# EFFICIENT PROTEIN STRUCTURAL CLASS PREDICTION VIA CHAOS GAME REPRESENTATION AND RECURRENT NEURAL NETWORKS

*Zervou Michaela Areti*$^{1,2}$ *Effrosyni Doutsi* $^{2}$ *Panagiotis Tsakalides*$^{1,2}$

$^{1}$ Foundation for Research and Technology-Hellas, Institute of Computer Science, Heraklion, Greece
$^{2}$ University of Crete, Department of Computer Science, Heraklion, Greece
{zervou, doutsi, tsakalid}@ics.forth.gr

## ABSTRACT

Predicting the structural class of a protein from its amino acid sequence is among the most significant problems in bioinformatics, especially for proteins with a low sequence similarity. While current methods using recurrent neural networks achieve a notable accuracy in this task, their approach relies on extracting a large quantity of features, impacting the efficiency and reliability of the prediction. In this work, we introduce an efficient and accurate classification scheme based on chaos game representation and recurrent neural networks. The proposed scheme achieves comparable results with state-of-the-art methods, while using a significantly lower-dimensional representation of the feature space.

***Index Terms***— Protein Structural Class Prediction, Gated Recurrent Unit, Chaos Game Representation, Multidimensional Time-series

## 1. INTRODUCTION

Structural class prediction of proteins with low-sequence similarity constitutes one of the most important challenges in bioinformatics. The structure of a protein is associated with its biological function and is determined by its amino acid sequence via the process of protein folding [1]. Structural class prediction provides a better understanding of the protein's biological activity and it is helpful for analyzing protein function, interactions, and regulation. Based on the protein's folding patterns, the following four structural classes exist: (i) all-$\alpha$, where the structural domains are mainly composed of $\alpha$-helices and a small amount of $\beta$-strands; (ii) all-$\beta$, which is mostly formed by $\beta$-strands and a few isolated $\alpha$-helices; (iii) $\alpha + \beta$, forming $\alpha$-helices and mostly anti-parallel $\beta$-strands; and (iv) $\alpha/\beta$ consisting of $\alpha$-helices and almost all parallel $\beta$-strands [2].

A variety of machine learning based algorithms have been developed, with significant effort being given on improving the prediction accuracy for proteins with low sequence similarity [3, 4, 5, 6]. It has been recognized that the performance of the prediction model is significantly impacted by the size and dimensionality of the data. A recent work by Panda *et al.* [7] proposes a deep recurrent neural network architecture that verifies for the first time the structural class prediction of low-sequence similarity proteins in a low-dimensional feature space for large and significantly small datasets. Their approach requires a sophisticated feature extraction process that involves (i) a pretrained model that maps words into embeddings known as word2vec SkipGram model [8], (ii) a representation of the biochemical properties of each amino acid based on Atchley's factors II, IV, V [9], and (iii) the electron ion interaction potential (EIIP) of each amino acid [10]. All three representations arise after further processing in a 18-dimensional feature space for each protein sequence. The resulting feature space is evaluated by a deep neural network architecture consisting of a Gated Recurrent Unit (GRU) [11] and two dense layers, which is optimized with respect to the model's parameters separately for each one of the six different datasets examined.

This work introduces a novel, simple and efficient architecture based on a time-series representation of the protein sequences and recurrent neural networks. The proposed architecture outperforms the work of Panda *et al.* in terms of computational complexity and feature dimensionality while achieving a comparable classification performance. In particular, as depicted in Fig. 1, the protein sequence is transformed into a two-dimensional time-series via the processes of Chaos Game Representation (CGR) [12]. Meanwhile, a significantly lower-dimensional feature representation is derived in an automated-fashion through the data processing pipeline. Finally, a simple neural network architecture, consisting of a GRU and a single dense layer, is employed to process directly on the multidimensional time-series data of varying lengths and the respective features.

All in all, the key contributions of this work are the following: (i) protein structural class prediction is verified for the first time in a significantly lower-dimensional (5-dimensional) feature space via recurrent neural networks, (ii) through our sequence-to-time-series procedure, the respective

This work was funded by the TITAN ERA Chair project (contract no. 101086741) within the Horizon Europe Framework Program of the European Commission and the Hellenic Foundation for Research and Innovation (HFRI) Ph.D. Fellowship grant no. 5647.

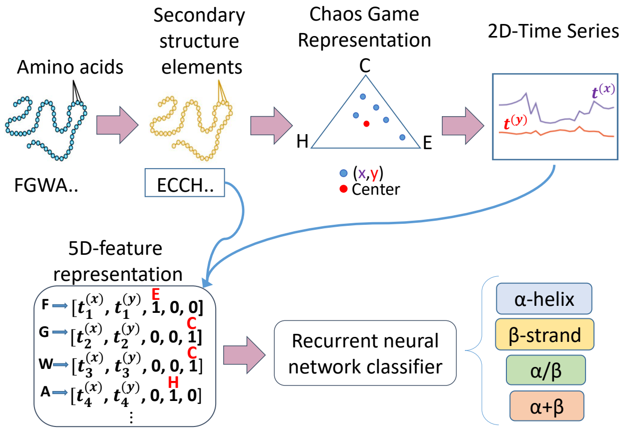
Secondary structure elements
Chaos Game Representation
2D-Time Series
Amino acids
C
H
E
$t^{(x)}$
$t^{(y)}$
FGWA..
ECCH..
● (x,y)
● Center
5D-feature representation
F → $[t_1^{(x)}, t_1^{(y)}, 1, 0, 0]$ E
G → $[t_2^{(x)}, t_2^{(y)}, 0, 0, 1]$ C
W → $[t_3^{(x)}, t_3^{(y)}, 0, 0, 1]$ C
A → $[t_4^{(x)}, t_4^{(y)}, 0, 1, 0]$ H
⋮
Recurrent neural network classifier
α-helix
β-strand
α/β
α+β

**Fig. 1**. General proposed sequence-to-time-series architecture for protein secondary structure classification.

features are derived in an automated, information-recycling manner, (iii) the network architecture is simple yet efficient, (iv) the experiments are performed for datasets of varying sizes with the same network parameter configuration, indicating that the herein proposed architecture can generalize for this type of protein data.

## 2. MATERIALS AND METHODS

### 2.1. Dataset description

This work employs publicly available benchmark datasets of proteins with low-sequence similarity. As depicted in Table 1, 25PDB, FC699 and 640 are large datasets compared to 498, 277 and 204 which are significantly smaller. The protein samples of each dataset are categorised based on their structural class to $\alpha$-fold, $\beta$-fold, $\alpha/\beta$-fold and $\alpha+\beta$-fold classes which refer to the secondary structure of a protein. The sequence similarity is also provided in Table 1. FC699 dataset has a higher sequence similarity considering the rest of the datasets, and is also highly imbalanced with respect to the number of samples on the $\alpha/\beta$ class. Finally, it is important to mention that each dataset contains protein sequences of varying lengths.

| Dataset | Protein samples per class | | | | Number of samples | Sequence similarity | Length range |
|---|---|---|---|---|---|---|---|
| | $\alpha$ | $\beta$ | $\alpha+\beta$ | $\alpha/\beta$ | | | |
| **25PDB [13]** | 443 | 443 | 441 | 346 | 1673 | 25% | [13,697] |
| **FC699 [14]** | 130 | 269 | 377 | 82 | 858 | 40% | [46,808] |
| **640 [15]** | 138 | 154 | 171 | 177 | 640 | 25% | [36,823] |
| **498 [16]** | 107 | 126 | 136 | 129 | 498 | 25% | [9,537] |
| **277 [16]** | 70 | 61 | 81 | 65 | 277 | 25% | [31,255] |
| **204 [17]** | 52 | 61 | 46 | 45 | 204 | 25% | [16,360] |

**Table 1**. Dataset description.

### 2.2. PSI-blast based Secondary Structure Prediction

The pipeline followed in this work is described in Fig. 1. As depicted, 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 [18]. This simplification reduces the dimensionality of the data from 20 amino acids to three structural elements and thus reduces the overall computational complexity as well.

### 2.3. Time-series Generation via Chaos Game Representation

In order to transform the unidimenisional sequence of characters into a two-dimensional time series, chaos game representation (CGR) [12] is employed. In essence, CGR is able to graphically represent the sequence while preserving its original structure. In this work, the updated 3-state sequence is represented in a unit equilateral triangle with its three vertices referring to the three secondary structure types helix (H), coil (C), and strand (E). Then the triangle centroid $(t_0^{(x)}, t_0^{(y)})$ is defined and the $(t_1^{(x)}, t_1^{(y)})$ coordinates of the first point of the plot are evaluated as the midpoint distance between the center of the triangle and the vertex corresponding to the first letter of the 3-state sequence. The following successive elements in the 3-state sequence are displayed as the midpoint distance between the previous plotted point and the vertex representing the element being plotted as follows,

$$\begin{aligned} t_i^{(x)} &= \tfrac{1}{2}(t_{i-1}^{(x)} + v_i^{(x)}), \quad \text{for} \quad i = 1, \ldots, N \\ t_i^{(y)} &= \tfrac{1}{2}(t_{i-1}^{(y)} + v_i^{(y)}), \quad \text{for} \quad i = 1, \ldots, N \end{aligned} \tag{1}$$

where $v_i^{(x)}$ and $v_i^{(y)}$ are respectively the $x$ and $y$ coordinates of the vertex corresponding to the $i$-th secondary structure element of a protein with sequence length $N$. Finally, as presented in Fig. 1, the obtained graphical representation, the so called CGR, of the 3-state sequence is decomposed into a two-dimensional time-series, where $T^{(1)} = \{t_1^{(x)}, t_2^{(x)}, \ldots, t_N^{(x)}\}$ and $T^{(2)} = \{t_1^{(y)}, t_2^{(y)}, \ldots, t_N^{(y)}\}$.

## 3. EXPERIMENTAL SETUP

This section describes in detail the parameter setting of our recurrent neural network architecture as well as the classification procedure and metrics. The herein work is compared and contrasted with the one of Panda *et al.* [7], since, to the best of our knowledge, it is the only one to use recurrent neural networks for the problem of structural classification of low-sequence similarity proteins. The model is implemented on a desktop computer equipped with NVIDIA's GPU model Quadro P4000, with 8Gb available RAM.

### 3.1. Network Architecture

The designed architecture is developed in Python programming platforms, by exploiting the open-source machine learning framework PyTorch. In more detail, the network con-

|  | **This work** | **Panda *et al.*** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Dataset** | **25PDB, FC699, 640.<br>498, 277, 204** | **25PDB** |  |  | **FC699** |  |  | **640** |  |  | **498** |  |  | **277** |  |  | **204** |  |  |
| **Number of features** | 5 | 18 |  |  | 18 |  |  | 18 |  |  | 18 |  |  | 18 |  |  | 18 |  |  |
| **Batch size** | 1 | 100 |  |  | 100 |  |  | 200 |  |  | 100 |  |  | 50 |  |  | 50 |  |  |
| **Number of layers** | 2 | 3 |  |  | 3 |  |  | 3 |  |  | 3 |  |  | 3 |  |  | 3 |  |  |
| **Hidden states per layer** | 64 | 124 | 64 | 64 | 32 | 32 | 32 | 128 | 64 | 20 | 128 | 64 | 64 | 32 | 16 | 16 | 32 | 32 | 32 |
| **Dropout per layer** | 0.1 | 0.3 | 0.2 | 0.2 | 0.5 | 0.3 | 0.2 | 0.5 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.2 | 0.5 | 0.2 | 0.2 |
| **Learning rate** | 0.001 | 0.01 |  |  | 0.01 |  |  | 0.01 |  |  | 0.01 |  |  | 0.01 |  |  | 0.01 |  |  |

**Table 2**. Network parameters employed in this work and the work of Panda *et al.* [7].

sists of a GRU and a fully connected layer that is followed by a softmax classifier with negative sampling. The number of hidden states and the learning rate are evaluated in a grid search manner. In particular, the learning rate and the number of hidden states are set to $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ and $\{16, 32, 64, 128, 256\}$ respectively. The grid search parameter tuning is conducted 20 times, and the most frequent values that result in high accuracy across all datasets are chosen. Table 2 provides information on the network parameters employed in the work of Panda *et al.* and in the herein study. As provided, while it is common to pad or crop sequences to a common length, this work keeps each time series length unaltered and therefore, the batch size is set to 1 to process each data sample separately. It is important to note that we have also designed architectures with two and three dense layers which also led to lower performance.

Regarding the optimization process of the network, the Adam optimizer is employed with a constant learning rate of 0.01 and exponential decay rates for the $1^{st}$ and $2^{nd}$ moment estimates been equal to $\beta_1 = 0.9$ and $\beta_2 = 0.999$, respectively. $L_2$ regularization is also introduced and evaluated in the grid search process with the weight decay values set to $\{0, 10^{-4}, 10^{-3}, 10^{-2}\}$, where the zero value indicates that the parameter is excluded. The grid search revealed that no substantial improvement is achieved on network's performance with the presence of weight decay, and thus it is set to zero. Moreover, categorical cross entropy and categorical accuracy are employed as loss functions and performance measures during training. Finally, an early stopping criterion is used in the training process to ensure that it is terminated when the validation loss does not decrease for 40 consecutive epochs. The neural network's input is a feature representation consisting of the two-dimensional time-series and the secondary structure element description (H, E and C) of each time instance that is presented as three extra vectors in the form of one-hot-encoding for each of the three secondary structure states. All in all, in this work we provide a general and simple framework for classifying low-sequence similarity proteins independently of the size of the dataset.

### 3.2. Classification Setup $\&$ Performance Metrics

In this work, the data are split in a stratified fashion into 80%-10%-10% training, validation and testing non-overlapping sets respectively. The model's performance is reported based on the lowest validation loss achieved during the training process. The best performing model is then evaluated on the test set to assess the final performance of the method. The classification procedure is repeated for 20 random data splits and the average performance with standard derivation is reported. The performance of the proposed architecture is evaluated in terms of overall classification accuracy, precision and recall. On the contrary, there are no records concerning the classification setup in the study of Panda *et al.* where a single run of their model is reported.

## 4. EVALUATION RESULTS

The performance of the proposed framework in terms of average precision and recall is presented in Table 3. Precision represents the prediction quality of the classifier, whereas, recall can be viewed as a measure of quantity of correctly classified samples. As provided on Table 3, the proposed classification scheme is less confident considering the prediction of $\alpha/\beta$ and/or $\alpha + \beta$ classes across datasets. An intuitive explanation is that the $\alpha/\beta$ and $\alpha + \beta$ folds consist of $\alpha$-helices and almost all parallel/antiparallel $\beta$-strands by their nature, so it is very likely to be interpreted as one among each other or an $\alpha$ or $\beta$ structure. In particular, considering the 25PDB dataset, $\alpha/\beta$ class is misclassified as $\beta$ and $\alpha + \beta$-fold as given by the precision of the model for this particular class. Considering the datasets 640 and 277, the model misclassifies $\alpha/\beta$ class mainly to the $\alpha + \beta$ class, as these particular datasets contain more samples in those classes as described in Table 1. Similarly, the previous hold for the 498 and 204 dataset, however, the overall performance of proposed model is quite sufficient. Finally, for the FC699 dataset we observe that the model achieves high quality prediction for the first three classes and is less accurate considering the final class. This is due to the fact that (i) the protein sequences have higher similarity comparing to the rest of the datasets and (ii) the dataset is highly unbalanced with respect to the last class as shown in Table 1. The overall classification accuracy of the proposed classification scheme in comparison with the study of Panda *et al.* is depicted in Table 4. As mentioned in Section 3.2 Panda *et al.* report a single run of their model with no discussion around the classification process. In this work, we perform a 20-fold cross-validation and report the average

| Class | 25PDB | | FC699 | | 640 | | 498 | | 277 | | 204 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Precision | Recall | Precision | Recall | Precision | Recall | Precision | Recall | Precision | Recall | Precision | Recall |
| $\alpha$ | 87.8 | 90.7 | 94.7 | 95.2 | 93.0 | 87.4 | 94.0 | 89.5 | 86.1 | 89.6 | 98.3 | 96.7 |
| $\beta$ | 87.9 | 79.2 | 94.1 | 92.9 | 83.2 | 87.0 | 96.3 | 95.2 | 91.5 | 83.7 | 95.3 | 99.2 |
| $\alpha + \beta$ | 79.8 | 77.8 | 93.5 | 97.2 | 78.7 | 85.3 | 89.2 | 95.3 | 78.5 | 89.1 | 90.4 | 97.1 |
| $\alpha/\beta$ | 66.4 | 70.8 | 76.6 | 77.0 | 71.1 | 62.9 | 88.6 | 85.3 | 73.6 | 55.3 | 70.4 | 80.2 |

**Table 3**. Performance evaluation of the proposed architecture in terms of average precision and recall per class.

| | Overall Classification Accuracy % | | |
|---|---|---|---|
| | Panda *et al.* | This work | |
| | | Best run | Average (std) |
| **25PDB** | 84.2 | **85.6** | 79.8 (3.0) |
| **FC699** | 93.1 | **97.6** | 92.9 (3.5) |
| **640** | **94.3** | 87.6 | 80.2 (4.2) |
| **498** | 95.9 | **98.0** | 91.4 (5.0) |
| **277** | 94.5 | **94.6** | 80.2 (9.2) |
| **204** | 85.3 | **97.5** | **93.8 (7.6)** |

**Table 4**. Comparison between the proposed architecture and the work of Panda *et al.* in terms of overall classification accuracy.

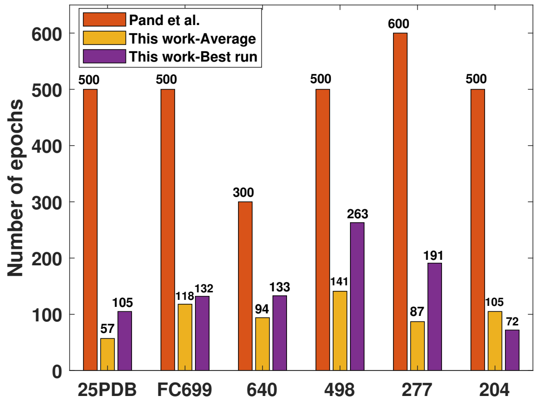

**Fig. 2**. Number of epochs required for the work of Panda *et al.* and the proposed model considering the average and best run.

as well as the respected standard deviation among the 20 individual runs. Furthermore, we report the highest achieved accuracy among the 20-folds. As shown in Table 4, the proposed model achieves comparable and even higher average classification accuracy compared to the work of Panda *et al.* for the FC699 and 204 datasets repsectively. On the contrary, for 640 and 277 datasets the average classification accuracy is significantly lower. This is due to the fact that those specific datasets contain in total more $\alpha + \beta$ and $\alpha/\beta$ classes which as discussed previously are not properly distinguished by the proposed classifier. Similarly, for the 25PDB and 498 datasets, the average performance of the model is not as high as in the work of Panda *et al.* since the proposed model is less confident in predicting $\alpha + \beta$ and $\alpha/\beta$ classes. On the other hand, considering the best model among the 20 different data splits, we observe that it outperforms the work of Panda *et al.* in every case besides the 640 dataset. It is important to mention that the configuration of the proposed model provided the highest achieved accuracy for 640 dataset compared to the rest configurations studied during the grid search parameter tuning procedure.

Regarding the convergence time of our proposed network, as described in Section 3.1 an early stopping criterion is utilized in the training process. Experimental evaluation verified that further training results to a similar performance across all datasets. Fig. 2 displays the number of epochs required for the model of Panda *et al.* and the proposed model to converge. As shown, the proposed model requires significantly less epochs on average to provide accurate prediction results. We also report the number of required epochs for the best model achieved per dataset. As observed, for the majority of the datasets, the number of epochs in this case is higher than the average due to the fact that the number of epochs per run varies significantly. Overall, the experimental evaluation indicates that the proposed architecture is able to predict the structural class of low-sequence similarity proteins for both large and small datasets providing comparable results with the state-of-the-art with a significantly lower number of required epochs.

## 5. CONCLUSIONS

In this work we design and implement a novel classification scheme for secondary structure classification of proteins with low-sequence similarity incorporating chaos game representation and recurrent neural networks. The suggested design takes advantage of the information derived from the sequence-to-time-series technique in an automated, information-recycling fashion, resulting in a relatively low-dimensional (5-dimensional) feature space. Despite the low-dimensionality of the feature space, the analysis on real protein data indicated the superiority of the suggested scheme, demonstrating that the proposed architecture is capable of classifying the secondary structure of proteins with low- sequence similarities for both large and considerably small datasets.

## 6. REFERENCES

[1] Christian B Anfinsen, "Principles that govern the folding of protein chains," *Science*, vol. 181, no. 4096, pp. 223–230, 1973.

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

[3] Taigang Liu, Yufang Qin, Yongjie Wang, and Chunhua 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, pp. 15, 2016.

[4] Mehta Apurva and Himanshu 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, pp. 107164, 2020.

[5] Michaela Areti Zervou, Effrosyni Doutsi, Pavlos Pavlidis, and Panagiotis 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.

[6] Xiao-Juan Zhu, Chao-Qin Feng, Hong-Yan Lai, Wei Chen, and Lin Hao, "Predicting protein structural classes for low-similarity sequences by evaluating different features," *Knowledge-Based Systems*, vol. 163, pp. 787–793, 2019.

[7] Bishnupriya Panda and Babita 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] Chris McCormick, "Word2vec tutorial-the skip-gram model," *Apr-2016.[Online]. Available: http://mccormickml. com/2016/04/19/word2vec-tutorial-the-skip-gram-model*, 2016.

[9] William R Atchley, Jieping Zhao, Andrew D Fernandes, and Tanja Drüke, "Solving the protein sequence metric problem," *Proceedings of the National Academy of Sciences*, vol. 102, no. 18, pp. 6395–6400, 2005.

[10] 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.

[11] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

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

[13] Lukasz A Kurgan and Leila 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.

[14] KE Chen, Lukasz A Kurgan, and Jishou 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.

[15] Jian-Yi Yang, Zhen-Ling Peng, and Xin Chen, "Prediction of protein structural classes for low-homology sequences based on predicted secondary structure," *BMC bioinformatics*, vol. 11, no. 1, pp. S9, 2010.

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

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

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