# Network-based Machine Learning Approach for Structural Domain Identification in Proteins

Anirudh Tiwari and Nita Parekh

February 3, 2021

# Network-based Machine Learning Approach for Structural Domain Identification in Proteins

Anirudh Tiwari
Center for Computational Natural Sciences and Bioinformatics
International Institute of Information Technology Hyderabad, India
anirudh@research.iiit.ac.in

Nita Parekh*
Center for Computational Natural Sciences and Bioinformatics
International Institute of Information Technology Hyderabad, India
nita@iiit.ac.in

***Abstract*— In the era of structural genomics, with a large number of protein structures becoming available, identification of domains is an important problem in protein function analysis as it forms the first step in protein classification. In the proposed network-based machine learning approach, NML-DIP, a combination of supervised (SVM) and unsupervised (k-means) machine learning techniques are used for domain identification in proteins. The algorithm proceeds by first representing protein structure as a protein contact network and using topological properties, viz., length, density, and interaction strength (that assesses inter- and intra-domain interactions) as feature vectors in the first SVM to distinguish between single and multi-domain proteins. A second SVM is used to identify number of domains in multi-domain proteins. Thus, it does not require a prior information of the number of domains. The domain boundaries are identified using k-means algorithm and confirmed with CATH annotation. Performance of the proposed algorithm is evaluated on four benchmark datasets and compared with four state-of-the-art domain identification methods. Its performance is comparable to other domain identification tools and works well even when the domains are non-contiguous. Available at: https://bit.ly/NML-DIP.***

***Keywords*— Structural domain identification in proteins, k-means, SVM, graph theory**

## I. Introduction

Proteins are comprised of domains, folds and motifs, which form its basic building blocks. Genetic recombinant techniques allow reorganization of domains, called domain shuffling, resulting in different combinations of domains in different proteins. This along with swapping and insertion of domains result in complex architectures that are responsible for new protein functions in evolution. Hence, efforts to understand protein evolution and its function have mainly focused on domains as these fold into a stable, semi-independent 3D structure and perform a unique function conserved over evolution. Protein domains are also very useful in analyzing mechanisms of protein folding and their stability and structural transformations in various conditions. Being the basic units of protein folding, function and evolution, identification of domains is the first step in understanding functional and structural aspects of proteins.

Although domain boundaries can be determined by visual inspection, there definitely exists a need for developing accurate methods for automatic domain identification with increasing numbers of solved protein structures. The problem of dividing a protein structure into domains is challenging due to the lack of an unambiguous definition of domain. The most common definition of domains based on structural aspects is that these are compact stable modules containing a hydrophobic core and fold independent of the rest of the protein while the evolutionary and functional aspects of the definition suggest that these can occur in different combinations and perform a specific function [1]. Deviations are observed in a number of proteins, such as a domain may be very small and may not contain a hydrophobic core, may occur as a large single structural unit, or two domains together may perform a specific function, instead of each one having its own unique function. This makes domain assignment computationally a difficult task. Non-contiguous domains (occurring because of domain insertion) further adds to the difficulty in developing an automated solution.

Though several methods have been proposed to predict domains they all have notable limitations. Some methods fail to correctly partition non-contiguous domains or are unable to distinguish between single and multi-domain proteins, while some require specifying the number of domains the protein must be split into. Most domain databases use more than one approach along with manual inspection for correct assignment of domains, for e.g., CATH [2] uses four domain assignment methods, namely, DETECTIVE [3], DOMAK [4], PUU [5] and the method by Islam *et al* [6]. If all the four methods are unanimous in their prediction, then the domains are automatically assigned, else a manual judgment is made about the best definition among the four. While classification of domains in CATH is based on structural integrity, SCOP [7] focuses on functional and evolutionary aspects, and as a result they differ in about 20% of domain assignments. With over 60% of proteins being single domain, and about 25% of multi-domain proteins being non-contiguous, there exists a need for reliable domain identification methods that can handle simple as well as complex domain architectures. An excellent review by Holland *et al* [1] provides comparison of various methods for domain assignment.

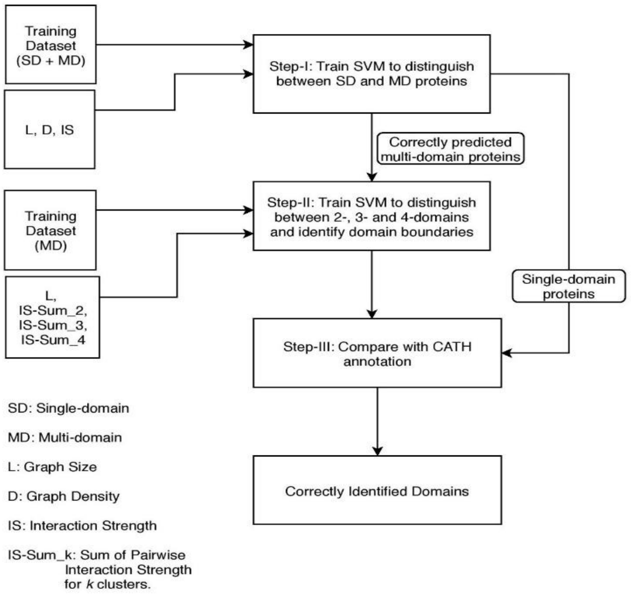
## II. Methodology

The flowchart depicting various steps of the proposed Network-based Machine Learning algorithm for Domain Identification in Proteins (NML-DIP) is shown in Figure 1. Here, protein structure is modeled as a Protein Contact Network (PCN), wherein each amino acid residue acts as a node and an edge is drawn between two residues if they are within 7Å [8]. Next, we employ a combination of supervised (SVM) and unsupervised (k-means) machine learning approaches using graph properties as feature vectors for the SVMs. Various steps in the algorithm are briefly discussed below.

**Step-1: Distinguishing single *vs* multi-domain proteins:** First step in domain classification problem is to identify whether a protein is single or multi-domain protein. For this, a support vector machine (SVM) is trained using network properties of PCN as feature vectors, defined below.

**Length of Protein:** It is defined as the number of nodes in a PCN. Here the underlying assumption is that the length of protein is expected to increase with increase in the number of domains, with notable exceptions.

**Graph Density:** It is defined as ratio of the number of edges, E, observed to the number of possible edges, ½ V(V-1), in a graph of size V, given by (1). Since domains are compact globular structures and a single domain protein are expected to be more compact compared to a multi-domain protein, it is a good measure in quantifying the compactness of a protein.



Training Dataset (SD + MD)
L, D, IS
Step-I: Train SVM to distinguish between SD and MD proteins
Correctly predicted multi-domain proteins
Training Dataset (MD)
Step-II: Train SVM to distinguish between 2-, 3- and 4-domains and identify domain boundaries
L, IS-Sum_2, IS-Sum_3, IS-Sum_4
Single-domain proteins
Step-III: Compare with CATH annotation
Correctly Identified Domains
SD: Single-domain
MD: Multi-domain
L: Graph Size
D: Graph Density
IS: Interaction Strength
IS-Sum_k: Sum of Pairwise Interaction Strength for k clusters.

Fig. 1: Workflow of the proposed algorithm, NML-DIP.

$$D = \frac{2|E|}{|V|(|V|-1)} \qquad (1)$$

**Interaction Strength:** It captures inter-domain interactions given by (2), by computing the number of inter- and intra-domain interactions obtained on splitting a protein chain/cluster into two clusters by k-means algorithm.

$$IS = \left(\frac{N_{xy}}{N_x+N_y}\right) \times 100 \qquad (2)$$

Here $N_{xy}$: number of inter-cluster interactions, $N_x$ and $N_y$: number of intra-cluster interactions in clusters *x* and *y*, respectively [9]. Two nodes are said to be interacting if the distance between them is ≤ 7Å.

**Step-2: Identifying Number of Domains, *k*:** In this step, a second SVM is used for classifying multi-domain proteins into 2-, 3- or 4-domains. Using k-means, protein chain is split into *k*-clusters (*k* = 2 - 4), and interaction strength (IS) is computed for every cluster-pair in each *k* split. Sum of pairwise Interaction Strengths, IS-Sum_k = $\Sigma_{i\neq j}$IS-$D_iD_j$, between clusters is computed for each *k*-split, *k* = 2, 3 and 4, and is expected to increase with increasing *k* for an incorrect split, where IS-$D_iD_j$ denotes the interaction strength between domains *i* and *j*. If a *k*-domain protein is split into *n* clusters, $n > k$ , IS-Sum_n is expected to be significantly higher than IS-Sum_k as in this case one of the true domains would be incorrectly split. To capture this information, the second SVM is trained on four features: Length of protein and three interaction strengths, IS-Sum_2, IS-Sum_3 and IS-Sum_4, corresponding to the three splits, *k* = 2, 3, and 4, respectively.

**Step-3: Comparison with CATH Annotation:** To assess the reliability of our predictions, number of domains, *k*, and the domain boundaries are compared with CATH annotation. True prediction is reported if fraction of correctly predicted residues is ≥ 75% compared to CATH annotation [10]. The algorithm also works if the user has prior information about the number of domains. In this case k-means algorithm is run on the given protein structure for user input *k*, and the program outputs the domain boundaries.

**Implementation Details**

Python scripts were written to parse the PDB files, construct protein contact network (PCN), compute feature vectors and apply SVM and k-means using Scikit-learn module (0.22.1). The algorithm works in two modes: the user may provide a file in PDB format (with or without the information about the chains) and the domain annotation (number of domains and domain boundaries) will be provided for all chains (or the specified chain). If the user is interested in identifying only the domain boundaries, then the user may provide a PDB file with chain ID and number of domains, k, as input. The algorithm is computationally very efficient and on an Intel(R) Core (TM) i7-8565U CPU@1.80GHz system with 8 GB RAM, it takes ~ 2 - 3 seconds to execute and identify domains in a protein of size ~ 450 residues.

## III. Results & Discussions

**Dataset Construction**

A brief description of training and test datasets used in this analysis is given below.

**Training dataset:** A dataset of 3000 proteins comprising 1500 chains each of single and 500 chains each of type 2-, 3- and 4-domain proteins (1500 multi-domain proteins), is constructed for training the two SVMs. Care was taken to include at least one representative domain defined by unique C, A and T categories in CATH [2], where C represents secondary structure class of the domain, A the Architecture and T the 3-dimensional Topology. Also, it was ascertained that for the selected protein chains, CATH and SCOP agreed on the domain assignments.

**Test datasets:** Performance of the proposed approach is evaluated on four test datasets, summarized in Table I, and briefly described below.

**Benchmark_3 dataset [10]:** It consists of proteins chains for which domain assignment in CATH [2] and SCOP [11] databases agree and each protein chain is a representative of unique topology group in CATH database. Since only half this dataset is made publicly available by the authors, it is small (132), with very few 3-and 4- domain proteins.

**ASTRAL SCOP dataset [11]:** It is a non-redundant dataset at sequence level and consists of proteins having sequence similarity < 30%. Though largest with 6290 protein chains, it is not truly non-redundant at topological level.

TABLE I. THREE TEST DATASETS USED FOR PERFORMANCE EVALUATION.

| Test Datasets | 1-d | 2-d | 3-d | 4-d | Total |
|---|---|---|---|---|---|
| **Benchmark_3** | 55 | 53 | 21 | 3 | 132 |
| **ASTRAL SCOP** | 4048 | 1599 | 489 | 154 | 6290 |
| **NR_Dataset** | 761 | 331 | 139 | 47 | 1278 |

**Non-redundant dataset (NR_dataset):** To address the issues of class imbalance and redundancy, we constructed a non-redundant dataset (NR_Dataset), similar to the training dataset, in accordance with the approach proposed by Holland *et al* [10]. First, protein chains for which domain assignment agreed between CATH and SCOP were selected, resulting in 88,986 chains. These chains were grouped into 1313 topology classes represented by unique Class, Architecture and Topology in CATH. From each topology group, a *k* domain protein (*k*=1-4) was randomly picked.

**Single *vs* Multi-domain Classification:** A step-by-step evaluation of our algorithm on the non-redundant NR_dataset is summarized in Table II. An overall prediction accuracy of ~87% is observed in classifying single *vs* multi-domain proteins. We observed that length of the protein is a very crucial parameter and majority of incorrectly classified single domain proteins are much larger compared to the average single domain length. For e.g., the misclassified single domain protein 1XIM (A) is of length 394, much larger than single-domain average (∼ 154 ). We observed that 11 (Benchmark_3), 183 (ASTRAL SCOP) and 82 (NR_Dataset) single-domain proteins wrongly classified as multi-domain proteins were larger in size. Similarly, we also observed that most wrongly predicted multi-domain proteins were smaller in size compared to the average (∼ 434). For e.g., a 2-domain protein 1YUA (chain B) is of length 122, much smaller than the multi-domain average. We observed 6 (Benchmark_3), 108 (ASTRAL SCOP) and 34 (NR_Dataset) misclassified as single-domain proteins because of their lengths smaller than average. Compactness is another important feature in domain identification and is captured by the graph properties, density and interaction strength. A number of single domain proteins wrongly classified in the four test datasets had lower Interaction Strength and/or lower density than the average value for single-domains. For e.g., apart from being a large protein, 1XIM (A) has lower density (= 0.039) and interaction strength (=3.973) values compared to the single-domain protein average, ∼ 0.08 and ∼ 11.6, respectively.

Additionally, to evaluate the performance of SVM on the three datasets, the metrics Precision, Recall, F1 Score and Matthews Correlation Coefficient (MCC) were evaluated and are summarized in Table III. It may be observed that these metrics are high and comparable across the three datasets. For the NR_Dataset these values are higher, especially the MCC value, clearly indicating the importance of a balanced and no-redundant dataset performance evaluation.

**Identifying Number of Domains:** In the second step of the algorithm, another SVM is used for *de novo* detection of number of domains in multi-domain proteins. It may be noted from Table II that prediction accuracy of the second SVM in correctly identifying the number of domains on NR_Dataset is ~ 85%, comparable to the first SVM. It is worth noting that the performance is equally good for non-contiguous as for contiguous proteins, while many algorithms fail to detect non-contiguous domains. Performance of the algorithm is good in the detection of 2-domain (~97%) and 4-domain (~81%) proteins. However, the prediction accuracy dropped to ~63% for 3-domain proteins with majority of them incorrectly labeled as 2-domain proteins.

Performance metrics of the second SVM are summarized in Table IV. As in the previous case, we again observe a better performance on NR_Dataset compared to the other three datasets, further reinforcing the importance of the dataset in testing. Low Recall values for 3-domain proteins indicate that a number of these are missed by the algorithm. We observe majority of these wrongly classified as 2-domain proteins, resulting in lower Precision values for 2-domain proteins. In contrast, the Precision value for 3-domain proteins is significantly high indicating low False Positives in this case. High MCC value for the NR_Dataset indicates the reliability of the predictions.

TABLE II: PERCENTAGE OF CORRECTLY CLASSIFIED PROTEINS SHOWN FOR VARIOUS STEPS OF THE PROPOSED ALGORITHM ON NR_DATASET. C, NC: CONTIGUOUS AND NON-CONTIGUOUS MULTI-DOMAIN PROTEINS.

| Protein | No. of Proteins | SVM-1 | Multi-Domain Proteins | | SVM-2 | Domain Boundary Prediction |
|---|---|---|---|---|---|---|
| | | | No. of Proteins | No. of Domains | | |
| Single | 761 | 92.77 | - | - | - | - |
| Multi | 517 | 78.92<br>(C: 76.4,<br>NC: 82.2) | 331 | 2 | 96.65<br>(C: 96.5,<br>NC: 96.8) | 87.01<br>(C: 87.05,<br>NC: 87) |
| | | | 139 | 3 | 63.11<br>(C: 62.9,<br>NC: 63.3) | 76.62<br>(C: 82,<br>NC: 75) |
| | | | 47 | 4 | 80.85<br>(C: 88.2,<br>NC: 76.7) | 86.84<br>(C: 86.7,<br>NC: 87) |
| Total | 1278 | 87.17 | 517 | - | 84.80<br>(C: 86.5,<br>NC: 82.7) | 84.68<br>(C: 86,<br>NC: 83) |

TABLE III: PERFORMANCE EVALUATION OF SVM IN DISTINGUISHING BETWEEN SINGLE- AND MULTI-DOMAIN DOMAINS ON FOUR TEST DATASETS.

| Benchmark_3 | Precision | Recall | F1 Score | MCC |
|---|---|---|---|---|
| Single-Domain | 0.88 | 0.80 | 0.84 | 0.65 |
| Multi-Domain | 0.75 | 0.85 | 0.80 | |
| Average | 0.83 | 0.82 | 0.83 | |

| ASTRAL SCOP | Precision | Recall | F1 Score | MCC |
|---|---|---|---|---|
| Single-Domain | 0.87 | 0.84 | 0.86 | 0.63 |
| Multi-Domain | 0.74 | 0.79 | 0.76 | |
| Average | 0.83 | 0.82 | 0.83 | |

| NR_Dataset | Precision | Recall | F1 Score | MCC |
|---|---|---|---|---|
| Single-Domain | 0.86 | 0.92 | 0.89 | 0.73 |
| Multi-Domain | 0.88 | 0.78 | 0.83 | |
| Average | 0.87 | 0.87 | 0.87 | |

TABLE IV: PERFORMANCE METRICS FOR SVM-II IN DETECTING THE NUMBER OF DOMAINS ON FOUR TEST DATASETS.

| Benchmark_3 | Precision | Recall | F1 Score | MCC |
|---|---|---|---|---|
| 2-domain | 0.81 | 1 | 0.89 | 0.64 |
| 3-domain | 0.91 | 0.55 | 0.68 | |
| 4-domain | 0.5 | 0.33 | 0.4 | |
| Average | 0.83 | 0.82 | 0.80 | |

| ASTRAL SCOP | Precision | Recall | F1 Score | MCC |
|---|---|---|---|---|
| 2-domain | 0.78 | 0.92 | 0.84 | 0.49 |
| 3-domain | 0.68 | 0.40 | 0.50 | |
| 4-domain | 0.76 | 0.63 | 0.69 | |
| Average | 0.75 | 0.77 | 0.75 | |

| NR_Dataset | Precision | Recall | F1 Score | MCC |
|---|---|---|---|---|
| 2-domain | 0.81 | 0.96 | 0.88 | 0.72 |
| 3-domain | 0.92 | 0.63 | 0.75 | |
| 4-domain | 0.92 | 0.80 | 0.86 | |
| Average | 0.86 | 0.85 | 0.84 | |

Below we discuss the possible reason why SVM failed in correctly classifying 3-domain proteins considering few representative examples. We observe a common pattern in majority of incorrectly classified 3-domain proteins that one of the domains is spatially distant from other two. In Figure 2 (IA), representative example of a 3-domain protein 1IRA (Y) is shown, wherein one of the domains, depicted in ‘red is spatially distant from the remaining two domains shown in ‘blue’ and ‘green’. In this case, the SVM classified it as a 2-domain protein, merging blue and green coloured domains (shown in green on the right panel, (IB)). Because of their spatial proximity, the number of interactions between blue and green domains is much higher compared to that between red and green domains or red and blue domains. This results in IS-Sum_2 values comparable to that of average 2-domain proteins and the SVM ends up classifying them as 2-domain proteins. Few other examples of similar type of 3-domain proteins that are wrongly classified as 2-domain are 3FFK (chain A), 1BI3 (chain A), 3C18 (chain A), etc.

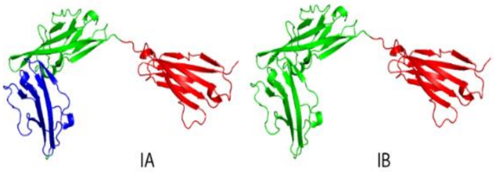


Fig. 3: Representative example of a 3-domain proteins in which one domain is spatially distant from other two in 1IRA (Y) is shown. On the left (labeled ‘A’) CATH annotation is depicted with three domains shown in ‘red’, ‘green’ and ‘blue’, while on the right (labeled ‘B’) the prediction from our algorithm is depicted with two domains is close proximity merged and represented in ‘green’ colour.

**Comparison of Domain Boundaries with CATH:** The proteins for which number of domains are correctly identified in the previous step, domain boundaries are extracted for the correct *k*-split and true prediction is reported if the fraction of correctly predicted residues is ≥ 75% compared to CATH annotation. The results are summarized in the last column of Table II. Thus, we see that a simple unsupervised algorithm such as k-means is able to capture very well the structural features of domains, resulting an overall prediction accuracy of ~ 85% on the NR_Dataset.

**Comparison with Other Domain Identification Methods**

Prediction accuracy of NML-DIP is compared with four state-of-the-art structure-based domain identification tools, namely, CA Algorithm [12], DDomain [13], DomainParser2 [14] and PDP [15] on two datasets: Benchmark_3 and ASTRAL SCOP, shown in Figure 5. It may be noted that the performance of NML-DIP is comparable to other four approaches and the performance is marginally better for the larger ASTRAL SCOP dataset. In Table V the overall performance of NML-DIP on the non-redundant NR_Dataset is given. We observe that the accuracy in detecting single domains on NR_Dataset is significantly higher and the overall prediction accuracy of 3-domain proteins is also improved on this dataset compared to the other two datasets.

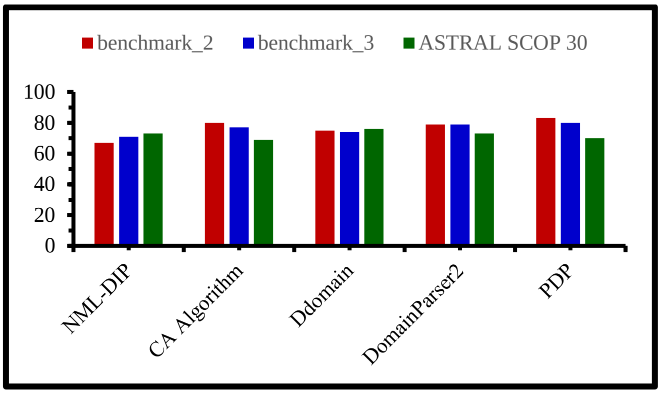


Fig. 5: Comparing overall domain prediction accuracy of NML-DIP with four state-of-the-art algorithms on three datasets. The results for the four algorithms is reproduced from Feldman [12].

TABLE V: PERFORMANCE OF NML-DIP ON NR_DATASET.

| | **1-d** | **2-d** | **3-d** | **4-d** | **Overall** |
|---|---|---|---|---|---|
| **NML-DIP** | 92.77% | 60.73% | 42.45% | 68.09% | 78.09% |

## IV. CONCLUSION

It may be noted that the domain identification problem is very similar to community detection in a social network in the sense that the number of domains or communities is not known a priori and also both exhibit large number of connections within a community/domain than between communities/domains), allowing us to borrow techniques from social network theory to address biological problems. Further, since numerous definitions have been proposed to define a domain, an approach not dependent on the domain knowledge is desirable. With this observation we proposed here a combination of graph theory (for feature selection) and machine learning approach for domain identification. We show that using graph properties as feature vectors in SVM algorithm provides a reliable approach for domain identification. The prediction accuracy of our algorithm on ASTRAL SCOP dataset is comparable with other tools. This suggests that a combination approach can really help in improving the sensitivity of domain detection algorithms.

## REFERENCES

[1] Veretnik S, Wodak S, Gu J, Identifying Structural Domains in Proteins. In *Structural Bioinformatics*, Gu J, Bourne PE, Eds. Second Edition, Wiley-Blackwell, 485–513 (2009).

[2] Orengo, CA, Michie AD, Jones S, Jones DT, Swindells MB, Thornton JM, CATH – a Hierarchic Classification of Protein Domain Structures. Structure, 5 (8): 1093–1109 (1997).

[3] Swindells MB, A Procedure for Detecting Structural Domains in Proteins. Protein Science 4 (1): 103–12 (1995).

[4] Siddiqui AS, Barton GJ, Continuous and Discontinuous Domains: An Algorithm for the Automatic Generation of Reliable Protein Domain Definitions. Protein Science, 4 (5): 872–84 (1995).

[5] Holm L, Sander C, Parser for Protein Folding Units. Proteins, 19 (3): 256–68 (1994).

[6] Islam SA, Luo J, Sternberg MJ, Identification and Analysis of Domains in Proteins. Protein Engineering, 8 (6): 513–26 (1995).

[7] Hubbard TJ, Ailey B, Brenner SE, Murzin AG, Chothia C, SCOP: A Structural Classification of Proteins Database. *Nucl. Acids Res.,* 27 (1): 254–56 (1999).

[8] Chakrabarty B, Parekh N, NAPS: Network Analysis of Protein Structures. *Nucl. Acids Res.*, 44: W375–W382 (2016).

[9] Yalamanchili HK, Parekh N, Graph Spectral Approach for Identifying Protein Domains. In Bioinformatics and Computational Biology, Rajasekaran S, Eds. Lecture Notes in Computer Science, vol 5462. Springer, Berlin, Heidelberg, 437–48 (2009).

[10] Holland TA, Veretnik S, Shindyalov IN, Bourne PE, Partitioning protein structures into domains: why is it so difficult? J Mol Biol, 361(3): 562–590 (2006).

[11] Fox NK, Brenner SE, Chandonia JM, SCOPe: Structural Classification of Proteins—Extended, Integrating SCOP and ASTRAL Data and Classification of New Structures. *Nucl. Acids Res.,* 42: D304–D309 (2014).

[12] Feldman HJ, Identifying Structural Domains of Proteins Using Clustering. BMC Bioinformatics, 13 (1): 286 (2012).

[13] Zhou H, Xue B, Zhou Y, DDOMAIN: Dividing Structures into Domains Using a Normalized Domain–Domain Interaction Profile. Protein Science, 16 (5): 947–55 (2007).

[14] Guo JT, Xu D, Kim D, Xu Y, Improving the Performance of DomainParser for Structural Domain Partition Using Neural Network. *Nucl. Acids Res.,* 31 (February): 944–52 (2003).

[15] Alexandrov N, Shindyalov I, PDP: Protein Domain Parser, Bioinformatics. 19 (3): 429–30 (2003).