Classification of Protein Structures Using Deep Learning Models

Georgina Mirceva, Andreja Naumoski and Andrea Kulakov
Faculty of computer science and engineering, Ss. Cyril and Methodius University in Skopje, Skopje, Macedonia
georgina.mirceva@finki.ukim.mk, andreja.naumoski@finki.ukim.mk, andrea.kulakov@finki.ukim.mk

***Abstract* - Protein molecules are very important in the organisms because they participate in different processes. Understanding the way they interact in the cells is of high importance. In order to understand that, solving the task of protein classification could be really helpful thus providing valuable knowledge about the similar proteins that belong to same class. In this paper we focus on solving the task of protein classification. First, we extract some features of the proteins thus obtaining feature vectors, and then by using deep learning architecture, we create prediction model that could be used for classifying protein structures. We present some experimental results of the obtained classification models.**

Keywords - protein structure; protein classification; feature extraction; deep learning

I. Introduction

Proteomics is a research area that studies the protein molecules, which are among the most important compounds of the humans' bodies. Namely, proteins are involved in many processes that occur in their cells. Proteins play very important roles, like enzymes in biochemical reactions, as well as transport of the oxygen to the cells. Some proteins have signaling role, for example insulin triggers the uptake of glucose from blood. They also have defensive role and play as antibodies that neutralize foreign molecules. Therefore, there is a high interest in the proteomics community to understand the structure of proteins molecules as well as the functions that they have in the processes in the organisms.

The development in the technology provides different techniques that could be used to determine the structure of the protein molecules. Using these techniques, the structure of proteins is determined, and they are later deposited in the Protein Data Bank (PDB) [1], [2], which is the main repositorium for this purpose. On Fig. 1, the growth of the number of structures that are deposited in PDB is presented. The determined data about protein structures is presented in pdb files, which contain data about the primary, secondary and tertiary structure of proteins. These data itself are not very useful, meaning that the most important thing regarding proteins is to discover the functions that they may have in the living organisms.

The literature provides various approaches for determination of proteins' functions. Some of the approaches are based on the premise that similar proteins, those that belong to same class, have the same functions. Due to this, solving the problem of protein classification is very important. Therefore, a plethora of methods have been introduced for classification of protein structures. Nevertheless, using them there is still large gap between the protein molecules that are determined and stored in PDB compared to the number of proteins whose functions have been discovered. This clearly shows that there is a great need for development of computational methods that would provide fast and accurate classification of protein structures.

SCOP (Structural Classification Of Proteins) [3] is one of the most important methods that is introduced for this purpose. It is considered as very accurate since the decisions in which class a given protein belongs is made manually by human experts by making visual inspections. Due to the same reason, it is not very fast, so the need for automatic or semi-automatics methods is obvious. CATH (Class, Architecture, Topology and Homologous superfamily) [4] is one of the most well-known methods from this category. Namely, CATH tries to classify the proteins automatically, and if the automatic classification is not suitable for some protein, then it is examined manually by human expert. In this way, using CATH the proteins are classified in semi-automatic manner.

Another group of methods perform alignment of protein sequences in order to classify the corresponding proteins. Needleman–Wunch [5], BLAST [6] and PSI-BLAST [7] are among the most well-known methods from this group. However, these methods would recognize as similar proteins whose sequences are similar but are not suitable for cases where the proteins have similar structure although their sequences are not similar. Namely, it is better to make alignment of protein structures instead of aligning their sequences, like CE [8], MAMMOTH [9] and DALI [10]. There are also methods that combine sequence and structure alignment [11], [12].

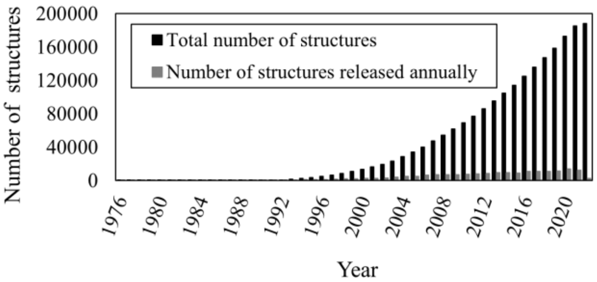

Figure 1. Number of structures deposited in the Protein Data Bank

However, the methods based on alignment, no matter whether they perform alignment of sequences or structures or maybe both, take too long to classify novel proteins. To overcome this, another group of methods make comparison of protein structures by making comparison of their features vectors that are first extracted and are later used for making prediction model. There are such methods that consider features of the sequences of the proteins [13] or their structures [14]. The extracted feature vectors contain the most relevant characteristics of the examined sequences or structures. This way, the prediction would not be based on the entire data, but only the most informative features are maintained. After feature extraction, the classification model could be created using various machine-learning algorithms. Due to the reduction of the data that are processed, in this way the creation of the classification model would be significantly faster, and moreover the testing phase would take less time to classify the novel proteins.

In this research, our aim is to provide method that would provide fast and accurate classification of protein structures. We put our focus on the last-mentioned group of methods, where feature vectors are extracted to represent the main characteristics of the protein structures. In our previous research studies, we have introduced various approaches for extraction of feature vectors of proteins, mainly focusing on their tertiary structure. In [15], comparative analysis is made where we compared the approaches that we have proposed that are suitable for making comparison between protein structures. These approaches could be used for protein retrieval, where the inspected protein is compared with the proteins stored in the database, and the most similar proteins are identified. These approaches could be further used for the task at hand, which is classifying protein structures in corresponding classes based on their feature vectors.

In this research paper, we introduce a novel method for solving the protein classification task. For that purpose, first, extraction of the feature vectors is made. To do that, we selected the protein ray-based descriptor as one of the most accurate, compact, and easiest to extract, according to the results that we obtained in [15]. The protein ray-based descriptor contains features about the geometrical characteristics of the protein structure, particularly it presents how the protein skeleton (backbone) is placed in the 3D space regarding the center of mass. After feature extraction, then using deep learning architecture, we generate prediction models. We make examination how the number of hidden layers, as well as the number of hidden nodes in these layers have influence on the accuracy of the classification models.

The remaining of this research paper is structured this way. First, in Section 2 we give detailed presentation of the method that is proposed in this study. We describe how the protein ray-based descriptor is extracted, and also we give description about the deep learning architecture that is used for creating models. In section 3, the experimental results that are obtained using different settings for the neural network are shown, and the influence of the different settings are discussed. Finally, section 4 concludes the paper and shows directions for further improvements.

II. THE PROPOSED METHOD

In this research paper, we introduce a novel method for classifying protein structures based on their tertiary structure. The proposed method has two steps. In the first step, feature vectors are extracted. For this purpose, we use our protein ray-based descriptor [15]. Then, in the second step, using deep learning architecture we create prediction model for classifying unknown protein structures. For creating classification models, we use a fully connected neural networks with 3 up to 5 hidden layers.

Fig. 2 shows the training phase and thus presents how the prediction model is created. Also, it shows the testing phase where a given query protein is classified in a corresponding class. The data about both training and testing proteins are stored in their corresponding PDB files that are deposited in the PDB database [2]. In the PDB files, data regarding the primary, secondary and tertiary structure is contained. In this research, we consider only data about the tertiary structure of proteins, meaning that the geometry of the protein is examined.

The training phase starts with extraction of the feature vectors for all proteins in the training set. The extracted protein ray-based descriptors are used as samples for training the classification model in the next stage. Then, the neural network model is trained by tuning the weights in the model. Once the classification model is created, then novel proteins could be classified.

The classification of a given query protein (protein from the test set) is made in the following manner. First, its feature vector is extracted using the same approach that is used for extraction of the feature vectors of the training proteins. Then, the extracted feature vector is presented as

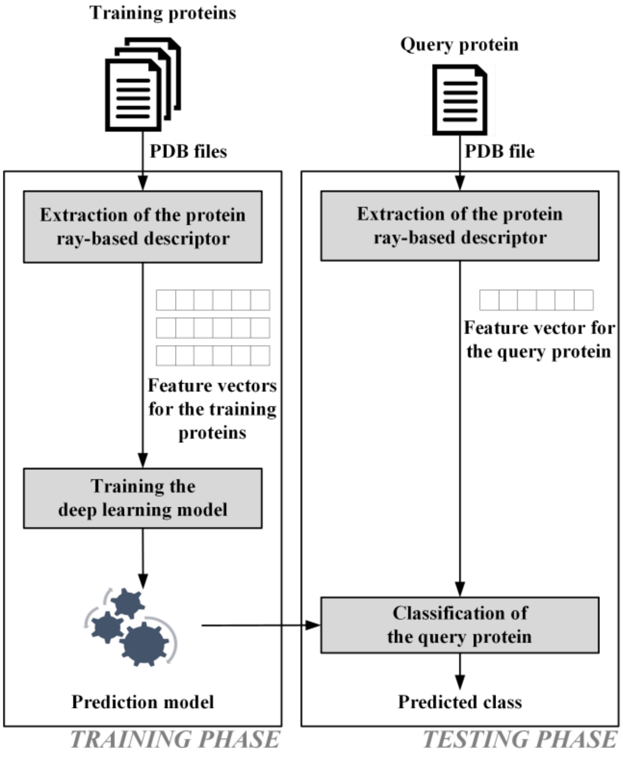
Training proteins
PDB files
Extraction of the protein ray-based descriptor
Feature vectors for the training proteins
Training the deep learning model
Prediction model
TRAINING PHASE
Query protein
PDB file
Extraction of the protein ray-based descriptor
Feature vector for the query protein
Classification of the query protein
Predicted class
TESTING PHASE

Figure 2. Training and testing phases of the proposed method

input in the neural network models in order to obtain the corresponding output for that input vector. The output corresponds to the decision of the class in which the query protein is classified. In this study, the classes in which the proteins are classified correspond to the domains in the SCOP hierarchy. For that, we consider part of the knowledge from the SCOP database that contains data how proteins are classified based on the SCOP method. Using SCOP, the classification is made in hierarchical manner, and in this research, we selected to consider only the domain level that could be considered as the most important level in the hierarchy in order to make distinction between the proteins based on the functions that they have. In this way, each query protein from the test set is classified in a given SCOP domain in which it is likely that it belongs to.

A. *Extraction of the Protein Ray-based Descriptor*

Protein molecules are commonly composed of several protein chains, which are folded in some particular way in the 3D space. Since we use the SCOP database [3] as a standard of truth, which contains data about the SCOP domains in which the protein chains belong, that means that the protein chains are considered as samples in the dataset used in this research. The protein chains are further composed of amino acid residues that are connected and form the protein skeleton (backbone). The amino acid residues in a given chain are folded in a particular way thus forming the secondary and tertiary structure of the corresponding protein chain. The amino acid residues in each chain are further composed of several atoms positioned in a particular manner in the 3D space based on the type of the amino acid. The C_α atoms of the amino acids connect two consecutive amino acid residues, thus forming the protein skeleton.

In [15], we presented several approaches for protein structure retrieval. Some of the approaches considered all atoms of the amino acid residues, while some of the approaches took into account only the C_α atoms. The results given in [15] show that it is more appropriate to consider only the C_α atoms, while by considering the remaining atoms, the accuracy declines. Using the protein ray-based descriptor, only the coordinates in 3D space of the C_α atoms are taken into account, thus in this way the exact position of the protein skeleton is known. Using these data, we obtain a 3D model for the protein, namely the protein skeleton is a 3D object that is placed in the 3D space.

This 3D model is scaled, thus the Euclidean distance between the most distant C_α atom with respect to the center of mass is equal to 1. In this way, scale invariance of the feature vectors is provided. As it is known in 3D object retrieval research area, besides scale invariance, it is also important to provide invariance to translation as well as rotation, meaning that if a given protein chain is translated in the coordinate system or rotated for some angle, the same feature vector should be extracted as in the case when these transformations are not performed. However, the way how the protein ray-descriptor is extracted, delivers invariance to translation and rotation.

Protein chains have different number of C_α atoms, so another challenge that should be solved is how to represent the protein chains with feature vectors with a same length. To solve this, the protein skeleton is interpolated using a predefined number of interpolation points, thus for each protein chain the same number of interpolation points would be used no matter how many C_α atoms they have.

In our previous study [15], we used protein ray-based descriptors using two approaches for interpolation of the protein skeleton. One approach was to interpolate the skeleton uniformly using interpolation points that are equidistant along the skeleton, while the other approach was to use more interpolation point in the parts of the skeleton where the consecutive C_α atoms are more distant. The results presented in [15] clearly showed that with uniform interpolation the extracted feature vectors contain more relevant features, thus in this research paper we decided to apply uniform interpolation of the protein skeleton.

The uniform interpolation of the protein skeleton starts with calculating the length of the skeleton using

$$L = \sum_{i=1}^{N_\alpha - 1} d_{Euclidean}(i, i+1), \qquad (1)$$

where $d_{Euclidean}(i,i+1)$ denotes the Euclidean distance between i-th and $(i+1)$-th C_α atoms, while N_α expresses how many C_α atoms exist in the inspected protein chain.

In [15], we made analysis in order to determine the optimal number of interpolation points that should be used. The results in [15] showed that it is most appropriate to use 64 interpolation points. Namely, by increasing the number of interpolation points above this value, there is not significant increase in the retrieval's accuracy. On the other hand, using lower number of interpolation points lead to significantly lower results. Therefore, in this research paper the number of interpolation points is set to N=64. The interpolation points are determined such that they are equidistant along the curve of the protein skeleton. The distance between two consecutive interpolation points equals $L/(N\text{-}1)$, where L is the length of the skeleton that was calculated using Eq. (1).

Once the interpolation points are determined, next the extraction of the feature vectors follows. As the name of the feature vector indicates, it is inspired from the ray descriptor [16] that is originally introduced for comparing 3D objects. It is called ray descriptors because rays are "emitted" from the center of mass towards the representative points of the object (the interpolation points in our case), and the features are extracted by calculating the Euclidean distance between the representative points and the center of mass. With this kind of feature vector, the invariance to translation and rotation is provided. The obtained protein ray-based descriptor for a given protein chain shows how the skeleton of that chain is placed in the 3D space with respect to the center of mass.

B. Deep Learning Models

In this research paper we created classification models using deep learning. A fully connected neural network was used by using 3, 4 or 5 hidden layers. Because we are using fully connected neural network, that means that the hidden layers are dense layers where each neuron is connected with the neurons from the previous and next layer.

In the input layer, we have 64 neurons, which is the dimension of the features vectors. In the hidden layers, ReLU activation function was used for each neuron. The output layer contains 150 neurons, so each neuron corresponds to one of the possible classes (SCOP domains in this case). In the output layer, softmax activation function is used. Stochastic gradient descent (SGD) was used as an optimization algorithm in order to optimize the objective function. SGD was also used as bias updater. As an updater, we used the Adam optimizer. The learning rate was set to 0.001.

We made experiments using 3, 4 and 5 hidden layers. Also, we created models by using 10, 20, 30, 40, 50, 100, 150, 200, 250 and 300 neurons in each of the hidden layers. Most of the models were trained for 10 epochs, but for the best setting we also tried using 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90 and 100 epochs, in order to find out whether the models are underfitted or overfitted.

We used the implementation provided in the WekaDeeplearning4j package [17] for the Weka software, which is based on the Deeplearning4j Java library. The remaining parameters were set to the default values as defined in the WekaDeeplearning4j package.

III. Results and Discusion

For evaluation of the proposed method, we used data from the PDB files of the proteins that are considered in the dataset. Namely, we used the data regarding the 3D coordinates of the C_α atoms of the protein chains that are used for training and testing purposes, using the knowledge from the PDB repositorium [2]. Regarding class labels, the SCOP database [3] was used as a standard of truth. As stated before, in this research paper we focus on the domain level in the SCOP hierarchy, thus the domains of the examined proteins correspond to the classes in which the proteins would be classified.

In total, 6145 protein chains are used, which are uniformly distributed in 150 SCOP domains. These proteins chains correspond to the samples used in the dataset, while these 150 SCOP domains represent the possible classes. Next, we divide this dataset into training and testing set in a ratio 90:10. In this way 5531 protein chains are used for forming the training set and the remaining 614 protein chains are used for testing purposes in the evaluation. In the division of the dataset into training and testing set, the uniform distribution is preserved.

Next, we will present the results that are obtained for the classification accuracy. Since the test set is balanced, this evaluation measure is an appropriate measure that would fairly present the predictive power of the models.

First, we made experiments by using neural network models with 3, 4 and 5 hidden layers, each layer containing 10, 20, 30, 40, 50 or 100 neurons. In these experiments, the models were trained for 10 epochs. The results from these experiments are presented in Table 1. From the results, we can note that in general when using 3 hidden layers the best results are obtained. For most of the cases, the inclusion of the additional hidden layers lead to decrease in the classification accuracy. In Table 2, we also give evidence about the training and testing times (expressed in seconds) from the same experiments, which are needed to train the models and make predictions for the entire test set. From these results it is evident that as the model is more complex by adding additional hidden layers or additional neurons in each hidden layer, the time needed to train the model rises. Similar conclusion can be made also for the testing time. Based on these experiments, we selected that it is more appropriate to use 3 hidden layers, thus leading to simpler models, while still having higher classification accuracy compared to the cases where 4 and 5 hidden layers are used. So, in the remaining experiments we used 3 hidden layers.

We made additional experiments in order to determine the optimal number of neurons per hidden layer. In Table 3, we present the results that are obtained for the classification accuracy using 100, 150, 200, 250 and 300 neurons in each hidden layer. This table also shows the training and testing times from these experiments. From the results given in this table, we can note that the models that contain more than 100 neurons in each hidden layer have lower accuracy than the model with 100 neurons in each hidden layer. Besides that, also as more neurons are added in each layer, the training of the models takes longer because the neural network is more complex. Since the models are more complex, also the testing time rises.

TABLE I. Classification Accuracy Obtained Using Different Number of Hidden Layers and Different Number of Neurons in the Hidden Layers

Number of neurons in each hidden layer	**Number of hidden layers**		
	3	**4**	**5**
10	**88.76**	83.22	79.64
20	**95.77**	90.23	91.86
30	**96.25**	96.09	93.65
40	94.46	93.65	**94.63**
50	94.95	95.60	**95.77**
100	**97.07**	95.44	95.11

TABLE II. TRAINING AND TESTING TIME IN SECONDS FOR THE MODELS OBTAINED USING DIFFERENT NUMBER OF HIDDEN LAYERS AND DIFFERENT NUMBER OF NEURONS IN THE HIDDEN LAYERS

Training time			
Number of neurons in each hidden layer	**Number of hidden layers**		
	3	**4**	**5**
10	**57.98**	83.77	95.91
20	**71.57**	97.56	106.71
30	**72.57**	92.12	109.61
40	**70.39**	90.54	102.43
50	**73.69**	96.27	105.42
100	**74.17**	86.16	106.42
Testing time			
Number of neurons in each hidden layer	**Number of hidden layers**		
	3	**4**	**5**
10	**0.23**	0.32	0.32
20	0.36	**0.24**	0.38
30	**0.31**	0.47	0.32
40	**0.27**	0.28	0.34
50	0.31	**0.30**	0.34
100	**0.23**	0.37	0.26

In the previous experiments, we trained the model for 10 epochs. We also made experiment using different number of epochs. In Table 4, the results obtained by training the model for 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 30, 40, 50, 60, 70, 80, 90 and 100 epochs are shown. This table shows results regarding the classification accuracy of the obtained models, as well as the training and testing times. In these experiments we used 3 hidden layers, each containing 100 neurons, which showed as best setting according to the results from the previous tables. The results shown in Table 4 show that when the number of epochs is increased from 1 up to 10, there is a significant increase in the classification accuracy. Then, by increasing the number of epochs from 10 up to 80, the accuracy slightly increases, and then by increasing the accuracy over 80 there is a slight decrease. The highest accuracy is obtained when training the model for 80 epochs. Regarding the training time, as the number of epochs increases, also the training time increases almost linearly. On the other hand, there is no significant pattern that could be determined regarding testing time, the differences are due to differences in the current CPU usage. This is due to the fact that all these models have the same complexity, thus the testing time is comparable.

TABLE III. THE RESULTS OBTAINED USING 3 HIDDEN LAYERS AND DIFFERENT NUMBER OF NEURONS IN THE HIDDEN LAYERS

Number of neurons in each hidden layer	**Classification accuracy**	**Training time**	**Testing time**
100	**97.07**	**74.17**	**0.23**
150	94.30	89.71	0.44
200	95.28	100.21	0.28
250	94.63	161.36	0.52
300	95.60	245.81	0.38

TABLE IV. THE RESULTS OBTAINED USING 3 HIDDEN LAYERS, 100 NEURONS IN THE HIDDEN LAYERS AND DIFFERENT NUMBER OF EPOCHS

Number of epochs	**Classification accuracy**	**Training time**	**Testing time**
1	93.49	7.12	0.32
2	92.67	13.62	0.27
3	95.77	19.9	0.36
4	94.14	28.55	0.23
5	95.60	35.98	0.21
6	95.28	44.83	0.31
7	95.11	54.12	0.26
8	96.58	69.99	0.18
9	97.07	68.88	0.24
10	97.07	75.68	0.27
15	96.09	123.36	0.24
20	97.39	150.48	0.17
30	97.23	231.68	0.36
40	97.23	322.53	0.31
50	97.23	407.82	0.24
60	97.23	548.59	0.26
70	97.56	627.54	0.25
80	**97.72**	724.42	0.29
90	97.56	810.94	0.22
100	96.91	860.72	0.19

In our previous research [15], we made an analysis where several approaches for comparing protein structures were compared with several well-known methods (DALI and CE). The results made in [15] showed that even the protein ray-based descriptor is fast and simple, it provides an accurate prediction of the similar proteins, while still been comparable with the state-of-the-art methods that are time consuming.

We also made analysis to compare these models with the models that we created in [18], where different classification methods were used in combination with the protein ray-based descriptor to classify protein structures. The results are shown in Table 5. From the results we can note that the best model obtained in this research outperforms most of the models that are used in this comparison, except knn. However, the testing time with knn is higher. If this analysis is made on larger dataset, it is expected that the difference in testing time with knn would be much higher, since knn do not create models but make comparison of the query with all training samples.

TABLE V. THE RESULTS OBTAINED USING DIFFERENT CLASSIFICATION MODELS

Classification model	Classification accuracy	Training time	Testing time
This research	97.720	724.42	0.29
C4.5	92.997	0.34	0.01
Naïve Bayes	94.625	0.06	0.74
Bayes Net	96.417	0.51	0.20
knn	98.534	0	0.42
SVM	97.557	17.35	2.08

IV. CONCLUSION

In this paper we proposed a novel method for classifying protein structures based on their geometrical features. The method contains two steps. In the first step, the protein ray-based descriptors are extracted for each training protein chain. Then, in the second step using deep learning architecture we create prediction models for solving the task at hand. In this study, we used a fully connected neural network models with 3 up to 5 hidden layers. The evaluation of the proposed method is done using knowledge from the SCOP database. Different experiments were performed and the classification accuracy, as well as training and testing times were examined.

First, the influence of the number of hidden layers was examined, and the results showed that the simpler models with fewer layers attain better accuracy. Also, the influence of the number of neurons in each hidden layer over the classification accuracy was analyzed. The results showed that it is most suitable to use hidden layers with 100 neurons in each layer. Finally, we made experiments by training the network using different number of epochs. The results showed that as the number of epochs is increased from 1 up to 10, the accuracy increases, then using higher number of epochs there is less significant increase. When the number of epochs is more than 80, the classification accuracy starts to decrease because the neural network is overfitted. The highest accuracy was obtained when training the model for 80 epochs. We also examined the time needed for training and testing the models, and the results showed that as the models are getting more complex, training and testing times increase.

As future improvements, we will mention several possible directions. Besides the protein ray-based descriptor, also some other feature vectors could be used. Regarding the neural network model, also other settings for the parameters could be examined, including other activation functions and optimization algorithms. Besides fully connected neural network, also some other deep learning architectures could be used. We also plan to use deep learning architectures based on fuzzy logic. Finally, also some other classification algorithms from machine learning could be used besides the algorithms for training deep learning models, including algorithms based on classical sets as well as fuzzy sets.

ACKNOWLEDGMENT

This work was partially financed by the Faculty of computer science and engineering at the “Ss. Cyril and Methodius University in Skopje”, Skopje, Macedonia.

REFERENCES

[1] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, “The protein data bank,” Nucleic Acids Res., vol. 28, no. 1, pp. 235–242, January 2000.

[2] RCSB Protein Data Bank, http://www.rcsb.org, 2019.

[3] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, “Scop: a structural classification of proteins database for the investigation of sequences and structures,” J. Mol. Biol., vol. 247, no. 4, pp. 536–540, April 1995.

[4] C. A. Orengo, A. D. Michie, D. T. Jones, M. B. Swindells, and J. M. Thornton, “CATH – a hierarchic classification of protein domain structures,” Structure, vol. 5, no. 8, pp. 1093–1108, August 1997.

[5] S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” J. Mol. Biol., vol. 48, no. 3, pp. 443–453, March 1970.

[6] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, “Basic local alignment search tool,” J. Mol. Biol., vol. 215, no. 3, pp. 403–410, October 1990.

[7] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, “Gapped BLAST and PSI-BLAST: a new generation of protein database search programs,” Nucleic Acids Res., vol. 25, no. 17, pp. 3389–3402, September 1997.

[8] I. N. Shindyalov and P. E. Bourne, “Protein structure alignment by incremental combinatorial extension (CE) of the optimal path,” Protein Eng., vol. 11, no. 9, pp. 739–747, September 1998.

[9] A. R. Ortiz, C. E. Strauss, and O. Olmea, “Mammoth: an automated method for model comparison,” Protein Sci., vol. 11, no. 11, pp. 2606–2621, November 2002.

[10] L. Holm and C. Sander, “Protein structure comparison by alignment of distance matrices,” J. Mol. Biol., vol. 233, no. 1, pp. 123–138, September 1993.

[11] S. Cheek, Y. Qi, S. S. Krishna, L. N. Kinch, and N. V. Grishin, “SCOPmap: automated assignment of protein structures to evolutionary superfamilies,” BMC Bioinformatics, vol. 5, pp. 197–221, December 2004.

[12] C. H. Tung and J. M. Yang, “fastSCOP: a fast web server for recognizing protein structural domains and SCOP superfamilies,” Nucleic Acids Res., vol. 35, W438–W443, July 2007.

[13] K. Marsolo, S. Parthasarathy, and C. Ding, “A multi-level approach to SCOP fold recognition,” IEEE Symposium on Bioinformatics and Bioeng., pp. 57–64, October 2005.

[14] P. H. Chi, Efficient protein tertiary structure retrievals and classifications using content based comparison algorithms, PhD thesis, University of Missouri-Columbia, 2007.

[15] G. Mirceva, I. Cingovska, Z. Dimov, and D. Davcev, “Efficient approaches for retrieving protein tertiary structures,” IEEE/ACM Trans. Comput. Biol. Bioinform., vol. 9, no. 4, pp. 1166–1179, July/August 2012.

[16] D. V. Vranic, 3D Model Retrieval, Ph.D. Thesis, University of Leipzig, 2004.

[17] S. Lang, F. Bravo-Marquez, C. Beckham, M. Hall, and E. Frank, “WekaDeeplearning4j: a Deep Learning Package for Weka based on DeepLearning4j,” Knowledge-Based Systems, vol. 178, no. 15, pp. 48-50, August 2019.

[18] G. Mirceva and A. Kulakov, “Protein classification by using four approaches for extraction of the protein ray-based descriptor,” CIIT 2020, Macedonia, 2020.