Secondary Structure Classification of Low-homology Proteins with Graph Neural Networks

Michaela Areti Zervou[1,2], Effrosyni Doutsi[1], Panagiotis Tsakalides[1,2]
[1]*Institute of Computer Science, Foundation for Research and Technology-Hellas, Heraklion, Greece*
[2]*Department of Computer Science, University of Crete, Heraklion, Greece*
E-mails: zervou@ics.forth.gr, edoutsi@ics.forth.gr, tsakalid@ics.forth.gr

***Abstract*—Acquiring knowledge of the dynamics and properties that force each protein into a unique secondary structure is highly important in medicine and biotechnology. The majority of existing prediction methods rely on complicated processes to extract numerous representative features that are later paired with state-of-the-art classifiers. A recent study proposes a recurrent neural network architecture demonstrating for the first time the potential of deep learning architectures for secondary structure classification of low-homology proteins. According to the latest state-of-the-art, proteins can be successfully described as graphs for the task of classification providing substantial results while at the same time addressing serious challenges including dealing with datasets with different sample sizes, among others. Thus, this study employs an efficient graph representation of low-homology proteins along with graph neural networks which have exhibited cutting-edge performance in a variety of graph-based applications, including protein sequence classification. Overall, this work proposes a novel, simple and fast architecture based on graph neural networks, with a significantly reduced representative feature set, when compared to the literature, that derives directly through the sequence-to-graph procedure. Experimental evaluation on real protein datasets demonstrates the superiority of our proposed architecture, with respect to accuracy and overall complexity when compared against the state-of-the-art.**

***Index Terms*—Secondary structure classification, Graph neural networks, Multidimensional Horizontal visibility graph, Nonlinear time series analysis**

I. Introduction

The prediction of protein structural classes from amino acid sequences is a challenging problem as it is profitable for determining protein function and regulation. In essence, the biological function of a protein is associated with its structure, which is specified by its amino acid sequence via the process of protein folding. Along these lines, the protein structure can be classified into four structural categories based on the protein's folding patterns known as (i) all α-fold, (ii) all β-fold, (iii) the $\alpha + \beta$-fold, and the (iv) α/β-fold [1]. Developing, however, a computationally efficient yet reliable secondary structure prediction scheme remains a pressing issue, especially for sequences for which not enough homologous information is provided from current protein sequence databases. To this end, a plethora of machine learning based algorithms has been developed for the structural class prediction of low-homology proteins based on a wide range of protein sequence representations [2]–[6]. The aforementioned approaches provide high-precision results by combining manually-tailored features with cutting-edge classifiers such as Support Vector Machines (SVM), Random Forests (RF), and Fisher's Linear Discriminant Analysis (FDA), among others. Recently, Panda *et al.* proposed in [7] a deep neural network approach demonstrating for the first time the potential of deep learning architectures in the task of low-homology protein structural classification, particularly for small datasets. Nevertheless, the performance of their work, as in aforementioned studies, comes at the expense of utilizing complex procedures for meaningful feature extraction for the representation of the proteins. Specifically, in order to provide a representative feature set for each protein the authors employed three independent procedures, namely, (i) a SkipGram based sequence representation where each amino acid sequence is converted to a series of trigrams [7], (ii) an Atchley's factors II, IV, V representation that derives from a set of over 50 amino acid properties by dimensionality reduction to identify clusters of amino acid properties that co-vary [8] and (iii) an Electron–Ion-Interaction-Potential representation that represents the electron interaction potential with an ionic subsystem of a crystal quadratic in ionic displacement [9]. All three representations result in a 18-dimensional feature vector for each protein sequence that is later is evaluated by a deep Gated Recurrent neural network [10].

Along these lines, the herein work proposes a novel and efficient protein structure prediction architecture based on graph neural networks, with a significantly reduced feature set multitude that derives directly from the sequence-to-graph pipeline without further feature extraction techniques needed. In particular, as depicted in Figure 1, in this pipeline every amino acid in the protein sequence is initially predicted as one of the three secondary structural elements, namely H (helix), E (strand) and C (coil) using the PSI-PRED tool [11]. Then, by employing the Chaos Game Representation (CGR) [12] technique, the updated 3-state sequence is converted into a two-dimensional time series. In brief, as depicted in Figure 1,

This work was funded by the Hellenic Foundation for Research and Innovation (HFRI) and the General Secretariat for Research and Technology (GSRT) under grant agreement No. 1725 (V4-ICARUS) and HFRI PhD Fellowship grant no. 5647.

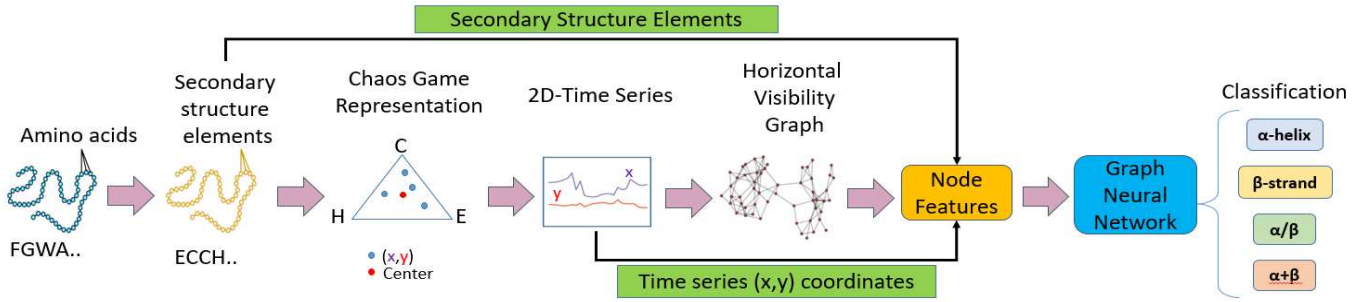

Fig. 1: General proposed sequence-to-graph architecture for protein secondary structure classification.

the concept of CGR is defined in a triangle where the three vertices correspond to the 3 secondary structural element letters H, E and C. Then, the first (x, y) point of the plot is placed half way between the center of the triangle and the vertex corresponding to the first letter of the 3-state sequence, whereas, the $i-th$ point of the plot is then placed half way between the $(i-1)-th$ point and the vertex corresponding to the $i-th$ letter. The obtained plot is then called CGR of the 3-state sequence and the x- and y-coordinates of each point on the CGR are considered as the two-dimensional time series. Each time instance of the two-dimensional time series is later considered as a graph node resulting in a single graph representation of the protein sequence based on the Multidimensional Horizontal Visibility Graph (mdHVG) method; a simple yet accurate algorithm that we proposed in [13]. The secondary structural states along with the x and y-coordinates are utilized as the three and only node features that derive directly from the sequence-to-graph process. To the best of our knowledge, there is no prior work in the literature that employs graph neural networks for the protein structural class prediction and especially when combined with mdHVG algorithm. The contributions of this paper are summarized below:

- The protein secondary structural class prediction is verified for the first time in a significantly low-dimensional (3-dimensional) feature space with a novel feature representation method and graph neural networks.
- Through the sequence-to-graph technique, node characteristics are derived directly in an information-recycling manner. As a result of the effectiveness of our suggested framework, no further study for obtaining representative attributes is necessary.
- The experiments are performed for datasets of varying sizes indicating that the herein proposed architecture can generalize for this type of protein data.

The rest of the paper is organized as follows: Section II is an introduction to the data and methods utilized in this work. Section III describes the experimental setup, the classification procedure and the evaluation metrics. Section IV is a discussion on the experimental results, and lastly, Section V draws the conclusions of this work and gives directions for future extensions.

Dataset	Protein Samples per Class				Total Number of samples	Sequence similarity
	α	β	$\alpha+\beta$	α/β		
25PDB	443	443	441	346	1673	25%
640	138	154	171	177	640	25%
FC699	130	269	377	82	858	40%
498	107	126	136	129	498	25%
277	70	61	81	65	277	25%
204	52	61	46	45	204	25%

TABLE I: Dataset description based on structural class information, total number of samples and sequence similarity.

II. Materials & Methods

A. Dataset & Data Preprocessing

This work employs the following publicly available and widely used low-similarity benchmark datasets, namely, the 25PDB [14], FC699 [15], 640 [16], 498, 277 [17], and 204 [18]. As depicted in Table I, 25PDB, FC699 and 640 are large datasets compared to 498, 277 and 204 which are significantly smaller especially for training neural networks. The protein samples of these datasets are categorised based on their structural class to α-fold, β-fold, α/β-fold and $\alpha+\beta$-fold classes. In this work, instead of dealing with the protein primary structure, the PSI-PRED tool [11] is utilized to predict the role of each amino acid in the protein secondary structure. Particularly, PSI-PRED transforms the initial amino acid sequence into another sequence of equal length that now consists of only three states that describe its secondary structure, namely coils (C), strands (E) and helices (H). This simplification not only reduces the dimensionality of our data from 20 amino acids to three structural elements but also the overall computational complexity. Thereafter, CGR [12] is employed in order to transform a sequence of secondary structural elements into a two-dimensional time series [16], [19] that is later represented as a unique planar graph via the process of mdHVG as is described in details in the following section.

B. Multidimensional Horizontal Visibility Graphs

The multidimensional Horizontal Visibility Graph (mdHVG) is a novel algorithm capable of mapping a multidimensional time series into a representative graph as we proposed in [13]. It is experimentally shown that mdHVG is much more efficient than its predecessor Horizontal Visibility Graph

	This work						Panda et al.					
	25PDB	**FC699**	**640**	**498**	**277**	**204**	**25PDB**	**FC699**	**640**	**498**	**277**	**204**
Input features	3	3	3	3	3	3	18	18	18	18	18	18
Batch size	16	16	16	16	16	64	100	100	200	100	50	50
MLP layers	3	3	3	3	3	3	3	3	3	3	3	3
Hidden layer nodes per layer	64	256	128	64	64	256	128	32	128	128	32	32
	64	256	128	64	64	256	64	32	64	64	16	32
	64	256	128	64	64	256	64	32	20	64	16	32
Drop out per layer (%)	40	10	0	30	30	50	30	50	50	20	20	50
	40	10	0	30	30	50	20	32	20	20	20	20
	40	10	0	30	30	50	20	20	20	20	20	20
Learning rate	0.001	0.001	0.001	0.001	0.001	0.0001	0.01	0.01	0.01	0.01	0.01	0.01
Weight decay	0.001	0.001	0.01	0.001	0.001	0.001	0.0	0.0	0.0	0.0	0.0	0.0
Pooling ratio (%)	90	90	80	90	90	80	-	-	-	-	-	-

TABLE II: Network parameters employed in [7] and this work.

(HVG) in terms of computational complexity. In addition, mdHVG algorithm is scalable, simple to implement, and suitable for the analysis of large, heterogeneous and non-stationary time series of varying length.

Specifically, consider a multidimensional time series $\{t_i^{(d)}\}_{i=1}^N$ with d=1,..,D being the number of dimensions. A planar mdHVG graph is constructed with a set of edges E as follows: two nodes n_i and n_j share an edge $e_{ij} \in E$ if and only if for each intermediate time instance, t_k, $t_i < t_k < t_j$ holds that,

$$E = \{e_{ij} |\ s(t_k^{(d)}) < \min\{s(t_i^{(d)}), s(t_j^{(d)})\}, \forall d, d = 1, .., D\} \quad (1)$$

where $s(t_k^{(d)})$, $d = 1, .., D$ is the time series intensity in the d-dimension.

In essence, the algorithm processes directly a multidimensional time series by generating a unique planar graph, considering only the inter-visibility between the pairs of time series intensities among different dimensions exploiting both intra- and inter-data correlations.

C. Graph Neural Network with Hierarchical Graph Pooling & Structure Learning

Graph Convolutional Neural Networks (GCNs) [20], have achieved state-of-the-art performance in a variety of graph-related applications, including protein sequence classification [21]. Hierarchical Graph Pooling with Structure Learning (HGP-SL) [21] is a revolutionary graph pooling operator that combines graph pooling and structure learning into a single module to build hierarchical representations of graphs. In further detail, the graph pooling procedure selects a subset of nodes dynamically to construct an induced subgraph for the future layers. A structure learning technique is also provided to learn a revised graph structure for the pooled graph at each layer in order to preserve the integrity of the graph's topological information.

The node selection procedure in the k-th layer in GCNs for the m-th input graph G_m, is formally defined as the Manhattan distance between the node representation itself and the one constructed from its neighbors as,

$$p = \gamma(\mathcal{G}_m) = ||(I_m^k - (D_m^k)^{-1} A_m^k) H_m^k||_1 \quad (2)$$

where A_m^k and H_m^k are the adjacent and node feature representations matrices respectively, D_m^k represents the diagonal degree matrix of A_m^k, and I_m^k is the identity matrix. Following the evaluation of the node information score, the nodes v_m^k are re-ordered in the graph depending on their scores, and then for a pooling ratio r a subset of the $\lceil r \times v_m^k \rceil$ top-ranked node indices is selected resulting to a new pooled subgraph G_m^k. The adjacency and feature matrices A_m^k and H_m^k are also updated based on the top-scored indices and represent the next layer's node feature and graph structure information.

Considering the structure learning mechanism, a similarity score between two nodes v_m^q and v_m^w is calculated through a sparse attention mechanism,

$$\begin{aligned} S_m^k(q,w) &= max\{0, E_m^k(q,w) - \tau(E_m^k(q,:))\}, \\ E_m^k(q,w) &= \sigma(\vec{a}[H_m^k(q,:)||H_m^k(w,:)]^T) + \lambda A_m^k(q,w) \end{aligned} \quad (3)$$

where $\vec{a} \in R^{1\times 2d}$, with d being the output dimension of the layer, is a weight vector that parametrizes a single layer neural network, $\sigma(\cdot)$ is the activation function, $||$ represents the concatenation operation and λ is a trade-off parameter between the two nodes.

This convolution and pooling operations are repeated several times. Then, a readout function is applied to aggregate node representations to make a fixed size representation, which goes through Multi-Layer Perceptron (MLP) layers for graph classification.

III. Experimental Setup

This section describes in details the parameter setting of our graph neural network architecture as well as the classification procedure. The work presented here is compared and contrasted with the one of Panda *et al.* [7], since it is the first to use neural networks for low-homology protein structural classification. All frameworks are implemented on a desktop computer equipped with a CPU processor (Intel Core i7-7700) clocked at 2.8GHz, and a 12 GB RAM.

A. Network Architecture & Parameter Tuning

The proposed graph neural network is implemented in PyTorch, the Adam optimizer is utilized to optimize the model and L_2 regularization is employed to prevent overfitting during training. The MLP consists of three fully connected layers that

Class	25PDB		FC699		640		498		277		204	
	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.	Sens.	Spec.
α	90.1±3.2	95.9±1.3	86.5±7.3	99.1±0.8	81.9±9.4	96.6±3.1	93.3±5.4	98.3±1.3	89.7±8.8	96.6±2.5	96.9±5.6	100
β	77.9±4.6	95.3±1.2	91.2±5.8	95.7±2.7	96.2±6.6	94.4±2.4	82.7±6.3	98.3±2.2	87.9±11.0	98.0±2.1	98.6±2.9	97.9±2.5
$\alpha+\beta$	64.3±11.8	90.7±3.0	94.9±3.1	84.9±5.4	77.1±11.5	81.5±3.7	91.3±9.2	93.3±2.3	78.5±10.8	85.9±6.6	84.4±17.7	90.4±4.4
α/β	62.0±7.9	83.8±2.8	19.6±15.6	97.2±2.0	46.0±8.5	88.6±4.9	77.43±9.9	95.4±4.9	61.4±19.8	90.9±5.7	62.9±16.0	95.0±4.8
Overall Accuracy	87.2±1.0		92.7±1.1		85.7±1.5		94.7±2.0		89.7±3.6		93.6±2.9	

TABLE III: Performance evaluation of the proposed graph neural network architecture on large (25PDB, FC699, 640) and small datasets (498, 277, 204) in terms of averaged sensitivity, specificity and overall accuracy.

is followed by a softmax classifier. The number of neurons, the learning rate, the weight decay and the pooling ratio are evaluated in a grid search manner. In particular, a grid search parameter tuning is conducted 20 times for a fixed data split, and the most frequent values with the highest accuracy are chosen per dataset. Categorical cross entropy and categorical accuracy are employed as loss functions and performance measures in all configurations. Finally, an early stopping criterion is utilized in the training process so that it is terminated when the validation loss does not reduce for 10 consecutive epochs. Table II provides information on the network parameter set employed in [7] and in the herein study.

B. Classification

In the work of Panda *et al.* [7] there are no records about the classification set up. The highest attained classification accuracy, in particular, is reported without any comment on whether or not the classification method is repeated for a number of random splits. On the contrary, in our work, the data are split in a stratified fashion into 80%-10%-10% for training, validation and testing respectively. The classification procedure is repeated for 15 random data splits for large datasets and 30 for the smaller ones, and the average performance with standard derivation is reported. The input of the neural network specifically is the adjacency matrix of each protein along with its feature matrix that consists of the secondary structure state and the CGR x- and y-coordinates for each node of the graph.

C. Performance metrics

The performance of the proposed architecture is evaluated in terms of overall classification accuracy, sensitivity and specificity as defined in the work of Panda *et al.* [7]. Sensitivity, also referred as the true positive rate or recall, is an important metric that measures the percentage of true samples being correctly identified as true, whereas specificity, or the true negative rate, is the percentage of the actual negative samples being identified as negative. Finally, the number of epochs required for the network to converge is also examined.

IV. Evaluation Results

The performance of the herein proposed framework in terms of overall classification accuracy, sensitivity and specificity is reported in Table III. As provided, the proposed classification scheme is more stable for larger datasets (25PBD, FC699 and 640) since the standard deviation among 15 random data splits is minimal (±1.0, ±1.1 and ±1.5). On the other hand, for smaller datasets (498, 277 and 204), even if we doubled the number of random data splits, the standard deviation is substantially higher (±2, ±3.6, and ±2.9), indicating the algorithm's difficulty in generalizing on such little data. Furthermore, with the exception of dataset 498, the network mostly fails to accurately categorize the α/β structural class, since the true positive rate, or else sensitivity, is substantially low. This could be due to the fact that by its nature the α/β-fold consists of α-helices and almost all parallel β-strands [1] so it is very likely to be interpreted as an α or β structure. This also stands for $\alpha+\beta$ class that consists of α-helices and almost all antiparallel β-strands. For the FC699 dataset in particular the extremely low percentage of sensitivity for the α/β fold is also influenced by the fact that the dataset is quite unbalanced with lesser data belonging to the α and β classes.

Dataset	Overall Classification Accuracy				
	This work				Panda et al.
	Average	Worst-case	Best-case	Average without Pooling	
25PDB	87.2	85.8	**88.8**	78.1	84.2
FC699	92.7	91.3	**95.6**	83.9	93.1
640	85.7	82.6	88.4	75.0	**94.3**
498	94.7	91.5	**98.0**	74.9	95.9
277	89.7	77.7	**96.4**	77.3	94.5
204	93.6	87.8	**98.8**	80.7	85.3

TABLE IV: Comparison between the herein proposed architecture, with and without graph pooling, and the work of Panda *et al.* in terms of overall classification accuracy. The best performed scheme per dataset is highlighted and indicated in bold.

Table IV depicts the comparison between the herein proposed classification scheme, with and without graph pooling, and the work of Panda *et al.* in terms of overall classification accuracy. As mentioned in SectionIII-B Panda *et al.* provide their optimal result with no much explanation. Thus, we compare and contrast our results based on the average, the best and the worst performance of our framework in terms of standard deviation. Considering the case where no graph pooling is conducted, we have also performed an optimal parameter selection for the network in a grid search manner for a fair comparison. As provided, the graph pooling operation enhances indeed the performance of the to a great extend. Along these lines, considering the 25PDB and 204 datasets the herein proposed architecture with graph pooling outperforms the work of Panda *et al.* even in the worst-case scenario. On the contrary, the performance of our scheme on the 640 dataset is

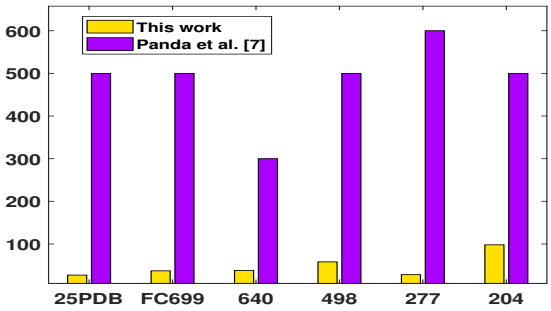

Fig. 2: Number of epochs required for the network to converge. In our work the average number of epochs is reported.

significantly lower even on the best-case scenario. According to Table I, this could be due to the fact that those specific datasets contain in total more $\alpha + \beta$ and α/β classes that are very likely to be misclasified as an α or β fold due to their nature as previously mentioned. Finally, for the remaining datasets, FC699 and 277, our best-case scenario achieves slightly higher classification results compared to Panda *et al.* indicating that overall, despite the low dimensional representation of data, the system has embedded global knowledge about the protein sequences. Overall, the experimental evaluation indicates that graph neural networks perform rather well in terms of protein secondary structure categorization for both large and small datasets.

Regarding the convergence time of our proposed network, as described in Section III-A an early stopping criterion is utilized in the training process. In particular, experimental evaluation verified that after that certain upper limit of training, further overtraining results to a similar performance across all datasets. Figure 2 shows the minimum number of epochs needed for our proposed architectures to reach to an optimal solution. The number is averaged across the number of random data splits and a comparison to the work of Panda *et al.* is also provided. As shown, besides 204 dataset that requires nearly 100 epochs to reach to an optimal solution, for the rest datasets a minimum of 30 epochs is enough.

V. Conclusions and Future Work

In this work we design and implement a novel classification scheme for secondary structure classification of low-homology proteins incorporating graph neural networks to the process for the first time. More importantly, the proposed architecture exploits the information deriving from the sequence-to-graph procedure in an information-recycling manner resulting in significantly low-dimensional (3-dimensional) feature space. The study on real protein data revealed the superiority of the proposed scheme, despite the low dimensional representation of data, indicating that graph neural networks are capable of categorizing the secondary structure of low-homology proteins for both large and significantly small datasets. As a result of the effectiveness of our suggested framework, no further study for obtaining representative attributes is necessary.

An extension of this work will consider a framework that will process directly on the primary amino acid protein sequence without employing intermediate tools such as PSI-PRED and CGR.

References

[1] M. Levitt and C. Chothia, "Structural patterns in globular proteins," *Nature*, vol. 261, no. 5561, pp. 552–558, 1976.

[2] M. Apurva and H. Mazumdar, "Predicting structural class for protein sequences of 40% identity based on features of primary and secondary structure using random forest algorithm," *Computational Biology and Chemistry*, vol. 84, p. 107164, 2020.

[3] M. A. Zervou, E. Doutsi, P. Pavlidis, and P. Tsakalides, "Structural classification of proteins based on the computationally efficient recurrence quantification analysis and horizontal visibility graphs," *Bioinformatics*, vol. 37, no. 13, pp. 1796–1804, 2021.

[4] T. Liu, Y. Qin, Y. Wang, and C. Wang, "Prediction of protein structural class based on gapped-dipeptides and a recursive feature selection approach," *International journal of molecular sciences*, vol. 17, no. 1, p. 15, 2016.

[5] M. Yuan, Z. Yang, G. Huang, and G. Ji, "A novel feature selection method to predict protein structural class," *Computational biology and chemistry*, vol. 76, pp. 118–129, 2018.

[6] X.-J. Zhu, C.-Q. Feng, H.-Y. Lai, W. Chen, and L. Hao, "Predicting protein structural classes for low-similarity sequences by evaluating different features," *Knowledge-Based Systems*, vol. 163, pp. 787–793, 2019.

[7] B. Panda and B. Majhi, "A novel improved prediction of protein structural class using deep recurrent neural network," *Evolutionary Intelligence*, vol. 14, no. 2, pp. 253–260, 2021.

[8] W. R. Atchley, J. Zhao, A. D. Fernandes, and T. Drüke, "Solving the protein sequence metric problem," *Proceedings of the National Academy of Sciences*, vol. 102, no. 18, pp. 6395–6400, 2005.

[9] V. Veljkovic, I. Cosic, D. Lalovic *et al.*, "Is it possible to analyze dna and protein sequences by the methods of digital signal processing?" *IEEE Transactions on Biomedical Engineering*, no. 5, pp. 337–341, 1985.

[10] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[11] D. T. Jones, "Protein secondary structure prediction based on position-specific scoring matrices," *Journal of molecular biology*, vol. 292, no. 2, pp. 195–202, 1999.

[12] H. J. Jeffrey, "Chaos game representation of gene structure," *Nucleic acids research*, vol. 18, no. 8, pp. 2163–2170, 1990.

[13] M. A. Zervou, E. Doutsi, and P. Tsakalides, "Visibility graph network of multidimensional time series data for protein structure classification," in *2021 29th European Signal Proc. Conf. (EUSIPCO) (EUSIPCO)*. IEEE, 2021, pp. 1216–1220.

[14] L. A. Kurgan and L. Homaeian, "Prediction of structural classes for protein sequences and domains—impact of prediction algorithms, sequence representation and homology, and test procedures on accuracy," *Pattern Recognition*, vol. 39, no. 12, pp. 2323–2343, 2006.

[15] K. Chen, L. A. Kurgan, and J. Ruan, "Prediction of protein structural class using novel evolutionary collocation-based sequence representation," *Journal of computational chemistry*, vol. 29, no. 10, pp. 1596–1604, 2008.

[16] J.-Y. Yang, Z.-L. Peng, and X. Chen, "Prediction of protein structural classes for low-homology sequences based on predicted secondary structure," *BMC bioinformatics*, vol. 11, no. 1, p. S9, 2010.

[17] G.-P. Zhou, "An intriguing controversy over protein structural class prediction," *Journal of protein chemistry*, vol. 17, no. 8, pp. 729–738, 1998.

[18] K.-C. Chou, "A key driving force in determination of protein structural classes," *Biochemical and biophysical research communications*, vol. 264, no. 1, pp. 216–224, 1999.

[19] M. Zervou, E. Doutsi, P. Pavlidis, and P. Tsakalides, "Efficient dynamic analysis of low-similarity proteins for structural class prediction," in *28th European Signal Proc. Conf. (EUSIPCO)*. IEEE, 2020, pp. 1328–1332.

[20] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[21] Z. Zhang, J. Bu, M. Ester, J. Zhang, C. Yao, Z. Yu, and C. Wang, "Hierarchical graph pooling with structure learning," *arXiv preprint arXiv:1911.05954*, 2019.