Bioinformatics, 37(13), 2021, 1796–1804
doi: 10.1093/bioinformatics/btab407
Advance Access Publication Date: 28 May 2021
Review

Structural bioinformatics

Structural classification of proteins based on the computationally efficient recurrence quantification analysis and horizontal visibility graphs

Michaela Areti Zervou [1,2], Effrosyni Doutsi[2,*], Pavlos Pavlidis[2] and Panagiotis Tsakalides[1,2]

[1]Department of Computer Science, University of Crete, Heraklion 700 13, Greece and [2]Signal Processing Lab, Institute of Computer Science, Foundation for Research and Technology-Hellas, Heraklion 700 13, Greece

*To whom correspondence should be addressed.

Associate Editor: Pier Luigi Martelli
Received on January 11, 2021; revised on April 13, 2021; editorial decision on May 23, 2021; accepted on May 27, 2021

Abstract

Motivation: Protein structural class prediction is one of the most significant problems in bioinformatics, as it has a prominent role in understanding the function and evolution of proteins. Designing a computationally efficient but at the same time accurate prediction method remains a pressing issue, especially for sequences that we cannot obtain a sufficient amount of homologous information from existing protein sequence databases. Several studies demonstrate the potential of utilizing chaos game representation along with time series analysis tools such as recurrence quantification analysis, complex networks, horizontal visibility graphs (HVG) and others. However, the majority of existing works involve a large amount of features and they require an exhaustive, time consuming search of the optimal parameters. To address the aforementioned problems, this work adopts the generalized multidimensional recurrence quantification analysis (GmdRQA) as an efficient tool that enables to process concurrently a multidimensional time series and reduce the number of features. In addition, two data-driven algorithms, namely average mutual information and false nearest neighbors, are utilized to define in a fast yet precise manner the optimal GmdRQA parameters.

Results: The classification accuracy is improved by the combination of GmdRQA with the HVG. Experimental evaluation on a real benchmark dataset demonstrates that our methods achieve similar performance with the state-of-the-art but with a smaller computational cost.

Availability and implementation: The code to reproduce all the results is available at https://github.com/aretiz/protein_structure_classification/tree/main.

Contact: edoutsi@ics.forth.gr

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Protein structural class prediction is one of the most important and challenging issues in computational biology. Obtaining knowledge of protein function and regulation is highly important and useful in medicine and biotechnology (Noble et al., 2004), especially for drug design, enzymes composition and interpretation of disease related phenotypes. Based on their folding patterns, proteins can be classified into four structural categories, namely, (i) all-α, where the structural domains are mainly composed of α-helices and a small amount of β-strands (a.k.a. β-sheets), (ii) all-β, that is mostly formed by β-strands and a few isolated α-helices, (iii) $\alpha + \beta$, forming α-helices and mostly anti-parallel β-strands and (iv) α/β consisting of α-helices and almost all parallel β-strands (Levitt and Chothia, 1976; Orengo et al., 1997).

The last few decades, the accelerated evolution of genomics has lead to a substantial volume of amino acid sequence data of proteins. Therefore, an emerging need of efficient architectures for protein structure classification has arisen. Along these lines, a plethora of machine learning based algorithms has been developed for protein structural class prediction. Some extensively utilized methods for the representation of protein samples via feature extraction are the amino and pseudo-amino acid composition (Chou, 2001; Nakashima et al., 1986), the identification of binding motifs in protein–protein interactions (Guharoy and Chakrabarti, 2007), the PSI-BLAST profile (Liu et al., 2010) and the predicted secondary

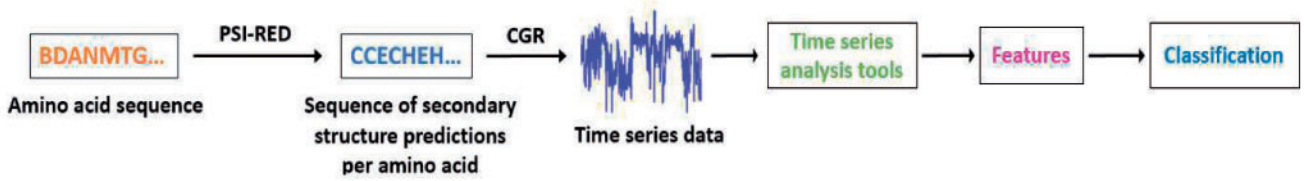

Fig. 1. General protein structural class prediction scheme

structure information (Liu and Jia, 2010), among others (Spänig and Heider, 2019). However, the performance of these techniques suffers when low-similarity proteins are encountered. Thus, significant effort has been put to improve the prediction accuracy for proteins for which we cannot obtain a sufficient amount of homologous information (Apurva and Mazumdar, 2020; Liang et al., 2015; Wang et al., 2015; Yang et al., 2008; Yu et al., 2017; Zhang et al., 2012; Zhu et al., 2019).

Numerous studies (Jiang et al., 2019; Olyaee et al., 2016; Yang et al., 2009, 2010) demonstrate the potential of transforming the amino acid sequence into a time series and then, utilizing powerful time series analysis techniques such as Recurrence Quantification Analysis (RQA) (Eckmann et al., 1987), Horizontal Visibility Graphs (HVG) (Lacasa et al., 2008) or a combination of both, for extracting meaningful information from the data. The general procedure for protein structure classification is briefly depicted in Figure 1. According to the scheme, every amino acid in the protein sequence is initially predicted as one of three secondary structural elements, namely H (helix), E (strand) and C (coil) using the PSI-PRED (Jones, 1999) tool and then, by employing the Chaos Game Representation (CGR) (Jeffrey, 1990) technique it is possible to generate the time-series which are processed to create the set of features. Although the aforementioned studies lead to high-precision results, their enhanced performance comes at the expense of time and memory complexity as well as with an inefficient processing of the multidimensional data. This is mainly due to the fact that (i) a pair of coordinates has to be processed separately, resulting in a large number of features and (ii) the fine-tuning of the RQA parameters for each time series is highly time demanding.

To overcome these limitations, this work processes directly on the two-dimensional time series by introducing (i) the utilization of the Generalized multidimensional Recurrence Quantification Analysis (GmdRQA) (Zervou et al., 2019) as a sophisticated, nonlinear analysis tool of multidimensional time series data, and (ii) a data-driven estimation of the RQA parameters employing the Average Mutual Information (AMI) (Fraser and Swinney, 1986) and the False Nearest Neighbors (FNNs) (Kennel et al., 1992) methods. The aforementioned mechanisms enable to exploit both the intra- and inter-data correlations among the two time series in an automated fashion, resulting in lower time and memory complexity. The contributions of this article are summarized below:

i. The utilization of GmdRQA greatly reduces the number of features as it is applied directly to the two-dimensional time series.
ii. The introduction of a data-driven parameter selection employing the FNN and AMI algorithms.
iii. The design of a novel feature extraction scheme consisting of HVG and the Data-Driven unidimensional RQA (DD-RQA) or the Data-Driven GmdRQA (DD-GmdRQA) frameworks that enable the discovery of representative patterns, increasing the overall accuracy of the protein structural class prediction.
iv. The significant reduction of the computational complexity of both the feature selection and the classification processes.

The rest of the article is organized as follows: Section 2 is a background on the most notable methods that have been used in the literature for the generation of time series from an amino acid sequence and gives details on the proposed GmdRQA architectures that consist of two data-driven parameter-tuning algorithms, namely AMI and FNN. Section 3 introduces the proposed time delay embedding parameter selection architectures, the classification procedure and the performance of the proposed architectures in terms of classification accuracy, time complexity and feature multitude. Finally, Section 4 draws the conclusion of this work and gives directions for future extensions.

2 Materials and methods

The purpose of this section is to briefly introduce the most significant mechanisms that have been used in the literature in order to (i) transform the amino acid sequence into time series and (ii) extract informative features for an accurate protein structural class prediction using machine learning algorithms. Specifically, the PSI-PRED and the CGR methods are first introduced as they are employed to convert an amino-acid sequence into a time series. Then, two time series analysis techniques, the RQA and the HVG, are described in details to extract information-rich characteristics.

2.1 PSI-blast based secondary Diction

There are several works in the literature (Jiang *et al.*, 2019; Olyaee *et al.*, 2016; Yang *et al.*, 2010) that instead of dealing with the protein primary structure they use the PSI-blast based secondary structure PREDiction (PSIPRED) tool that predicts the role of each amino acid in the protein secondary structure. Only 20 amino acids are known as proteinogenic, meaning they participate in the synthesis of a protein primary structure. Particularly, PSI-PRED transforms the initial amino acid sequence to a 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 reduces the dimensionality of our data from 20 amino acids to three structural elements, thus easing the overall computational complexity. Hence, in this work, we have decided to use as input data the prediction of the protein secondary structure.

2.2 Time series generation via Chaos Game Representation

In order to transform the unidimenisional sequence of characters into a two-dimensional time series, CGR is employed. CGR was first proposed as a scale-independent representation of genomics (Almeida et al., 2001; Jeffrey, 1990) and has been also recently utilized in protein classification studies (Löchel et al., 2020). In essence, CGR is able to graphically represent the sequence while preserving its original structure. Specifically, a sequence is represented in a unit equilateral triangle. Its three vertices refer to the three secondary structure types namely helix (H), coils (C) and strands (E), with xy-plane coordinates (0,0), (0.5,$\sqrt{3}/2$) and (1,0), respectively. The CGR graph, as shown in Figure 2 (Left) is obtained through the following procedure: Initially, the triangle centroid $(t_0^{(x)}, t_0^{(y)}) = (0.5, \sqrt{3}/6)$ is defined and then, the $(t_1^{(x)}, t_1^{(y)})$ coordinates of the first element of the sequence are calculated as the halfway distance point between the center of the triangle and the vertex representing this element. Accordingly, the remaining consecutive elements in the secondary structure sequence are plotted as the midpoint between the previous plotted point and the vertex representing the element being plotted as follows:

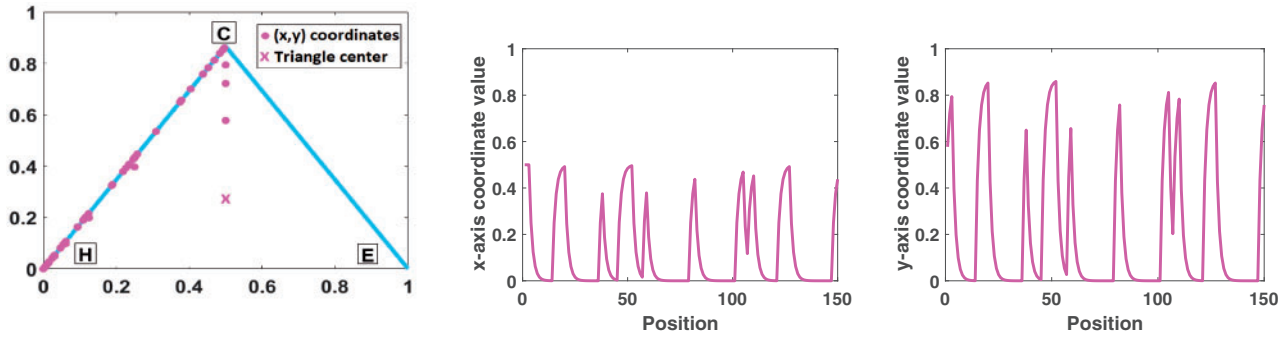

Fig. 2. Left: Chaos Game Representation of protein's 1ASH predicted secondary structure ('*CCCHH*...*HHCCC*') based on the three secondary structural elements, C (coil), E (strand) and H (helix). This graph indicates that the sequence consist of only coil and helix elements. Middle: This time series represent the *x*-coordinates of the points. Right: This time series represent the *x*-coordinates of the points

$$t_i^{(x)} = \frac{1}{2}\left(t_{i-1}^{(x)} + v_i^{(x)}\right), \quad \text{for} \quad i = 1, \ldots, N$$
$$t_i^{(y)} = \frac{1}{2}\left(t_{i-1}^{(y)} + v_i^{(y)}\right), \quad \text{for} \quad i = 1, \ldots, N \tag{1}$$

where $v_i^{(x)}$ and $v_i^{(y)}$ are respectively the x and y coordinates of the vertex corresponding to the ith secondary structure element of a protein consisting of N amino-acids. Finally, as depicted in Figure 2 (Middle) and Figure 2 (Right), the CGR graph is decomposed into two time series that consist accordingly of the x and y coordinates so that $t^{(x)} = \{t_1^{(x)}, t_2^{(x)}, \ldots, t_N^{(x)}\}$ and $t^{(y)} = \{t_1^{(y)}, t_2^{(y)}, \ldots, t_N^{(y)}\}$.

2.3 Time series analysis

2.3.1 Recurrence quantification analysis

The RQA (Eckmann *et al.*, 1987) is exploited to perform a sophisticated non-linear analysis of the time series data. RQA is capable of treating non-stationary and short data series, as it comprises a set of appropriate quantitative measures for the analysis of recurrences, typically small-scale structures. As a result, RQA enables the detection of critical transitions in the system's dynamics (e.g. deterministic, stochastic, random). More specifically, a recurrence plot (RP) is derived depicting those times at which a state of a dynamical system recurs. In particular, the recurrence of a state that occurs at time i and at a different time j is represented within a two-dimensional square matrix with ones (recurrence) and zeros (non-recurrence), where both axes are time axes. In other words, RPs reveal all the times when the phase space trajectory of the dynamical system visits roughly the same area in the phase space. To this end, RPs enable the investigation of an m-dimensional phase space trajectory through a two-dimensional representation of its recurrences.

Given a time series of length N, $\{t_i\}_{i=1}^N$, a phase space trajectory can be reconstructed via time-delay embedding,

$$\mathbf{x}_i = [t_i, t_{i+\tau}, \ldots, t_{i+(m-1)\tau}], \quad i = 1, \ldots, N_s, \tag{2}$$

where m is the embedding dimension, τ is the time delay and $N_s = N - (m-1)\tau$ is the number of states. Having constructed a phase space representation, an RP is defined as follows:

$$\mathbf{R}_{i,j} = \Theta(\varepsilon - ||\mathbf{x}_i - \mathbf{x}_j||_p), \quad i, j = 1, \ldots, N_s, \tag{3}$$

where $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^m$ are the states, ε is a threshold, $||\cdot||_p$ denotes a general ℓ_p norm and $\Theta(\cdot)$ is the Heaviside step function, whose discrete form is defined by

$$\Theta(w) = \begin{cases} 1, & \text{if} w \geq 0 \\ 0, & \text{if} w < 0 \end{cases}, \; w \in \mathbb{R}. \tag{4}$$

The resulting matrix $\mathbf{R}$ exhibits the main diagonal, $\mathbf{R}_{i,i} = 1$, $i = 1, \ldots, N_s$, also known as the *line of identity* (LOI). Typically, several linear (and/or curvilinear) structures appear in RPs, which give hints about the time evolution of the high-dimensional phase space trajectories. A major advantage of RPs is that they can also be applied to rather short and even non-stationary data. The visual interpretation of RPs, which is often difficult and subjective, is enhanced by means of several numerical measures for the quantification of the structure and complexity of RPs (Zbilut and Webber, 1992). These quantification measures provide a global picture of the underlying dynamical behavior during the entire period covered by the data. This work utilizes 10 of the RQA quantitative measures (Marwan et al., 2007) that can be found in Supplementary Data File.

2.3.2 Horizontal visibility graph

In recent years, complex network theory has been popularized in the analysis of biological problems (Zhao et al., 2018). A simple and fast computational method, known as HVG (Lacasa *et al.*, 2008), maps time series into graphs. HVG is invariant under affine transformations of the series data and its main focus lies on the time series structural properties (periodicity, fractality, etc.). Specifically, let a time series $\{t_i\}_{i=1}^N$. The HVG algorithm assigns each sample point t_i as a node n_i of a graph G. Then, two nodes n_i and n_j in the network are connected if the geometrical rule in (5) is satisfied,

$$n_k < \min(n_i, n_j), \; i < k < j. \tag{5}$$

In essence, two nodes n_i and n_j share an edge when a horizontal line can be drawn among them without intersecting, in terms of magnitude, any intermediate node. In this study, the resulted HVG $G = (V, E)$, $N = |V|$, $M = |E|$, with N and M being the number nodes and edges respectively, is undirected, unweighted and connected. The graph properties are represented by the measures described Supplementary Data File that are later employed for classification purposes.

2.4 Dataset description

This work employs the 25PDB dataset that includes 1673 proteins of varying length with 25% sequence homology. The length wise distribution of the protein sequences is comparable for the all-α, all-β and $\alpha + \beta$ folds, whereas for the α/β fold the length is observed to be generally higher. Particularly, in the all-α, all-β and $\alpha + \beta$ folds, there is a higher proportion of small protein sequences with less than 100 residues compared to the α/β fold. On the other hand, the number of sequences that consist of more than 300 residues is higher for the α/β and $\alpha + \beta$ folds against all-α and all-β folds. The proteins are categorized based on their structural class as following: 443 proteins belong to the all-α, 443 to the all-β, 346 to the α/β and 441 to $\alpha + \beta$ fold.

2.5 Generalized multidimensional recurrence quantification analysis

Multidimensional RQA (Wallot et al., 2016) extracts the underlying dynamics of the system by mapping the time series in a higher dimensional phase space of trajectories by constructing state vectors $\mathbf{u}_i$ via time delay embedding. The generalized multidimensional

GmdRQA framework transforms state vectors u_i into state matrices X_i to represent the time-delay embedding (Zervou *et al.*, 2019). This is due to the fact that state matrices are considered more appropriate for describing multidimensional signals from a mathematical perspective, enabling them to model the correlations not only within a signal but also between different signals. More specifically, given a multidimensional time series $\{t_i^{(D)}\}_{i=1}^{N}$, where D stands for the data dimensionality, the corresponding phase space representation is reconstructed as follows:

$$\begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_{N_s} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1^{(1)} & \mathbf{x}_1^{(2)} & \cdots & \mathbf{x}_1^{(D)} \\ \mathbf{x}_2^{(1)} & \mathbf{x}_2^{(2)} & \cdots & \mathbf{x}_2^{(D)} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{x}_{N_s}^{(1)} & \mathbf{x}_{N_s}^{(2)} & \cdots & \mathbf{x}_{N_s}^{(D)} \end{pmatrix}, \tag{6}$$

where $\mathbf{x}_i^{(d)} = (t_i^{(d)}, t_{i+\tau}^{(d)}, \ldots, t_{i+(m-1)\tau}^{(d)})$, $i = 1, \ldots, N_s$, $d = 1, \ldots, D$, m being the embedding dimension, τ the delay and $N_s = N - (m-1)\tau$ the number of states. The state vectors $\mathbf{u}_i$ can be transformed into state matrices of the form

$$\mathbf{X}_i = \begin{pmatrix} \mathbf{x}_i^{(1)} & \mathbf{x}_i^{(2)} & \cdots & \mathbf{x}_i^{(k)} \\ \mathbf{x}_i^{(k+1)} & \mathbf{x}_i^{(k+2)} & \cdots & \mathbf{x}_i^{(2k)} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{x}_i^{(l-1)(k+1)} & \mathbf{x}_i^{(l-1)(k+2)} & \cdots & \mathbf{x}_i^{(D)} \end{pmatrix}, \tag{7}$$

where $k = \lfloor \sqrt{D} \rfloor$, $l = \lfloor D/k \rfloor$ and $i = 1, \ldots, N_s$. Subsequently, the Generalized multidimensional Recurrence Plot (GmdRP) is defined according to (3). Having constructed the corresponding RP of the multidimensional system, the features described in Supplementary Data File can also be employed.

2.6 Estimation of embedding parameters

Identifying the optimal RQA/GmdRQA parameters for the reconstruction of the phase space is extremely crucial. To the best of our knowledge, parameter tuning is usually done in a grid search manner which is time demanding. In the herein work, the estimation of the embedding parameters is performed in a Data-Driven (DD) fashion where the AMI and the FNN methods are utilized to evaluate the optimal time delay τ and minimal sufficient value of the embedding dimension m, respectively.

2.6.1 Average Mutual Information Algorithm

The AMI (Fraser and Swinney, 1986) is a measure of non-linear correlation between the given signal $\{\mathbf{t}_i\}_{i=1}^{N}$ and a time delayed version of this signal by τ samples $\{\mathbf{t}_{i+\tau}\}_{i=1}^{N}$ and is expressed as,

$$I(t_i, t_{i+\tau}) = \sum_{i,j} p_{jk}(\tau) \log\left(\frac{p_{jk}(\tau)}{p_j p_k}\right) \tag{8}$$

where p_j is the probability that t_i is in bin j of the histogram constructed from the data points in t, and $p_{jk}(\tau)$ is the probability that t_i is in bin j and $t_{i+\tau}$ is in bin k.

Determining a proper value for τ implies that the coordinates of the phase space embedded signal will be maximally independent. As proposed by Fraser and Swinney (1986), this is guaranteed by choosing as the optimal value for the time lag τ the position of the first minimum of $I(t_i, t_{i+\tau})$. Nonetheless, it is possible that the AMI function does not acquire a local minimum. Therefore, Kantz and Schreiber (2004) introduced a criterion where the optimal value is considered to be the lowest value of τ for which the AMI function descents below the value 1/e, $e \approx 2.71$. Furthermore, in this study, the max lag τ as well as the number of bins for calculating the histogram are set to 10.

2.6.2 False Nearest Neighbor Algorithm

The embedding dimension m is an estimate of the dimensionality of the dynamics of the time series. The FNN (Kennel *et al.*, 1992) method is based on the assumption that two points that are close in the sufficient embedding dimension should continue to be close as the dimension increases. A criterion for recognizing embedding errors is a considerable increase in the distance between two neighboring points while moving from dimension m to $m+1$.

Specifically, given an embedded time series in a m-dimensional phase space with a time delay τ and two of its coordinate vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ that are adjacent at a time instance, the squared Euclidean distance between them when moving from m into $(m+1)$ dimensions is,

$$\mathfrak{D}_{m+1}^2 = \mathfrak{D}_m^2 + (\mathbf{x}_i - \mathbf{x}_j)^2 \tag{9}$$

where $\mathfrak{D}_m$ is the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ and is defined as,

$$\mathfrak{D}_m = \sqrt{\sum_{c=0}^{m-1} (\mathbf{x}_i - \mathbf{x}_j)^2} \tag{10}$$

If the one-dimensional time series is already properly embedded in m dimensions, then the distance $\mathfrak{D}$ between $\mathbf{x}_i$ and its nearest neighbor $\mathbf{x}_j$ should not appreciably change by some distance criterion $\mathfrak{D}_{tol}$ so that $\mathfrak{D} < \mathfrak{D}_{tol}$. Moreover, the distance of the nearest neighbor when embedded into the next higher dimension should be less than some criterion A_{tol} such that $\mathfrak{D}_{m+1} < A_{tol}$. According to Wallot and Mønster (2018), the next settings of the thresholds $\mathfrak{D}_{tol} \approx 10$ and $A_{tol} = 2$ are recommended. The procedure is repeated for the nearest neighbor of each coordinate vector until one of the stopping criteria is met. In particular, the optimal m is reached when (i) FNN drops to 0, or (ii) subsequent embeddings have the same number of false neighbors which implies that their difference is less than a threshold $T_{tol} = 2$, or (iii) the point before which the number of FNNs starts to increase again.

3 Results

This section initially describes the experimental evaluation for choosing the optimal set of parameters, m, τ and ε, of the herein proposed RQA parameter selection schemes employing a Gaussian-kernel Support Vector Machine (SVM) classifier as well as Fishers Linear Discriminant Analysis (FDA). Then the performance of each proposed architecture is compared to state-of-the-art in terms of overall classification accuracy, feature multitude and running time complexity. It is important to note that in this work the classification procedure is not considered for the measurement of computational complexity. All the experiments are implemented in MATLAB, on a desktop computer equipped with a CPU processor (Intel Core i5-4590) clocked at 3.30 GHz and 8 GB RAM.

3.1 RQA parameter selection

The goal of this section is to show that the GmdRQA is capable of *reducing* the number of features by half and *increasing* the computational efficiency of the system while achieving the same prediction accuracy. In particular, a parameter selection based on Grid Search (GS) is evaluated as it is the only technique suggested in the literature for the protein prediction problem. In addition to GS, two more case studies are proposed and evaluated: (i) the embedding dimension m and time delay τ parameters are found for each protein in a Data-Driven (DD) fashion when AMI and FNN are employed, and (ii) the Most Frequent (MF) m and τ values are computed as the product of a statistical analysis of the optimal set of parameters. These three parameter selection approaches are employed by RQA and GmdRQA, resulting in six different algorithms. Last but not least, HVG is also combined with each one of the six aforementioned algorithms. The general framework of the proposed protein structural class prediction architecture is depicted in Figure 3.

3.1.1 Grid search parameter tuning

The performance of GS-RQA and GS-GmdRQA schemes is initially evaluated when the range of embedding dimension m and time delay

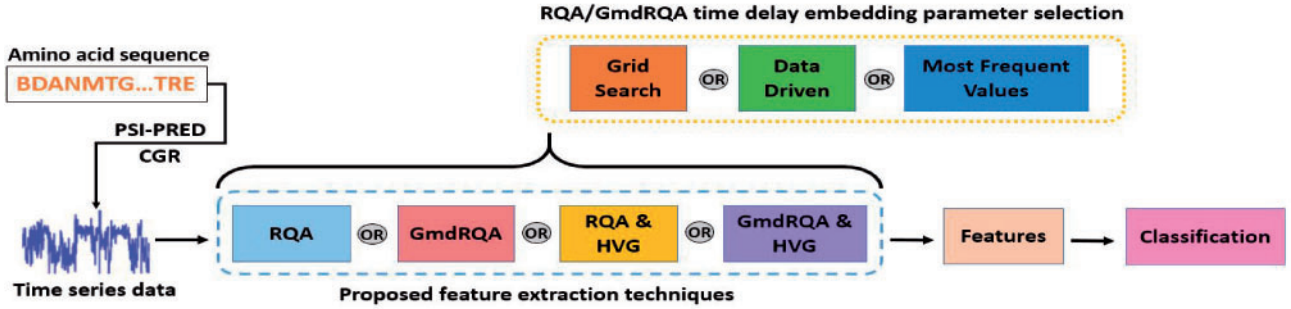

Fig. 3. General proposed RQA time delay embedding parameter selection scheme

τ varies between one and eight. In the case that both m and τ are equal to eight, the phase space cannot be constructed for small proteins, hence no results are reported and the grid search procedure is terminated. To assess the classification accuracy of GS-RQA and GS-GmdRGA, we generated 30 randomly shuffled datasets and for each dataset, the classification accuracy is reported for all combinations of m and τ for every individual shuffling. The pair that is reported to most frequently achieve the highest accuracy among all the different random data shuffles is employed as the optimal. Based on this procedure, the optimal parameters for GS-RQA are $m=4$ and $\tau=1$ when FDA is employed and $m=2$ and $\tau=4$ for SVM. On the other hand, for GS-GmdRQA, the optimal parameters for FDA are $m=2$ and $\tau=4$ while for SVM the optimal parameters are $m=5$ and $\tau=1$. The implementation of GS-RQA and GS-GmdRQA for the rest of this work uses the herein evaluated optimal parameters.

3.1.2 Data-driven parameter tuning

The following case study involves the data-driven fine-tuning time delay embedding parameter selection. In this scenario, the optimal set of parameters is evaluated in an automated fashion. Particularly, the time delay embedding parameters are evaluated per protein using the FNN and AMI algorithms. However, in this work, for proteins of length lower than 45 residues, time delay embedding parameters are set to be $\tau=1$ and $m=2$ that are the minimum values for both parameters. For the DD-RQA scheme, 20 features (10 features $\times$ 2 dimensions) are extracted in total, whereas for the DD-GmdRQA architecture, the number of features is reduced to 10 since the two-dimensional time series is processed concurrently.

3.1.3 Statistical parameter tuning

In the case of MF-RQA and MF-GmdRQA, the m and τ parameter values are selected as the most frequent values among the optimal sets of m and τ that derive when FNN and AMI are applied on each single protein included in the training phase to later avoid overfitting. Again, for proteins of length lower than 45 residues, the time delay embedding parameters are set as $\tau=1$ and $m=2$ that are the minimum values for both parameters. The main difference between GS-RQA/GS-GmdRQA and MF-RQA/MF-GmdRQA is that for the grid search process (GS-RQA/GS-GmdRQA), m and τ assume the same value for both time series dimensions, whereas for the most frequent parameter searching (MF-RQA/MF-GmdRQA), m and τ are evaluated separately per dimension. Along these lines, concerning MF-RQA the most frequent values are $(m,\tau)=(4,5)$ and $(m,\tau)=(2,5)$ for the t_x and t_y time series, respectively. On the other hand, in MF-GmdRQA, $(m,\tau)=(2,6)$ for both dimensions. For the rest of this work, the implementation of MF-RQA and MF-GmdRQA utilizes the aforementioned optimal parameters accordingly.

3.1.4 Neighborhood ε

Following the empirical rule proposed by Krämer et al. (2018) the neighborhood threshold ε can be chosen as a percentage of the distribution of all the pairwise distances that describe the phase space trajectory. Figure 4 provides the average classification accuracy as a function of the neighborhood threshold ε for both the RQA and GmdRQA schemes using the FDA and SVM classifiers. As depicted, the average classification accuracy of GS-RQA and GS-GmdRQA plateaus for ε values above 30% for both classifiers for all study cases. As the focus of this study is to evaluate the influence of time and delay embedding parameters on the proposed architectures we select the same neighborhood percentage in all three models. Therefore, since a higher neighborhood percentage increases the computational cost, $\varepsilon=30\%$ is selected as the minimum percentage that provides high classification results. The DD-RQA architecture performs its best when ε lies between 30% and 50%. Thus, $\varepsilon=30\%$ is chosen. In addition, for the case of DD-GmdRQA, it is clear that $\varepsilon=30\%$ provides the highest accuracy over all percentages for both SVM and FDA. Finally, the performance of MF-RQA and MF-GmdRQA for FDA and SVM is better when ε lies between 20–50% and 30–50%, respectively. Consequently, we select $\varepsilon=20\%$ for FDA and $\varepsilon=30\%$ for SVM.

3.2 Classification

The first step in the classification process is to perform a z-score normalization of the feature matrix. Previous studies (Jiang *et al.*, 2019; Olyaee *et al.*, 2016; Yang *et al.*, 2009, 2010) recommend the leave-one-out cross-validation process. However, splitting the data into 70–30% for training and testing respectively, yields nearly the same overall accuracy in considerably less time. Therefore, to reduce the running time complexity of the classification process, in this work the data are randomly split into 70–30% for training and testing and the procedure is repeated 150 times. Along these lines, a Gaussian-kernel SVM, a Linear-kernel SVM, as well as Fisher's Linear Discriminant Analysis (FDA) algorithm are applied separately on the normalized feature matrix for discriminating between the four structural classes all-α, all-β, α/β and $\alpha+\beta$. Concerning the Gaussian-kernel SVM classifier, the regularization parameter C and kernel width parameter γ can take all positive values log-scaled in the range $[10^{-3},10^{3}]$. Since each experiment is repeated 150 times, the average score of all the metrics mentioned above along with the respected standard deviation is reported.

Initially, the RQA and GmdRQA architectures are examined for each of the data-driven and the statistical time delay embedding parameter selection schemes (see Sections 3.1.1 and 3.1.2). Particularly, Tables 1 and 2 indicate the performance of the proposed DD-RQA, DD-GmdRQA, MF-RQA and MF-GmdRQA feature extraction frameworks in terms of sensitivity, specificity and Overall Accuracy (OA) for the Gaussian-kernel SVM and FDA, respectively. As shown, proteins that belong to the α-fold and β-fold class are better predicted in most test cases for both SVM and FDA classifiers. Moreover, it is given that MF-RQA outperforms MF-GmdRQA in addition to DD-RQA and DD-GmdRQA. This leads to the conclusion that a more generalized approximation of the time-delay embedding parameters enhances the system's ability to learn information-rich patterns that best capture the underlying data dynamics.

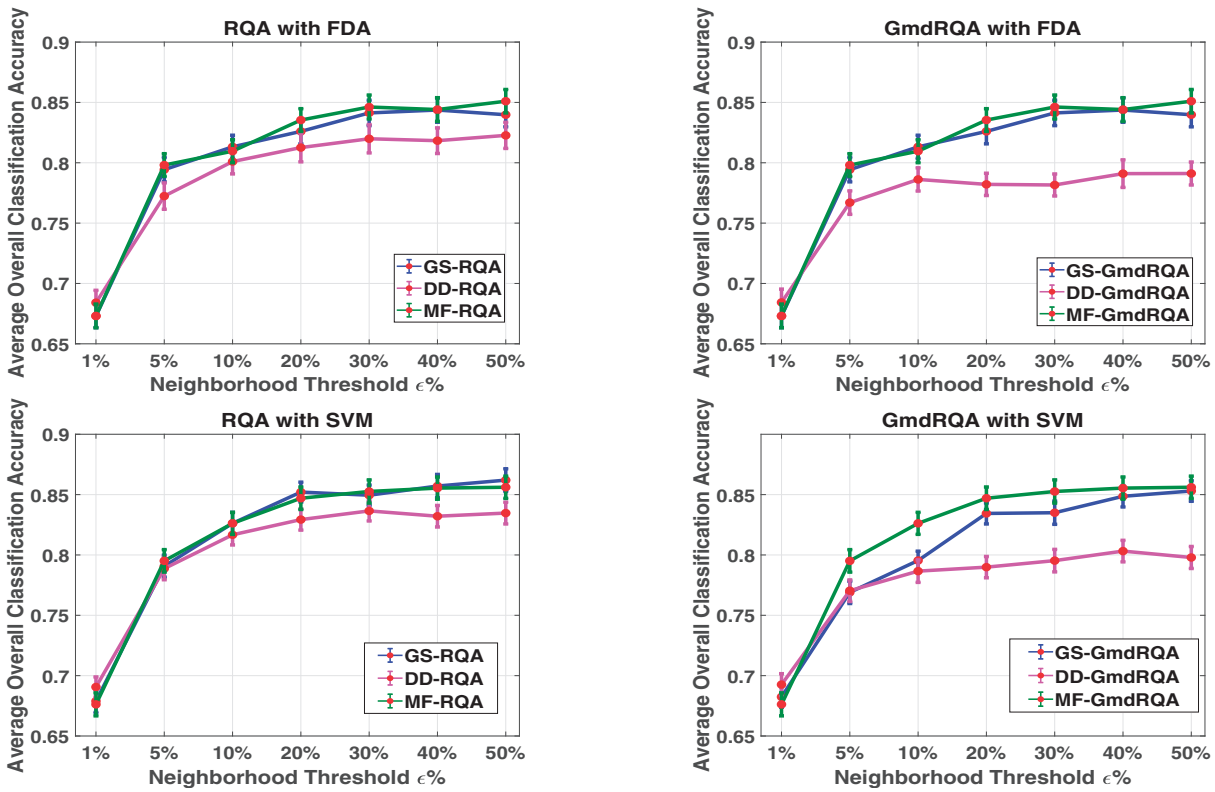

Fig. 4. Mean classification accuracy as function of the percentage of the neighborhood threshold ε for RQA and GmdRQA with FDA (Top Left, Top Right) and SVM (Lower Left, Lower Right)

Table 1. Performance evaluation of (i) the data-driven fine-tuned unidimensional RQA (DD-RQA), (ii) the data-driven fine-tuned generalized multidimensional RQA (DD-GmdRQA), (iii) the most frequent fine-tuned unidimensional RQA (MF-RQA) and (iv) the most frequent fine-tuned generalized multidimensional RQA (MF-GmdRQA) scheme using SVMs

SVM

	DD-RQA		DD-GmdRQA		MF-RQA		MF-GmdRQA	
	Sensitivity	Specificity	Sensitivity	Specificity	Sensitivity	Specificity	Sensitivity	Specificity
α	78.1 ± 3.1%	89 ± 1.7%	63.8 ± 4.2%	83.9 ± 2.2%	88.8 ± 2.6%	89.8 ± 1.6%	85.7 ± 2.8%	88.9 ± 1.6%
B	63 ± 4.1%	90 ± 1.7%	59.7 ± 4%	89.6 ± 1.9%	65.5 ± 4%	93.4 ± 1.5%	63.5 ± 3.8%	90.5 ± 1.9%
α/β	60.8 ± 4.6%	93.4 ± 1.1%	58.9 ± 4.8%	91.2 ± 1.4%	67.5 ± 4.3%	91.9 ± 1.5%	60.7 ± 4.3%	92.1 ± 1.5%
$\alpha+\beta$	65.4 ± 3.8%	83.6 ± 1.8%	53.8 ± 4.3%	80.4 ± 2.3%	60.9 ± 4.1%	85.8 ± 1.9%	55.5 ± 4.4%	83.7 ± 2%
OA	83.6 ± 0.8%		79.5 ± 0.8%		**85.4 ± 0.9%**		83.3 ± 0.9%	

Note: The optimal parameter set for (iii) and (iv) are evaluated in Section 3.1.3. We would like to keep the highlight as these cells indicate the methods and the metrics we used.

Table 2. Performance evaluation of (i) the data-driven fine-tuned unidimensional RQA (DD-RQA), (ii) the data-driven fine-tuned generalized multidimensional RQA (DD-GmdRQA), (iii) the most frequent fine-tuned unidimensional RQA (MF-RQA) and (iv) the most frequent fine-tuned generalized multidimensional RQA (MF-GmdRQA) scheme using Fisher's linear Discriminant Algorithm (FDA)

FDA

	DD-RQA		DD-GmdRQA		MF-RQA		MF-GmdRQA	
	Sensitivity	Specificity	Sensitivity	Specificity	Sensitivity	Specificity	Sensitivity	Specificity
α	73.2 ± 4%	87.1 ± 2.5%	42.2 ± 4.4%	93 ± 1.5%	85.2 ± 2.9%	92.8 ± 1.4%	78.3 ± 3.3%	93.2 ± 1.1%
B	53.8 ± 6.8%	92.6 ± 2.2%	63.1 ± 3.8%	85.7 ± 1.8%	55.3 ± 5.1%	96.2 ± 1.5%	64.8 ± 3.8%	91.3 ± 1.3%
α/β	60.9 ± 4.2%	91.9 ± 1.4%	55.7 ± 4.7%	90.9 ± 1.4%	64.7 ± 4.7%	91.9 ± 1.5%	59.6 ± 5%	92.4 ± 1.4%
$\alpha+\beta$	68.9 ± 4.1%	80.2 ± 2.6%	64.7 ± 4.1%	72 ± 2.6%	70.4 ± 3.9%	77.6 ± 2.7%	64.3 ± 4.1%	79 ± 2.4%
OA	82.1 ± 1%		78.2 ± 1%		**84.5 ± 0.9%**		83.5 ± 1%	

Note: The optimal parameter set for (iii) and (iv) are evaluated in Section 3.1.3. We would like to keep the highlight as these cells indicate the methods and the metrics we used.

Table 3. Summarized predicted quality results for the 25PDB dataset

	Method										Number of Features	Overall Accuracy			Computational Complexity (min)
	PSI-PRED	CGR	RQA			GmdRQA			HVG	Other		Gaussian-kernel SVM	Linear-kernel SVM	FDA	
			GS	DD	MF	GS	DD	MF							
Yang *et al.* (2009)		X	X								16	–	–	65.08	–
Yang *et al.* (2010)	X	X	X								16	–	–	81.40	–
Experiment 1	X	X	X								20	**86.20**	86.08	84.37	427.86
	X	X		X							20	83.60	83.76	82.14	**8.18**
	X	X			X						20	85.40	**86.17**	**84.52**	11.15
Experiment 2	X	X				X					10	**85.30**	**84.28**	**84.40**	232.80
	X	X					X				10	79.51	79.50	78.23	**4.38**
	X	X						X			10	83.32	83.98	83.54	8.33
Olyaee *et al.* (2016)	X	X	X							X	24	–	–	**90.08**	–
Jiang *et al.* (2019)	X	X	X						X	X	30	**95.33**	–	–	–
Experiment 3	X	X	X						X		37	86.16	87.29	85.92	430.76
	X	X		X					X		37	85.95	86.98	**86.69**	**11.08**
	X	X			X				X		37	**86.20**	**87.85**	86.25	14.06
Experiment 4	X	X				X			X		27	85.37	86.70	87.08	235.71
	X	X					X		X		27	84.39	**86.97**	86.30	7.28
	X	X						X	X		27	**86.27**	86.85	**87.23**	11.24

Note: The minimum computational time as well as the highest achieved accuracy for Gaussian-kernel SVM, Linear-kernel SVM and FDA per experiment, are indicated with bold letters. Blue highlight refers to the most efficient architecture in terms of classification accuracy and running time complexity per experiment. The classification process is excluded for the measurement of the computational complexity that refers only to the feature extraction procedure.

The scheme of RQA with grid search parameter selection has been widely employed in previous studies (Olyaee *et al.*, 2016; Yang *et al.*, 2009, 2010). However, none of the corresponding works report the optimal parameter set-up. Hence, in this work, a GS-RQA and a GS-GmdRQA framework are re-implemented. Along these lines, as depicted in Table 3, four different experiments involving the GS-RQA/GmdRQA, DD-RQA/GmdRQA, MF-RQA/GmdRQA and HVG are conducted. In more detail, Experiment 1 indicates that the GS-RQA framework with Gaussian-kernel SVM achieves the highest classification accuracy (86.20%). Furthermore, GS-RQA provides better results than those reported in the respective works of Yang *et al.* (2009) and Yang *et al.* (2010). Specifically, the main difference between our proposed GS-RQA and Yang *et al.* (2009) is the absence of PSI-PRED tool from their architecture. On the other hand, in the work of Yang *et al.* (2010) PSI-PRED is employed but the number of extracted features is 16, whereas we have extracted 20 features. Moreover, we cannot conclude on the effect of the grid search parameter values on accuracy results since Yang *et al.* (2010) do not report their grid search parameter values. Nevertheless, the performance of GS-RQA is comparable to MF-RQA with Linear-kernel SVM, which reaches an overall classification accuracy of 86.17%. It is clear that GS-RQA improves only slightly the overall classification accuracy (86.20% versus 86.17%) but with the cost of an extremely high running time (427.86 versus 11.15 min). Thus, we conclude that MF-RQA is the most efficient architecture in terms of running time complexity with comparable classification accuracy in Experiment 1. Finally, it is also worth mentioning that the best algorithm in terms of computational complexity is MF-RQA. Thereafter, GmdRQA is examined in Experiment 2. As presented, GmdRQA utilizes the minimum number of features that has been so far proposed in the literature (Jiang *et al.*, 2019; Olyaee *et al.*, 2016; Yang *et al.*, 2009, 2010), namely 10, realizing a maximum overall accuracy of 85.30% when GS-GmdRQA with Gaussian-kernel SVM classifier is employed. However, we note that the performance of this scheme comes again with the cost of high computational complexity. Thus, MF-GmdRQA is highlighted as the best trade-off in terms of classification accuracy (83.98%) for the Linear-kernel SVM classifier and running time efficiency. The most important outcome of the previous two experiments though is that the overall classification accuracy is similar for the RQA and GmdRQA schemes with the later reducing the computational cost approximately to half, in most cases. Therefore, Experiments 1 and 2 demonstrate that MF-GmdRQA is the most efficient scheme as it yields a high overall accuracy that exceeds the performance of the respective works of Yang *et al.* (2009) and Yang *et al.* (2010), utilizing the minimum number of features that has been reported so far considering the RQA framework (Jiang *et al.*, 2019; Olyaee *et al.*, 2016; Yang *et al.*, 2009, 2010). Thus MF-GmdRQA not only reduces time complexity due to its automated parameter estimation, but also improves the memory complexity.

Considering the works of Olyaee *et al.* (2016) and Jiang *et al.* (2019), RQA is combined with other feature extraction and time series analysis techniques that are not examined in the herein work. Particularly, Olyaee *et al.* (2016) combine RQA and Complex Networks whereas Xu et al. (2017) combined the multiscale coarse-grained RQA with HVG. In our work, HVG is also employed and combined with all the aforementioned RQA and GmdRQA schemes in Experiments 3 and 4. The main benefit of HVG against RQA is the absence of hyper-parameter tuning that results in low running time complexity. In particular, our implementation requires 2.9 min to reproduce the HVG framework proposed by Zhao *et al.* (2018). As depicted in Experiment 3, the combination of HVG with GS-RQA, DD-RQA and MF-RQA slightly improves the overall classification accuracy with the HVG and DD-RQA being the optimal scheme in terms of classification accuracy (86.69%) and time and memory complexity. The performance of the combination of HVG and GmdRQA frameworks is then presented in Experiment 4. The combination of HVG and DD-GmdRQA achieves similar overall classification accuracy with the rest architectures, however, it is more favorable in terms of computational complexity.

Compared to the best performing scheme of this work, i.e. DD-GmdRQA and HVG, the work of Olyaee *et al.* (2016) achieves higher overall accuracy, namely 90.08%, utilizing 24 features that derive from the combination of RQA and Complex Networks. However, the tuning of the RQA hyperparameters is performed in a grid search manner which is extremely time consuming based on the experiments presented in this work for GS-RQA. The same assertion can be also made for the work of Jiang *et al.* (2019) although the

Table 4. Comparison with the recent state-of-the-art for the protein structural class prediction problem for the 25PDB dataset

	Number of features	Overall accuracy (%)
Apurva and Mazumdar (2020)	65	**91.6**
Liu et al. (2015)	224	90.3
Dehzangi et al. (2013)	200	90.1
Liang *et al.* (2015)	224	88.4
This work	**27**	86.9
Ding *et al.* (2014)	36	85.8
Yuan *et al.* (2018)	200	84.2

Note: The best performed method along with the method with the less features are highlighted.

stage of parameter selection and definition is not stated in their work. Specifically, they utilize multiscale coarse-grained RQA (Xu *et al.*, 2017) along with HVG and achieve an overall accuracy of 95.33%. However, multiscale coarse-grained RQA considers the spatial proximity of the phase space of time series adding an extra hyperparameter to the system and hence further escalates the time complexity. In addition, the work of Jiang *et al.* (2019) implements an extremely time consuming classification procedure. In particular, they perform a leave-one-out cross-validation using a Gaussian-kernel SVM classifier with a grid search parameter selection of the regularization parameter C and kernel width parameter γ that can take all positive values log-scaled in the range $[2^{-10}, 2^{10}]$. On the contrary, the implementation of FDA in our work is performed in an automated fashion and requires merely 2.98 s, whereas our proposed Gaussian-kernel SVM classification scheme requires 24.87 s for the exact same setup described in Section 3.2.

Finally, DD-GmdRQA and HVG architecture is compared against the recent state-of-the-art architectures for protein structural class prediction for the 25PDB dataset. As presented, our DD-GmdRQA and HVG scheme requires the minimum number of features to achieve an overall classification accuracy of 86.9% that is quite similar to the work of Liang *et al.* (2015), Ding et al. (2014) and Yuan et al. (2018) that achieve 88.4%, 85.8% and 84.2% respectively. The rest approaches reach a higher overall classification accuracy, however, on the cost of high feature multitude and complex feature extraction techniques that escalate the overall computational complexity (Table 4).

4 Conclusion

In this work, we designed and implemented novel data-driven protein structural class prediction architectures based on the representation of secondary structure data in higher-dimensional phase spaces using RQA, GmdRQA and a combination of HVG with the two aforementioned techniques. In particular, the herein proposed work addresses the problem of efficient time delay embedding parameter selection which has a significant positive impact on the overall classification accuracy and the running time complexity. Four efficient data-driven evaluation approaches are suggested, namely DD-RQA, DD-GmdRQA, MF-RQA and MF-GmdRQA. The experimental evaluation on real data revealed the superiority of the HVG & DD-GmdRQA-based framework in extracting and exploiting the underlying temporal dynamics of the data generating processes, resulting in lower time and memory complexity in terms of feature multitude, when compared against the state-of-the-art RQA frameworks. Moreover, the herein presented frameworks are applicable to any protein type since they only rely on the dynamical behavior of the given system. An extension of this work will consider a GmdRQA-HVG framework that utilizes the observed secondary structure and later a framework that will process directly on the primary amino acid protein sequence without employing the secondary structure information. Lastly, a future goal is to predict the three-dimensional structure of the proteins with deep learning algorithms, exploiting the graphs exported with CGR.

Funding

This research work was supported by the Hellenic Foundation for Research and Innovation (HFRI) and the General Secretariat for Research and Technology (GSRT), under HFRI faculty [1725] and by the Stavros Niarchos Foundation within the framework of the project ARCHERS.

Conflict of Interest: none declared.

References

Almeida,J.S. *et al.* (2001) Analysis of genomic sequences by chaos game representation. *Bioinformatics*, **17**, 429–437.
Apurva,M. and Mazumdar,H. (2020) Predicting structural class for protein sequences of 40% identity based on features of primary and secondary structure using random forest algorithm. *Comput. Biol. Chem.*, **84**, 107164.
Chou,K.-C. (2001) Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins Struct. Funct. Bioinf.*, **43**, 246–255.
Dehzangi,A. *et al.* (2013) Exploring potential discriminatory information embedded in pssm to enhance protein structural class prediction accuracy. In: *IAPR International Conference on Pattern Recognition in Bioinformatics*, pp. 208–219. Springer, Nice, France.
Ding,S. *et al.* (2014) A protein structural classes prediction method based on predicted secondary structure and psi-blast profile. *Biochimie*, **97**, 60–65.
Eckmann,J. *et al.* (1987) Recurrence plots of dynamical systems. *Europh. Lett.*, **4**, 973–977.
Fraser,A.M. and Swinney,H.L. (1986) Independent coordinates for strange attractors from mutual information. *Phys. Rev. A*, **33**, 1134–1140.
Guharoy,M. and Chakrabarti,P. (2007) Secondary structure based analysis and classification of biological interfaces: identification of binding motifs in protein–protein interactions. *Bioinformatics*, **23**, 1909–1918.
Jeffrey,H.J. (1990) Chaos game representation of gene structure. *Nucleic Acids Res.*, **18**, 2163–2170.
Jiang,H. *et al.* (2019) Protein tertiary structure prediction based on multiscale recurrence quantification analysis and horizontal visibility graph. In: *International Symposium on Neural Networks*, pp. 531–539. Springer, Moscow, Russia.
Jones,D.T. (1999) Protein secondary structure prediction based on position--specific scoring matrices. *J. Mol. Biol.*, **292**, 195–202.
Kantz,H. and Schreiber,T. (2004) *Nonlinear Time Series Analysis*, Vol. 7. Cambridge University Press.
Kennel,M.B. *et al.* (1992) Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Phys. Rev. A*, **45**, 3403–3411.
Krämer,K.H. *et al.* (2018) Dimension-scalable recurrence threshold estimation. *arXiv preprint arXiv:1802.01605*.
Lacasa,L. *et al.* (2008) From time series to complex networks: the visibility graph. *Proc. Natl. Acad. Sci. USA*, **105**, 4972–4975.
Levitt,M. and Chothia,C. (1976) Structural patterns in globular proteins. *Nature*, **261**, 552–558.
Liang,Y. *et al.* (2015) Prediction of protein structural classes for low-similarity sequences based on consensus sequence and segmented PSSM. *Comput. Math. Methods Med.*, **2015**, 1–9.
Liu,T. and Jia,C. (2010) A high-accuracy protein structural class prediction algorithm using predicted secondary structural information. *J. Theor. Biol.*, **267**, 272–275.
Liu,T. *et al.* (2010) Prediction of protein structural class for low-similarity sequences using support vector machine and psi-blast profile. *Biochimie*, **92**, 1330–1334.
Liu,T. *et al.* (2015) Prediction of protein structural class based on gapped-dipeptides and a recursive feature selection approach. *Int. J. Mol. Sci.*, **17**, 15.
Löchel,H.F. *et al.* (2020) Deep learning on chaos game representation for proteins. *Bioinformatics*, **36**, 272–279.
Marwan,N. *et al.* (2007) Recurrence plots for the analysis of complex systems. *Phys. Rep.*, **438**, 237–329.
Nakashima,H. *et al.* (1986) The folding type of a protein is relevant to the amino acid composition. *J. Biochem.*, **99**, 153–162.
Noble,M.E. *et al.* (2004) Protein kinase inhibitors: insights into drug design from structure. *Science*, **303**, 1800–1805.
Olyaee,M.H. *et al.* (2016) Predicting protein structural classes based on complex networks and recurrence analysis. *J. Theor. Biol.*, **404**, 375–382.

Orengo,C.A. *et al.* (1997) Cath—a hierarchic classification of protein domain structures. *Structure*, **5**, 1093–1109.

Spänig,S. and Heider,D. (2019) Encodings and models for antimicrobial peptide classification for multi-resistant pathogens. *BioData Min.*, **12**, 1–29.

Wallot,S. and Mønster,D. (2018) Calculation of average mutual information (AMI) and false-nearest neighbors (FNN) for the estimation of embedding parameters of multidimensional time series in matlab. *Front. Psychol.*, **9**, 1679.

Wallot,S. *et al.* (2016) Multidimensional recurrence quantification analysis (MDRQA) for the analysis of multidimensional time-series: a software implementation in matlab and its application to group-level data in joint action. *Front. Psychol.*, **7**, 1835.

Wang,J. *et al.* (2015) Prediction of protein structural classes for low-similarity sequences using reduced PSSM and position-based secondary structural features. *Gene*, **554**, 241–248.

Xu,M. *et al.* (2017) Multiscale recurrence quantification analysis of order recurrence plots. *Physica A*, **469**, 381–389.

Yang,J.-Y. *et al.* (2008) Protein structure classification based on chaos game representation and multifractal analysis. In: *2008 International Conf. on Natural Computation*, Vol. **4**. IEEE, Jinan, Shandong, China, pp. 665–669.

Yang,J.-Y. *et al.* (2009) Prediction of protein structural classes by recurrence quantification analysis based on chaos game representation. *J. Theor. Biol.*, **257**, 618–626.

Yang,J.-Y. *et al.* (2010) Prediction of protein structural classes for low-homology sequences based on predicted secondary structure. *BMC Bioinformatics*, **11**, S9.

Yu,B. *et al.* (2017) Prediction of protein structural class for low-similarity sequences using chou's pseudo amino acid composition and wavelet denoising. *J. Mol. Graph. Modell.*, **76**, 260–273.

Yuan,M. *et al.* (2018) A novel feature selection method to predict protein structural class. *Comput. Biol. Chem.*, **76**, 118–129.

Zbilut,J. and Webber,C. (1992) Embeddings and delays as derived from quantification of recurrence plots. *Phys. Lett. A*, **171**, 199–203.

Zervou,M.A. *et al.* (2019) Automated screening of dyslexia via dynamical recurrence analysis of wearable sensor data. In *2019 IEEE International Conf. on Bioinformatics and Bioengineering (BIBE)*, pp. 770–774. IEEE, Athens, Greece.

Zhang,S. *et al.* (2012) Using principal component analysis and support vector machine to predict protein structural class for low-similarity sequences via PSSM. *J. Biomol. Struct. Dyn.*, **29**, 1138–1146.

Zhao,Z.-Q. *et al.* (2018) Low-homology protein structural class prediction from secondary structure based on visibility and horizontal visibility network. *Am. J. Biochem. Biotechnol.*, **14**, 67–75.

Zhu,X.-J. *et al.* (2019) Predicting protein structural classes for low-similarity sequences by evaluating different features. *Knowledge Based Syst.*, **163**, 787–793.