Acta Biochimica Polonica

Vol. 51 No. 1/2004

161–172

QUARTERLY

*Review*

# Comparison of proteins based on segments structural similarity[✪]

Dariusz Plewczynski[1,2✉], Jakub Pas[2], Marcin von Grotthuss[2] and Leszek Rychlewski[2]

[1]*Interdisciplinary Center for Mathematical and Computational Modeling Warsaw University Warszawa, Poland;* [2]*Bioinformatics Laboratory BioInfoBank Institute, Poznań, Poland*

Received: 23 October, 2003; revised: 24 January, 2004; accepted: 29 January, 2004

**Key words**: 3D-hit, protein structure comparison, protein structure, structural hashing, liveBench, toolShop

**We present here a simple method for fast and accurate comparison of proteins using their structures. The algorithm is based on structural alignment of segments of Cα chains (with size of 99 or 199 residues). The method is optimized in terms of speed and accuracy. We test it on 97 representative proteins with the similarity measure based on the SCOP classification. We compare our algorithm with the LGscore2 automatic method. Our method has the same accuracy as the LGscore2 algorithm with much faster processing of the whole test set, which is promising. A second test is done using the ToolShop structure prediction evaluation program and shows that our tool is on average slightly less sensitive than the DALI server. Both algorithms give a similar number of correct models, however, the final alignment quality is better in the case of DALI. Our method was implemented under the name 3D-Hit as a web server at http://3dhit.bioinfo.pl/ free for academic use, with a weekly updated database containing a set of 5000 structures from the Protein Data Bank with non-homologous sequences.**

The three dimensional structure of proteins is highly conserved during evolution (Chothia & Lesk 1986). Comparison of 3D structures makes it possible to establish distant relation-

[✪]This work was supported by the EU grants (“ELM” QLRI-CT2000-00127 and “MiFriend” QLRT-2001-02884) and the State Committee for Scientifc Research grants (KBN Poland E-617/SPB/5.PR UE/DZ 438/2003-2004 and E-617/SPB/5.PR UE/DZ 439/2003).

[✉]To whom correspondence should be addressed: D. Plewczynski, Bioinformatics Laboratory BioInfoBank Institute, Limanowskiego 24A, 60-744 Poznań, Poland; phone: (48 61) 865 3520; fax: (48 61) 864 3350; e-mail: darman@bioinfo.pl

ships, even between protein families distinct in terms of sequence comparison alone. This is why structural alignment of proteins increases our understanding of more distant evolutionary relationships (Bujnicki, 2000; Johnson *et al.,* 1990). The link between structural classification and sequence families enables us to study functions of various folds, or whole proteins. It is a very promising part of bioinformatics, so the interest in the structural information of proteins is justified.

The main problem in the area is to distinguish between the similarities of 3D structures of proteins that come from evolutionary relationships and those that arise from common properties, or chemical constraints on protein folding. Not only in the case of sequence similarity but also for structural similarity, there is a twilight zone, where one cannot determine whether the similarity arises from biological relationship or from physical constraints. A structural motif may happen to be more similar to another one, and it is very difficult to compute the relative probability of such a situation. Genetic mechanisms rarely produce changes in the topological connectivity between secondary structure elements (Pointing & Russel, 1995). Sometimes (as in the case of formation of $\beta$ sheets and their packing into three dimensional layers) chemical forces may drive the formation of large structural motifs which gives false (non-evolutionary) similarities of structures (Murzin, 1994). In such a case there may be no similarity in the chain ordering of secondary structure elements. In order to avoid this problem and properly recognize the fold classification it is crucial to analyze longer parts, or whole C$\alpha$ chain of a protein. On the other hand, longer parts of a main chain give the larger structural differences between members of protein families which can change relative similarity probabilities within a group. This obstacle makes it difficult to establish a general similarity measure for all known protein families.

Nevertheless, it is important to somehow solve the problem of protein structure similarity using available databases. A detailed description of a protein provides information about positions of all atoms, the main chain and side chains. From this variety of data we take only the coordinates of C$\alpha$ atoms. We believe that the most important information is contained in the backbone of a protein. The remaining atoms give additional information, but it is not crucial for determining the general structural similarity within a pair of proteins.

The size of the Protein Data Bank (Berman *et al.,* 2000) is growing rapidly (doubling every 18 months). This amount of data needs fast yet accurate automatic algorithms to deal with the structural information. Algorithms should be fast enough to enable a structure–structure search and alignment over all proteins in the databases in real time. We provide here such a tool, which is fully automated, and can be used for fast pre-filtering of large structural databases of proteins. After such pre-processing a more detailed analysis of structures (including side chain information) and sequences of studied proteins can be performed.

Our algorithm assigns a score to a pair of proteins based on their structural similarity. This score is computed using a structural alignment of fragments of these proteins chosen as the most similar. The segments have a specified length (equal for all the proteins) to ensure the same scale of scores for various proteins. The similarity matrix used in the alignment is binary and is built by assigning 1 to those pairs of C$\alpha$ atoms from the two proteins chains that are closer than a given cut-off, and zero otherwise. The distances are measured after three dimensional superimposition of the segments. The gap penalties for the Smith-Watermann alignment are optimized using a specially prepared set of proteins (87 proteins). Information about the sequences of the query proteins is also important to speed up the process of computing the score (almost 20 fold). An added value in the method is its speed and accuracy (in comparison with existing algorithms).

The layout of the paper is as follows. First we describe the available servers which calculate structural similarity of proteins. In the next section we describe our method for fast and accurate structural alignment of proteins. Then we present a detailed description of our results on a test dataset (97 proteins from the SCOP database with various levels of similarity) in comparison with the LGscore2 method. In the next section we describe our Web server, which can be used for querying proteins in terms of structural similarity and a database of representative proteins. In the summary we sketch the perspectives for further development of the method.

## STRUCTURAL COMPARISON OF PROTEINS

We would like to start by describing shortly the existing internet servers which can be used in comparing structures of proteins. These methods provide search over protein databases (like PDB) and enable identification of statistically significant structural similarities. There are a number of classification schemes for protein structures available *via* the Internet. All of them use the same source of data (the Protein Data Bank; Berman *et al.*, 2000), but they differ in their basic procedures. This is because the methods are based on different assumptions about what constitutes significant similarities between proteins. The authors managed to establish human-based, semiautomatic or fully algorithmic methods with well-tuned statistical significance thresholds. The basic idea is, first, rapid identification of pair alignments of secondary structure elements, clustering them into groups, and scoring the best substructure alignment. The first two methods (SCOP and CATH) provide discrete, hierarchical classifications based on structural classes. The next method (VAST) is based on continuous distribution of domains in the fold space. FSSP/DALI provides two levels of description — a coarse-grained one and one with a fine-grained resolution. The last two methods (CE and LGscore2) are based on a different idea. They focus on the local geometry rather than global features such as orientation of secondary structures and overall topology (as in the case of VAST or DALI). The former algorithms mostly attempt to make a global optimization of the alignment path for some similarity measure. They use dynamic programming (CATH) Monte Carlo (FSSP/DALI) graph theory (Alexandrov & Fischer, 1996) three dimensional scanning of structures like in the package "3D-SCAN" of the WHAT-IF program (de Filippis *et al.*, 1994) or 3D clustering (Fischer *et al.*, 1992). Dynamic programming approaches solve the optimization task exactly, but are dependent on a target function, which focuses on specific parts of the protein molecule. The Monte Carlo and 3D clustering algorithms allow a better choice of target function, but they are very sensitive to the technical details of the numeric implementation. The search space for these algorithms may be extremely large and difficult to handle. In contrast the CE and LGscore2 algorithms are faster and more robust in finding an accurate 3-D structure alignment. Now we describe shortly the existing classical methods of structural comparison of proteins.

### SCOP — structural classification of proteins (Murzin *et al.*, 1995) provided by the MRC Laboratory of Molecular Biology and Centre for Protein Engineering on the internet site http://scop.berkeley.edu.

It covers augmented manual classification in four similarity categories: class, fold, superfamily and family. The emphasis is on defining functionally related superfamilies. It provides a detailed and comprehensive description of the structural and evolutionary relationships between protein structures. The database is constructed manually by visual inspection by a protein expert (A. Murzin). The comparison of structures is done with some automation and considers the evolutionary evidence provided by sequences structures and functions of the proteins. This database is frequently used as a valuable resource for comparative benchmarking.

**CATH — class, architecture, topology and homologous superfamily (Orengo *et al.*, 1994; 1997) is a hierarchical classification of protein domain structures provided by the University College London (http://www.biochem.ucl.ac.uk/bsm/cath/).**

It provides the complete PDB fold classification by domains and links to other sources of information. This hierarchical database of protein domain structures focuses on the definition of four architectural types. It includes class (C) — derived from secondary structure content; architecture (A) — which describes the gross orientation of the secondary structures; topology (T) — according to their topological connections and numbers of secondary structures; and homologous superfamily (H) — highly similar in terms of structure and function.

**VAST — vector alignment search tool (Gibrat *et al.*, 1996) is maintained by the National Center for Biotechnology Information (http://www.ncbi.nlm.nih.gov/Structure/VAST/vast.shtml).**

It performs all-on-all structure comparisons using the VAST algorithm. The output is a neighbors' list. It also contains the complete PDB representative structure comparison structure alignments and a structure superposition tool. The search space for alternative secondary structure elements depends on the length of proteins. The basic idea is to take into account the size of this search space. Then it calculates statistical *p*-value for the best substructure superposition in the same way as in the case of BLAST for sequence analysis. The VAST algorithm can be used in threading experiments because of its smaller RMSD differences between the query and the template (in spite of a shorter length of the alignments).

**FSSP — fold classification based on structure — structure alignment of proteins (Created by Liisa Holm and Chris Sander (Holm & Sander, 1994; 1998)) and maintained by the European Bioinformatics Institute (http://www.ebi.ac.uk/dali/fssp/).**

This classification is also based on an all-against-all automatically maintained and continuously updated comparison of structures in the Protein Data Bank using an automatic structure alignment program (called DALI). It continuously processes all new structures released by the PDB. The basic concept for this algorithm is a neighborhood in a fold space. The exhaustive all-against-all 3D-structure comparison of protein structures uses the DALI search engine. All chains and sequence homologs are divided into representative sets. Than an all-against-all structure comparison is performed on representatives of these subsets. This method contains the complete PDB database a fold tree definitions of domains lists of sequence neighbors and structure superposition. The output information provides also insight into the definition of structurally conserved cores and multiple alignments of distantly related protein families. The DALI alignment because of its longer extension at higher RMSD can be used in efficient homology modeling.

**CE — combinatorial extension of the optimal path (Shindyalov & Bourne, 1998) is maintained by the Research Collaboratory for Structural Bioinformatics (http://cl.sdsc.edu/ce.html).**

It is also based on the complete PDB and provides representative structure comparison, structure alignments, and a structure superposition tool. Combinatorial Extension (CE) determines an optimal alignment between aligned fragment pairs (AFPs — pairs of segments from the proteins being compared with high structural similarity). AFPs are determined from the local geometry averaged over eight C$\alpha$ positions. To prevent a combinatorial explosion the authors use some heuristics. Final alignments are done by dynamic programming. Combinations of AFPs that represent possible continuous alignment paths are selectively extended or discarded leading to a single optimal alignment. The algorithm is accurate in finding the optimal structure alignment and hence suitable for database scanning. CE provides a significant reduction of the search space and empirically establishes a reasonable target function. This function assumes that the alignment path is continuous when including gaps and that there is only one such optimal match.

**LGscore2 is a single program (Cristobal *et al.,* 2001) maintained now by Arne Elofsson (http://www.sbc.su.se/˜arne/lgscore/).**

LGscore2 is used to calculate the significance of the similarity between two structures after structural superposition. It detects the most "significant" non-continuous segments of a model. The similarity between two proteins is measured using the algorithm by Levitt and Gerstein with structural *p*-values as defined in (Levitt & Gerstein, 1998). After such a structural superposition the most significant subset is found giving the structural similarity score for a pair of proteins.

Our method presented here provides an interesting complement to the current structure comparison programs. Nevertheless, inferring from the structure prediction field, a consensus approach based on a combination of various structure similarity search procedures, would probably be more robust and sensitive. Our algorithm provides fast and continuous description of structural similarity between proteins. It can be further used for clustering of large protein databases using structure information, thereby giving opportunity to search for new structural domains and motifs. One can also impose on this pure- structural method some sequential information, gaining better benchmark results together with new insights in analyzing proteins from the twilight-zone of low sequence similarity.

## SEGMENT STRUCTURAL SUPERIMPOSITION

Our method is based on structural alignment of two proteins. We use the standard Smith-Watermann dynamic programming algorithm with a gap penalty. To describe the structural similarity of two proteins it is enough to compare only parts of their chains, which are called segments. They are chosen as the most similar substructures of C$\alpha$ chains with a fixed size (sequential 99 amino acids in the first iteration of our algorithm, or 199 residues in the second iteration).

The procedure of choosing the right pair of segments from two proteins chains is not straightforward, because of time constraints. First of all we must choose the central parts of the segments — "seeds", and then decide if these seeds are similar enough to proceed with further analysis of whole segments. By seeds we understand very short parts of the C$\alpha$ chains with the length of 13 amino acids. If the structural similarity of the two seeds from the pair of being compared proteins is high enough we start to analyze two longer continuous parts of the main chains centered on the seeds. The overall score for the comparison of a pair of structures is equal to the best score for the whole set of pairs of segments. In order to speed up our algorithm we make several preprocessing steps:

- The first reduction of the computing time of our algorithm is reached by requesting identity of the amino acids in the centers of the two seeds. We discard those seeds which have different amino acids in their centers. This condition speeds up the program about 20 times without loosing the accuracy.

- The second step is a comparison of the distance measured between the ends of one seed with the distance between the ends of the seed from the second protein. If the difference is larger than `ENDS = 3.0 Å` we stop analyzing the pair of segments centered on this pair of seeds. This speeds up our algorithm next few times.
- In the third step we compare the whole three dimensional structures of the seeds that have the same amino acids in the center and similar ends-distance. Here we make rotation and translations of both 13 aa chains in order to minimize the RMSD between them. If the resulting RMSD is small enough we carry on the analysis of large segments centered on this pair of seeds. We take `RMSD = 3.0 Å` as the cut-off value for this filter. This condition removes up to 25% of input cases.
- After analyzing seeds we end up with a rotation matrix and a translation vector for a Cartesian-space superimposition of the two seeds. In the following step we proceed with the structural analysis of the whole segments of 99 amino acids centered on the chosen pair of seeds. We rotate and translate these large segments based on the rotation matrix and translation vector from the minimization procedure of the RMSD between the two seeds. Then we define the similarity matrix `matrix1` for the dynamic programming algorithm in the following way. If two C$\alpha$ atoms taken from the superimposed segments are closer in space than `SEGRMSD1 = 5.0 Å` we assign 1 as their "structural distance", 0 is taken otherwise. We compute for the similarity matrix the number of non-zero values. If it is less than `SEGHITS = 35` we discard this case. This removes about 30% of incoming cases.
- Based on the similarity matrix `matrix1` we make global structural alignment `align1` between our pair of segments. If the score of the alignment is greater than `SEGHITS = 35` we pass this pair of segments on to the next filter. This removes most of the incoming cases (about 99%).
- In the end of this procedure we have a pair of promising segments. Now using a finer distance cut-off we compare the structures of these segments. Again we rotate and translate whole segments with a new more accurate rotation matrix and translation vector constructed using all previously aligned pairs of residues from the two protein segments. If the distance between two C$\alpha$ atoms from the two segments is smaller than `SEGRMSD2 = 3.0 Å` we assign 1 in the structural similarity matrix `matrix2`, and 0 otherwise. Using this similarity matrix we make a structural alignment with gaps `align2` between two 99 aa segments and store its score. The gap penalty for opening is equal to `paraGap = 1.0`, and the gap extension cost `paraExt = 0.1`. These values are computed during the optimization procedure on a representative set of proteins.

If one iteration of this procedure is performed with segments of 99 residues the resulting score for the pair of proteins is equal to the best structural score of `align2` for all possible pairs of similar segments which passed all previous filters. If any pair of segments has not passed all the filters, we assign 0 as the overall score. The overall score (best score of alignment of two segments) is always less than 99 (because of the maximal length of the segment), and larger or equal to 0.

In the case of the 3D-Hit internet server (see also short description in Plewczynski *et al.*, 2002) we perform also a second iteration of our procedure. We align longer parts of protein chains (199 residues) based on the rotation matrix and translation vector of superimposition of the shorter segments (99 residues). Then we calculate how many pairs of C$\alpha$ atoms are closer in space than `SEGRMSD1 = 5.0 Å`. If this number is larger then `SEGHITS2 = 70` we carry on with the procedure. In the same way as in the first iteration we make structural alignments `align1` (with `SEGHITS2 = 70` as the score cut-off) and `align2`. The best resulting score (for convenience divided by 2) for all compared pairs of longer segments is the overall score of comparison of the two protein structures.

## 3D-HIT SERVER

The Web server based on two iterations of our algorithm is available free for academic use on the Internet at http://3dhit.bioinfo.pl/. Now we describe the server's structure for a sample query. We start the search for a query protein given by the PDB file containing at least the C$\alpha$ atom coordinates. To improve the speed of the algorithm we make the initial clustering and hashing procedures. We use two databases of proteins structures — smaller for initial clustering of short fragments one can meet in proteins, and larger as the core structural database for searching. The clustering procedure is used to establish structural similarity classes of small protein fragments (seeds), and can be performed on the smaller representative database of protein structures. Then using those structural clusters we simplify the search for structural homologs of the query protein by the hashing procedure of the larger non-redundant subset of PDB database of template proteins.

◆ First of all we prepare structural clusters of short protein fragments (seeds of 13 amino acids). This clusterization procedure is based on a representative set of 1507 proteins taken from the PDB database. Each cluster represents a group of possible short seeds of C$\alpha$ protein chains, with the RMSD value for each pair of seeds in the cluster smaller than `RMSD = 3.0 Å`. This clustering procedure is made only once and stored for future use during each search over the whole large non-redundant subset of PDB database for the query protein. The smaller database, of proteins used in the clustering is different from server's main database. This procedure only finds possible structural architectures of short fragments in proteins. It is independent of the chosen small database, providing it is large enough to ensure proper general statistic. For test purposes we make two cluster databases: one for 1024 proteins, and other for 1507 proteins. The resulting clusters are basically the same for both sets but differently populated.

◆ In the hashing procedure we connect each cluster with a subset of seeds from all proteins taken form the large database of about 5000 proteins. The hashing procedure is made only once after each update of the main server's database, and stored for use in each similarity search. Each short seed from a protein connected to a given cluster has the RMSD smaller than `RMSD = 3.0 Å` with the mean cluster representative.

◆ The query protein itself is divided into short seeds, and each seed is compared with the cluster database. For each cluster with RMSD distance between the seed and mean cluster representative less than `RMSD = 3.0 Å` we make a search over all hashes of this cluster in the large database.

◆ For each pair of seeds (from the query protein and hash seed from the large database) we compute the RMSD. If it is less than `RMSD = 3.0 Å` and the amino acids in the center of both seeds are the same, we proceed with two iterations of the structural comparison algorithm. We take the pair of longer segments of 99 amino acids (or 199 in the second iteration) centered on these two seeds. One seed is taken from the query protein, the other from the template protein *via* the hash database. The method used in assigning a score for these two segments is explained in the previous section. The result of the structural comparison of these two segments is then stored in a table.

- After searching over all hashes from the large protein database, all clusters from the smaller database of clusters, and all seeds from the query protein, we assign a score to each pair (template and query) of proteins. The overall score for the structural comparison of each template protein from the large database and the query protein is equal to the largest score of comparison of all pairs of segments (from the template and the query protein). The global alignment between these two proteins is also computed and stored in a table. It is built straightforward by concatenating fragmentary structural alignments of all segments from this template protein and the query protein.
- The overall result of the search over the whole database for the query protein is the list of proteins taken from the large database, with the total score of comparison larger than `DBCUT = 40` for one iteration of our method. This list is then sent by e-mail to the user of the server.

The clustering procedure gives additional speed-up of processing of the whole database of 5000 protein chains. It takes only up to 1 minute to search over all template proteins for one query protein. This allows us to propose on-line service available on the Web (Plewczynski *et al.,* 2002). As the result of searching the whole database of known proteins we print a list of template proteins with the structural similarity to the query protein above a given cut-off (`DBCUT = 40` for the main score). The purpose of the similarity threshold is to distinguish between the similarity based on the evolutionary relationship, similarity by chance, or pure chemical similarity of some parts of protein chains. The result for each template and query protein pair is presented as a text field containing the following information:

- template protein name;
- the internal number of a hit with SCOP classification of the template protein;
- template protein length;
- main score for the comparison between the query and the template protein;
- amino acid identity between the two proteins (in %);
- global structural alignment for the query and template, with the start and end of the aligned parts of these proteins. Sign '–' indicate deletion, lowercase stands for insertion;
- some additional data like the number of C$\alpha$ atoms from the query protein close in space to the corresponding ones from the template protein within cut-off `SEGRMSD1 = 5.0` Å score of first alignment `align1` the same numbers for cut-off `SEGRMSD2 = 3.0` Å and second alignment `align2`.

It should be stressed that best results are obtained for proteins longer than the chosen segment size (99/199 amino acids). Smaller proteins have ill-defined score for segment comparison because of the shorter length of possible alignment. We have not addressed this problem by, e.g. rescaling raw similarity scores. On the other hand, for two large proteins similarity of segments can generate a lot of alternative competing structural alignments, because the search space for alternative alignments grows rapidly with the size of proteins. To sort through such alternatives one has to somehow prepare scoring schemes which rapidly identify the best structure alignment. In our case the global alignment is done simply by concatenating the local alignments, summing up the local score if some parts of these alignments overlap. It is, however, important, and left to the next publication, to prepare proper weighting of scores with the length of compared proteins, especially in the case of those shorter than 99/199 residues. The same problem is the major difference between the LGscore and LGscore2 algorithms. We hope that such rescaling of scores will improve the results of our method.

## DATABASE

An important part of our web server is the large database of template proteins, which provides implementation of our comparing algorithm for a wide audience over the Internet. For practical reason (time and computer resources limitations) we cannot take the whole PDB database. There is a considerable redundancy in the Protein Data Bank in terms of structure and sequence similarities. Numerous proteins in PDB represent the same structural or sequential family. Our aim is complete and economical use of this database. That is why in order to speed up the process of searching for close structural homologs we take only a specific subset of the whole PDB as the template database for our searching engine. We chose about 5 000 non-redundant proteins, which are collected from the PDB and clustered at a 90% identity level. This database is prepared by the PIECES server (http://www.fccc.edu/research/labs/dunbrack/pieces/) and contains a subset of sequences (with full PDB representation) culled from the entire PDB according to structure quality and maximum mutual sequence identity. The database is updated weekly to incorporate fresh structures appearing in the PD.

## RESULTS OF TESTS

The first test of our method is performed using one iteration structural similarity assignment on a test list of 97 proteins (including easy and hard targets). From this list we construct two sets of protein pairs. The first set consists of pairs of proteins which are similar to each other (according to the SCOP classification belonging to the same family). The second one is built from pairs that have a weaker similarity.

The overall result of this benchmark is equal to:

$$B = \sum_j \frac{1}{N_i},$$

where $N_i$ is the number of false-positives, i.e. hits from the second subset of the test list that have a score above the computed score of $j$ pair from the first subset. If $N_i = 0$ we take 1 as the value of the fraction. $B$ is approximately equal to the number of good hits (from the first subset) with score above the highest score for bad hits (from the second subset). The value of $B$ for the LGscore2 method is 123.66, and the result for one iteration of our method is 120.31. The difference (2%) is negligible so both algorithms give the same results on the test list.

A comparison between our method and the LGscore2 algorithm is presented in Fig. 1. Both methods have the same accuracy. The added value is almost 10 times faster execution in comparison with the much slower LGscore2. The whole database search (comparison with about 5 000 proteins) takes only less than 1 minute on a standard PC.

The second benchmark was conducted for the two-iteration version of our method using our Web server. The 3D-Hit server was coupled to the ToolShop (Rychlewski, 2001) structure prediction and evaluation program. The program compares protein structure prediction servers using the structure similarity server DALI (Holm & Sander, 1994; 1998) as a reference. That is why we can directly compare the DALI server with the 3D-Hit server.

The evaluated set included 100 query proteins (easy and hard targets). The numbers of correct models generated by the DALI and 3D-Hit servers were very similar as evaluated by all model assessment methods. The final alignment quality was better in the case of the DALI server in comparison with our method.

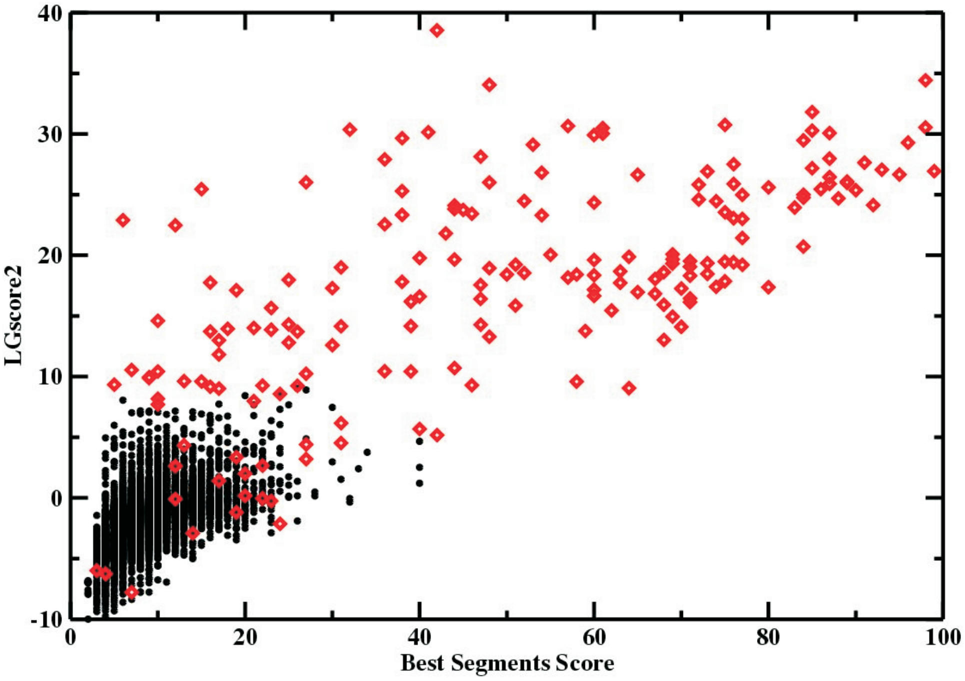

**Figure 1. Comparison between the one iteration version of our method and the LGscore2 structural alignment.**

The results are presented for a test set of 97 proteins. Diamonds represent scores for pairs of proteins that are similar in terms of SCOP classification index. Black circles represent scores for pairs of proteins from different subgroups of SCOP. Our method gives the overall result similar to LGscore2. Protein pairs that have no segments which are able to pass our filters (structural similarity score 0) are not shown.

In the case of distant structural cousins the results vary with the evaluation method used. 3D-Hit gets better ratings in all categories (especially in the specificity analysis) in the case of MaxSub (Siew *et al.,* 2000). Due to differences in the evaluation strategy, this picture changes in the case of the other two methods, Touch (Bujnicki *et al.,* 2001) and LGscore (Cristobal *et al.,* 2001). Touch evaluates similarities in contact space, which is more similar to the idea of the DALI server. LGscore focuses on closer structural similarity than 5 Å distance between pairs of aligned residues.

To conclude, the Web server based on our method is on average slightly less sensitive than the DALI server, nevertheless it represents an interesting complement to the currently available structure comparison programs (CE, Shindyalov & Bourne, 1998; VAST, Gibrat *et al.,* 1996). Inferring from the structure prediction field, a consensus approach based on a combination of various structure similarity search procedures would be probably more sensitive.

## SUMMARY

The main advantage of our algorithm is its speed and accuracy in comparison with previous methods. Its speed is the result of the pre-filtering procedure (introduction of short seeds) and simplicity of the method (we compare segments of protein backbones). The accuracy is tuned by choosing the size of segments (99 or 199 residues), and adjusting distance cut-offs and gap penalties using a small representative subset of proteins structures.

The Web server based on the proposed method performs a 3D similarity search in the database of known structures using the atomic coordinates of a 3D protein model as input. The size of the database is practically unlimited because of the speed of the algorithm. For practical reasons we use a representative set of about 5 000 proteins from PDB database. The complete search of the database takes about 1 minute on a standard PC (2 GHz) computer. It is possible to use this server as an on-line resource.

The method is useful also for analyzing similarity of proteins in a structural context, for defining structurally meaningful patterns or segments, and in general for studying protein evolution folding and design. Our clustering procedure can be useful for searching basic structural blocks of proteins (like I-sites protein motifs library, Bystroff & Baker, 1998; BLOCKs server by de Brevern *et al.,* 2000; or generalized secondary structure elements).

The main reason for developing structural classifications of proteins is to maximize the information return from experimental structure determination. If we know the structure of a query protein we are able to list all its structurally homologous proteins. That information provides an insight into the unknown function and role of the protein in the living organism. It is also very useful for evolutionary unification of protein families and analysis of folding principles. In molecular modeling it is important to compare sequences at functionally important sites, not whole proteins. That is why choosing only part of a protein for a detailed analysis, as in our method, seems to be the right choice giving proper accuracy and fast processing of the whole database of proteins.

Future developments of our method can cover also automatic search of functional evidence of plausible evolutionary relationships, based on structural and sequential similarity of proteins. The structural information linked to sequential data and associated functions, can be a rich source of biologically interesting observations. Strong structural similarity despite low overall sequence similarity is a hint of possible distant evolutionary relationship. These relations are undetectable using only sequential information, so those proteins can have unsuspected functional properties. These similarities define also the conserved structural core of protein families, which is a critical information in identifying distant homologs by fold recognition techniques. This direction of research is very promising because of great impact on the drug industry.

It is also worth to point out that the clustering procedure is alone a very interesting part of this research. In many applications it is useful to work with discrete classification. To provide such disjoint clustering one can use all-on-all structure comparison to derive interesting results. For example one can prepare a fold tree of proteins (dendrogram type). In that case one needs to use an average linkage procedure and hierarchical clustering. The basic structural motifs of our long segments are then defined by cutting the fold tree at a chosen cutoff to ensure that similar proteins share the same cluster.

## REFERENCES

Alexandrov NN, Fischer D. (1996) Analysis of topological and nontopological structural similarities in the PDB: new examples with old structures. *Proteins.*; **25**: 354–65.

Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. (2000) The Protein Data Bank. *Nucleic Acids Res.*; **28**: 235–42.

de Brevern AG, Etchebest C, Hazout S. (2000) Bayesian probabilistic approach for predicting backbone structures in terms of protein blocks. *Proteins.*; **41**: 271–87.

Bujnicki JM. (2000) Phylogeny of the restriction endonuclease-like superfamily inferred from comparison of protein structures. *J Mol Evol.*; **50**: 38–44.

Bujnicki JM, Elofsson A, Fischer D, Rychlewski L. (2001) LiveBench-2: Large-scale automated evaluation of protein structure prediction servers. *Proteins.*; **45**: 184–91.

Bystroff C, Baker D. (1998) Prediction of local structure in proteins using a library of sequence-structure motifs. *J Mol Biol.*; **281**: 565–77.

Chothia C, Lesk AM. (1986) The relation between the divergence of sequence and structure in proteins. *EMBO J.*; **5**: 823–6.

Cristobal S, Zemla A, Fischer D, Rychlewski L, Elofsson A. (2001) A study of quality measures for protein threading models. *BMC Bioinformatics.*; **2**: 5.

de Filippis V, Sander C, Vriend G. (1994) Predicting local structural changes that result from point mutations. *Protein Eng.*; **7**: 1203–8.

Fischer D, Bachar O, Nussinov R, Wolfson H. (1992) An efficient automated computer vision based technique for detection of three dimensional structural motifs in proteins. *J Biomol Struct Dyn.*; **9**: 769–89.

Gibrat JF, Madej T, Bryant SH. (1996) Surprising similarities in structure comparison. *Curr Opin Struct Biol.*; **6**: 377–85.

Holm L, Sander C. (1994) The FSSP database of structurally aligned protein fold families. *Nucleic Acids Res.*; **22**: 3600–9.

Holm L, Sander C. (1998) Touring protein fold space with Dali/FSSP. *Nucleic Acids Res.*; **26**: 316–9.

Johnson MS, Sutcliffe MJ, Blundell TL. (1990) Molecular anatomy: Phyletic relationships derived from three-dimensional structures of proteins. *J Mol Evol.*; **30**: 43–59.

Levitt M, Gerstein M. (1998) A unified statistical framework for sequence comparison and structure comparison. *Proc Natl Acad Sci U S A.*; **95**: 5913–20.

Murzin AG. (1994) New Protein Folds. *Curr Opin Struct Biol.*; **4**: 441–9.

Murzin AG, Brenner SE, Hubbard T, Chothia C. (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol.*; **247**: 536–40.

Orengo CA, Jones DT, Thornton JM. (1994) Protein superfamilies and domain super-folds. *Nature.*; **373**: 631–4.

Orengo CA, Michie AD, Jones S, Jones DT, Swindells MB, Thornton JM. (1997) CATH – a hierarchic classification of protein domain structures. *Structure.*; **5**: 1093–108.

Plewczynski D, Pas J, von Grotthuss M, Rychlewski L. (2002) 3D-Hit: fast structural comparison of proteins. *Appl Bioinformatics.*; **1**: 223–5.

Pointing CP, Russel RB. (1995) Swaposins: circular permutations within genes encoding saposin homologues. *Trends Biochem Sci.*; **20**: 179–80.

Rychlewski L. (2001) ToolShop: prerelease inspections for protein structure prediction servers. *Bioinformatics.*; **12**: 1240–1.

Shindyalov IN, Bourne PE. (1998) Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng.*; **11**: 739–47.

Siew N, Elofsson A, Rychlewski L, Fischer D. (2000) MaxSub: an automated measure for the assessment of protein structure prediction quality. *Bioinformatics.*; **16**: 776–85.