar, but shifted to smaller distances. For close distances of less than 12Å we observe distinct peaks for certain aligned distances. These peaks represent typical distances within secondary structure elements and within super-secondary structures. The histograms help identify distance thresholds for C_α and all-atom distance matrices that are qualitatively equivalent in terms of overall number of distances included in the ILP as well as in terms of inclusion of biological features. We optimized parameters for the distance thresholds $d_t \in \{7.5$Å, 8Å,$\ldots$, 10Å$\}$ and $d_t \in \{5$Å, 5.5Å,$\ldots$,7Å$\}$ for C_α and all-atom distance matrices, resp.

We carry out a parameter sweep in order to optimize the scoring function parameters θ_R, Δ_t and c. We use 7 nodes equipped each with two quad core 2.33 GHz Intel Xeon processors and 8 GB of main memory running 64 bit Linux. On each node we compute 4 PAUL alignments in parallel using OpenMP. We choose a maximum time limit of 90 CPU s and a maximum number of $1\,000$ Lagrange iterations for each computation. In a first

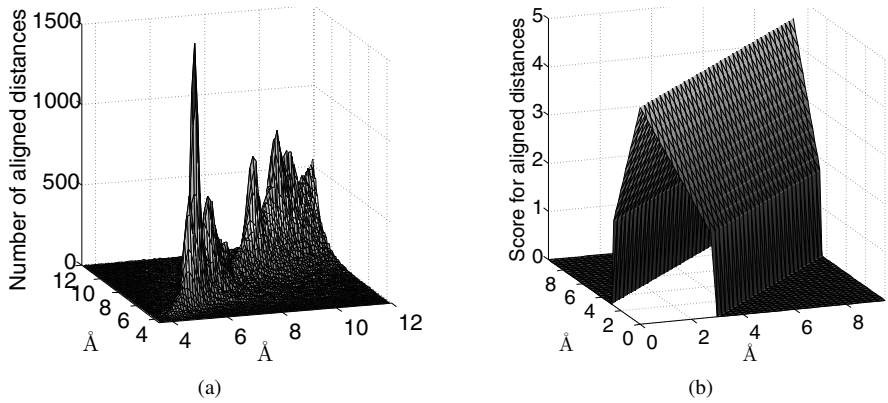


Figure 2: (a) Number of aligned C_α distances over all HOMSTRAD training set alignments. (b) Scoring function for C_α distance matrices ($d_t = 9.5$, $\theta_R = 4.5$ and $\Delta_t = 3$).

broad sweep we choose 10 values for the steepness of the scoring function, θ_R, in such a way that the angle is divided equally. We then refine the sweep by focusing on an interval of good parameter values and equally divide this interval again, obtaining 10 more values for θ_R. For the maximum distance difference we evaluate $\Delta_t \in \{1.5, 2, \ldots, d_t - 1.5\}$ for C_α matrices and $\Delta_t \in \{0.5, 1, \ldots, d_t - 0.5\}$ for all-atom matrices. The sequence penalty ranges over $c \in \{0, -\frac{1}{2}\theta_R, \ldots, -3\theta_R\}$. Since aligning two identical distances receives a maximum score of θ_R, the range of penalty values covers the cases of aligning two residues if they maintain at least 0, 1, 2, or 3 aligned distances of maximum score. We compute the average alignment accuracy for each parameter set $(\theta_R,\ \Delta_t, c)$, resulting in more than 150 evaluations of the full training set for each distance threshold. We then apply 10-fold cross-validation in order to assess the performance of PAUL on the HOMSTRAD training set alignments. We use the best parameter set for C_α and all-atom distances resp. to align the HOMSTRAD test set alignments and compare PAUL performance to DALI.

Experimental setup of computational study. We use the parameters optimized on the HOMSTRAD data set to compare PAUL with the state-of-the-art structural alignment programs DALI, MATRAS, FATCAT, SHEBA, CA and CE on a second, distinct data set, the SISY set [MDL07, L^+]. This set is assembled from SISYPHUS [APHM07], a manually curated database for alignments of proteins with non-trivial relationships. It consists of 130 very diverse reference alignments: the lengths of the protein chains vary greatly, from 32 up to 1 283 residues, as do the lengths of the number of aligned residues, from 17 to 372. For aligning the SISY set we use a maximum runtime of 30 minutes per alignment, in order to exploit the benefit of using a high distance threshold. Note that, depending on the pair of proteins, the actual runtime in which we observe improvements is usually a lot shorter (see HOMSTRAD), but in order to proof the optimality of a solution a longer runtime is needed. However, in terms of speed our method is not yet competitive to others, therefore we did not compare runtimes.

	PAUL	MATRAS	DALI	FATCAT	SHEBA	CA	CE
average %	**72.93**	71.83	69.44	62.40	59.43	51.45	50.13
median %	**92.48**	91.42	90.96	78.30	84.41	59.55	57.43

Table 1: Results on the SISY data set. Average and median alignment accuracies of different state-of-the-art structural alignment algorithms. Overall best values denoted in bold.

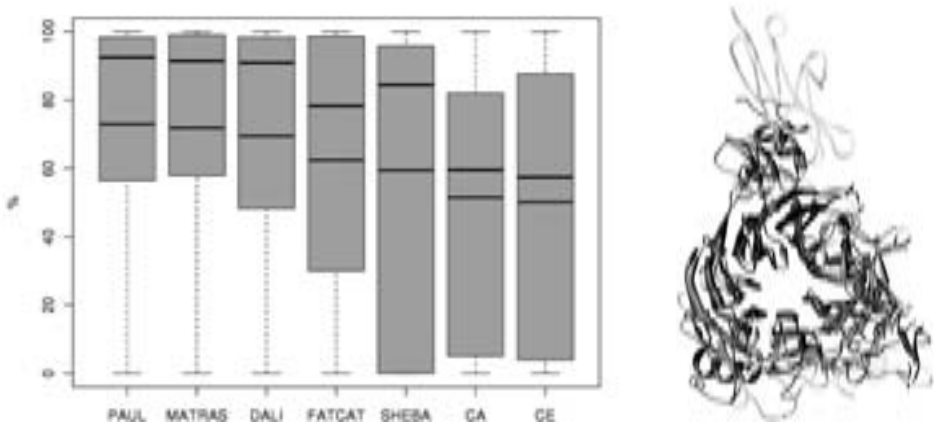
3 Results

Optimized scoring function. The best distance thresholds for C_α distance matrices were 8Å with an alignment accuracy of 87.56% followed by 8.5Å with 87.34% and closely followed by 9.5Å with 87.30%. The corresponding optimized parameters are similar, for $d_t = 9.5$ they are $\theta_R = 4.5$, $\Delta_t = 3$ and $c = -4.5$. For all-atom distance matrices the best parameters are $d_t = 5.5$, $\theta_R = 3.5$, $\Delta_t = 3$ and $c = -1.75$ with an alignment accuracy of 87.23%. Although the parameters vary over a large range, the resulting optimized parameters are of the same order of magnitude for all distance thresholds d_t, with alignment accuracies between 86.77 and 87.56% for C_α distances and between 85.17 and 87.23 for all-atom distances. The result of the 10-fold cross-validation over all distance thresholds and parameter sets amounts to 86.76% for C_α and 86.42% for all-atom distances, compared to 85.08% alignment accuracy achieved by DALI. For a visualization of the optimized scoring function for C_α distances refer to Fig. 2(b).

We test our optimized parameters on the HOMSTRAD test set. For C_α distance matrices PAUL reaches an average alignment accuracy of 85.86% for $d_t = 8$, of 85.49% for $d_t = 8.5$, and of 86.56% for $d_t = 9.5$; all-atom distance matrices with $d_t = 5.5$ reach 86.22%. The alignment accuracy for C_α distances and $d_t = 9.5$ is slightly higher than the average alignment accuracy of DALI alignments, which amounts to 86.32%. Based on these results, we decide to use C_α distance matrices with $d_t = 9.5$ for the evaluation on the SISY data set.

Results on SISY data set. We investigate the alignment accuracy in terms of percentages of correctly aligned residues on the more challenging SISY data set and compare PAUL's performance to six other state-of-the-art structural alignment algorithms. Table 1 contains the average and median alignment accuracies for the set of 130 alignments. Fig. 3(a) shows the distributions of the percentages of alignment accuracies for PAUL and each of the other structural alignment methods using box-and-whisker plots. Fig. 3(b) visualizes a difficult SISY alignment, for which PAUL outperforms the other structural alignment methods.

We observe that PAUL alignments shows higher average and median accuracy than any other method. Furthermore, according to two-sided Wilcoxon signed-rank tests with paired observations, PAUL matches the SISY gold standard alignments significantly better than SHEBA, CA, and CE. Additionally, we investigate the correlation between alignment accuracy values using Pearson correlation coefficients. These are around 0.5 for any pair of methods and are thus generally low, whereas the correlation between PAUL and MATRAS has a Pearson correlation coefficient of 0.56 and between PAUL and DALI of 0.49.



(a) (b)

Figure 3: (a) Box-and-whisker plots display median and quartiles of the distributions of percentages of alignment accuracies for the SISY set for PAUL, MATRAS, DALI, FATCAT, SHEBA, CA and CE. Additionally, blue lines denote the average alignment accuracies. (b) PAUL alignment of semaphorin 4D, PDB 1olz chain A, (grey) with hepatocyte growth factor, PDB 1shy chain B, (purple). Protein lengths are 621 and 499 residues, resp. The proteins are oriented according to the optimal superposition of the matching residues given by PAUL. The alignment given by PAUL is mostly correct, with an alignment accuracy of 94.74%; the other methods generate alignments with lower accuracy (DALI 86.84%, FATCAT 81.58%, MATRAS 57.89%, SHEBA 10.53%, CA 2.63%, CE 0%).

4 Discussion

We suggest a novel structural alignment algorithm that is based on aligning small inter-residue distances using techniques from combinatorial optimization. By considering each combination of distances explicitly in our integer linear program we are able to solve the structural alignment problem on single-residue level and potentially to optimality without applying heuristics. This has several advantages. First of all, only a method that provides provably optimal alignments with respect to the scoring function allows to question, assess, and validate the underlying model, which is, in the case of structural alignment, the measure that evaluates structural similarity. Provably optimal solutions allow to attribute poor alignments to the measure of structural similarity that we maximize. For heuristic methods, however, a corrupt alignment might be suboptimal and then may have to be attributed to a poor search algorithm.

In order to be able to handle the combinatorial complexity in an explicit, non-heuristic manner, we have to restrict our method to sparse distance matrices and accept a significantly longer runtime than other, heuristic methods. PAUL's running time highly depends on protein length and protein similarity and may vary significantly. Therefore, in terms of improving the running time and estimating the status of the solution process, a lot of work still needs to be done. However, using the SISY set, we show that our scoring function and problem formulation is capable of finding difficult similarities, and on the HOM-

STRAD data set we show that this can also be done in shorter time scales, because PAUL achieves higher alignment accuracies than DALI—on the training set, as determined by cross-validation, as well as on the test set. Furthermore, we demonstrate that by aligning only small inter-residue distances, we still can compute alignments as good as or better than alignments computed by DALI, a heuristic structural alignment method that aligns complete inter-residue distance matrices.

There are two aspects that influence the performance of PAUL. Firstly, this is the suitability of scoring function parameters. Optimal or close to optimal parameters are of the same order of magnitude for different distance thresholds d_t. This denotes a common distance difference Δ_t, at which a majority of pairs of distances is structurally non-significant as well as a common preference for differentiated scoring of aligned distances, denoted by θ_R. The second aspect is the distance threshold d_t itself and the resulting computation time. Including more distances in the problem description renders the computation of a good alignment gradually more difficult, inefficient and thus time-consuming. This effect has to be counterbalanced by a gain of accuracy in describing protein structure, which leads to an overall higher alignment accuracy. Remarkably, different distance thresholds d_t and thus different numbers of distances in the ILP led to similar alignment accuracies on HOMSTRAD alignments. Therefore, we find that higher distance thresholds increase alignment accuracy, however, only when combined with a significantly longer runtime. In order to assess the importance of the penalty c we use different penalties to compute SISY alignments. We find that PAUL almost always finds an alignment as good as the best alignment from any of the other six methods. Therefore PAUL almost never fails due to algorithmic problems, but the balance between global and local alignment is crucial.

On the challenging SISY set, PAUL reaches the highest average and median alignment accuracies. This illustrates the soundness of our approach and its capability to detect structural similarity even in difficult cases. An example is given in Fig. 3(b). For more than 30% of the alignments PAUL achieves the maximum alignment accuracy that is reached by the seven structural alignment methods. In addition to its good performance, PAUL computes 21 alignments to provable optimality and thus with maximum score with respect to the scoring function. On the SISY set PAUL alignment accuracies correlate poorer to DALI than to MATRAS alignment accuracies, despite the common approach of aligning inter-residue distances. This might be attributed to qualitatively different scoring functions, to the fact that PAUL aligns only sparse distance matrices, and to the restriction of DALI to compute scores based on fragments and not on single-residue level. PAUL as well as DALI alignments benefit from a high degree of flexibility, because the approach of aligning distances instead of computing rigid superpositions allows to detect similarities of high RMSD, for which other algorithms need to introduce twists. The results on the SISY set thus demonstrate that PAUL is a beneficial tool for high-quality alignments, on its own as well as when complementing other structural alignment methods.

Acknowledgements. We thank Peter Lackner for providing the SISY data set and the results of the structural alignment methods with which we compare PAUL and Ingolf Sommer for valuable comments and initiation of the authors' cooperation. This work has been partly supported by DFG grant KL 1390/2-1. Computational experiments were sponsored by the NCF for the use of supercomputer facilities, with financial support from NWO.

References

[APHM07] 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, Jan 2007.

[BFNW93] O Bachar, D Fischer, R Nussinov, and H Wolfson. A computer vision based technique for 3-D sequence-independent structural comparison of proteins. *Protein Eng*, 6(3):279–288, Apr 1993.

[CBZ08] G Csaba, F Birzele, and R Zimmer. Protein structure alignment considering phenotypic plasticity. *Bioinformatics*, 24(16):98–104, Aug 2008.

[CCI+04] 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.

[Doo86] R F Doolittle. *Of URFs and ORFs: a primer on how to analyze derived amino acid sequences*. University Science Books, Mill Valley, CA, USA, 1986.

[HS93] L Holm and C Sander. Protein structure comparison by alignment of distance matrices. *J Mol Biol*, 233(1):123–138, Sep 1993.

[JL00] J Jung and B Lee. Protein structure alignment using environmental profiles. *Protein Eng*, 13(8):535–543, Aug 2000.

[K+] G W Klau et al. `http://planet-lisa.net`. Accessed 21 May 2009.

[Kaw03] T Kawabata. MATRAS: A program for protein 3D structure comparison. *Nucleic Acids Res*, 31(13):3367–3369, Jul 2003.

[Kec93] J D Kececioglu. The maximum weight trace problem in multiple sequence alignment. In *Proc 4th Annual Symposium on Combinatorial Pattern Matching (CPM 93)*, volume 684 of *LNCS*, pages 106–119. Springer–Verlag, 1993.

[L+] P Lackner et al. `http://biwww.che.sbg.ac.at/RSA/`. Accessed 8 May 2009.

[MBC08] M Menke, B Berger, and L Cowen. Matt: local flexibility aids protein multiple structure alignment. *PLoS Comput Biol*, 4(1), Jan 2008.

[MDBO98] K Mizuguchi, C M Deane, T L Blundell, and J P Overington. HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Sci*, 7(11):2469–2471, Nov 1998.

[MDL07] G Mayr, F S Domingues, and P Lackner. Comparative analysis of protein structure alignments. *BMC Struct Biol*, 7:50–50, 2007.

[Ros97] B Rost. Protein structures sustain evolutionary drift. *Fold Des*, 2(3):19–24, 1997.

[RSWD09] J Rocha, J Segura, R C Wilson, and S Dasgupta. Flexible structural protein alignment by a sequence of local transformations. *Bioinformatics*, 25(13):1625–1631, Jul 2009.

[SB98] I N Shindyalov and P E Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng*, 11(9):739–747, Sep 1998.

[TO89] W R Taylor and C A Orengo. Protein structure alignment. *Journal of Molecular Biology*, 208(1):1–22, Jul 1989.

[YG03] Y Ye and A Godzik. Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics*, 19 Suppl 2:246–255, Oct 2003.

[YYS+04] A F Yakunin, A A Yee, A Savchenko, A M Edwards, and C H Arrowsmith. Structural proteomics: a tool for genome annotation. *Curr Opin Chem Biol*, 8(1):42–48, Feb 2004.

[ZS05] Y Zhang and J Skolnick. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res*, 33(7):2302–2309, 2005.