

METHODOLOGY ARTICLE

Open Access



Efficient and automated large-scale detection of structural relationships in proteins with a flexible aligner

Fernando I. Gutiérrez^{1,2}, Felipe Rodriguez-Valenzuela¹, Ignacio L. Ibarra^{1,3}, Damien P. Devos^{2,3*} and Francisco Melo^{1*}

Abstract

Background: The total number of known three-dimensional protein structures is rapidly increasing. Consequently, the need for fast structural search against complete databases without a significant loss of accuracy is increasingly demanding. Recently, TopSearch, an ultra-fast method for finding rigid structural relationships between a query structure and the complete Protein Data Bank (PDB), at the multi-chain level, has been released. However, comparable accurate flexible structural aligners to perform efficient whole database searches of multi-domain proteins are not yet available. The availability of such a tool is critical for a sustainable boosting of biological discovery.

Results: Here we report on the development of a new method for the fast and flexible comparison of protein structure chains. The method relies on the calculation of 2D matrices containing a description of the three-dimensional arrangement of secondary structure elements (angles and distances). The comparison involves the matching of an ensemble of substructures through a nested-two-steps dynamic programming algorithm. The unique features of this new approach are the integration and trade-off balancing of the following: 1) speed, 2) accuracy and 3) global and semiglobal flexible structure alignment by integration of local substructure matching. The comparison, and matching with competitive accuracy, of one medium sized (250-aa) query structure against the complete PDB database (216,322 protein chains) takes about 8 min using an average desktop computer. The method is at least 2–3 orders of magnitude faster than other tested tools with similar accuracy. We validate the performance of the method for fold and superfamily assignment in a large benchmark set of protein structures. We finally provide a series of examples to illustrate the usefulness of this method and its application in biological discovery.

Conclusions: The method is able to detect partial structure matching, rigid body shifts, conformational changes and tolerates substantial structural variation arising from insertions, deletions and sequence divergence, as well as structural convergence of unrelated proteins.

Keywords: Protein structure comparison, Protein structure search, Flexible structural alignment

* Correspondence: damienpdevos@gmail.com; fmelo@bio.puc.cl

²Centre for Organismal Studies (COS), Heidelberg University, Heidelberg, Germany

¹Departamento de Genética Molecular y Microbiología, Facultad de Ciencias Biológicas, Pontificia Universidad Católica de Chile, Alameda 340, Santiago, Chile

Full list of author information is available at the end of the article

Background

Structural comparison between proteins is a fundamental and common practice in structural biology with many applications, such as the identification of new domains, the classification into structural families and the detection of evolutionary relationships between protein structures that cannot be found by sequence comparisons. For example, the homology between prokaryotic and eukaryotic cytoskeletal filaments (FtsZ/Tubulin and MreB/Actin) or the paralogy between proteins such as hemoglobin and myoglobin where only revealed once the 3D structures of these proteins were solved and compared [1, 2]. Since the determination of the first structures in the 1970s to the present day, the number of solved protein structures in the Protein Data Bank (PDB) has continued to grow at an exponential rate, with more than one hundred thousand structures available today. To facilitate the organization and analysis of this large amount of information, different structure comparison methods and tools have been developed [3]. However, the rise in number of known structures makes the comparison of query structures against the database increasingly costly (both for time and computational requirements) using existing tools.

Depending on the representation of proteins, current structural alignment methods use two main approaches: methods based at the level of residues or C α atoms (DALI, Structural, TopMatch, MAMMOTH, CE, MUSTANG, FATCAT, TM-align) [4–11] or based on secondary structure representations (VAST, SSAP, GANGSTA+, QP tableau search) [12–15]. One of the major advantages of methods based on secondary structure representations is that they are generally faster, as there is typically at least one order of magnitude fewer secondary structure elements than residues within a protein. However, residue-based methods are generally more accurate [16].

Structure comparison methods are increasingly successful at detecting more divergent relationships [3]. Significant improvements have also been achieved in terms of speed when searching against large databases [17]. Despite this success, current structural comparison tools have a few major drawbacks that limit their utility for detecting cases of remote homology where protein structures might have diverged considerably. First, they treat proteins as rigid bodies and cannot accommodate the large structural variations observed over long evolutionary divergence, for example, the relationship between the nucleoporins and vesicle coats [18]. Additional structural variations that might be due to protein flexibility or allosteric transitions are difficult to detect with the current methods. Finally, they are usually restricted to the comparison of individual domains and do not consider multi-domain proteins. How many distant structural

relationships remain undetected because the tools are not sensitive enough? Our goal was to detect protein structure similarities that are beyond the reach of current tools based on rigid body superposition and, at the same time, to be able to do it efficiently and with competitive accuracy.

To that end, we have developed an efficient flexible aligner tool to compare protein structures based on matrices that contain a simple description of the geometrical arrangement of secondary structure elements. Arthur Lesk was the first to describe a tabular representation, which comprises the information about the relative orientation of the elements of secondary structure (interaxial angle) using a coarse-grained and discrete double quadrant codification [19]. The concept is that the sequential order of secondary structure elements and the geometry of interacting pairs capture the essence of the protein fold. The secondary structure elements and their respective angles and distances can be encoded in a matrix. The secondary structure elements are recorded in order of appearance along the main diagonal of the matrix. Each off-diagonal position contains the angles and distances between the pairs of secondary structure elements. The comparison of these matrices allows a faster structural matching than when using a protein representation at the residue/atomic level. However, secondary structure geometry matrices comparison is an NP-hard problem. Various implementations to solve this problem have been presented, including quadratic and linear integer programming [15, 20, 21]. Those methods are very precise at extracting maximally similar submatrices, but this is at the expense of speed when comparing against a large number of matrices such as the complete PDB database. In 2008, Konagurthu proposed the TableauSearch method to detect similarities between matrices using two steps of dynamic programming [20]. TableauSearch is faster than previous methods, but this comes at the expense of accuracy and of lacking the ability to find local matches as compared to global ones [15]. This method is not limited to element pairs that are in contact and uses the scheme previously proposed by Lesk described above [19].

We present and release here a new computer application called MOMA (from MOrphing & MAtching). This tool relies on a new algorithm that incorporates several innovations, which are: 1) it considers the continuous value of the angles instead of the discrete and coarse-grained quadrant codification proposed by Lesk and implemented in TableauSearch; 2) the incorporation of a user-defined maximum distance cutoff to consider contacts between secondary structure elements, 3) a modified two-step dynamic programming algorithm that allows for the maximization of the rigid union of several local and compatible structural

matches and 4) a new procedure to solve the integration of several rigid and globally incompatible local matches into a flexible and global solution. This new algorithm, as implemented in MOMA computer

application, results in a fully automated and highly efficient global flexible structural aligner, which is able to find structural similarity between distantly related proteins with high accuracy.

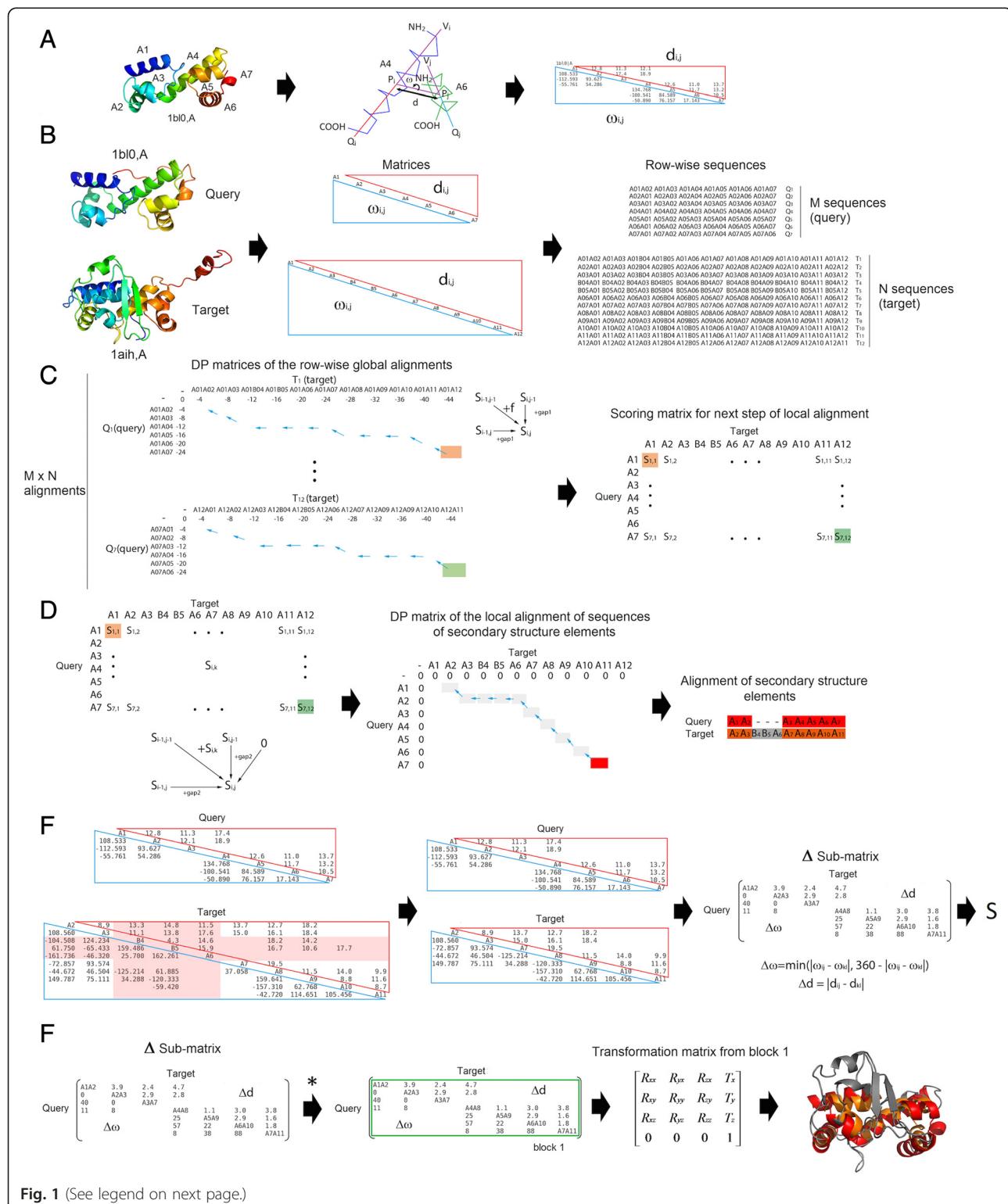


Fig. 1 (See legend on next page.)

(See figure on previous page.)

Fig. 1 Flowchart of the method as implemented in MOMA. **a** Example of structure of MarA (PDB code 1bl0) and the matrix representation of its folding pattern. The relative orientation of any two secondary structural elements (for example, A4 and A6 helices) is specified by the angle (w) between the vectors along their axes (*left bottom of the matrix*). This is recorded only for those SSE pairs found in close proximity ($d < D$), as measured by the distance (d) between midpoints of the vectors (*upper right of the matrix*). **b** These matrices are built for the query (1BL0 chain A) and the target (1AIH chain A) structures. After that, row-wise matrices containing all possible SSE pairs in each structure are also built. Query and target proteins render matrices of $[M, M-1]$ and $[N, N-1]$ pairs, where M and N correspond to the total number of SSEs found in the query and target structures, respectively. **c** A first step of global or semi-global dynamic programming (DP) algorithm is executed to build DP matrices for each query row against each target row, thus generating a total of $M \times N$ DP matrices. In this step, scoring rules and restraints based on angular and distance information of all SSE pairs in each structure are used (see Methods for details). From each DP matrix, only the maximum score value is selected and recorded into a new scoring matrix that is going to be used in a second and final step of a dynamic programming algorithm. In the case of a global alignment, this value is obtained from the bottom right cell of the DP matrix. In the case of the semi-global alignment, this value is obtained from the most right column or the most bottom row of the DP matrix. **d** A local dynamic programming algorithm and the previously built scoring matrix are now used to align the secondary structure elements of the query and the target structures. **e** Unaligned SSE elements from the query and target structures are removed from the initial 2D matrices, thus rendering two matrices of identical dimensions, which can now be compared directly. A delta sub-matrix is built and from it a global matching score calculated (see Methods for details). **f** Finally, a new algorithm (*) is used to infer the list of all incompatible rigid local matches (*blocks*), which are independently superposed with the Kabsch algorithm. In this particular and simple example only one local match or block is found. Details of the algorithm for finding local matching blocks are provided as Supplemental Material (Additional file 1: Figure S1). The resulting superposition is represented with aligned elements in red (*query*) and orange (*target*). Residues not aligned are displayed in grey color

Results and discussion

Overview of the new method

This article describes a fully automated and highly efficient method for the flexible comparison of two protein structure chains. The method relies on the matching of secondary structure elements between the protein chains, based on a two-step dynamic programming algorithm that combines local and global matching procedures. The results obtained when applying this method consist on a single and global structural alignment that integrates all rigid local matches found between the two input protein structures. A general overview of the method is provided in Fig. 1. A detailed description of each step of the method is provided in Methods section.

Calibration of parameter values

The results of our method, as implemented in MOMA, strongly depend on the value of three parameters, which are the constant that limits the score calculated from the angular difference (C) and the gap-opening penalties for the two steps of dynamic programming (g_1 and g_2). By optimization of the different combinations of these parameters, we found that the best results were obtained with a C constant value of 45 and a gap-opening penalty of -4 for both steps of dynamic programming (Additional file 1: Table S1). With these parameter values, only 2 out of 100 alignments from HOMSTRAD database have a QS index smaller or equal to 0.5 and the average QS index was 0.9436 (Additional file 1: Table S2). The failure of MOMA to correctly align the corresponding SSE pairs in these two cases is due to an inaccurate assignment of secondary structure elements by DSSP computer program. In some cases, DSSP does not assign the exact start and end points of SSEs. In other cases, long helices and

strands with some bending are split into two or more non-contiguous SSEs [21].

Another relevant parameter in the matrix comparison step of our method is the distance cutoff (D) used to define SSE pairs in contact [15]. We tested different values of distance thresholds in the HOMSTRAD set to define the best performing one (Additional file 1: Figure S1 and Table S1). If the distance cutoff value was smaller or equal than 12 Å, several matrices could not be aligned because too few SSE pairs were considered (ie. few contacts are found near the main diagonal of the matrix). Most of the information required to identify a folding pattern is contained in adjacent positions near the main diagonal in the matrices [22].

On the other hand, if the distance cutoff was set to values greater than 20 Å, the average QS index decreased (Additional file 1: Table S1). Therefore, a value of 20 Å was finally used as the maximum distance cutoff to define a contact between two SSEs.

After fixing the previous parameter values, and to evaluate if the raw score reported by MOMA was better than the relative similarity score, we then carried out searches using the seven most common folds as a query against a subset of 19,602 domains from ASTRAL 2.03 (95% sequence-identity cutoff; for details see Methods). The ROC curve analysis of these two scores showed that the relative similarity is slightly better than the raw score (Additional file 1: Figure S2 and Table S3). Thus, we defined relative similarity as the measure to be used for fold assignment by default in our method, as implemented in MOMA.

Testing the new method

As a first test of our method with the fixed parameter values described above, we used as a query the seven

most common folds and searched against the 19,602 domains in ASTRAL 95 % sequence identity dataset. ROC analysis of structure similarity matching results shows that, irrespectively of the query, the method has an excellent performance in terms of accuracy at the fold, family and superfamily levels (Additional file 1: Table S3). Execution time increases exponentially with the total number of SSE elements assigned in the structures (Additional file 1: Figure S4).

Benchmarking with other methods

The representative set of 100 protein queries was compared against the ASTRAL 2.03 40 % sequence identity dataset (which contains a total of 11,121 domains; for details see Methods) with SHEBA, YAKUSA, QP tableau search, GANGSTA+, Structal, TopMatch and MOMA computer programs. The performance of these methods was assessed by ROC curve analysis based on the normalized scores reported by each of them and adopting the SCOP classification as the gold standard [23]. We also measured the execution time required by these computer programs to perform a search against the full ASTRAL dataset of 11,121 domains with the 100 query structures.

In terms of AUC and maximum accuracy values, both at the fold and superfamily levels, Structal, TopMatch and MOMA are the best performing methods, followed by GANGSTA+, QP tableau search, SHEBA and Yakusa (Table 1; Additional file 1: Figure S3). In terms of accuracy, at the fold and superfamily levels, MOMA has the best performance among methods that use a geometric secondary structure representation of 3D protein structure such as QP tableau search and GANGSTA+, or when compared to currently the fastest methods for 3D structure matching such as YAKUSA and SHEBA. MOMA requires a variable amount of time to complete the search, which depends on the number of SSEs

present in the matrix (Additional file 1: Figure S4), but in this large benchmark set MOMA is much faster than all tested methods (at least by one or two-three orders of magnitude faster than most of the tested methods) (Table 1).

A detailed analysis of ROC curves reveals that SHEBA is a more specific classifier than MOMA, GANGSTA+ and QP tableau search, exhibiting a very low rate of false positives at the fold and superfamily levels. However, these methods have a higher sensitivity when compared to SHEBA. GANGSTA+ has an excellent performance and is better than QP tableau to search for proteins with the same fold, but QP tableau search is better than GANGSTA+ at a rate of false positives >0.6 for the superfamily level.

At the fold level, Yakusa is always worst than SHEBA, QP tableau search, GANGSTA+ and MOMA. However, Yakusa has a slight advantage than SHEBA at a rate of false positives >0.5 for the superfamily level.

The statistical analysis of the AUC curves reveals that the difference observed in the performance of MOMA with other computer programs is statistical significant at the 95 % confidence level (Additional file 1: Table S4).

As for the running time of each method, MOMA is the fastest of the methods tested. For example, it takes only 8 min and 28 s to search the 100 queries against the whole ASTRAL 40 %, while all other methods take more than 45 min, hours or even days of execution time (Table 1). We note that Structal, GANGSTA+, QP tableau search, and SHEBA are infeasible to run queries on very large datasets, such as the PDB database, which was one of the goals that motivated us to develop this method. Although QP tableau search can calculate the exact solution of the comparison of two matrices and GANGSTA+ can generate non-sequential protein structure alignments based in SSEs, MOMA has a better performance and is much faster than these two methods.

Table 1 Performance benchmark analysis of MOMA with different methods

Methods	AUC		ACC		*fp		*tp		time
	Fold	Superfamily	Fold	Superfamily	Fold	Superfamily	Fold	Superfamily	
Structal	0.956	0.969	0.902	0.919	0.076	0.060	0.880	0.898	10d 21h (1,842x)
TopMatch	0.955	0.974	0.883	0.911	0.121	0.069	0.887	0.891	2d (339x)
MOMA	0.940	0.956	0.872	0.889	0.139	0.113	0.884	0.891	8m 28s (1x)
GANGSTA+	0.916	0.911	0.845	0.851	0.101	0.058	0.791	0.761	5d 6h 49m (895x)
QP tableau search	0.877	0.918	0.791	0.831	0.224	0.188	0.805	0.850	2d 7h 27m (391x)
SHEBA	0.870	0.889	0.841	0.875	0.052	0.042	0.734	0.793	6h 51m (48x)
FATCAT flexible	0.837	0.911	0.743	0.825	0.220	0.211	0.706	0.862	27d 2h 38m (4,614x)
YAKUSA	0.790	0.858	0.727	0.794	0.155	0.088	0.609	0.677	48m (5.7x)

Area under ROC curve (AUC), maximal accuracy (ACC), false positive (fp) and true positive (tp) rates for each method are reported (*these values are calculated at the same threshold that gives the maximum accuracy reported as ACC). The execution time needed to compare the 100 queries against the 11,121 domains in the ASTRAL SCOP 40% sequence identity dataset is shown in the last column of the table. Execution times are reported in seconds (s), minutes (m), hours (h) and days (d) (in parenthesis, the speed gain factor of MOMA when compared to other methods is displayed, where "x" means number of times faster)

Biological applications

Rigid body shift caused by a rearrangement of domains

A well-known case that illustrates an example of rigid body movement between two structural domains is provided by the comparison of structures of calmodulin with and without Ca^{2+} ion (PDB codes and 2bbm and 1fcf, respectively). Both structures have 4 EF-hands, which consist of a helix-loop-helix motif that interact with Ca^{2+} and are organized into two distinct globular domains (N-terminal and C-terminal domains) [24]. These two domains are connected by a linker that is unstructured. This specific case is difficult to align due to the flexibility of the 6 loops and of the central linker. In the calmodulin- Ca^{2+} structure, the two calcium-binding domains are wrapped around a binding peptide in a “close” conformation while in the Ca^{2+} -free structure, a rotation around the axis of the linker leaves the two domains in an “open” conformation. Other flexible aligners such as Flexprot [25] and FATCAT [10], required the introduction of four or more rigid-body movements (twists) around pivot points (hinges) to obtain a good superposition of these two structures. In a single step, MOMA is able to automatically detect the conserved N-terminal and C-terminal domains, as shown in the matrix alignment, despite the different relative orientation of the two domains (Fig. 2).

Simple but significant structural rearrangement

The case of two functionally unrelated proteins illustrates the capacity of MOMA to obtain the global alignment of two structurally similar domains whose relative

orientation is not conserved. The putative oxidoreductase from *Pseudomonas putida* (PDB code 3l6d) and the human Cytokine-Like Nuclear Factor N-Pac (PDB code 2uyy) share two almost identical structural domains, which are separated by a connecting linker (Fig. 3). This linker is composed by two or three helices in bacterial and human proteins, respectively. The differential number of helices present in the linkers orients the two domains differently in the bacteria and human proteins. This simple structural rearrangement is a challenging problem for structural similarity detection methods, because the orientation of the two domains is different in both proteins. Rigid structure comparison tools can only identify the matching of these domains as two separate solutions, in the rare cases where more than one solution is reported (ie. TopMatch).

The power of MOMA resides in the fact that the structural similarity between both structural domain pairs is automatically detected and reported in a single step. In addition, the source of the conformational difference is also readily detected and highlighted in the alignment matrix (ie. helix 20 of 2uyyA cannot be aligned to a missing corresponding helix in 3l6dA).

Complex structural rearrangement

A more impressive example of structural rearrangement detection occurs in the case of Sec31 subunit from the COPII coated vesicle complex and the nucleoporin Nic96. Despite a lack of detectable sequence similarity [26], it is now generally accepted that coated vesicles proteins and nucleoporins have a common origin [18, 26].

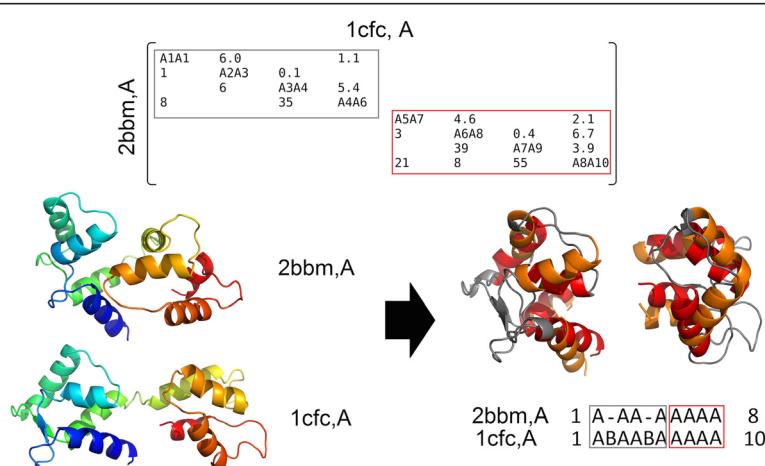
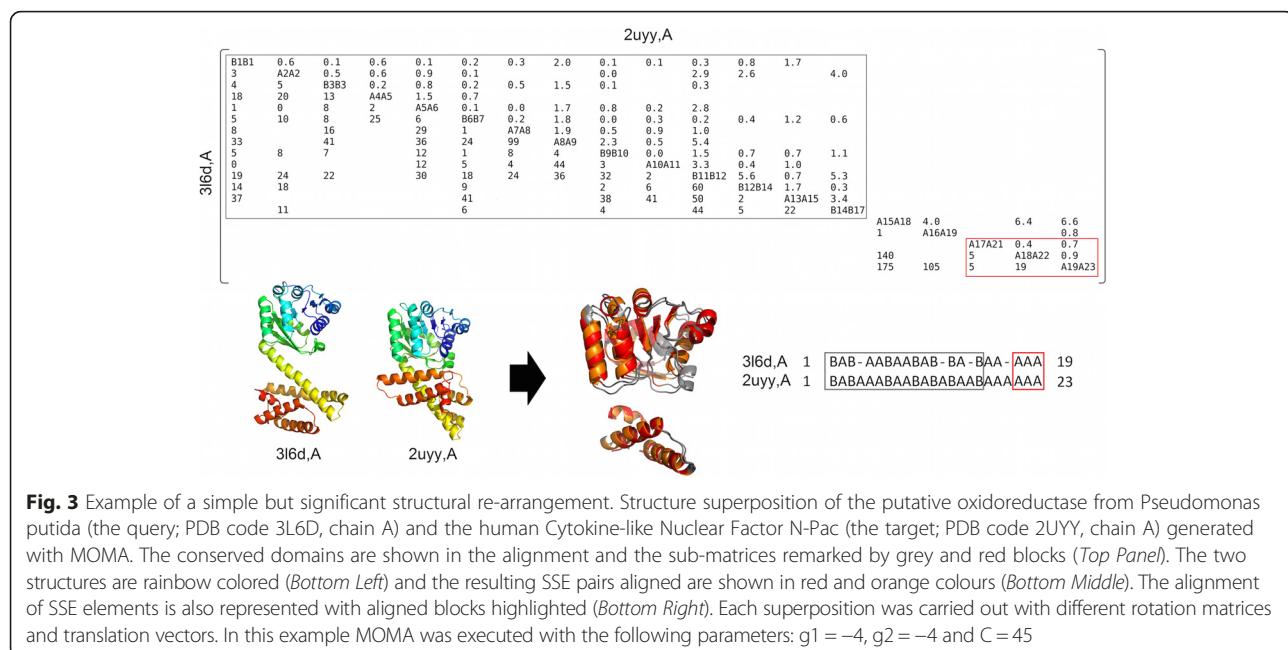
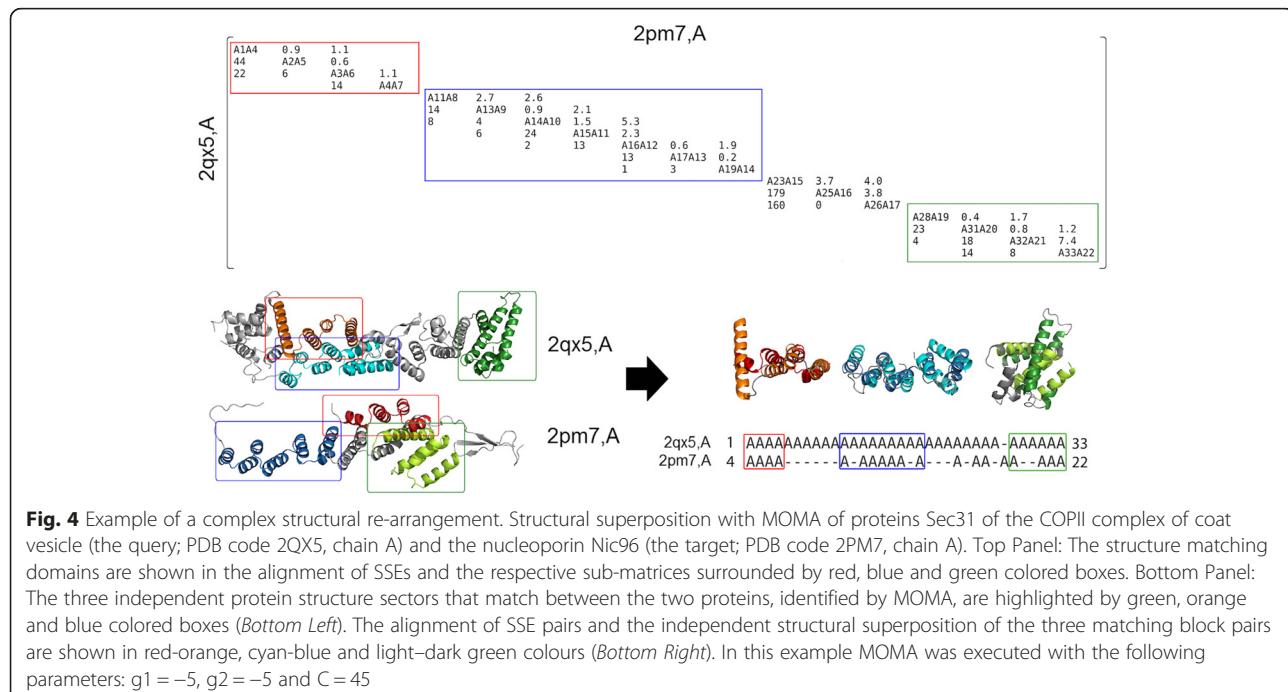


Fig. 2 Example of a rigid body shift caused by the rearrangement of two structural domains upon ligand binding. Structure superposition generated with MOMA of the Calmodulin-target peptide complex (the query; PDB code 2BBM, chain A) and the calcium-free Calmodulin (the target; PDB code 1CFC, chain A). Top Panel: The conserved domains are shown in the alignment of SSEs and the respective sub-matrices surrounded by grey and red blocks. Bottom Panel: The two structures and the superposition of their aligned domain pairs are shown respectively in rainbow color representation (Left) and with the SSE pairs structurally aligned in red and orange colours (Right). Non-aligned residues are shown in grey. The alignment of SSE elements is also represented with aligned blocks highlighted (Bottom Right). The structural superposition of these two domain pairs required different rotation matrices and translation vectors. In this example MOMA was executed with the following parameters: $g1 = -4$, $g2 = -4$ and $C = 90$



However, considerable divergence has occurred since the event of gene duplication, up to a point that sequence similarity cannot be detected any longer, even by the most recent and powerful methods [27]. This sequence divergence has had important consequences on the structural conformation, interactions and cages formed in these two proteins [27]. This is the type of structural divergence that we aimed to detect efficiently and automatically, and thus the main motivation behind the development of the new

method reported here. Nic96 (PDB code 2qx5) and Sec31 (PDB code 2pm7) are mainly composed of pairs of α -helices that are stacked on each other, hence termed SPAH domain (for Stacked Pairs of Alpha-Helices; also referred to as α -solenoid domain). Both proteins adopt a roughly linear shape that can be divided into three sections of conserved local structure (Fig. 4). However, those three conserved sections are preceded, followed and separated by other sections that exhibit considerable structural



deviation. Sections 1 and 2 are separated by a compact globular U-turn in Nic96, while this linker is unstructured in Sec31. The linker between sections 2 and 3 is composed by 9 α -helices in Nic96, but only by 3 α -helices in Sec31. These substantial structural modifications imply that section 1 is interacting only with section 2 in Nic96, while in Sec31 section 1 interacts almost exclusively with section 3. The relative orientation between the three blocks is also very different in both proteins. Despite these considerable global structural differences, the local structural similarity of the three blocks is clear and represents a legacy of their common ancestry [26]. To the best of our knowledge, MOMA is the only existing tool that is able to readily detect this intricate structural conservation in an automated fashion, which was the initial motivation of this work. The result obtained for this example case with MOMA clearly illustrates the power and potential for biological discovery of the new method reported here.

Strengths and weaknesses of the method

The speed, accuracy and flexible alignment capability of the method described here are their distinctive strengths. The method, as implemented in MOMA computer tool, is able to detect distant structural relationships in proteins in an automated fashion and efficiently, which makes it suitable to search the complete PDB for biological discovery. Among the weaknesses is the fact that MOMA is a single chain and topology-dependent protein structure alignment tool (ie. it depends on the connectivity order of SSEs). Few other tools, such as TopMatch and Structal have the capability of aligning protein structures in a topology-independent manner, but this comes at the cost of a longer execution time (these computer programs are 2–3 orders of magnitude slower than MOMA). TopMatch is the only tool currently available that is capable of aligning multiple protein chains, but the alignments are rigid and not flexible, which is a drawback in order to find domain movements or significant structural re-arrangements as exemplified here.

Structal was the most accurate tool in our benchmark (Table 1; Additional file 1: Table S4). A detailed analysis of the benchmark differences observed between Structal and MOMA shows that out of the 4,340 and 3,882 positive cases reported by Structal and MOMA, respectively, a total of 3,618 positive cases are common to both methods. There are 722 and 264 positive cases reported only by Structal and MOMA, respectively. Out of the 722 positive cases that Structal reports and MOMA fails to detect, 36.1 % is because of topological rearrangements and 16.7 % is because there are too short or very few SSEs in the structures. In 11.1% of the cases, MOMA fails to detect the positive cases because of large differences in secondary structure definitions between

the target and the query structures. It is noteworthy to mention that the use of STRIDE [28] or DSSP to assign SSEs produced, in a general basis, no significant difference on the performance of MOMA in our benchmark test (Additional file 1: Table S5). However, the accuracy of our method does depend directly on the assignment of SSEs, as well as on its use to represent protein structures and on its intrinsic topology-dependency. On the other hand, this simplified representation translates into a significant gain of speed without an important loss in accuracy (MOMA was 1,842 times faster than Structal in our benchmark test, but only 3 % less accurate). Finally, it is important to mention that the method described here produces structural alignments of secondary structure elements and not structural alignments at the residue-level. Therefore, if required, MOMA could be used in a first stage for fast database search on the task of fold or superfamily assignment and then, afterwards and only for positive matches, a more sophisticated software tool able to incorporate topology re-arrangements and to provide residue-level structure alignment, could be executed in a nested and sequential manner.

It is noteworthy to mention that this new method is not only restricted to protein structure comparison and could be implemented for many other applications that require the maximization of global shape matching between two three-dimensional objects with significant conformational variation, provided that those objects can be represented with vectors of different types which are relevant to describe the shape of the object, but with the limitation that vector order is a constraint of the method (ie. the method is topology-dependent).

Conclusions

We have developed a new structural comparison algorithm based on the spatial arrangement of secondary structure elements and shown that it allows the efficient retrieval of similar folding patterns in database searches. MOMA exhibits a high sensitivity to detect distant structural similarities without compromising its performance at identifying proteins that share a common fold.

In this regard, the development of a new combined global/semi-global and local structural alignment method that relies on a two-level nested dynamic programming algorithm and involves a new scoring scheme based on the continuous angular difference of SSE pairs close in 3D space instead of the previously used discrete quadrant codification, significantly improved the accuracy to find global similarities based on local matches in protein structures.

Methods

Protein structure and benchmark datasets

We used different protein structure datasets to first optimize the value of some parameters and then to evaluate the implementation of our method. First, to calibrate internal parameter values of the program, we used a subset of 100 pairwise structural alignments obtained from HOMSTRAD database [29] as previously described [30]. We kept only those alignments with a percentage sequence identity equal or less than 25 % and an average sequence length equal or greater than 150 residues (Additional file 1: Table S6). In this calibration process, a measure of similarity (the QS index) was maximized (see below). Second, to define the similarity score used and reported by our method, we used a small set of seven protein structures that represent the most common folds according to TOPS database [15, 20]. These seven proteins were used as a query to search against the ASTRAL SCOPe 2.03 95 % sequence identity protein domain database that contains 19,602 entries [31] (released October 2013). Receiver operating characteristic (ROC) curve analysis was performed and the area under the curve (AUC) measure was used to define the best performing score for classifying at the fold, family and superfamily level the query structures (see below).

Finally, to evaluate the performance of MOMA and other methods at classifying protein structures at the fold and superfamily levels, we used a representative set of 100 proteins extracted from the ASTRAL SCOPe 2.03 95 % sequence identity protein domain database described above (19,602 entries). These 100 proteins were used as a query to search for common structural matches against a non-redundant subset obtained from ASTRAL SCOPe 2.03 protein domain database [31] (released October 2013) with a 40 % sequence identity cutoff, which contains a total of 11,121 entries, none of them being any of the 100 query proteins. In this benchmark, we also carried out ROC curve analysis to assess and compare the performance of the methods (see below). All datasets described in this paper are available as supplementary data at: <http://melolab.org/supdat/moma>.

Computer software and methods

We used the DSSP program [32] to assign the secondary structure of proteins and the Numpy Python library to calculate the vectors and interaxial angles between the secondary structure elements. Moreover, we evaluated and compared MOMA against six methods based on their performance at classifying protein structures with similar folds or belonging to the same superfamily. The tested software implementing different methods were TopMatch [6], SHEBA [33], Yakusa [34], QP tableau search [15], Structural [5, 35], FATCAT [10] and

GANGSTA+ [14]. These computer programs were used with their default parameter values. All calculations were carried out using an Intel Core i7 2.64 GHz processor with 12 GB RAM memory and Ubuntu 13.04 Linux operating system.

Method description

To construct a 2D matrix from the 3D structure of a protein, the secondary structural elements (SSE) are assigned with the DSSP program, version 2.0.4 [32]. Only α -helices and β -strands with more than four and three residues, respectively, are considered in the analysis. Different types of α -helices (π , β_{10} and α) are treated equivalently and always assigned as a common α -helix type. Next, each secondary structure element is represented as a vector from its amino to carboxyl terminus by linear square fitting of an axis through the $C\alpha$ coordinates with the singular value decomposition method [36].

After that, the interaxial angle between each pair of SSE vectors and the Euclidean distance between the midpoints of the axes is computed (Fig. 1a). The interaxial angle (ω) is the shortest rotation (clockwise or anti-clockwise) required for the reorientation of the nearest vector that eclipses the farther vector, its value is restricted between -180° and 180° and was calculated as previously described [21]. Finally, the angle and distance between each pair of SSEs are recorded in the two halves of a 2D matrix: 1) the angle half-matrix and 2) the distance half-matrix. Two SSEs are only considered to be in contact if the distance between the midpoints of their linear axes is below a user-defined cutoff (see below). The diagonal positions are labeled by the elements of secondary structure, numbered by order of appearance in the amino acid sequence, from NH₂ to COOH terminus (where 'A' stands for α -helix and 'B' for β -strand). All off-diagonal positions in the matrix are either blank, if the SSE pairs are not in contact, or they contain the observed angle or distance value of the corresponding SSE pair (Fig. 1a).

To compare 2D matrices of different size, we implemented a different method than that of TableauSearch [20] for submatrix matching. Our method aligns the two matrices with a nested dynamic programming algorithm. The first step of the method is aimed at discovering putatively equivalent SSE pairs by comparing each row in the query matrix with each row in the target matrix, with a global or semi-global alignment and a constant gap opening penalty value model (denominated g1). The rows are treated as linear sequences of SSE pairs (Fig. 1b). Therefore, each element in a row represents a pair of different SSEs in a protein. If the query and target structures contain M and N elements of secondary structure, then a total of M and N rows are generated

from the query and target structures, respectively. Consequently, in this step of the method, a total of $M \times N$ global or semi-global alignments are calculated (Fig. 1c).

Semi-global alignment is similar to global alignment, in the sense that it attempts to align the two sequences entirely. The difference between both methods lies in the way the alignments are scored. Semi-global alignment assigns no cost to opening end gaps in the alignment [37]. This alignment type selection depends on the difference in the number of SSEs identified in the query and target structures (ie. the size difference of the matrices). If the maximum ratio of the number of SSEs from the two structures is greater than two, a semi-global alignment is calculated; otherwise, a global alignment is built. We defined a scoring function that takes into account the value of interaxial angle (in degrees) calculated for each pair of SSEs, implicitly incorporating the distance between the two vectors. This function was defined as follows:

$$f(\omega_{ij}, \omega_{kl}, d_{ij}, d_{kl}, E_i, E_j, E_k, E_l) = \begin{cases} 0, & d_{ij} > D \text{ or } d_{kl} > D \\ -C, & E_i E_j \neq E_k E_l \\ -C, & \Delta\omega > 2C \\ C - \Delta\omega, & \text{otherwise} \end{cases} \quad (1)$$

$$\Delta\omega = \min(|\omega_{ij} - \omega_{kl}|, 360 - |\omega_{ij} - \omega_{kl}|) \quad (2)$$

where E_x stands for an element of secondary structure in relative position x from NH₂ to COOH terminus in the protein chain, which can adopt two possible labels or values: A for alpha helix and B for beta strand; $E_i E_j$ and $E_k E_l$ are SSE pairs in the query and target structure, respectively; ω_{ij} and ω_{kl} are the interaxial angles between the $E_i E_j$ pair in the query structure and between the $E_k E_l$ pair in the target structure, respectively; d_{ij} and d_{kl} are the distances between the $E_i E_j$ pair in the query structure and between the $E_k E_l$ pair in the target structure, respectively; $\Delta\omega$ is the minimal angular difference between ω_{ij} and ω_{kl} , and C is an angular constant (in degree units). D is the maximum distance allowed to define that two SSEs are in contact (in Angstroms). This function is subjected to several constraints. The first constraint, $d_{ij} < D$ and $d_{kl} < D$, is introduced in order to avoid false positives when pairs of SSEs in two proteins have a similar interaxial angle, but are found at very different distances in the two structures [15] or found at very large distances in both the query and target structures. It is expected that in these cases there is no direct association between the SSE pairs in the two structures that should be used to infer fold similarity. This restriction is applied if at least one of the pairs is not in contact, as defined by the maximal distance cutoff D (a user-defined parameter). The second constraint, $E_i E_j = E_k E_l$, ensures

that two SSE pairs of different types should not be matched (for example, helix-helix with strand-helix or with strand-strand) and the third constraint, $\Delta\omega < 2C$, ensures that the function takes values between C and $-C$. Finally, the adopted constant gap opening penalty values for the two levels of the dynamic programming algorithm were those resulting from an optimization process using one of the benchmark datasets (see section 2.6 below and Additional file 1).

The optimal score value obtained from each query and target row alignment (Fig. 1c) is taken to generate the scoring matrix that is used in the second alignment step (Fig. 1d), but this time with the local Smith-Waterman dynamic programming algorithm [38]. Here, a different constant gap opening penalty value can be adopted (denominated g_2), which is another user-defined parameter required by our method. The alignment of SSE elements between the query and target structures is generated by the usual backtracking procedure (Fig. 1d).

At this point, it is important to mention that this alignment contains the union of all local structurally matching SSEs between the query and target structures, concordant to optimized, but not yet integrated global information of structurally matching SSE pairs. Therefore, the current alignment cannot be directly interpreted as a global structure alignment of two rigid bodies. In the case of highly related proteins this alignment will be accurate, but in the case of proteins with domain movements, rigid body shifts or partial structure matching, the identification of the structural regions to be matched as rigid body shifts by unique geometrical transformations is still needed.

The next step of the method consists on removing all rows and columns corresponding to non-aligned SSEs from both 2D initial matrices, the query and the target, thus rendering two matrices of identical size and shape that can be now compared directly and efficiently, in a one-to-one cell-to-cell manner (Fig. 1e). A unique 2D difference sub-matrix is now produced (called ΔSM or delta sub-matrix), which contains in the diagonal the labels for only those matching SSE pairs between the query and target structure, along with their differences in angle (upper middle triangle) and distance (lower triangle). Only the difference values for SSE pairs below a maximum parameter value, named ΔD , are reported in this difference matrix.

Structural matching score and similarity measures

A score of overall and integrated structural similarity for the query and target structures is calculated from the 2D difference sub-matrix (Fig. 1e). This score represents an estimation of the global integration of local matches. We calculate a measure of integrated structural similarity based on a Gaussian function that considers the angular

difference observed in the matrix. This raw score can be defined as:

$$S = \frac{\sum_i^N e^{-r_i^2}}{\sigma^2}, \quad r_i^2 = \Delta\omega^2 \quad (3)$$

where r_i^2 is the squared angular difference observed between two SSE pairs below distance threshold D, N is the total number of the SSE pairs aligned and σ is the scale parameter that determines the reduction rate of the score as a function of increasing angular difference. If the target structure is structurally equivalent with the query structure (ie. similar matrices), the score is equal to the total number of SSE pairs aligned. With increasing spatial deviation of the angular difference of SSE pairs aligned, the score approaches to 0.

In addition to score S, for comparing proteins of different size, we implemented two normalization functions. One of these functions is the relative similarity, S_r [39], which constitutes a global similarity measure between two proteins, and is defined by:

$$S_r = 100 \times \frac{2S}{n_q + n_t} \quad (4)$$

where n_q and n_t are the number of SSE pairs that are in contact in the query and target matrices, respectively, and S is the raw score described above. Another normalization function is the relative cover C_r [30] which represents the cover of the structural match in the smallest protein with respect to the largest protein [39], and it was implemented in the following function:

$$C_r = \frac{100 \times S}{\min(n_q, n_t)} \quad (5)$$

The integration of the information from all these score similarity measures allows the detailed assessment of structure similarity between two protein chains, from a local and global perspective, at once.

Inference of compatible local structural matching

To obtain a flexible and global superposition of two structures, a complete list of rigid local sub-matches between the two structures must be generated (Fig. 1f). Each rigid local sub-match follows a specific geometric transformation (ie. a specific rotation matrix and translation vector pair). To that end, we have implemented an algorithm that infers all local and rigid matches from the 2D difference sub-matrix. The only constraint imposed by this algorithm is that a minimum local match must contain at least three pairs of SSE elements. Briefly, the algorithm follows the diagonal below and adjacent to the main diagonal, checking for the observed $\Delta\omega$ values. To

initiate a new local matching block, a non-null $\Delta\omega$ value equal or smaller than 90° is needed. If the next value is equal or smaller than 90°, the algorithm extends the matching block. If the observed $\Delta\omega$ value is absent (ie. null), then the block is trimmed. Matching blocks smaller than 3×3 are not considered. If the $\Delta\omega$ value is larger than 90°, then the adjacent left-row and bottom-column cell values are checked for non-null values equal or smaller than 90°. If this is not fulfilled, the matching block is trimmed. The detailed pseudocode of this algorithm is provided as supplementary material (Additional file 1: Figure S5).

Integrated visualization of structural matches

Finally, the local matching blocks are superposed in 3D following independent geometrical transformations. To achieve this, the coordinates of the SSE vectors belonging to each local matching block are first extracted. Then, both sets of coordinates are superposed using a particular implementation of the Kabsch algorithm [40], which is based on Lagrange multipliers to solve the optimal superposition problem. This algorithm implementation was proposed by Kearsley and provides an analytical solution based on quaternions to generate the three-dimensional superposition with minimal root mean square deviation [41]. The end result is the flexible global superposition of two structures (Fig. 1f).

Parameterization of the method

The gap-opening penalties defined in the steps of dynamic programming, C constant and maximum distance cutoff are the most important parameters to compare the SSE matrices. To calibrate these parameters in our method, we aligned 100 homologous protein pairs from HOMSTRAD dataset, carrying out several tests with different combinations of parameter values.

We used the Sorensen-Dice similarity index (QS) [42] to compare the precision of the method to detect equivalent pairs of SSEs in matrix alignments, using as gold standard the HOMSTRAD superpositions. The QS index was defined as:

$$QS = \frac{2 \times M}{A + B} \quad (6)$$

where A and B are the number of SSE pairs aligned that were reported by MOMA and HOMSTRAD, respectively, and M is the number of SSE pairs aligned in common. QS index lies between 0 (all SSE pairs aligned by MOMA are different from those reported by HOMSTRAD superposition) and 1 (SSE pairs aligned by MOMA are equal to those reported by HOMSTRAD). In each test, we calculated the average QS index to

determine the best combination of parameter values (Additional file 1: Table S2).

Performance assessment

We performed standard receiver operating characteristic (ROC) curve analysis and adopted the area under the ROC curve (AUC) as the accuracy measure for each method [43]. In these tests, SCOP classification (same fold, superfamily or family) was used as the gold standard to define true positive and true negative instances. Given a protein query and considering the list of hits above a score threshold returned by a search against the datasets, we counted a hit as a true positive (TP) if the structure target had the same SCOP classification level as the protein query. Otherwise, it was classified as a false positive (FP). The statistical significance of the observed differences in classifier performance was calculated with StAR web server (<http://melolab.org/star>) as previously described [44].

Additional file

Additional file 1: Supplementary data, including: **Table S1.** Average QS values for best combinations of MOMA parameter values; **Table S2.** Comparison of structural alignments generated by MOMA with those defined in HOMSTRAD; **Table S3.** Benchmark test to assess the performance of MOMA; **Table S4.** Statistical analysis for the benchmark of MOMA with other methods; **Table S5.** Statistical analysis for the benchmark of MOMA with different methods to assign secondary structure; **Table S6.** Set of 100 distant homologous protein pairs obtained from HOMSTRAD database; **Figure S1.** Calibration of distance cutoff using the HOMSTRAD set with the best combination of parameter values; **Figure S2.** ROC curves for the small set of seven most common folds according to TOPS database; **Figure S3.** ROC curves of classification at the SCOP fold and superfamily level; **Figure S4.** Execution time of MOMA; **Figure S5.** Algorithm used for extracting the rigid local matches. (PDF 2793 kb)

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

FIG programmed and implemented the method, executed all computer calculations reported in this work and contributed to the writing of this manuscript and preparation of Tables and Figures. FR-V implemented the Kabsch algorithm for optimal three-dimensional superposition of matching structures and contributed to the preparation of some Figures. ILI contributed with the testing and improvement of the algorithm for integrating local structure matches into a single global solution. DPD and FM conceived and supervised this research, structured and wrote this manuscript with the help of FIG. All authors read and approved the final manuscript.

Acknowledgements

This work was supported by FONDECYT Chile research [grant 1141172, to F.I.G., F.R-V., I.L.I. and F.M.], and by Heidelberg University Frontier [grant 28577, project #D.801000/12.074, to D.P.D and F.I.G., respectively]. I.L.I was funded by the Marie Curie Initial Training Network PERFUME (PERoxisome Formation, Function, Metabolism) grant (grant agreement number 316723).^a

Author details

¹Departamento de Genética Molecular y Microbiología, Facultad de Ciencias Biológicas, Pontificia Universidad Católica de Chile, Alameda 340, Santiago, Chile. ²Centre for Organismal Studies (COS), Heidelberg University,

Heidelberg, Germany. ³Centro Andaluz de Biología del Desarrollo (CABD), Universidad Pablo de Olavide, Sevilla, Spain.

Received: 19 August 2015 Accepted: 21 December 2015

Published online: 05 January 2016

References

- Erickson HP. Atomic structures of tubulin and FtsZ. *Trends Cell Biol.* 1998;8(4):133–7.
- van den Ent F, Amos LA, Löwe J. Prokaryotic origin of the actin cytoskeleton. *Nature.* 2001;413(6851):39–44.
- Hasegawa H, Holm L. Advances and pitfalls of protein structural alignment. *Curr Opin Struct Biol.* 2009;19(3):341–8.
- Holm L, Sander C. DALI: a network tool for protein structure comparison. *Trends Biochem Sci.* 1995;20(11):478–80.
- Gerstein M, Levitt M. Using iterative dynamic programming to obtain accurate pairwise and multiple alignments of protein structures. *Proc Int Conf Intell Syst Mol Biol.* 1996;4:59–67.
- Sippl MJ, Wiederstein M. Detection of spatial correlations in protein structures and molecular complexes. *Structure (London, England : 1993).* 2012;20(4):718–28.
- Ortiz AR, Strauss CEM, Olmea O. MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. *Protein Sci.* 2002;11(11):2606–21.
- Shindyalov IN, Bourne PE. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng.* 1998;11(9):739–47.
- Konagurthu AS, Whisstock JC, Stuckey PJ, Lesk AM. MUSTANG: a multiple structural alignment algorithm. *Proteins: Struct, Funct, Bioinf.* 2006;64(3):559–74.
- Ye Y, Godzik A. Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics.* 2003;19 suppl 2:ii246–55.
- Zhang Y, Skolnick J. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.* 2005;33(7):2302–9.
- Gibrat J-F, Madej T, Bryant SH. Surprising similarities in structure comparison. *Curr Opin Struct Biol.* 1996;6(3):377–85.
- Ornengo CA, Taylor WR. SSAP: sequential structure alignment program for protein structure comparison. *Computer methods for macromolecular sequence analysis.* 1996.
- Guerler A, Knapp EW. Novel protein folds and their nonsequential structural analogs. *Protein Sci.* 2008;17(8):1374–82.
- Stivala A, Wirth A, Stuckey PJ. Tableau-based protein substructure search using quadratic programming. *BMC bioinformatics.* 2009;10:153.
- Schwede T, Peitsch MC. Computational structural biology: Methods and applications. 1st ed. Singapore: World Scientific; 2008.
- Wiederstein M, Gruber M, Frank K, Melo F, Sippl MJ. Structure-based characterization of multiprotein complexes. *Structure.* 2014;22(7):1063–70.
- Brohawn SG, Leksa NC, Spear ED, Rajashankar KR, Schwartz TU. Structural evidence for common ancestry of the nuclear pore complex and vesicle coats. *Science.* 2008;322(5906):1369–73.
- Lesk AM. Systematic representation of protein folding patterns. *J Mol Graph.* 1995;13(3):159–64.
- Konagurthu AS, Stuckey PJ, Lesk AM. Structural search and retrieval using a tableau representation of protein folding patterns. *Bioinformatics (Oxford, England).* 2008;24(5):645–51.
- Konagurthu AS, Lesk AM. Structure description and identification using the tableau representation of protein folding patterns. *Methods in molecular biology (Clifton, NJ).* 2013;932:51–9.
- Kamat AP, Lesk AM. Contact patterns between helices and strands of sheet define protein folding patterns. *Proteins.* 2007;66(4):869–76.
- Murzin AG, Brenner SE, Hubbard T, Chothia C. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol.* 1995;247(4):536–40.
- Chen K, Ruan J, Kurgan L. Prediction of three dimensional structure of calmodulin. *Protein J.* 2006;25(1):57–70.
- Shatsky M, Nussinov R, Wolfson HJ. Flexible protein alignment and hinge detection. *Proteins: Struct, Funct, Bioinf.* 2002;48(2):242–56.
- Devos D, Dokudovskaya S, Alber F, Williams R, Chait BT, Sali A, et al. Components of coated vesicles and nuclear pore complexes share a common molecular architecture. *PLoS Biol.* 2004;2(12):e380.

27. Field MC, Sali A, Rout MP. Evolution: On a bender–BARs, ESCRTs, COPs, and finally getting your coat. *J Cell Biol.* 2011;193(6):963–72.
28. Frishman D, Argos P. Knowledge-based protein secondary structure assignment. *Proteins: Struct, Funct, Bioinf.* 1995;23(4):566–79.
29. Mizuguchi K, Deane CM, Blundell TL, Overington JP. HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Sci.* 1998;7(11):2469–71.
30. Slater AW, Castellanos JL, Sippl MJ, Melo F. Towards the development of standardized methods for comparison, ranking and evaluation of structure alignments. *Bioinformatics (Oxford, England).* 2013;29(1):47–53.
31. Fox NK, Brenner SE, Chandonia JM. SCOPe: Structural Classification of Proteins–extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Res.* 2014;42(Database issue):D304–309.
32. Kabsch W, Sander C. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers.* 1983;22(12):2577–637.
33. Jung J, Lee B. Protein structure alignment using environmental profiles. *Protein Eng.* 2000;13(8):535–43.
34. Carpentier M, Brouillet S, Pothier J. YAKUSA: a fast structural database scanning method. *Proteins.* 2005;61(1):137–51.
35. Kolodny R, Koehl P, Levitt M. Comprehensive evaluation of protein structure alignment methods: scoring by geometric measures. *J Mol Biol.* 2005;346(4):1173–88.
36. Wall ME, Rechtsteiner A, Rocha LM. Singular value decomposition and principal component analysis. In: *A practical approach to microarray data analysis.* Springer. 2003: 91–109.
37. Sung W-K. Algorithms in bioinformatics: A practical introduction: CRC Press; 2009. Broken Sound Parkway, NW Suite 300, Boca Raton, FL, 33487. USA.
38. Smith TF, Waterman MS. Identification of common molecular subsequences. *J Mol Biol.* 1981;147(1):195–7.
39. Sippl MJ. On distance and similarity in fold space. *Bioinformatics (Oxford, England).* 2008;24(6):872–3.
40. Kabsch W. A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Crystallogr A.* 1978;34(5):827–8.
41. Kearsley SK. On the orthogonal transformation used for structural comparisons. *Acta Crystallogr A.* 1989;45(2):208–10.
42. Wolda H. Similarity indices, sample size and diversity. *Oecologia.* 1981;50(3):296–302.
43. Fawcett T. ROC graphs: Notes and practical considerations for researchers. *Mach Learn.* 2004;31:1–38.
44. Vergara IA, Norambuena T, Ferrada E, Slater AW, Melo F. StAR: a simple tool for the statistical comparison of ROC curves. *BMC bioinformatics.* 2008;9:265.

Submit your next manuscript to BioMed Central
and we will help you at every step:

- We accept pre-submission inquiries
- Our selector tool helps you to find the most relevant journal
- We provide round the clock customer support
- Convenient online submission
- Thorough peer review
- Inclusion in PubMed and all major indexing services
- Maximum visibility for your research

Submit your manuscript at
www.biomedcentral.com/submit



Supplementary Table S1. Average QS values for best combinations of MOMA parameter values.

g1	g2	D	Average QS
-4	-4	20	0.9436
-4	-3	20	0.9433
-3	-3	20	0.9432
-4	-5	20	0.9428
-6	-6	24	0.9423
-2	-3	20	0.9422
-2	-5	20	0.9422
-5	-11	28	0.9422
-6	-9	28	0.9422
-2	-4	20	0.9420
-2	-6	20	0.9420
-2	-8	20	0.9420
-3	-5	28	0.9420
-5	-10	28	0.9420
-6	-8	28	0.9420
-6	-7	24	0.9417
-6	-8	24	0.9417
-4	-1	20	0.9416
-5	-6	24	0.9416
-5	-12	28	0.9415
-5	-13	28	0.9415
-5	-14	28	0.9415
-6	-10	28	0.9415
-3	-1	20	0.9414
-6	-8	30	0.9414
-4	-8	30	0.9414
-6	-4	28	0.9413
-6	-5	28	0.9413
-6	-6	28	0.9413
-3	-7	20	0.9412

Only the 30 best average QS values are shown out of 4,000 combinations of parameter values tested. g1, g2: gap-opening penalties for the first and second step of the dynamic programming algorithm, respectively. These parameter values were varied in the range [-1, -20] with a step of -1. C: constant used to modulate the angular difference between a pair of secondary structure elements (SSEs) in the scoring scheme of the dynamic programming algorithm. This parameter value was 45. D: maximum distance threshold to define a contact between a pair of SSEs. This parameter value was varied in the range [12, 30] with a step of 2.

Supplementary Table S2. Comparison of structural alignments generated by MOMA with those defined in HOMSTRAD.

Query	Target	SSE pairs reported by MOMA	SSE pairs reported by HOMSTRAD	SSE pairs in common	QS index
154l	1qsaa	5	5	5	1.000
1a0i	1fvia	12	12	11	0.917
1a41	1a31a	12	11	11	0.957
1a48	1kuta	15	15	14	0.933
1a7j	1esma	13	13	13	1.000
1a9na	1d0ba	8	8	7	0.875
1ab4	1bjt	27	28	26	0.945
1al3	1i6aa	14	13	12	0.889
1am2	1at0	11	11	11	1.000
1amua	1lci	34	34	34	1.000
1b3qa	1bxda	8	8	8	1.000
1b63a	1h7sa	18	19	18	0.973
1b74a	1jfla	14	15	12	0.828
1b8ba	3pfla	22	23	21	0.933
1b9ia	1jg8a	20	20	20	1.000
1bf2	1ehaa	22	22	21	0.955
1bgla	1bhga1	14	14	14	1.000
1bgyb	1bgya	21	22	21	0.977
1bhga2	1dp0a	10	8	8	0.889
1bk0	1dc5	19	19	19	1.000
1bp12	1bp11	11	9	7	0.700
1bs9	1cex	11	11	10	0.909
1by5a	1fepa	29	29	29	1.000
1clc	1tf4b	16	16	16	1.000
1cqea	1mhla	20	19	19	0.974
1ct5a	1bd0a	16	16	16	1.000
1d2ra	2ts1	15	17	12	0.750
1dhpa	1nal1	20	20	20	1.000
1dpe	1jeta	30	30	28	0.933
1dqaa	1qaya	23	24	22	0.936
1dqwa	1dbta	15	15	15	1.000
1e42a	1qtsa	15	15	15	1.000
1eg2a	1boo	14	15	14	0.966
1egua	1cb8a	39	38	37	0.961
1ejea	1i0ra	9	11	9	0.900
1em2a	1jssa	12	12	12	1.000
1emsa	1fo6a	20	20	20	1.000
1erja	1gotb	29	29	28	0.966
1evsa	1lki	5	4	4	0.889
1ezia	1h7ea	13	13	13	1.000
1f8sa	1h83a	30	30	26	0.867
1fi4a	1h72c	20	18	17	0.895
1fioa	1ez3a	3	3	3	1.000
1fm2a	1e3aa	37	37	34	0.919
1fmta	2gar	11	11	11	1.000
1fo4a	1ffvc	19	19	19	1.000
1fs0g	1e79g	10	10	10	1.000
1ft1a	1dcea	15	15	15	1.000
1fxxa	1j54a	12	12	12	1.000
1g6ga	1qu5a	8	6	2	0.286
1g6sa	1ejda	33	32	32	0.985
1ga8a	1ll3a	18	18	15	0.833

Query	Target	SSE pairs reported by MOMA	SSE pairs reported by HOMSTRAD	SSE pairs in common	QS index
1gcua	1ofga	21	21	21	1.000
1ggxa	1gfla	12	12	12	1.000
1gln	1qtqa	18	17	16	0.914
1gr0a	1jkia	19	19	18	0.947
1gvfa	1dosa	19	19	19	1.000
1hu3a	1h6ka	10	9	9	0.947
1i7da	1cy9a	12	12	12	1.000
1igra2	1igra1	5	6	5	0.909
1iira	1f0ka	24	24	24	1.000
1iq0a	1f7ua	29	31	29	0.967
1iq8a	1k4ga	21	22	21	0.977
1jcua	1hrua	13	13	13	1.000
1jdia	1fua	13	13	13	1.000
1jeyb	1jeya	25	25	25	1.000
1jgta	1ct9a	31	31	31	1.000
1jj2e	1rl6a	14	14	14	1.000
1jj2j	1whi	6	6	6	1.000
1jmkc	1keza	11	11	11	1.000
1jqra	1bpya	13	13	13	1.000
1js8a	1bt3a	11	10	10	0.952
1k6da	1poia	16	16	16	1.000
1kas	1afwa	19	17	17	0.944
1kit2	1kit1	12	12	12	1.000
1l5ja	1c96a	26	26	26	1.000
18aa	1trka	16	16	16	1.000
1lam	1gyta	10	10	10	1.000
1lnsa	1ju3a	31	31	31	1.000
1m2vb	1m2oa	35	36	35	0.986
1moq2	1moq1	10	11	10	0.952
1n1ma	1h2wa	17	16	15	0.909
1n2za	1efdn	16	15	14	0.903
1pbe	1foha	23	23	23	1.000
1pbwa	1tx4a	10	10	10	1.000
1qgia	1chka	11	11	8	0.727
1qhta	1ih7a	20	20	19	0.950
1rdr	1khva	18	17	17	0.971
1rkd	1bx4a	22	22	22	1.000
1rmg	1bhe	22	22	11	0.500
1thfd	1qo2a	17	17	17	1.000
1uag	1fgs	14	15	13	0.897
1vsga	2vsga	10	10	7	0.700
1yaca	1nbaa	12	12	12	1.000
2abk	1mun	10	10	9	0.900
2pgi	1dqra	25	27	24	0.923
2sqca	1ft1b	13	12	12	0.960
2tmda	1oyc	19	19	19	1.000
2tysb	1tdj	21	21	20	0.952

MOMA alignments were calculated with the best combination of parameter values ($g1 = -4$, $g2 = -4$, $C = 45$, $D = 20$). The alignments highlighted in red color represent those cases where MOMA generated a bad superposition (QS index ≤ 0.5).

Supplementary Table S3. Benchmark test to assess the performance of MOMA.

SCOP id	Fold	SCOP levels	# SSE	AUC						Execution Time (s)	
				Raw score			Relative similarity				
				Fold	Superfamily	Family	Fold	Superfamily	Family		
d1ubia_	β-grasp	d.15.1.1	6	0.96	0.98	0.98	0.99	0.99	0.99	3.14	
d1tttb1	Key-barrel	b.43.3.1	8	0.95	0.93	0.96	0.96	0.95	0.98	3.82	
d1ae6h1	Immunoglobulin	b.1.1.1	10	0.94	0.96	0.98	0.97	0.98	0.98	4.59	
d1bhne	Plait	d.58.6.1	12	0.92	1.00	1.00	0.97	1.00	1.00	6.01	
d1h6rb_	GFP-like	d.22.1.1	14	1.00	1.00	1.00	1.00	1.00	1.00	7.20	
d1tima_	Tim-barrel	c.1.1.1	20	0.99	1.00	0.99	0.99	1.00	1.00	12.40	
d1f6dc_	NAD-binding fold	c.87.1.3	28	0.99	0.99	1.00	0.99	0.99	1.00	21.46	

Area under ROC curves and execution times of MOMA when using the small set of seven most common folds as queries against the 19,602 domains in ASTRAL 95% sequence identity dataset. Calculations were carried out on a computer with an Intel Core i7 2.64 GHz processor with 12 GB memory, running Ubuntu 13.04 Linux system. The total execution time is reported in seconds.

Supplementary Table S4. Statistical analysis for the benchmark of MOMA with other methods (fold and superfamily levels). This analysis was realized with StAR server (<http://melolab.org/star/home.php>). In the diagonal AUC values are shown (bold type). The values reported in the upper-right triangle correspond to AUC differences of each pairwise comparison of methods. In the lower-left triangle are P-values of Mann-Whitney U-statistic non-parametric test for the AUC differences calculated. The values in green indicate that the observed differences are statistical significant ($\alpha=0.05$), otherwise the values are shown in red.

Fold level

Methods	Structal	TopMatch	MOMA	GANGSTA+	QPtableau	SHEBA	Yakusa	FATCAT flex
Structal	0.9557	0.00019	0.01557	0.03900	0.07890	0.08551	0.16544	0.11838
TopMatch	0.93415	0.9555	0.01538	0.03882	0.07871	0.08532	0.16525	0.11819
MOMA	< 1E-05	0.00021	0.9401	0.02344	0.06333	0.06994	0.14987	0.10281
GANGSTA+	< 1E-05	< 1E-05	< 1E-05	0.9167	0.03990	0.04650	0.12643	0.07937
QPtableau	< 1E-05	< 1E-05	< 1E-05	< 1E-05	0.8768	0.00661	0.08654	0.03948
SHEBA	< 1E-05	< 1E-05	< 1E-05	< 1E-05	0.14672	0.8702	0.07993	0.03287
YAKUSA	< 1E-05	< 1E-05	< 1E-05	< 1E-05	< 1E-05	< 1E-05	0.7902	0.04706
FATCATflex	< 1E-05	< 1E-05	< 1E-05	< 1E-05	< 1E-05	< 1E-05	< 1E-05	0.8373

Superfamily level

Methods	Structal	TopMatch	MOMA	GANGSTA+	QPtableau	SHEBA	Yakusa	FATCAT flex
Structal	0.9696	0.00409	0.01319	0.05801	0.05197	0.08021	0.11199	0.05817
TopMatch	0.07733	0.9737	0.01729	0.06211	0.05606	0.08431	0.11608	0.06227
MOMA	< 1E-05	0.00009	0.9564	0.04482	0.03877	0.06702	0.09879	0.04498
GANGSTA+	< 1E-05	< 1E-05	< 1E-05	0.9116	0.00604	0.02220	0.05398	0.00016
QPtableau	< 1E-05	< 1E-05	< 1E-05	0.21668	0.9176	0.02825	0.06002	0.00621
SHEBA	< 1E-05	< 1E-05	< 1E-05	< 1E-05	< 1E-05	0.8894	0.03177	0.02204
YAKUSA	< 1E-05	< 1E-05	< 1E-05	< 1E-05	< 1E-05	< 1E-05	0.8576	0.05381
FATCATflex	< 1E-05	< 1E-05	< 1E-05	0.97525	0.18959	0.0018	< 1E-05	0.9114

Supplementary Table S5. Statistical analysis for the benchmark of MOMA with different methods to assign secondary structure (DSSP and Stride). Each set contains 10,000 pairs (5,000 positive and 5,000 negative cases) that were randomly selected from the original set (ASTRAL SCOP 2.3 40% sequence identity dataset). The statistical analysis was carried out with StAR server (<http://melolab.org/star>). The values in red indicate that the observed differences are not statistical significant ($\alpha=0.05$).

SCOP classification		Fold		Superfamily		Family	
Methods	DSSP	Stride	DSSP	Stride	DSSP	Stride	
Area under ROC curve (AUC)	0.9521	0.9521	0.9582	0.9567	0.9835	0.9833	
Accuracy	0.8846	0.8844	0.8933	0.8911	0.9376	0.9390	
Optimal Threshold (Sr)	22.7	22.3	23.4	25.8	30.9	30.3	
False positive rate	0.1148	0.1266	0.1100	0.0884	0.0518	0.0594	
True positive rate	0.8890	0.8954	0.8966	0.8706	0.9270	0.9374	
AUC differences	0.0009		0.0015		0.0002		
p-value	0.4685		0.1800		0.8017		

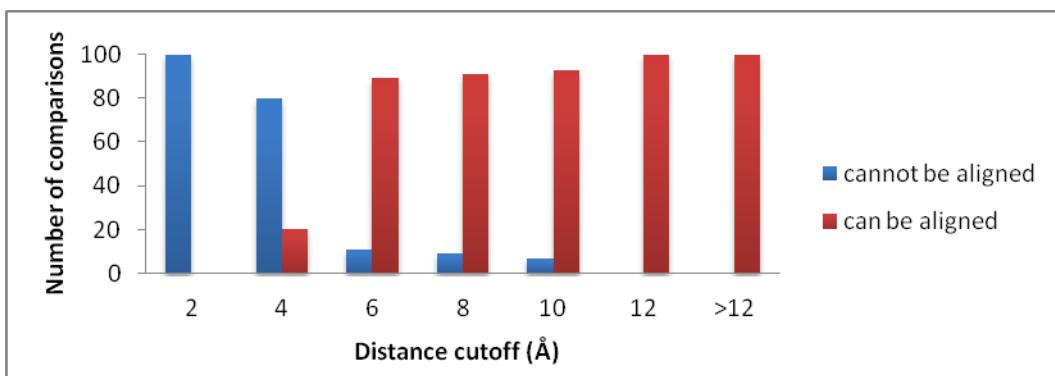
Sr: percentage of relative similarity (Sippl 2008)

Supplementary Table S6. Set of 100 distant homologous protein pairs obtained from HOMSTRAD database.

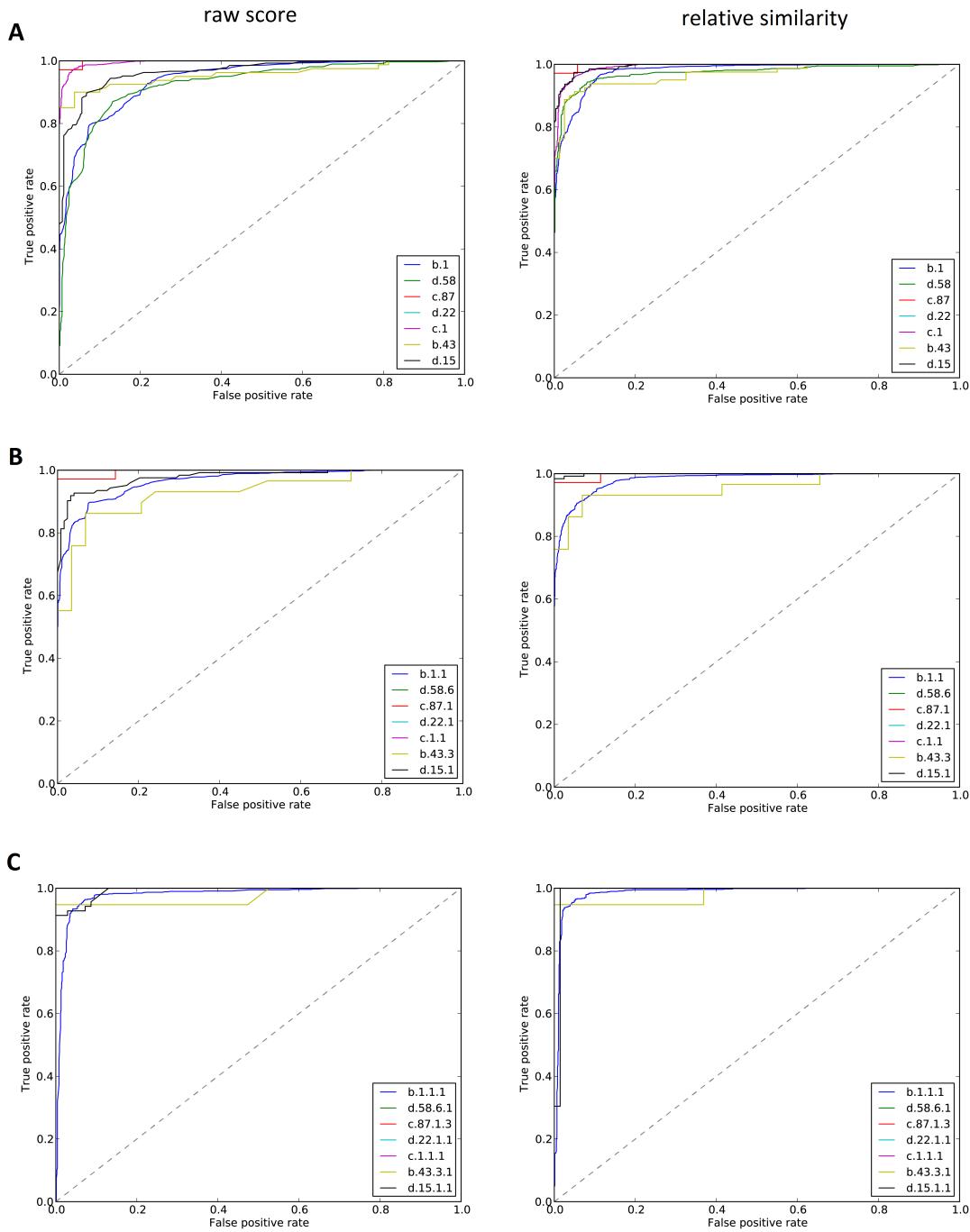
Query	Target	Number of SSE	HOMSTRAD family
154l	1qsaa	5	Transglycosylase SLT domain
1a0i	1fvia	12	DNA ligase
1a41	1a31a	11	Topoisomerase I core
1a48	1kuta	15	SAICAR synthetase
1a7j	1esma	13	Phosphoribulokinase/Uridine kinase
1a9na	1d0ba	8	Leucine rich repeats in splicesomal and internalin B
1ab4	1bjt	28	Type II DNA topoisomerase
1al3	1i6aa	13	LysR
1am2	1at0	11	Hint (Hedgehog/Intein)
1amua	1lci	34	AMP binding
1b3qa	1bxda	8	Histidine kinase
1b63a	1h7sa	19	DNA mismatch repair protein
1b74a	1jfla	15	Asp/Glu/Hydantoin racemase
1b8ba	3pfla	23	Glycine radical
1b9ia	1jg8a	20	DegT/DnrJ/EryC1/StrS
1bf2	1ehaa	22	Isoamylase and glycosyltrehalose trehalohydrolase
1bgla	1bhga1	14	Glycosyl hydrolase family 2
1bgyb	1bgya	22	Insulinase
1bhga2	1dp0a	8	Glycosyl hydrolase family 2, sugar binding domain
1bk0	1dc5	19	Iron/Ascorbate oxidoreductase
1bp12	1bp11	9	LBP/BPI/CETP
1bs9	1cex	11	Cutinase
1by5a	1fepa	29	TonB-dependent receptor proteins
1clc	1tf4b	16	Glycosyl hydrolases family 9
1cqea	1mhla	19	Animal haem peroxidase
1ct5a	1bd0a	16	Alanine racemase, N-terminal domain
1d2ra	2ts1	17	Tyrosyl-tRNA synthetase
1dhpa	1nal1	20	Dihydrodipicolinate synthetase
1dpe	1jeta	30	Bacterial extracellular solute-binding proteins, family 5
1dqaa	1qaya	24	HMG-CoA reductase
1dqwa	1dbta	15	Orotidine 5'-phosphate decarboxylases
1e42a	1qtsa	15	Alpha adaptin AP2, C-terminal
1eg2a	1boo	15	DNA methylase
1egua	1cb8a	38	Polysaccharide lyase family 8
1ejea	1i0ra	11	Flavin reductase like domain
1em2a	1jssa	12	START domain
1ems	1fo6a	20	Carbon-nitrogen hydrolase
1erja	1gotb	29	WD domain, G-beta repeat
1evsa	1lki	4	LIF/OSM
1ezia	1h7ea	13	Cytidylyltransferase
1f8sa	1h83a	30	Flavin containing amine oxidase
1fi4a	1h72c	18	GHMP kinases putative ATP-binding protein
1fioa	1ez3a	3	Syntaxin
1fm2a	1e3aa	37	Penicillin amidase
1fmta	2gar	11	Formyl transferase
1fo4a	1ffvc	19	FAD binding domain in molybdopterin dehydrogenase
1fs0g	1e79g	10	ATP synthetase g
1ft1a	1dcea	15	Protein prenyltransferase alpha subunit repeat

Query	Target	Number of SSE	HOMSTRAD family
1fxxa	1j54a	12	Exonuclease
1g6ga	1qu5a	6	FHA domain
1g6sa	1ejda	32	EPSP synthase
1ga8a	1ll3a	18	Glycosyl transferase family 8
1ggxa	1gfla	12	Green/red fluorescent protein
1gln	1qtqa	17	tRNAsynthetase 1c
1gr0a	1jkia	19	Inositol 1-phosphate synthetase
1gtta	1hyoa	11	FAA hydrolase
1gvfa	1dosa	19	Fructose-bisphosphate aldolase class II
1hu3a	1h6ka	9	MIF4G domain
1i7da	1cy9a	12	DNA topoisomerase
1igra2	1igra1	6	Receptor L domain
1iira	1f0ka	24	Glycosyltransferase family 28
1iq0a	1f7ua	31	tRNAsynthetase 1d
1iq8a	1k4ga	22	Queuine tRNA-ribosyltransferase
1jcua	1hrua	13	yrdC domain
1jdia	1fua	13	Class II Aldolase
1jeyb	1jeya	25	KU domain
1jgta	1ct9a	31	Asparagine synthase
1jj2e	1rl6a	14	Ribosomal protein L6
1jj2j	1whi	6	Ribosomal protein L14
1jmkc	1keza	11	Thioesterase domain
1jqra	1bpvy	13	DNA polymerase X
1js8a	1bt3a	10	Tyrosinase
1k6da	1poia	16	Coenzyme A transferase
1kas	1afwa	17	Thiolase
1kit2	1kit1	12	Sialidase N terminal
1l5ja	1c96a	26	Aconitase
1l8aa	1trka	16	Transketolase
1lam	1gyta	10	Cytosol aminopeptidase
1lnsa	1ju3a	31	Peptidase S15
1m2vb	1m2oa	36	Sec23/Sec24
1moq2	1moq1	11	Isomerase domain
1n1ma	1h2wa	16	Peptidase S9
1n2za	1efdn	15	Periplasmic binding protein
1pbe	1foha	23	PHBH-like
1pbwa	1tx4a	10	GTPase-activator protein for Rho-like GTPases
1qgia	1chka	11	Glycosyl hydrolase family 46
1qhta	1ih7a	20	DNA polymerase family B, C-terminal
1rdr	1khva	17	RNA dependent RNA polymerase
1rkfd	1bx4a	22	Ribokinase-like
1rmg	1bhe	22	Glycosyl hydrolase family 28
1thfd	1qo2a	17	Histidine biosynthesis protein
1uag	1fgs	15	Mur ligase
1vsga	2vsga	10	Trypanosome variant surface glycoprotein
1yaca	1nbaa	12	Isochorismatase
2abk	1mun	10	Endonuclease III
2pgi	1dqra	27	Phosphoglucose isomerase
2sqca	1ft1b	12	Prenyltransferase and squalene oxidase repeats
2tmda	1oyc	19	FMN oxidoreductase
2tysb	1tdj	21	Pyridoxal-phosphate dependent enzymes

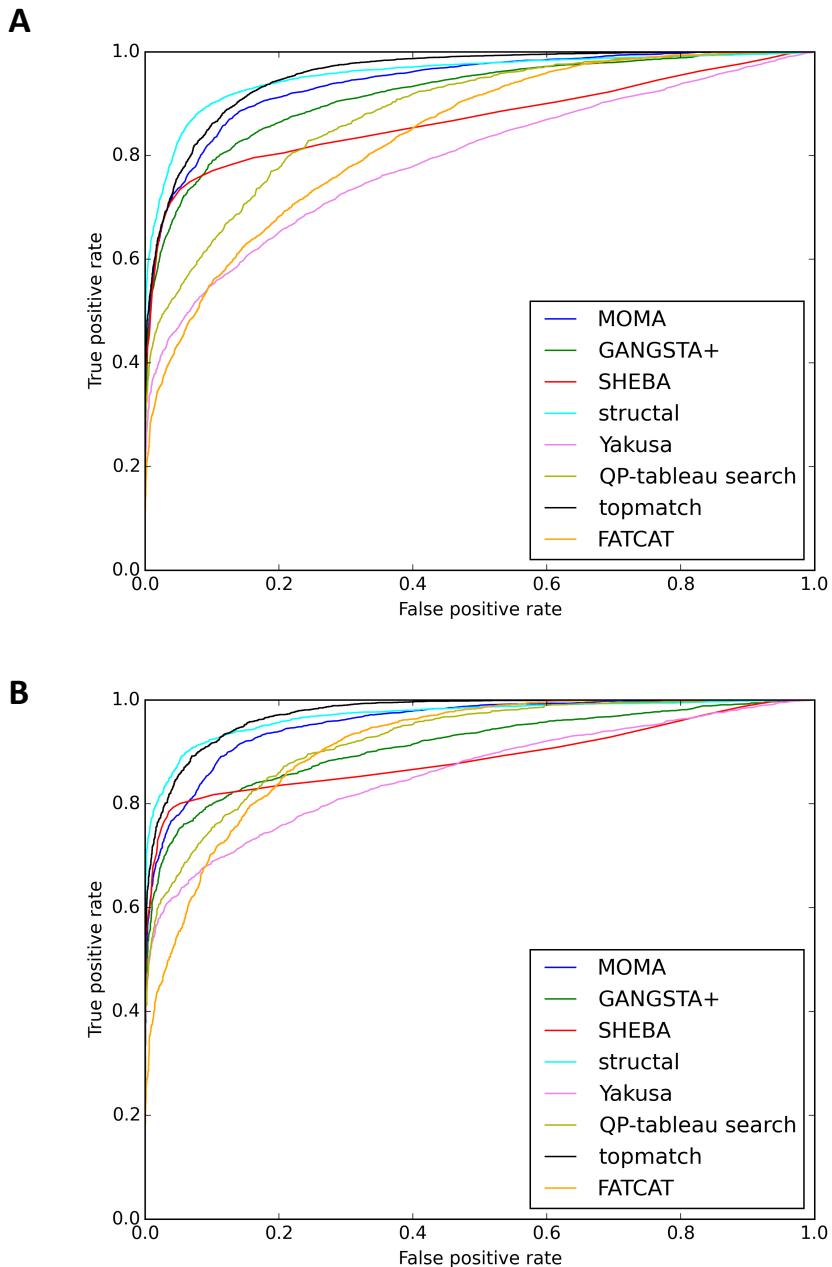
These alignments were used to calibrate some parameter values of MOMA. The total number of matching secondary structural element (SSE) pairs, according to HOMSTRAD structural superpositions and STOVCA structural alignment derivations with default parameter values, are shown. In the last column, the family name of each protein pair is displayed.



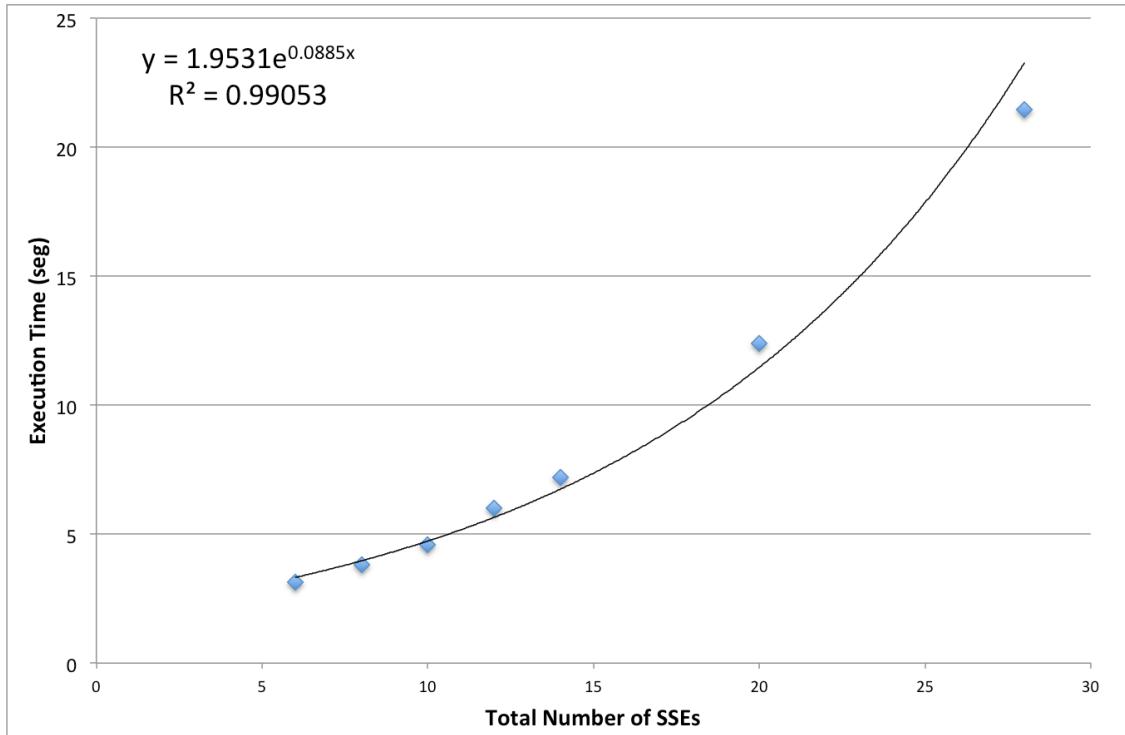
Supplementary Figure S1. Calibration of distance cutoff using the HOMSTRAD set with the best combination of parameter values. Histogram shows the number of comparisons carried out with MOMA varying the distance cutoff to define a contact between two secondary structure elements in the matrices. Blue and red bars indicate the comparisons that cannot and can be aligned using MOMA, respectively. The comparisons were carried out with the best combination of parameter values.



Supplementary Figure S2. ROC curves for the small set of seven most common folds according to TOPS database using raw score S (left) and relative similarity S_r (right) from MOMA. The searches were realized against ASTRAL 2.3 95% identity sequence, using the fold (A), superfamily (B) and family (C) levels as defined in SCOP database. The naming scheme of these proteins is from SCOP.



Supplementary Figure S3. ROC curves of classification at the SCOP fold (A) and superfamily level (B) for the large set of 100 proteins compared with other methods against ASTRAL SCOP 2.3 40% sequence identity dataset.



Supplementary Figure S4. Execution time of MOMA when varying the number of SSE considered, using the seven most common folds as a query. The searches were realized against ASTRAL 2.3 95% identity sequence, using the fold, superfamily and family levels as defined in SCOP database.

Algorithm: Method for generating blocks

Inputs: ΔSM , the Δ submatrix; N , the number of rows (Δ submatrix is a square matrix)

Output: B , a list of blocks

Require: $N > 0, C > 0$

```
1: C ← 90
2: B ← []
3: end_block ← 1
4: nb ← 0           # Size of block
5: i ← 0
6: while i < N do
7:     j ← 0
8:     if i = 0 then
9:         pi ← i      # Position i of block
10:        pj ← i     # Position j of block
11:    end if
12:    while j < i do
13:        if j = i - 1 then
14:            if  $\Delta\text{SM}_{i,j} \geq C$  and  $\Delta\text{SM}_{i,j} = \text{NULL}$  then
15:                end_block ← 1
16:                k ← j
17:                while k ≥ 0 do
18:                    if  $\Delta\text{SM}_{i,k} \neq \text{NULL}$  and  $\Delta\text{SM}_{i,k} < C$  then
19:                        end_block ← 0
20:                        break while
21:                    end if
22:                    k ← k - 1
23:                end while
24:                if end_block = 1 then
25:                    k = i + 1
26:                    while k < N do
27:                        if  $\Delta\text{SM}_{k,j} \neq \text{NULL}$  and  $\Delta\text{SM}_{k,j} < C$  then
28:                            end_block ← 0
29:                            break while
30:                        end if
31:                        k ← k + 1
32:                    end while
33:                end if
34:            end if
35:            nb ← nb + 1
36:        end if
37:        if end_block = 1 then
38:            if nb > 2 then
39:                push (pi, pj, nb) onto B
40:            end if
41:            nb ← 0
42:            pi ← i
43:            pj ← i
44:        end if
45:        j ← j + 1
46:    end while
47:    i ← i + 1
48: end while
49: if (nb + 1) > 2 then
50:     push (pi, pj, nb) onto B
51: end if
```

Supplementary Figure S5. Algorithm used for extracting the rigid local matches. Pseudocode of the algorithm used for the extraction of independent protein segments that will follow different geometrical transformations for the global flexible superposition of secondary structure elements. The algorithm extracts one or more contiguous rectangular blocks of SSE pairs.