# Protein structure comparison based on fold evolution

Natalja Kurbatova, Laura Mančinska, Juris Vīksna

Institute of Mathematics and Computer Science
University of Latvia
Raina bulvaris 29
LV-1459 Riga, Latvia
natalja@lu.lv

**Abstract:** The paper presents a protein structure comparison algorithm that is capable to identify specific fold mutations between two proteins. The search for such mutations is based on structure evolution models suggesting that, similarly as sequences, protein folds (at least partially) evolve by a stepwise process, where each step comprises comparatively simple changes affecting few secondary structure elements. The particular fold mutations considered in this study are based on the work by Grishin [Gr01].
The algorithm uses structure representation by 3D graphs and is a modification of a method used in *SSM* structure alignment tool [KH04a]. Experiments demonstrate that our method is able automatically identify 85% of examples of fold mutations given by Grishin. Also a number of tests involving all-against-all comparisons of CATH structural domains have been performed in order to measure comparative frequencies of different types of fold mutations and some statistical estimations have been obtained.

## 1 Introduction

Traditional viewpoint regarding protein sequence and structure similarity of homologous proteins is that protein structure is much better preserved than protein sequence and that sequence similarity of about 25% or more almost necessarily implies that protein structures will be almost identical. Whilst this is true in most of the cases, nevertheless it is possible to find pairs of proteins with highly similar sequences and at the same time noticeable structural differences. Probably the best known example is Janus protein designed by Dalal *et al.* [DBR97] in response to Paracelsus Challenge [RC94]. The authors have synthesised a pair of proteins with 50% sequence similarity and, at the same time, completely different folds. Although it is possible to argue that this is a designed protein, it still demonstrates the credibility of evolutionary events that preserve sequence similarity but change protein fold.

The existence of protein pairs with similar sequences and different structures also is implied by current models of protein evolution - although during the evolution protein structure is much more preserved than protein sequence, there should exist small sequence mutations that lead to noticeable structural changes. From biological perspective the problem is thoroughly studied by Grishin *et al.* [Gr01], [KG02]. The authors have identified

a set of possible fold mutations that could occur during protein evolution, each of the proposed mutations is confirmed by real biological examples. Similar sets of small fold changes are proposed and studied also by other authors, e.g. Matsuda *et al.* [Ma03] and Przytycka *et al.* [PSR02], – although these studies give less biological motivation and are more interested in the exploration of protein fold space under assumption that structures have evolved by a stepwise process, each step consisting of a small fold change belonging to the proposed set.

Most of the fold mutations proposed by Grishin presumably are the result of accumulated point-mutations (indels and substitutions of single amino acids) in protein sequence. Alternatively, structural changes can arise from circular permutations of protein fragments (the likely cause for this process is gene duplication followed by truncation of protein). This process has been studied by several authors [JL01], [PRT06], [Ul99], [WTB05] and also is confirmed by biological examples.

The process of structure evolution present us several interesting and challenging problems. Firstly, although we may know some real examples of pairs of proteins confirming one or another structural change, it could be useful to estimate the comparative frequencies with which different structural changes could occur. Secondly, it could be useful to have tools for structure comparison that can automatically identify such structural changes (and possibly estimate "evolution distance" on the basis of observed changes). Such tools can be useful either directly for comparison of two structures or for search of structural changes within database of known protein structures in order to gain better understanding of the nature of structure evolution.

Regarding the first problem, an attempt to estimate frequencies of different types of structural changes has been made by Viksna and Gilbert [VG07]. Although not conclusive, these results confirm often used assumption that most probable fold changes are indels of single helices and indels of single strands at one end of β-sheets. Rough estimates for frequencies of other types of structural changes also have been obtained.

To address the second problem, initially one need to chose a representation of protein structures that is most suited for such task. Traditionally there are two different approaches to protein structure comparison. The most often used approach is to describe the structures by 3D coordinates of backbone (and sometimes also side chain) atoms and then search for superposition of two structures that minimizes RMSD distance between selected pairs of atoms. Probably this is also the approach that can give the most precise results. Interestingly, a modification of this method have also been attempted for computing evolutionary distances between protein structures [Gr97]. However, atomic coordinate representation of structures doesn't contain any information about SSEs and hence is not well suited for identification of structural changes that are explicitly characterized in terms of SSEs.

Another alternative is a topological approach – the protein is described by some kind of graph; usually vertices of such graph represent either SSEs or individual atoms, edges can represent hydrogen bonds, distances between SSEs/atoms etc. The structure comparison problem then is translated to graph comparison, which most often means searching for a largest common subgraph. Initially the development of graph based methods probably was mainly motivated by looking for faster (but less exact) alternatives for structure compari-

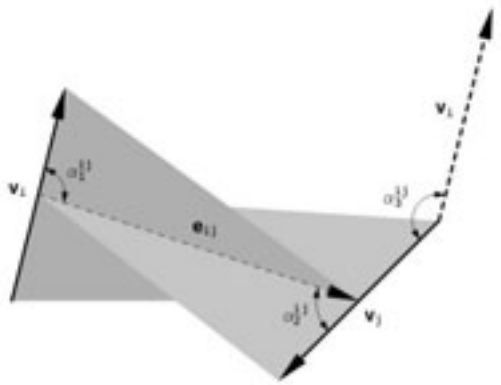

Figure 1: Construction of 3D graph

son, however the approach also has some intrinsic advantages, since it explicitly allows to include information about SSEs, hydrogen bounds etc. Such purely topological representation of protein structures [WTB05] was used in [VG07]. However, whilst the approach is justified for statistical analysis of probabilities of structural changes, the topological representation contains too little information for comparison results for a given particular pair of structures to be trusted.

Recently there have been several and successful attempts to combine these two approaches by introduction of so called 3D graphs [KH04b], [Wa02]. Basically "3D graph" means that structure is represented by a complete graph with vertex and edge labels that are sufficient to reconstruct the structure in 3 dimensions. Although there are few advantages of such approach if graph vertices represent individual atoms, the method becomes very useful for higher level structure representations, e.g. when edges represents SSEs. 3D graphs easily allow to incorporate also additional information about folds (types of SSEs, adherence of β-strands to particular sheets etc.). This makes 3D graph approach quite appropriate for structure comparisons that are able to identify particular structural changes between proteins and we have chosen this representation as a basis for our study.

## 2 Structure representation by 3D graphs and *SSM* algorithm

The notion of 3D graphs has been introduced in [Wa02] and the approach has been adapted for protein structure comparison by Krissinel *et al.* [KH04a], [KH04b]. We give a slightly simplified description of their method here. For a given protein structure the corresponding 3D graph is a complete undirected graph, with vertex set corresponding to the set of SSEs – the $i$-th SSE (according to their order from $C$ to $N$ terminus) is represented by vertex $i$. As SSEs are considered only β-strands with length 3 residues or more and α-helices

with length 5 residues or more. Each SSE is considered as a vector $v$ in 3D space (vectors being constructed according to recommendations from [SB97]) and each vertex is labelled with type $T$ of SSE (either α-helix or β-strand) and length $L$ of vector $v$. Edge between vertices $i$ and $j$ is labelled by distance $e_{ij}$ between the middle points of vectors $v_i$ and $v_j$ and 3 angles $\alpha_1^{ij}, \alpha_2^{ij}$ and $\alpha_3^{ij}$ describing the relative orientation of vectors in 3D space (see Figure 1).

For given two structures (3D graphs) *SSM* algorithm finds the largest common subgraph of these graphs. When searching for common subgraph, two vertices $i$ and $i'$ are compatible compatible if they have equal types and $\mid L_i - L_{i'} \mid \leq p_1(L_i + L_{i'}) + p_2$. Two edges $(i, j)$ and $(i', j')$ are compatible if $\mid e_{ij} - e_{i'j'} \mid \leq p_3(e_{ij} + e_{i'j'}) + p_4$ and $\mid \alpha_k^{ij} - \alpha_k^{i'j'} \mid \leq r_i$ for $k = 1...3$. After the largest common subgraph is found, *SSM* algorithm chooses two sets of $C_\alpha$ atoms belonging to SSEs matched by common subgraph and computes a 3D alignment with minimal RMSD for atoms belonging to these sets. Finally a RMSD-based $Q$-score is computed to characterize the quality of alignment.

The results depend from a set of 7 parameters $p_1...p_4$ and $r_1...r_3$, the values of which have been adjusted experimentally. In practice the searching for largest common subgraph is done with additional constraint that mapping induced by the subgraph have to preserve the vertex order (i.e. order of SSEs), since use of such constraint produces better result. As output the *SSM* algorithm produces ordered alignment of SSEs together with alignment score $Q$.

## 3 Types of fold mutations

The set of fold changes considered in this study is based on the set proposed by Grishin in [Gr01]. The definition of fold changes involves β-strands ($E$), α-helices ($H$), loops, β-hairpins (defined here as two adjacent β-strands that also hold adjacent positions in a β-sheet) ($S_2$) and 3-β-meanders (three adjacent β-strands that also hold adjacent positions in a β-sheet)($S_3$). We consider the following set of mutations (each of them can occur in both directions):

1. *Insertions (deletions):* loop $\longleftrightarrow \{E, H, S_2, S_3\}$,
2. *Substitutions:* $E \longleftrightarrow H$,
3. *Substitutions:* $S_2 \longleftrightarrow \{E, H\}$,
4. *Substitutions:* $S_3 \longleftrightarrow \{E, H\}$,
5. β-*hairpin swaps:* exchange of the order of β-hairpin's strands in a β-sheet.

In addition to fold changes proposed by Grishin, this set includes indels and substitutions of β-hairpins (the existence of such changes was suggested in [VG07]). Insertions of β-strands could be further classified in sheet extensions by β-strands and strand invasions/withdrawals – the insertion of a β-strand into existing β-sheet that requires H-bond

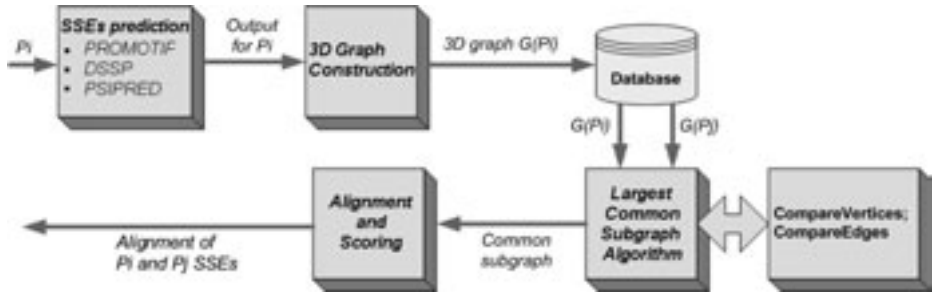


Figure 2: Flowchart of algorithm for structure comparison

breakage and formation of the H-bonding pattern on both sides of the inserted β-strand. However, we have omitted this distinction in this study (and results from [VG07] suggests that strand invasions/withdrawals are quite rare). The only structural change proposed in [Gr01] and not considered here is circular permutation of SSEs (although in principle such changes could be discovered by our approach, the first challenge here is to define in exact terms what we should be looking for).

## 4 Modified *SSM* algorithm for recognition of fold mutations

The general structure of our modification of *SSM* algorithm is given in Figure 2. The process can be divided in two stages – construction of 3D graph and alignment of two given 3D graphs. The general scheme is the same as for *SSM* method given in [KH04a], however there are modifications in 3D graph construction, largest common subgraph algorithm and functions providing edge and vertex comparisons.

The input data for 3D graph construction is a PDB file. In addition, a program is used for determining structures SSEs. In our experiments we have used both PROMOTIF [HT96] and DSSP [KS83] to evaluate the impact of accuracy of SSE predictions; unfortunately some ”prediction noise” is characteristic for both of these programs.

The construction of 3D graph is similar as for *SSM* algorithm, with the following modifications:

- We distinguish between 3 different types of SSEs: β-strands, α-helices and $3_{10}$-helices. Motivation for this is to decrease impact of ”noise” from SSE prediction (results from [VG07] suggest that helix types are the most affected by ”noise”). We also consider SSEs of any size. Vectors to SSEs are assigned using the methodology from [SB97]; however the method doesn't apply to short SSEs, in which case we take the vectors connecting centres of $C_\alpha$ atoms from the first and the last SSE residue.
- ”Virtual vertices” representing 3-β-meanders are included in 3D graph. For a 3-β-

meander composed from β-strands with the corresponding vectors $v_1$,$v_2$ and $v_3$ the vector $v_m$ connects the starting point of $v_1$ with the endpoint of $v_3$. Vector $v_m$ is used as reference to compute labels for edges connected to this virtual vertex.

- "Virtual vertices" representing β-hairpins are included in 3D graph. For a β-hairpin composed from β-strands with the corresponding vectors $v_1$ and $v_2$ the vector $v_h$ connects the starting point of $v_1$ with the starting point of $v_2$. Vector $v_h$ is used as reference to compute labels for edges connected to this virtual vertex

Although construction of 3D graph can be done on the fly for each structure comparison, for all-against-all comparison 3D graphs are created in advance and stored in database.

The part of algorithm for finding the largest common subgraph is based on backtracking search used in *SSM* [KH04b]. However in our case the mapping of some of the vertices can affect the mapping of some others (e.g. if a β-strand from 3-β-meander is involved in mapping, 3-β-meander as a whole can't be used). For that purposes two new procedures *HoldReferences* and *ReleaseReferences* have been added that locks/unlocks vertices depending from their availability for matching in current backtracking state. Also, to detect β-hairpin swaps for every pair of hairpins matching for β-strands in both direct and reverse order are checked.

When searching for common subgraph, vertex $i$ is considered compatible with a vertex $i'$ of the same type if $\mid L_i - L_{i'} \mid \leq p_1(L_i + L_{i'}) + p_2$ (for some parameters $p_i$); or with a "virtual vertex" if $\mid L_i - L_m \mid \leq p'_1(L_i + L_m) + p'_2$ (where $L_m$ is length of a 3-β-meander vector; less restrictive parameter $p'_i$ values are used for this case). The same approach is used also for matching of β hairpins. Two virtual vertices are not compatible. Compatibility requirement for edges remains the same as for original *SSM* – edges $(i, j)$ and $(i', j')$ are compatible if $\mid e_{ij} - e_{i'j'} \mid \leq p_3(e_{ij} + e_{i'j'}) + p_4$ and $\mid \alpha_k^{ij} - \alpha_k^{i'j'} \mid \leq r_i$ for $k = 1...3$; but less restrictive parameter values are used for edges involving 3-β-meanders and β-hairpins. The parameter values for vertices/edges involving 3-β-meanders and β-hairpins have been adjusted experimentally.

Matching of 3-β-meanders and and β-hairpins are scored with the same weight as other SSE's – the algorithm finds the common subgraph with the largest possible total number of virtual and non-virtual vertices.

We have implemented the algorithm in *C++* language in *Linux* environment. The running of algorithm is exponential, however in practice the results are obtained within few seconds for comparison of structures containing up to 70 SSEs.

# 5 Results

## 5.1 Method validation on known biological examples

To validate our method we have used it to detect fold mutations in pairs of protein structures published by Grishin [Gr01]. From 15 of such examples we successfully found fold

Figure 3: Example showing β-strands insertion (type $E^{ins}$) type for proteins **1fow** and **1hcrA**.

Figure 4: Example showing 3-β-meander substitution with α-helice (type $HS_3^{sub}$) for protein segments **1ldnA** (residues A20-A265) and **1npx** (residues 149-315).

mutations of different types in 13 cases. Some examples are shown in Figures 3 and 4. We use **PyMOL** software for ribbon-style representations of proteins [De02].

### 5.2 All-against-all comparisons of CATH protein domains

Experiments involving all-against-all comparisons of CATH ([Or97]) protein domains have been performed in order to check whether the method is capable to automatically detect fold mutations and to obtain some estimates about relative frequency of different types of fold changes.

For experiments we used two representative sets from CATH, which were proposed in [VG07] and are constructed with the aim to limit over representation of widely studied protein folds. These sets CATH2 and CATH3 were obtained from representative set CATH-95 (provided by CATH database and containing proteins with less than 95% sequence sim-

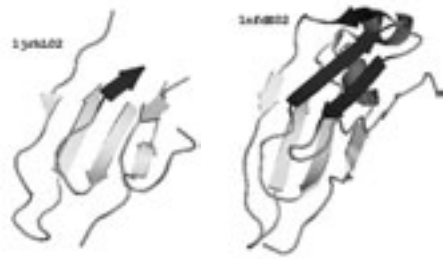

Figure 5: Example showing substitution 3-β-meander substitution with β-strand (type $ES_3^{sub}$) and α-helices insertion (type $H^{ins}$) in domains **1jrhL02** and **1nfdE02** from CATH2. Domains have 39% sequence similarity.

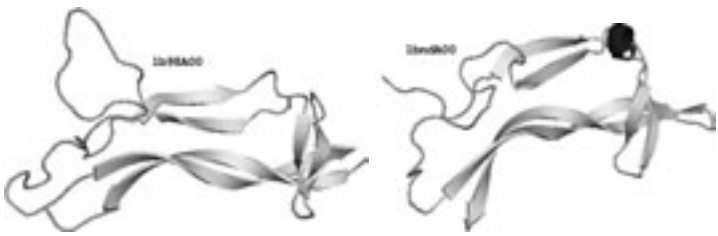

Figure 6: Example showing α-helice insertion (type $H^{ins}$) for domains **1b98A00** and **1bndA00** from CATH2. Domains have 51% sequence similarity.

ilarity) by additionally removing structures obtained with NMR technology as well as those with resolution greater than 3Å as well as reducing number of proteins from over-represented folds (for more technical details see [VG07]). The obtained representative set CATH2 contains 685 domains from CATH class 2 and representative set CATH3 contains 1182 domains from CATH class 3.

Sets CATH2 and CATH3 were treated separately, because few "short" evolutionary relations are to be expected between these groups, and also to check how mutation frequencies differ between these groups. In assessment of results we used as a reference also sequence similarity scores between CATH domains, computed by **ssearch** implementation of Smith-Waterman algorithm [SW81].

In analysis of results we looked at pairs of domains with sequence similarity 20% or more (taking in account also structure similarity these could be expected to be evolutionary related). Number of such pairs for CATH2 is 1226 and 1593 for CATH3 respectively. Manual checking of these pairs in many cases confirmed the detected fold changes (a noticeable obstacle in this manual checking was ambiguity in structure's division in SSEs, thus it is also difficult to give good estimation of the proportion of confirmed cases). Some of the discovered fold changes are given in Figures 5 (3-β-meander substitution with β-strand) and 6 (insertion of α-helice).

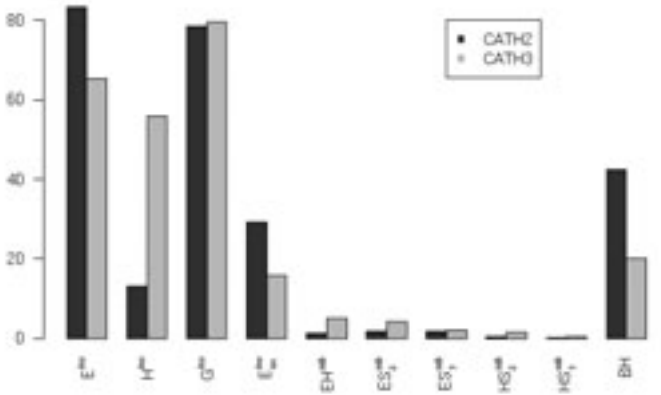


Figure 7: Probabilities for different types of fold mutations in CATH2 and CATH3 classes.

In contrast to the method used in [VG07] our algorithm was able to relate two structures with an arbitrary number of fold changes. Whilst being a certain advantage for discovery of structural changes, this poses some difficulties for statistical analysis of results (one have to decide on the weight of observed changes depending from total number of structures and structure size) and the appropriate scoring scheme still have to be developed. For this reason we provide here just overall probability of fold changes, probability value can be interpreted as "probability that a particular fold change occurs for proteins between which fold changes have been detected". This probability value was obtained by using a simple formula:

$$\frac{\textit{Number of pairs of domains with fold mutation of a given type}}{\textit{Total number of pairs of domains}}$$

Statistical estimations have been obtained for following types of fold mutations:

1. *Insertions:* $E^{ins}$ (β-strand insertion), $H^{ins}$ (α-helice insertion), $G^{ins}$ ($3_{10}$-helice insertion)
2. *Substitutions:* $EH^{sub}$ (β-strand substitution with α or $3_{10}$ helice), $ES_2^{sub}$ (β-strand substitution with β-hairpin), $ES_3^{sub}$ (β-strand substitution with 3-β-meander), $HS_2^{sub}$ (α or $3_{10}$ helice substitution with β-hairpin), $HS_3^{sub}$ (α or $3_{10}$ helice substitution with 3-β-meander).
3. β-*hairpin swaps:* $BH$

First of all, we computed probabilities for CATH2 class using different SSEs prediction programmes: DSSP and PROMOTIF. Differences between obtained results are in range of 10%. Manual checking of results lead to the conclusion that at least in case of CATH domains DSSP programme results are more convenient for insertions and substitutions, but PROMOTIF results are more accurate for β-hairpin swaps.

The most popular change in CATH2 class is β-strand insertion $E^{ins}$ - 83%. That means we found β-strand insertion $E^{ins}$ in 83% of cases (100% of all inspected pairs of domains). Almost in 30% of cases β-strand insertion takes place in existing β-sheet ($E^{ins}_{ex}$ type). This estimation correspond to one obtained in [VG07], where changes corresponding to our definition of $E^{ins}$ was the most popular.

We have 79% for $G^{ins}$ type (small $3_{10}$-helices insertion), but after manual checking of results we supposed that in most cases these changes are the result of the noise (SSEs prediction programme and our algorithm together). Besides estimations for $G^{ins}$ type are questionably similar in both cases for CATH2 and for CATH3. In turn, manual checking of $H^{ins}$ type confidenced us in biologically relevance of it – 56% for CATH3 and 13% for CATH2. At the same time $E^{ins}$ decreased almost for 20% for CATH3. This results seems quite logical since CATH3 class contains mixed α-β proteins, but CATH2 mostly contains proteins with β-strands.

In general we obtained much more insertions in comparison with substitutions. This result could be explained partly with noise of prediction programmes and necessarily of normalisation by number of SSEs in protein. From other point of view pairs of domains have sequence similarities more than 20% and checked examples with insertions at least $E^{ins}$ and $H^{ins}$ are biologically relevant. Taking into account all these observations we suggest that SSE addition/deletion happened more often than SSEs substitution at least in CATH classes 2 and 3.

Our results suggested also that β-hairpin swap probability is 42% in CATH2 and 20% in CATH2. This result have to be more carefully checked, but, again, difference between CATH2 and CATH3 make it at least interesting for future work.

## 6 Summary and Conclusions

We have developed an algorithm for protein structure comparison based on evolutionary changes. Our method is a modification of approaches described in [KH04a], [KH04b] for 3D graph construction and common subgraph isomorphism detection.

The method was found to be efficient and accurate to find evolutionary changes of specific types in protein structures comparing SSEs of two proteins. In contrast to topological approach described in [VG07] we are able to detect arbitrary number of fold mutations between two structures and to deal with substitutions and β-hairpin swaps. Since the comparison produces a score that allows to estimate structural similarity, it has the potential to discover proteins related by structural change and having small sequence similarity (in contrast to topological approach where discovered fold changes is verifiable only for on the basis of sequence similarity).

Experiments have been performed to validate the method on biologically confirmed fold changes and the method was able automatically identify 85% of examples given by Grishin [Gr01], [KG02]. All-against-all comparisons have been performed for subgroups of protein domains from CATH classes 2 and 3 in order to evaluate the capability of the method to discover new protein pairs related by structural changes and to estimate the comparative frequencies for different types of fold mutations. Although it is not as yet clear how to assess the validity of predicted fold mutations, manual checking confirm that algorithm in principle is able to discover fold changes of different types. Rough estimates for comparative frequencies for several types of fold mutations were also obtained. The experiments also highlighted the necessity to use more robust programs for SSE prediction and showed that the most problematic is assignment of $3_{10}$-helices.

## Acknowledgements

This work has been supported by the European Social Fund (ESF).

## References

[DBR97] Dalal, S., Balusubramanian, S. and Regan L., "Protein alchemy: Changing β-sheet into α-helix," *Nature Structural & Molecular Biology*, 4: 548-552, 1997.

[De02] DeLano, W.L., "The PyMOL Molecular Graphics System," *DeLano Scientific*, Palo Alto, CA, USA. *http://www.pymol.org*, 2002.

[Gr01] Grishin, N., "Fold change in evolution of protein structures," *Journal of Structural Biology*, 134: 167-185, 2001.

[Gr97] Grishin, N., "Estimation of evolutionary distances from protein spatial structures," *Journal of Molecular Evolution*, 45: 359-369, 1997.

[HT96] Hutchinson, E. G. and Thornton, J. M., "PROMOTIF–A program to identify and analyze structural motifs in proteins," *Protein Science*, 5: 212-220, 1996.

[JL01] Jung, J. and Lee, B., "Circularly permuted proteins in the protein structure database," *Protein Science*, 10: 1881-1886, 2001.

[KS83] Kabsch, W. and Sander, C., "Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features," *Biopolymers*, 22: 2577-2637, 1983.

[KG02] Kinch, L. and Grishin, N., "Evolution of protein structures and functions," *Current Opinion in Structural Biology*, 12: 400-408, 2002.

[KH04a] Krissinel, E. and Henrick, K., "Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions," *Acta Crystallographica*, D60: 2256-2268, 2004.

[KH04b] Krissinel, E. and Henrick, K., "Common subgraph isomorphism detection by backtracking search," *Software – practice and experience*, 34: 591-607, 2004.

[Ma03] Matsuda, K., Nashioka, T., Kinoshita, K., Kawabata, T. and Go, N., "Finding evolutionary relations beyond superfamilies: fold-based superfamilies," *Protein Science*, 12: 2239-2251, 2003.

[Or97] Orengo, C.A., Michie, A.D., Jones, D.T., Swindelis, M.B. and Thornton, J.M., "CATH – a hierarchic classification of protein domain structures," *Structure*, 5: 1093-1108, 1997.

[PRT06] Peisajovic, S., Rockah, L. and Tawfik, D., "Evolution of new protein topologies through multistep gene rearrangements," *Nature Genetics*, 32: 168-174, 2006.

[PSR02] Przytycka, T., Srinivasan, R. and Rose, G., "Recursive domains in proteins," *Protein Science*, 11: 409-417, 2002.

[RC94] Rose, G.D. and Creamer, T.P., "Protein folding: predicting predicting," *Proteins: Structure, Function and Genetics*, 19: 1-3, 1994.

[SB97] Singh, A. and Brutlag, D., "Hierarchical protein structure alignment using both secondary structure and atomic representations," *Proceedings of ISMB-97*, 284-293, 1997.

[SW81] Smith,T.F. and Waterman,M.S., "Identification of common molecular subsequences," *Journal of Molecular Biology*, 147: 195-197, 1981.

[Ul99] Uliel S., Fliess, A., Amir, A. and Unger, R., "A simple algorithm for detecting circular permutations in proteins," *Bioinformatics*, 15: 930-936, 1999.

[VG07] Viksna, J. and Gilbert, D., "Assessment of the probabilities for evolutionary structural changes in protein folds," *Bioinformatics*, 23: 832-841, 2007.

[Wa02] Wang, X., Shapiro B., Rigoutsos, I. and Zhang, K., "Finding patterns in three-dimensional graphs: algorithms and application to scientific data mining," *IEEE Transactions on Knowledge and Data Engineering*, 14: 731-749, 2002.

[WTB05] Weiner, J., Thomas, G. and Bornberg-Bauer, E., "Rapid motif-based prediction of circular permutations in multi-domain proteins," *Bioinformatics*, 21: 932-937, 2005.

[We99] Westhead, D., Slidel, T., Flores, T. and Thornton, J., "Protein structural topology: automated analysis and diagrammatic representation," *Protein Science*, 8: 897-904, 1999.