Published online 2 May 2012 *Nucleic Acids Research, 2012, Vol. 40, Web Server issue* ***W303–W309***
doi:10.1093/nar/gks362

CSA: comprehensive comparison of pairwise protein structure alignments

Inken Wohlers[1,*], Noël Malod-Dognin[2], Rumen Andonov[3] and Gunnar W. Klau[1]

[1]Life Sciences Group, Centrum Wiskunde & Informatica, Science Park 123, 1098 XG Amsterdam, The Netherlands, [2]INRIA Sophia Antipolis - Méditerranée, 2004 route des Lucioles, 06902 Sophia Antipolis Cedex and [3]INRIA Rennes - Bretagne Atlantique and University of Rennes 1, Campus de Beaulieu, 35042 Rennes Cedex, France

Received January 30, 2012; Revised March 29, 2012; Accepted April 10, 2012

ABSTRACT

CSA is a web server for the computation, evaluation and comprehensive comparison of pairwise protein structure alignments. Its exact alignment engine computes either optimal, top-scoring alignments or heuristic alignments with quality guarantee for the inter-residue distance-based scorings of contact map overlap, PAUL, DALI and MATRAS. These and additional, uploaded alignments are compared using a number of quality measures and intuitive visualizations. CSA brings new insight into the structural relationship of the protein pairs under investigation and is a valuable tool for studying structural similarities. It is available at http://csa.project.cwi.nl.

INTRODUCTION

Protein structural alignment is a key method for answering many biological questions that involve the transfer of information from well-studied proteins to less well-known proteins. Since structures are more conserved during evolution than sequences, structural alignment allows for the most precise mapping of equivalent residues. It is notably important for (i) detecting and investigating structural motifs, functional sites and common cores, and (ii) measuring similarity between proteins and bringing them in evolutionary relationship, e.g. by classification. Numerous web servers are available that offer individual methods for computing structural alignments (1–4).

Many structure-based scoring schemes have been proposed and there is no consensus which scoring is the best (5). Comparative studies find that scorings have individual strengths and weaknesses and that alignments produced by different methods can differ considerably (6). In the context of protein classification, there are first attempts to integrate information from alignments generated by different structural alignment methods (7,8).

Here, we present CSA (Comparative Structural Alignment), the first web server for computation, evaluation and comprehensive comparison of pairwise protein structure alignments at single residue level. CSA facilitates evaluating the agreement between alignments that maximize different established scoring schemes and helps detecting their strengths and weaknesses. It offers the computation of alignments using the scoring schemes of DALI (9), contact map overlap (CMO) (10), MATRAS (11) and PAUL (12). CSA uses our own, exact algorithm (13,14) that can be used with any inter-residue distance-based scoring scheme. Such a scheme scores the alignment of rows and columns of the two inter-residue distance matrices that represent the protein structures. Our algorithm does not optimize superposition-based scoring schemes. We choose CMO and PAUL scoring since they are tailored to the algorithm and DALI and MATRAS scoring because they are established and their programs and web servers (1,4) are widely used. CSA returns an optimal, i.e. top-scoring alignment, if found within the time limit, or otherwise an alignment with a quality guarantee that specifies how much improvement is at most possible. We denote this by calling our programs and its alignments DALIX and MATRASX, in which X indicates exact.

Optimality comes at the prize of higher running time, but is especially important when comparing alignments. A top-scoring, but biologically implausible alignment implies that the scoring scheme used is inadequate to detect the given structural relationship and a different scoring might be more advisable. In the case of pairwise structural alignment, in which primarily residue correspondences are of interest, and only secondarily the obtained similarity score, comparing alignments optimized with respect to different criteria thus brings additional insight.

In CSA, computed or uploaded alignments can be explored in terms of many inter-residue distance-, RMSD- and sequence-based scores and quality measures and with intuitive visualizations such that agreements and differences between alignments are easy to grasp. The user

*To whom correspondence should be addressed. Tel: +31 20 592 4014; Fax: +31 20 592 4199; Email: I.Wohlers@cwi.nl

can thus make educated decisions about the structural similarity of two proteins and, if necessary, post-process alignments by hand. Furthermore, a comparative analysis allows to differentiate between proteins with one clear-case alignment on which various scorings agree and proteins with ambiguous alignments for which it depends on the application which alignment is preferable.

MATERIALS AND METHODS

Structural alignment algorithm

The exact algorithm used in CSA is based on an integer linear programming (ILP) model of the structural alignment problem as described in (14). Solutions to the ILP are generated using the approach from (13). The algorithm combines branch-and-bound and Lagrangian relaxation, and can be seen as an iterative double dynamic programming method. The mathematical model supports a generic scoring scheme with positive and negative structural scores, sequence scores and affine gap costs. Many different scoring functions are special cases of this general scheme. Currently, CSA supports the scoring schemes of DALI (9), CMO (10), MATRAS (11) and PAUL (12). The performance of the Lagrangian approach strongly depends on the number of considered inter-residue distances (12). It has been extensively evaluated for CMO (13), PAUL (12) and DALI scoring (15).

Webserver implementation

The architecture of the web server is divided in a processing layer that computes (C++) and evaluates (Python) alignments and an output layer, which generates W3C-validated HTML websites, interacts with the user and displays all information (PHP and Javascript). The interface between the two layers is a MySQL database. The alignment engine for all our four currently supported scoring schemes is identical and implemented in C++ as a standalone program. The user may adjust the time limit of the computation. Furthermore, each scoring scheme has different parameters, for example, the use of C_α or C_β inter-residue distances.

Computed or user-uploaded (e.g. in FASTA format) alignments are read into a Python class and subsequently written to the MySQL database. A second Python class handles the computation of different scores. It obtains the required structural information from the PDB files with the help of the Biopython package Bio.PDB (16). Tasks related to superpositioning are also handled by this package. Visualizations of distance and distance difference matrices are generated using the Python Imaging Library.

The website functions have been implemented in separate modules, which makes it easy to integrate additional structural alignment methods. The modularity is illustrated by the use of a tab menu. All web server functions are extensively documented, which is denoted by a question mark next to the respective section titles or table headers. Additionally, a documentation puts instructions and explanations into context. Notably, we documented all structural alignment scorings that are used within CSA and we provide the corresponding formulas and references. In the output layer, structures and their superpositions are visualized in Jmol (http://www.jmol.org) and images are generated using the PHP package pChart (http://www.pchart.net/).

CASE STUDIES

We illustrate the functionality of CSA using two case studies that are accessible from its main page via the links 'Example 1' and 'Example 2'.

Benefits of visualization and comparison

The first case study deals with two proteins from the SISY data set (6,17), ubiquitin-binding protein CUE2 (PDB ID 1otr, chain A, 49 residues) and the CUE domain of activating signal cointegrator 1 complex subunit 2 (PDB ID 2di0, chain A, 71 residues). The proteins belong to the SISYPHUS (18) alignment AL00088995 of homologous proteins containing a CUE domain. The CUE domain is composed of a three helical bundle and it consists of 41 residues. It binds ubiquitin and is involved in protein degradation.

After specifying PDB IDs and chains on the main page of CSA, the user is redirected to the CSA evaluation environment. It is organized in tabs for the following tasks: overview on the protein structures, computing alignments using CMO, PAUL, DALI or MATRAS scoring, upload of external alignments, and the comparison of alignments.

The *Structures* tab lists PDB IDs, PDB file names, selected chains and their lengths and amino acid sequences. Links to the PDB (19) and to iHOP (20) are access points for additional information concerning the proteins and their function. Protein structures are visualized in Jmol. Their C_α and C_β distance matrices and contact maps are visualized. We compute CMO, PAUL, DALIX and MATRASX alignments using the default options, i.e. with a time limit of 30 CPUs. The setup of all four result pages is identical. Exemplary, we consider the CMO alignment page; parts of it are displayed in Figure 1.

Bounds on alignment score and similarity

Section *Optimized score* lists the resulting scores: the raw score $s(A, B)$ of proteins A and B (here, the number of common contacts), and a similarity score that normalizes $s(A, B)$ with respect to the self-similarity of the two proteins computed as $2s(A, B)/(s(A, A) + s(B, B))$. Our solver is an exact branch and bound search. At each step of the solving process, the solver borders the optimal solution using two values: a lower bound LB, which is the score of the best feasible solution found so far, and an upper bound UB that results from solving a Lagrangian relaxation of an integer programming formulation of the alignment problem (13,14). When an instance is optimally solved, then the relation $LB = s(A, B) = UB$ holds. Otherwise, $LB \leq s(A, B) \leq UB$, and the so called relative gap $(UB - LB)/LB$ quantifies the precision of the returned score LB. Such a quality guarantee is very useful in the context of large-scale database comparisons where

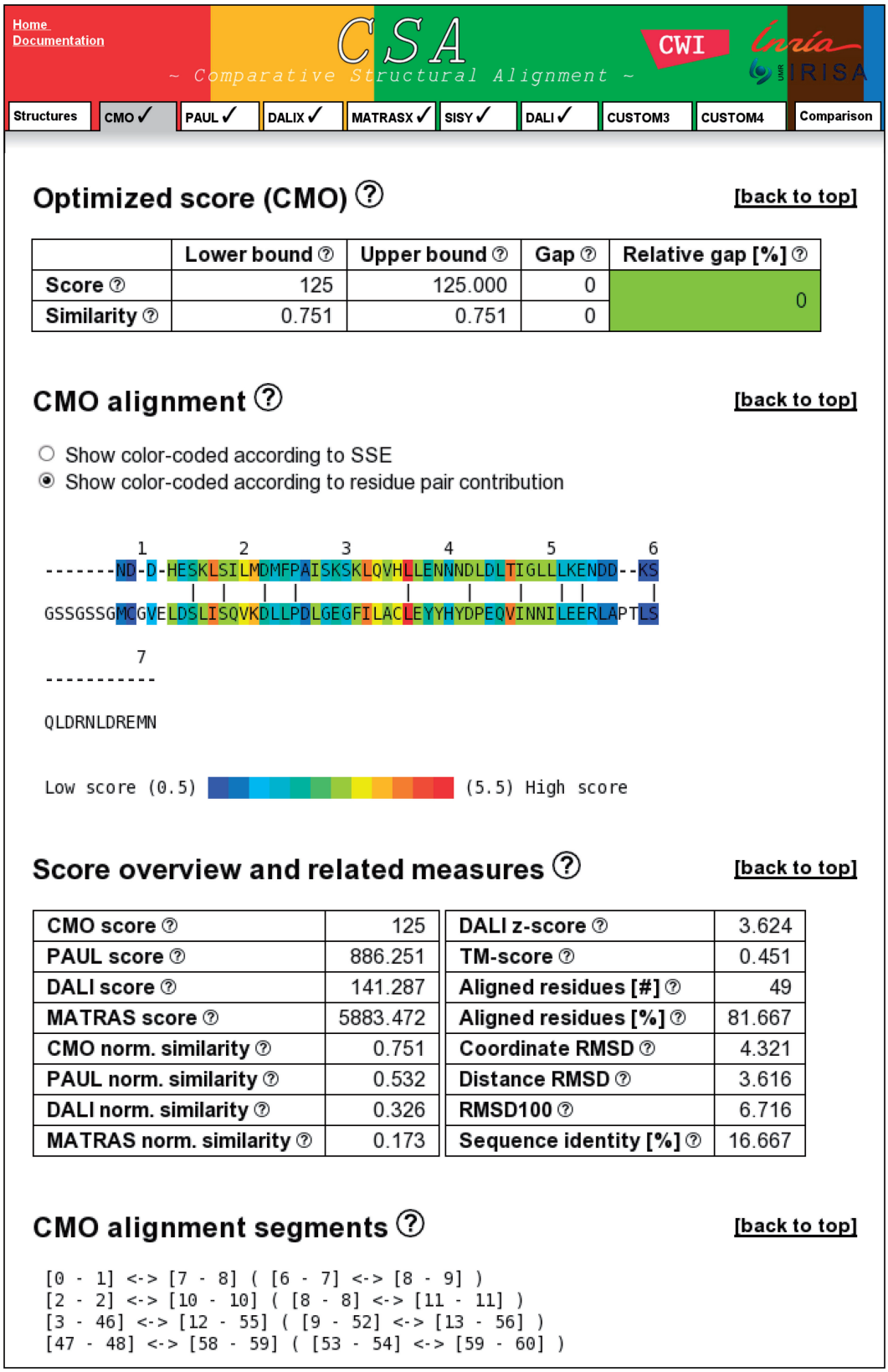

Figure 1. CSA website with different tabs and parts of the information displayed for the CMO alignment of 1otrA and 2di0A.

the execution time is usually bounded. It helps to quickly determine the progress of the computation as well as the similarity of the two proteins. If two proteins are dissimilar, the relative gap tends to be large, but the upper bound on the similarity score tends to be low from computation start on. In the considered example aligning 1otrA with 2di0A w.r.t. CMO yields 125 common contacts, and the corresponding similarity score on a scale from 0 to 1 is 0.751. The relative gap is 0%, indicating that the top-scoring alignment has been found.

Structural conservation and variation

The *Alignment* section displays the computed structural alignment. Residues are color-coded according to either SSE (helix, sheet, coil) as assigned by DSSP (21) or to residue pair score contribution. The second color-coding denotes for each residue pair how structurally prominent it is given the current alignment, compare Figure 1. For the two proteins containing the CUE domain, this indicates that the first identically aligned leucines are structurally preserved, and in fact this position is part of a motif for binding ubiquitin that consists of an invariant proline and two highly conserved leucines (22). Pairs of aligned residues with low score contribution highlight structural variations. In the CMO alignment, the N- and C-terminal regions are of little structural importance, as well as the residues in the region of the invariant proline within the CUE domain, because the proline is located in a turn. Such a visualization of residue score contribution can hint toward a manual modification of the alignment by removing aligned residues with low score. In fact, this is what happens in the top-scoring DALIX alignment of 1otrA and 2di0A, in which the four C-terminal residues with low CMO score are excluded from the alignment.

Comprehensive alignment-related data

Additional to the alignment, CSA displays the aligned segments, both using sequential and PDB residue numbering, compare Figure 1. Numerous raw alignment and similarity scores are listed, for example the number of aligned residues, sequence identity and root mean square deviation (RMSD). Furthermore, some statistics concerning the alignment computation are given. These are the number of residues and inter-residue distances considered during computation. They greatly influence the memory consumption of the algorithm: the more inter-residue distances are considered, the more memory is needed and typically the larger the running time. Using default values, CMO only considers distances smaller than 7.5 Å, PAUL considers distances smaller than 8.5 Å (for C_α distances, 9.5 Å), MATRAS scoring uses distances up to 50 Å and DALI scoring all distances. As a consequence, the exact DALIX and MATRASX solvers are extremely memory-demanding, and we currently restrict the computations of these two scores to protein pairs with a maximum average length. The allocation time for setting up all data structures is given, as well as the time actually spent on computing the alignment. The number of visited branch-and-bound nodes gives a good estimate on the progress of the computation. The proteins are superposed according to the alignment and visualized in Jmol. The trace of aligned residues and the distance difference matrix is plotted.

We upload an additional alignment in the tab for the first custom alignment. This alignment aligns only the 38 residues that belong to the respective CUE domain and that are structurally equivalent according to SISY. Furthermore, we upload a second custom alignment that has been generated by the DALI server (4), which uses a heuristic algorithm to find a good alignment according to the DALI score.

Improving, verifying optimality and assessing quality of heuristic alignments

Many different scorings and quality measures can be compared in the *Comparison tab*: the CMO, PAUL, DALI and MATRAS raw and similarity scores, DALI z-score (23), TM-score (24), number and percentage of aligned residues, coordinate and distance RMSD, RMSD100 (25), and sequence identity. For 1otrA and 2di0A, all six computed and uploaded alignments differ from each other. While CMO and PAUL alignment were computed to optimality in less than a second, the DALIX alignment has the potential to be improved by up to 12% and the MATRASX alignment by up to 24%. We also observe that the alignment that was computed by the DALI server and then uploaded is better with respect to DALI score than the alignment computed by our exact algorithm within 30 s. We thus increase the maximum running time for DALIX and MATRASX to 10 min. Now, both alignments are computed to provable optimality and our top-scoring DALIX alignment slightly improves the heuristic solution returned by the DALI server. DALIX and MATRASX alignments thus can be used to obtain quality guarantees for DALI or MATRAS alignments and in some cases also to either prove their optimality or to compute a better alignment.

Multi-criteria comparison and consensus alignment

Alignment trace comparison as introduced in (26) gives a visual overview about agreements and differences between alignments. Here, any subset of alignments can be shown. Using this visualization, we find that all alignments (except the SISY reference, which excludes three residues in the center of the domain) correctly align all 41 residues of the CUE2 domain, and that they differ in aligning the neighboring N- and C-terminal residues. A radar chart compares the different scores, compare Figure 2. This chart helps to quantify score differences and allows to decide whether one alignment is clearly preferable, i.e. better with respect to all criteria. The chart also allows to make an intuitive decision about which alignment is most appropriate in cases in which different scorings disagree as it is the case for 1otrA and 2di0A. Here, intuitively the DALIX alignment is the best choice since it performs good or best according to all criteria.

Two residue pair lists show aligned residues that appear in all, resp. in the majority, of the alignments. They each constitute a consensus alignment. In the case of aligning 1otrA and 2di0A, we see that such a consensus is useful: all alignments only agree in aligning the CUE2 domain.

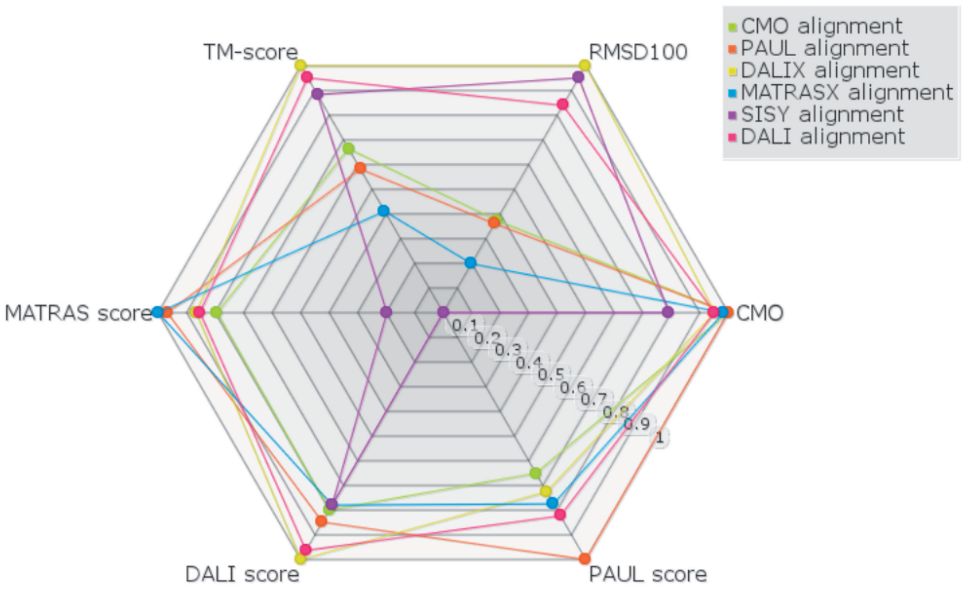

Figure 2. A radar chart for comparison of alignment scores for six different alignments of 1otrA and 2di0A. The closer a point is to 1, the better the corresponding score. CMO, PAUL, DALIX and MATRASX alignments have been computed by our exact algorithm and are provably optimal concerning their respective score. The SISY reference alignment aligns 38 residues of the CUE2 domain. The DALI alignment was computed by the DALI server and has slightly lower DALI score than the optimal DALIX alignment. The reference alignment is far behind in all scores except RMSD100 and TM-score, for which it performs quite well. The MATRASX alignment performs especially poor for these two measures. Intuitively, the DALIX alignment is most preferable since it has optimal DALI and close to optimal CMO, PAUL and MATRAS scores, as well as the best TM-score and RMSD100.

The consensus thus highlights the structurally conserved and biologically relevant region of the alignment.

Alignment of flexible proteins

We illustrate the usefulness of comparing structural alignments in the case of protein flexibility. This is a challenge for most structural alignment methods because flexible proteins typically do not superpose well unless the flexibility is accommodated for, e.g. by explicitly introducing a hinge.

Comparing flexible and rigid scoring schemes

We align two conformations of the calmodulin protein (PDB IDs 4cln, chain A and 2bbm, chain A, with a length of 148 residues). In structure 4clnA, calmodulin is bound to a ligand, in 2bbmA it is unbound. In bound conformation, a central helix is split and the components at the ends of the helix are moved toward each other. We align the two conformations using CMO and PAUL. We furthermore upload the alignments computed by TM-ALIGN (27) and by DAST (28), a local structural alignment method that determines the longest alignment with distance RMSD less than 4 Å. We find that both CMO and PAUL return the same alignment and correctly align the two conformations over their entire length. Figure 3 displays the two conformers superposed according to the TM-alignment and the alignment trace comparison. While PAUL and CMO align all residues of the two conformers, TM-ALIGN aligns only the C-terminal, rigidly superposable region (except the C-terminal residue). DAST also aligns the C-terminal region, but excludes and shifts further residues from the alignment. The radar chart as well as the distance difference matrices displayed in Figure 3 show why: while CMO, PAUL, DALI and MATRAS scoring by far favor the alignment of the entire conformers, TM-score as well as RMSD100 clearly favor the TM- and DAST alignment, which has a much smaller RMSD, but aligns only the C-terminal region.

Detecting flexibility and hinges

For each alignment we display the distance difference matrix. This is a symmetric square matrix with entries $|d_{ij}^A - d_{ij}^B|$ at position (i,j), where i is the i-th aligned position and j the j-th aligned position. Distance differences are visualized using a color gradient in which 0 Å is colored red; 2.5 Å, green; 5 Å, blue. Regions with low inter-residue distance differences correspond to rigidly superposable fragments. For the PAUL alignment of 4clnA and 2bbmA, red blocks in the distance difference matrix indicate that both the N- and C-terminal regions can be superpositioned very precisely. The distance differences between these two regions, however, are large, denoted by the blocks in blue color. The two regions can thus only be well superpositioned individually. A hinge is present at the residue bordering the two blocks (position 80) (29). TM-ALIGN and DAST align only the C-terminal region, thus avoiding any large distance differences. DAST is more restrictive in excluding large distance differences, it does not align a few residues that are still aligned by the TM-alignment and that have distance differences of about 5 Å, colored in blue.

Scores as CMO and PAUL, which implicitly ignore RMSD, are useful to gain information about flexible regions. While this feature is beneficial for flexible proteins it may also introduce flexibility where this is not

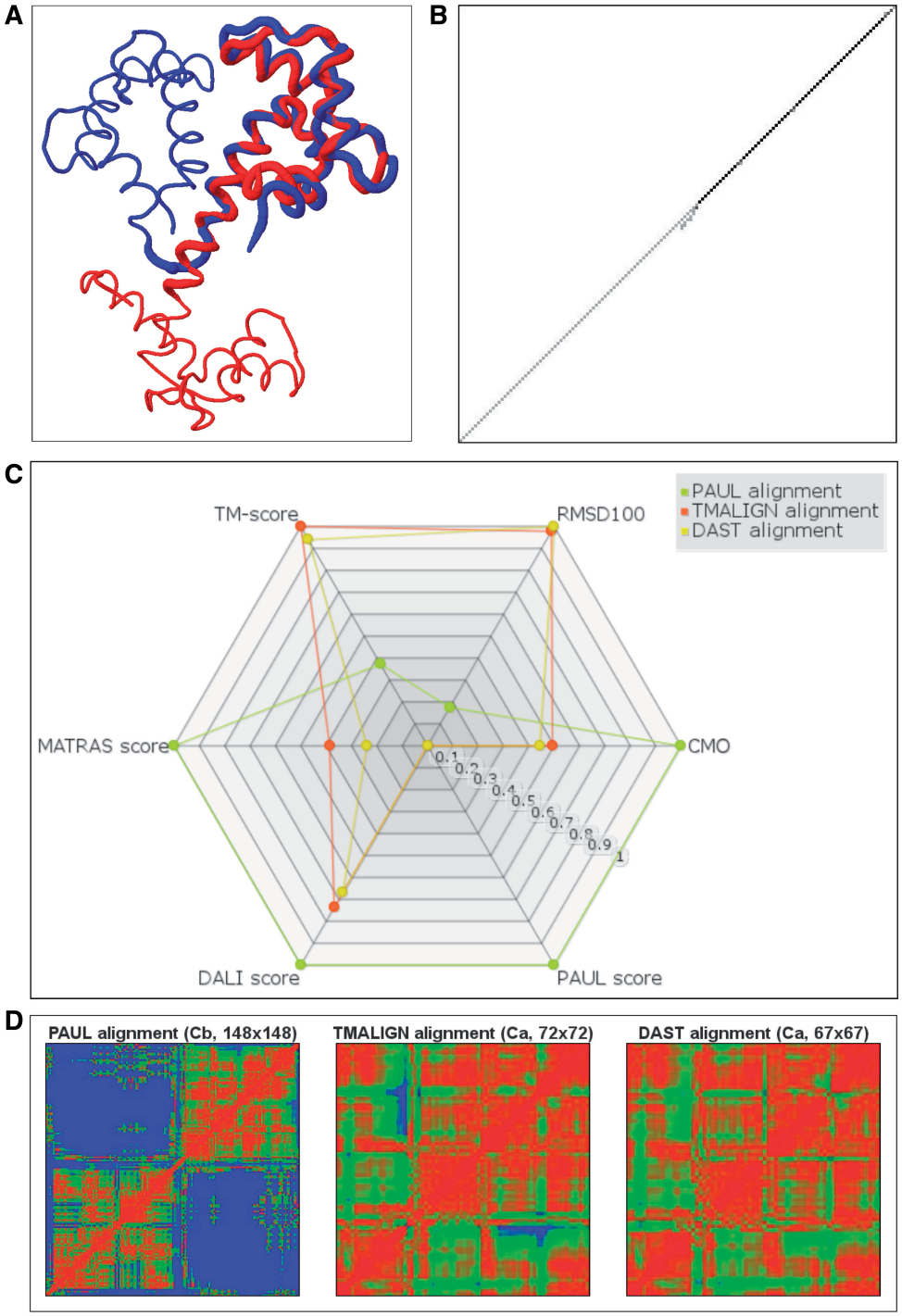

Figure 3. (**A**) The two calmodulin conformers (PDB IDs 4cln and 2bbm) superpositioned according to the TM-alignment, which aligns only one of the two regions that move relative to each other. (**B**) Comparison of the alignment traces. Each axis corresponds to one conformer. Black boxes denote residue pairs aligned by all three scorings, PAUL, TM-ALIGN and DAST. Light (intermediate) shades of gray denotes residue pairs aligned by only one (two) scorings. PAUL aligns all residues of the two conformers, TMALIGN and DAST the C-terminal region. (**C**) The radar chart illustrates the difference between scorings that are more in favor of a flexible alignment (CMO, PAUL, DALI and MATRAS) and scorings that are more in favor of a rigid superposition of low RMSD. (**D**) Distance difference matrices show the difference between the flexible PAUL alignment, that aligns all residues in spite of large distance differences (colored blue), and the TM- and DAST alignment that only align the C-terminal region.

appropriate. Protein similarities consisting in compact, well superposable fragments are therefore often better detected by maximizing scores like the TM- or the DAST score.

CONCLUSION

Different structural alignment scoring functions have different strengths and weaknesses. Which scoring to use depends on the application and on the structural

relationship of the investigated proteins. This different focus on handling various aspects of structural similarity is one reason why there are many different structural alignment scorings and programs and no consensus which combination is best.

We therefore consider it beneficial to compute alignments using different scoring schemes and algorithms and to compare them in order to gain insight into their structural relationship. The CSA web server provides the tools for such a comparison. CSA allows to compute alignments optimizing various scorings, returns a quality guarantee for the alignments and enables the user to additionally evaluate and compare uploaded alignments. In the most common case in which scorings and alignments disagree, it facilitates evaluating the agreement and differences between them and selecting the most suitable alignment.

ACKNOWLEDGEMENTS

We thank Maarten Dijkema for his help with setting up the server. Further, we thank Mathilde Le Boudic-Jamin for extensively testing CSA and Thomas Metzler for his idea of using a radar chart for comparison.

FUNDING

Deutsche Forschungsgemeinschaft [KL 1390/2-1, in part]. Funding for open access charge: INRIA Sophia Antipolis – Méditerranée.

Conflict of interest statement. None declared.

REFERENCES

1. Kawabata,T. (2003) MATRAS: a program for protein 3D structure comparison. *Nucleic Acids Res.*, **31**, 3367–3369.
2. Mosca,R. and Schneider,T.R. (2008) RAPIDO: a web server for the alignment of protein structures in the presence of conformational changes. *Nucleic Acids Res.*, **36**, 42–46.
3. Margraf,T., Schenk,G. and Torda,A.E. (2009) The SALAMI protein structure search server. *Nucleic Acids Res.*, **37**, 480–484.
4. Holm,L. and Rosenström,P. (2010) Dali server: conservation mapping in 3D. *Nucleic Acids Res.*, **38**, 545–549.
5. Hasegawa,H. and Holm,L. (2009) Advances and pitfalls of protein structural alignment. *Curr. Opin. Struct. Biol.*, **19**, 341–348.
6. Mayr,G., Domingues,F.S. and Lackner,P. (2007) Comparative analysis of protein structure alignments. *BMC Struct Biol.*, **7**, 50–50.
7. Camoglu,O., Can,T. and Singh,A.K. (2006) Integrating multi-attribute similarity networks for robust representation of the protein space. *Bioinformatics*, **22**, 1585–1592.
8. Barthel,D., Hirst,J.D., Blazewicz,J., Burke,E.K. and Krasnogor,N. (2007) ProCKSI: a decision support system for protein (structure) comparison, knowledge, similarity and information. *BMC Bioinformatics*, **8**, 416–416.
9. Holm,L. and Sander,C. (1993) Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.*, **233**, 123–138.
10. Godzik,A. and Skolnick,J. (1994) Flexible algorithm for direct multiple alignment of protein structures and sequences. *CABIOS*, **10**, 587–596.
11. Kawabata,T. and Nishikawa,K. (2000) Protein structure comparison using the Markov transition model of evolution. *Proteins*, **41**, 108–122.
12. Wohlers,I., Domingues,F.S. and Klau,G.W. (2010) Towards optimal alignment of protein structure distance matrices. *Bioinformatics*, **26**, 2273–2280.
13. Andonov,R., Malod-Dognin,N. and Yanev,N. (2011) Maximum contact map overlap revisited. *J. Comput. Biol.*, **18**, 27–41.
14. Wohlers,I., Andonov,R. and Klau,G.W. (2011) Algorithm engineering for optimal alignment of protein structure distance matrices. *Optimization Lett.*, **5**, 421–433.
15. Wohlers,I., Andonov,R. and Klau,G.W. (2012) *Optimal DALI protein structure alignment*. Technical Report RR-7915, INRIA.
16. Hamelryck,T. and Manderick,B. (2003) PDB file parser and structure class implemented in Python. *Bioinformatics*, **19**, 2308–2310.
17. Berbalk,C., Schwaiger,C.S. and Lackner,P. (2009) Accuracy analysis of multiple structure alignments. *Protein Sci.*, **18**, 2027–2035.
18. Andreeva,A., Prlić,A., Hubbard,T.J. and Murzin,A.G. (2007) SISYPHUS–structural alignments for proteins with non-trivial relationships. *Nucleic Acids Res.*, **35**, 253–259.
19. Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N., Weissig,H., Shindyalov,I.N. and Bourne,P.E. (2000) The Protein Data Bank. *Nucleic Acids Res.*, **28**, 235–242.
20. Hoffmann,R. and Valencia,A. (2005) Implementing the iHOP concept for navigation of biomedical literature. *Bioinformatics*, **21(Suppl. 2)**, 252–258.
21. Kabsch,W. and Sander,C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, **22**, 2577–2637.
22. Shih,S.C., Prag,G., Francis,S.A., Sutanto,M.A., Hurley,J.H. and Hicke,L. (2003) A ubiquitin-binding motif required for intramolecular monoubiquitylation, the CUE domain. *EMBO J.*, **22**, 1273–1281.
23. Holm,L. and Sander,C. (1998) Dictionary of recurrent domains in protein structures. *Proteins*, **33**, 88–96.
24. Zhang,Y. and Skolnick,J. (2004) Scoring function for automated assessment of protein structure template quality. *Proteins*, **57**, 702–710.
25. Carugo,O. and Pongor,S. (2001) A normalized root-mean-square distance for comparing protein three-dimensional structures. *Protein Sci.*, **10**, 1470–1473.
26. Godzik,A. (1996) The structural alignment between two proteins: is there a unique answer? *Protein Sci.*, **5**, 1325–1338.
27. Zhang,Y. and Skolnick,J. (2005) TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res*, **33**, 2302–2309.
28. Malod-Dognin,N., Andonov,R. and Yanev,N. (2010) Maximum cliques in protein structure comparison. In: Festa,P. (ed.), *SEA*, Vol. 6049 of *Lecture Notes in Computer Science*. Springer, pp. 106–117.
29. Emekli,U., Schneidman-Duhovny,D., Wolfson,H.J., Nussinov,R. and Haliloglu,T. (2008) HingeProt: automated prediction of hinges in protein structures. *Proteins*, **70**, 1219–1227.