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

Michaela Areti Zervou[a,b], Effrosyni Doutsi[b,*], Pavlos Pavlidis[b], Panagiotis Tsakalides[a,b]

[a]*Department of Computer Science, University of Crete, Heraklion, Greece*
[b]*Institute of Computer Science, Foundation for Research and Technology-Hellas, Heraklion, Greece*

Abstract

Protein structure 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 an efficient and accurate computational 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 (CGR) along with time series analysis tools such as recurrence quantification analysis (RQA), 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 multidimensional time series and reduce the number of features. In addition, two recently proposed data-driven algorithms, namely average mutual information (AMI) and false nearest neighbors (FNN), are utilized to define in a fast yet precise manner the optimal GmdRQA parameters. Finally yet importantly, the classification accuracy is improved by the combination of the aforementioned 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.

Keywords: Protein structure prediction, chaos game representation, nonlinear time series analysis, recurrence plots, horizontal visibility graphs, multi-class classification.

1. Introduction

Protein structure 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 [1], especially for drug design, enzymes composition and interpretation of disease related phenotypes. Proteins function properly when they adopt their final and stable shape which is also known as tertiary structure or fold. According to the Anfinsen's dogma the protein folding is mainly determined by its amino acid sequence [2]. 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 [3].

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 acid composition [4], the pseudo-amino acid composition [5], the dipeptide and tripeptide compositions [6], the PSI-BLAST profile [7], and the predicted secondary structure information [8], among others. 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 [9, 10, 11, 12, 13, 14, 15]. Numerous studies [16, 17, 18, 19, 18] 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) [20], Horizontal Visibility Graphs (HVG) [21] or a combination of both, extracting meaningful information from the data. Specifically, in order to achieve such a conversion, 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 [22] tool. Then, by employing the Chaos Game Representation (CGR) [23] technique, the updated 3-state sequence, that is of the same length as the original amino acid sequence, is converted into a graphical form where the x- and y-coordinates of each point on the graph are considered as two

*Corresponding author
Email address: edoutsi@ics.forth.gr, doutsi_efrosini@hotmail.com (Effrosyni Doutsi)

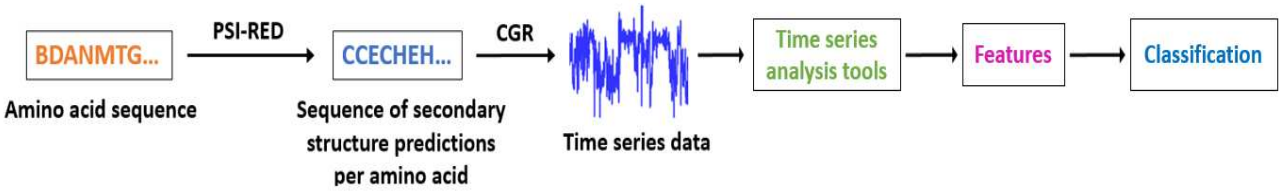


Figure 1: General protein structure prediction scheme.

individual time series. The aforementioned procedure is briefly depicted in Fig.1.

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 introduces (i) the utilization of the Generalized multidimensional Recurrence Quantification Analysis (GmdRQA) [24] as a sophisticated tool for a non-linear analysis of the multidimensional time series data, and (ii) a data-driven estimation of the RQA parameters employing the Average Mutual Information (AMI) [25] and the False Nearest Neighbors (FNN) [26] 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. Particularly, in this work, the amino acid sequence of a protein is transformed through the CGR procedure into a two-dimensional time series which are processed concurrently in order to exploit the correlations between the x- and y- coordinates that both describe the same protein. The contributions of this paper 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 instead of each time series separably.

(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 Generalized multidimensional RQA (DD-GmdRQA) frameworks that enable the discovery of representative patterns, increasing the overall accuracy of the protein structure prediction.

(iv) The significant reduction of the computational complexity of both the feature selection and the classification processes.

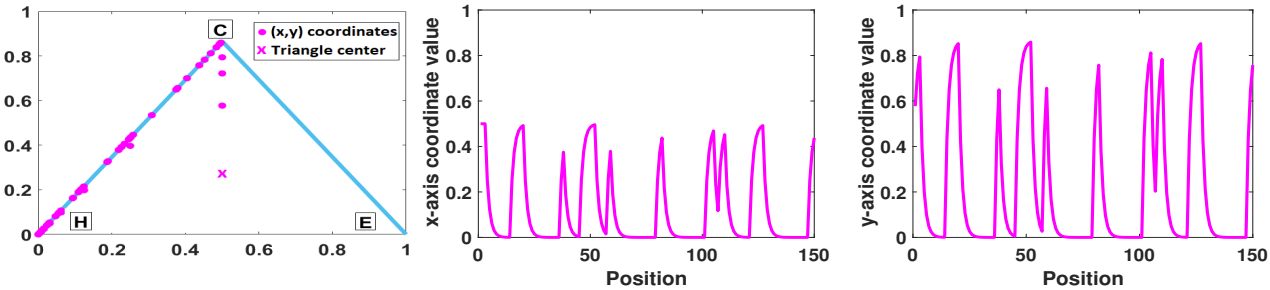
The rest of the paper is organized as follows: Section 2 is a background on the most notable methods that have been used in the literature for the generation of time series from an amino acid sequence and the quantitative time series analysis. Section 3 gives details on the proposed GmdRQA architecture that consists of two data-driven parameter-tuning algorithms, namely AMI and FNN. Section 4 introduces the proposed time delay embedding parameter selection architectures as well as the classification procedure. Moreover, the performance of the proposed architectures is evaluated in terms of classification accuracy, time complexity and feature multitude. Finally, section 5 draws the conclusion of this work and gives directions for future extensions.

2. Related Work

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 structure 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. Last but not least, this section summarizes the performance of state-of-the-art ML-based architectures which have adopted the aforementioned mechanisms for protein structure prediction.

2.1. PSI-blast based secondary structure PREDiction

Each protein is a long chain called polypeptide or polymer, which is formed when several monomers, known as amino acids, are joined together. Only 20 amino acids are known as proteinogenic, meaning they participate in the synthesis of a protein primary structure. In the literature, there are several works [17, 18, 19] 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. 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.



(a) Chaos Game Representation (b) Time series $t^{(x)}$ (c) Time series $t^{(y)}$

Figure 2: (a) Chaos Game Representation of protein's 1ASH predicted secondary structure ('$CCCHH\ldots HHCCC$') 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. (b) This time series represent the x-coordinates of the points. (c) This time series represent the x-coordinates of the points.

2.2. 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) is employed. CGR was first proposed as a scale-independent representation of genomics [23]. 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 Fig. 2a 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 centre 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,

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

where $v_i^{(x)}$ and $v_i^{(y)}$ are respectively the x and y coordinates of the vertex corresponding to the ith secondary structure element of a protein consisting of N amino-acids.

Finally, as depicted in Fig. 2b and Fig. 2c, 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 recurrence quantification analysis (RQA) [20] 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,

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

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 [27]. 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 [28] which are described in Appendix A.

2.3.2. Horizontal Visibility Graph

In recent years, complex network theory has been popularized in the analysis of biological problems [29]. A simple and fast computational method, known as horizontal visibility graph (HVG) [21], 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,

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

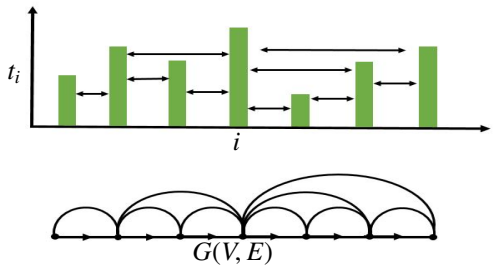


Figure 3: Horizontal Visibility Graph (HVG). Black dots refer to each $t(i)$ value represented with a green bar.

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 as presented in Fig 3. In this study, the resulted horizontal visibility graph $G = (V, E)$, $N = |V|$, $M = |E|$, 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 in Appendix B that are later employed for classification purposes.

2.4. State-of-the-art Architectures

The last decade has witnessed the introduction of RQA to the protein structure prediction problem. Yang *et al.* [16] first utilized RQA in their work to solve the problems caused by different protein sequence lengths. Specifically, they first transformed a protein into a time series via the CGR mechanism. It is important to mention that for each protein the CGR yields two time series and therefore, the procedure of the phase-space reconstruction and the generation of the RP is performed for each of the time series $t^{(x)}$ and $t^{(y)}$ independently. Then, for a given protein sequence, a total of 16 characteristic parameters is calculated with RQA. Last but not least, the Fisher's linear discriminant algorithm (FDA) is employed along with a leave-one-out cross-validation in order to predict the structural class for each protein.

Thereafter, a succeeding work of Yang *et al.*. [17] employs the PSI-PRED tool to predict the secondary structure for each protein sequence. Then, the CGR is exploited to represent the predicted secondary structure as two time series, from which a comprehensive set of 24 features is generated using RQA, K-string based information entropy, and segment-based analysis. Lastly, the FDA classifier is applied for the prediction of protein structural classes employing leave-one-out cross-validation. Later, Olyaee *et al.* [18] also employed PSI-PRED to predict secondary structure sequences which are then mapped into two time series by the CGR. Using RQA and the complex network theory on the respected time series, a total of 24 characteristic features can be calculated which are fed into the FDA classifier with a leave-one-out cross-validation.

Thenceforth, in the work of Zhao *et al.* [29] PSI-PRED is utilized as well to predict the secondary structure of proteins. The CGR is then applied on the predicted secondary structure and the two obtained time series construct a visibility and a horizontal visibility network, respectively, to generate a set of 17 features. Eventually, each protein structural class is predicted using Support Vector Machines (SVM) and FDA with a leave-one-out cross-validation. Finally, the work of Jiang *et al.* [19] proposed a combination of the multi-scale coarse-grained RQA and the HVG. In his work, PSI-PRED and CGR are employed as well, a multi-scale coarse granulation is applied on the time series, followed by the RQA and the HVG to generate the 30-D feature vector which is fed to an SVM classifier with a leave-one-out cross-validation.

3. Materials and Methods

This section introduces the dataset employed in this study and describes in detail the generalized multidimensional RQA (GmdRQA) along with the data-driven parameter estimation scheme that employs AMI and FNN.

3.1. Dataset Description

This work employs the 25PDB dataset [30] 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 α+β 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 categorised 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.

3.2. Generalized Multidimensional Recurrence Quantification Analysis

Multidimensional recurrence quantification analysis extracts the underlying dynamics of the system by mapping the time series in a higher dimensional phase space of trajectories by constructing state vectors u_i via time delay embedding. The generalized multidimensional recurrence quantification analysis (GmdRQA) framework transforms state vectors u_i into state matrices X_i to represent the time-delay embedding [24]. 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 by

$$\mathbf{GmdR}_{i,j} = \Theta\left(\varepsilon - \|\mathbf{X}_i - \mathbf{X}_j\|\right), \tag{8}$$

where ε is a threshold, $\|\cdot\|$ is the Frobenius norm, $\mathbf{X}_i$, $\mathbf{X}_j$ are the state matrices and $\Theta(\cdot)$ is the Heaviside step function whose discrete form is defined by (4). Having constructed the corresponding recurrence plot of the multidimensional system, the features described in Appendix A can also be employed.

3.3. 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.

3.3.1. Average Mutual Information Algorithm

The Average Mutual Information (AMI) [25] is a measure of nonlinear 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,

$$I(t_i, t_{i+\tau}) = \sum_{i,j} p_{jk}(\tau) \log\left(\frac{p_{jk}(\tau)}{p_j p_k}\right) \tag{9}$$

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 in [25], 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, [31] 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.

3.3.2. False Nearest Neighbor Algorithm

The embedding dimension m is an estimate of the dimensionality of the dynamics of the time series. The False Nearest Neighbor (FNN) [26] 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 neighbouring 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,

$$\mathfrak{D}_{m+1}^2 = \mathfrak{D}_m^2 + (\mathbf{x}_i - \mathbf{x}_j)^2 \tag{10}$$

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

$$\mathfrak{D}_m = \sqrt{\sum_{c=0}^{m-1} (\mathbf{x}_i - \mathbf{x}_j)^2} \tag{11}$$

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 [32] 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

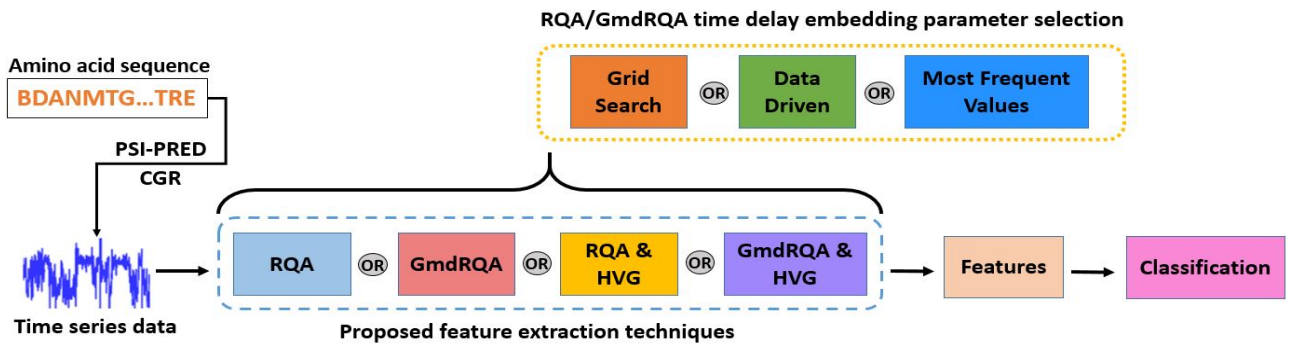


Figure 4: General proposed RQA time delay embedding parameter selection scheme.

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.

4. 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. 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 3.30GHz, and a 8 GB RAM.

4.1. 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 6 different algorithms. Last but not least, HVG is also combined with each one of the 6 aforementioned algorithms. The general framework of the proposed protein structure prediction architecture is depicted in Fig. 4.

4.1.1. 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 τ varies between 1 and 8. In the case that both m and τ are equal to 8, 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 10 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 SVM is employed and $m = 8$ and $\tau = 1$ for FDA. On the other hand, for GS-GmdRQA, the optimal parameters for SVM are $m = 8$ and $\tau = 1$ while for FDA the optimal parameters are $m = 7$ and $\tau = 1$. The implementation of GS-RQA and GS-GmdRQA for the rest of this work uses the herein evaluated optimal parameters.

4.1.2. 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×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.

4.1.3. 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

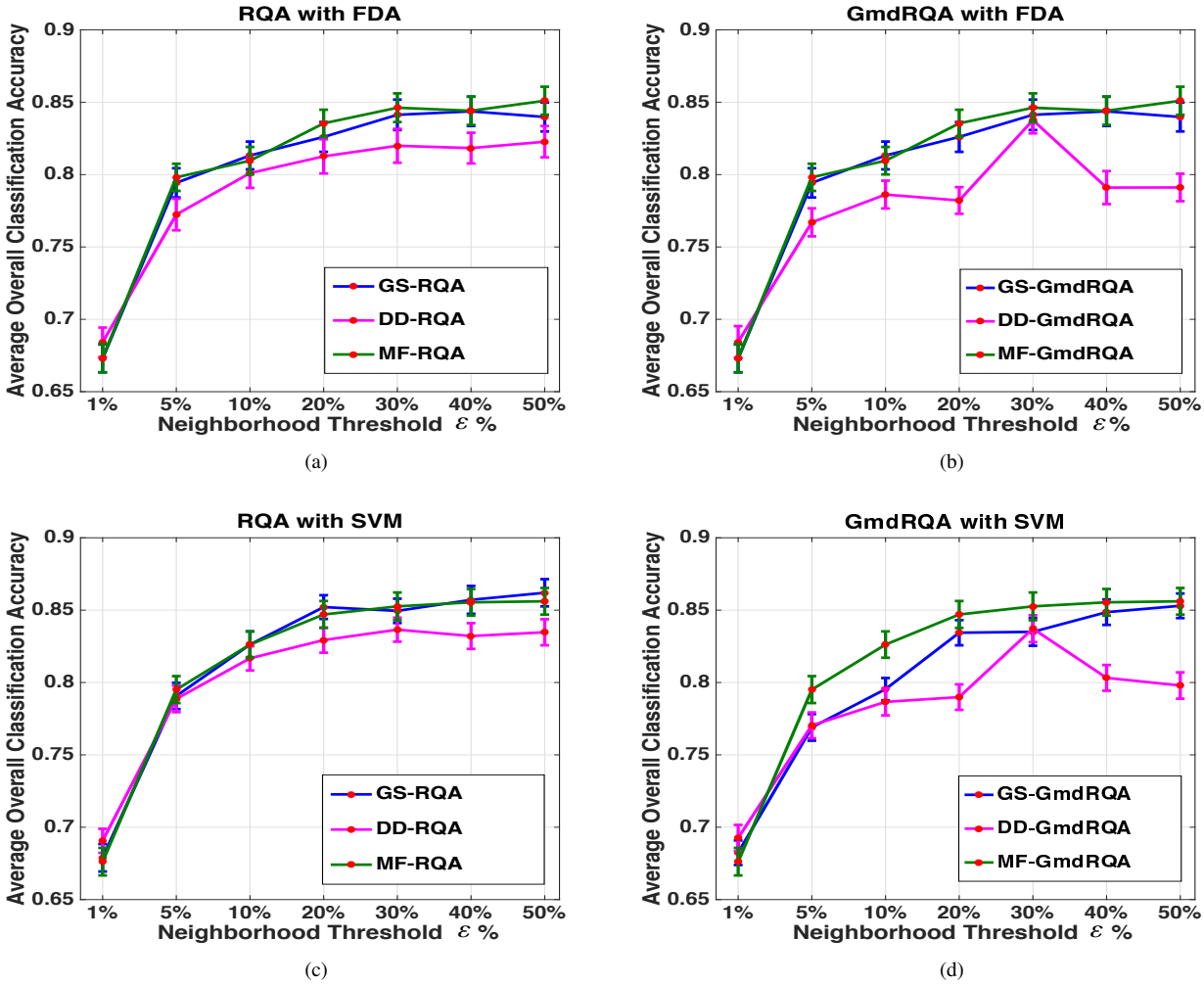


Figure 5: Mean classification accuracy as function of the percentage of the neighborhood threshold ε for RQA and GmdRQA with (a), (b) FDA and (c), (d) SVM.

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.

4.2. Neighborhood ε

Following the empirical rule proposed in [33] the neighborhood threshold ε can be chosen as a percentage of the distribution of all the pairwise distances that describe the phase space trajectory. Fig. 5 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%-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.

4.3. Classification

The first step in the classification process is to perform a z-score normalization of the feature matrix. Previous studies [16, 17, 18, 19] recommend the leave-one-out cross-validation process. However, splitting the data into 70%-30% for training

SVM								
	DD-RQA		**DD-GmdRQA**		**MF-RQA**		**MF-GmdRQA**	
	Sensitivity	**Specificity**	**Sensitivity**	**Specificity**	**Sensitivity**	**Specificity**	**Sensitivity**	**Specificity**
α	78.7 ± 3.7%	88.4 ± 1.8%	76.4 ± 4.1%	82.5 ± 2.2%	88.8 ± 2.6%	89.8 ± 1.6%	85.7 ± 2.8%	88.9 ± 1.6%
β	62.5 ± 4.1%	90.2 ± 1.7%	65.2 ± 3.9%	89.1 ± 1.7%	65.5 ± 4%	93.4 ± 1.5%	63.5 ± 3.8%	90.5 ± 1.9%
α/β	59.7 ± 4.6%	93.7 ± 1.2%	67.8 ± 4.6%	90.5 ± 1.6%	67.5 ± 4.3%	91.9 ± 1.5%	60.7 ± 4.3%	92.1 ± 1.5%
$\alpha+\beta$	64.9 ± 4.5%	83.1 ± 2.1%	61.1 ± 3.8%	94.6 ± 1.7%	60.9 ± 4.1%	85.8 ± 1.9%	55.5 ± 4.4%	83.7 ± 2%
OA	83.4 ± 0.9%		83.8 ± 0.9%		**85.4 ± 0.9%**		83.3 ± 0.9%	

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 Support Vector Machines (SVM). The optimal parameter set for (iii) and (iv) are evaluated in Section 4.1.3.

FDA								
	DD-RQA		**DD-GmdRQA**		**MF-RQA**		**MF-GmdRQA**	
	Sensitivity	**Specificity**	**Sensitivity**	**Specificity**	**Sensitivity**	**Specificity**	**Sensitivity**	**Specificity**
α	74.9 ± 4.7%	87.1 ± 2.8%	65.5 ± 4.2%	88.4 ± 1.9%	85.2 ± 2.9%	92.8 ± 1.4%	78.3 ± 3.3%	93.2 ± 1.1%
β	49.9 ± 9.7%	93.6 ± 2.5%	65.5 ± 4.2%	88.4 ± 1.9%	55.3 ± 5.1%	96.2 ± 1.5%	64.8 ± 3.8%	91.3 ± 1.3%
α/β	63.2 ± 4.6%	91.2 ± 1.5%	73.9 ± 4.6%	85.9 ± 1.9%	64.7 ± 4.7%	91.9 ± 1.5%	59.6 ± 5%	92.4 ± 1.4%
$\alpha+\beta$	68.5 ± 4.1%	80.1 ± 2.4%	57.7 ± 3.6%	100%	70.4 ± 3.9%	77.6 ± 2.7%	64.3 ± 4.1%	79 ± 2.4%
OA	82.1 ± 1.2%		83.6 ± 0.8%		**84.5 ± 0.9%**		83.5 ± 1%	

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), (ii) 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). The optimal parameter set for (iii) and (iv) are evaluated in Section 4.1.3.

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 Support Vector Machine (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 SVM classifier, the regularization parameter C and kernel width parameter γ can take all positive values log-scaled in the range $[10^{-3}, 10^{3}]$ [34]. Finally, the following metrics are utilized to evaluate the classification performance:

- Sensitivity, also known as the true positive rate, measures the proportion of actual positives that are correctly identified as such,

$$Sensitivity = \frac{\text{TP}}{\text{TP+FN}}$$

- Specificity, also known as the true negative rate, measures the proportion of actual negatives that are correctly identified as such,

$$Specificity = \frac{\text{TN}}{\text{TN+FP}}$$

- Overall accuracy is the most intuitive and frequently used performance parameter indicating the correctly classified samples against all samples,

$$Overall\ \ Accuracy = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}}$$

where TP indicates the correctly labeled positive samples by the classifier. FN indicates the samples incorrectly labeled as negative by the classifier. FP indicates the samples incorrectly labeled as positive by the classifier and TN indicates the correctly labeled negative samples by the classifier.

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 4.1.1 and 4.1.2). 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 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.

The scheme of RQA with grid search parameter selection has been widely employed in previous studies [16, 17, 18]. 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

	Method										Number of Features	Overall Accuracy		Computational Complexity (minutes)
	PSI-PRED	CGR	RQA			GmdRQA			HVG	Other				
			GS	DD	MF	GS	DD	MF				SVM	FDA	
Yang *et al.* [16]		X	X								16	-	65.08	-
Yang *et al.* [17]	X	X	X								16	-	81.40	-
Experiment 1	X	X	X								20	**86.20**	84.37	427.86
	X	X		X							20	83.69	82.18	**8.18**
	X	X			X						20	85.40	**84.52**	11.15
Experiment 2	X	X				X					10	**85.30**	**84.40**	232.80
	X	X					X				10	83.88	83.60	**4.38**
	X	X						X			10	83.32	83.54	8.33
Olyaee *et al.* [18]	X	X	X							X	24	-	**90.08**	-
Jiang *et al.* [19]	X	X	X						X	X	30	**95.33**	-	-
Experiment 3	X	X	X						X		37	**86.57**	86.29	430.76
	X	X		X					X		37	85.92	86.38	**11.08**
	X	X			X				X		37	85.64	**86.62**	14.06
Experiment 4	X	X				X			X		27	85.93	86.98	235.71
	X	X					X		X		27	**89.00**	**89.19**	7.28
	X	X						X	X		27	85.63	87.64	11.24

Table 3: Summarized predicted quality results for the 25PDB dataset. The minimum computational time as well as the highest achieved accuracy for 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.

and HVG are conducted. In more detail, Experiment 1 indicates that the GS-RQA framework with 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.*[16, 17]. Specifically, the main difference between our proposed GS-RQA and [16] is the absence of PSI-PRED tool from their architecture. On the other hand, in [17] 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.* [17] do not report their grid search parameter values. Nevertheless, the performance of GS-RQA is comparable to MF-RQA with SVM, which reaches an overall classification accuracy of 85.40%. It is clear that GS-RQA improves only slightly the overall classification accuracy (86.20% vs 85.40%) but with the cost of an extremely high running time (427.86 minutes vs 11.15 minutes). 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 DD-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 [16, 17, 18, 19], namely 10, realizing a maximum overall accuracy of 85.30% when GS-GmdRQA with SVM classifier is employed. However, we note that the performance of this scheme comes again with the cost of high computational complexity. Thus, DD-GmdRQA is highlighted as the best trade-off in terms of classification accuracy (83.88%) for the 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 DD-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.*[16, 17], utilizing the minimum number of features that has been reported so far considering the RQA framework [16, 17, 18, 19]. Thus DD-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.*[18] and Jiang *et al.*[19], RQA is combined with other feature extraction and time series analysis techniques that are not examined in the herein work. Particularly, in [18] authors combine RQA and Complex Networks whereas in [35] the multiscale coarse-grained RQA is combined with HVG. In our work, HVG is also employed and combined with all the aforementioned RQA and GmdRQA schemes in Experiment 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 minutes to reproduce the HVG framework proposed in [29]. As depicted in Experiment 3, the combination of HVG with GS-RQA, DD-RQA and MF-RQA does not provide any substantial improvement for both classifiers, with the score being quite similar to the results of Experiment 1. The performance of the combination of HVG and GmdRQA frameworks is then presented in Experiment 4. The combination of HVG and DD-GmdRQA outperforms the other architectures in terms of average overall classification accuracy and computational complexity. Particularly, HVG and DD-GmdRQA reaches the score of 89.19% ± 0.82% with the FDA classifier and 89% ± 0.80% with SVM.

Compared to the best performing scheme of this work, i.e. DD-GmdRQA and HVG, the work of Olyaee *et al.* [18] achieves a slightly 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.* [19] although the stage of parameter selection and definition is not stated in their work. Specifically, they utilize multiscale coarse-grained RQA [35] 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.* [19]. implements an extremely time consuming classification procedure. In particular, they perform a leave-one-out cross-validation using an 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 seconds, whereas our proposed SVM classification scheme requires 24.87 seconds for the exact same setup described in Section 4.3.

5. Conclusions and Future Work

In this work, we designed and implemented novel data-driven protein structure 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.

An extension of this work will consider a GmdRQA framework that processes directly on the primary amino acid protein sequence without employing intermediate tools such as PSIPRED. Lastly, a future goal is to predict the three-dimensional structure of the proteins with deep learning algorithms, exploiting the graphs exported with CGR.

Acknowledgments

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 grant no. 1725, and by the Stavros Niarchos Foundation within the framework of the project ARCHERS.

Appendix A. RQA Features

- **Recurrence rate:** Measures the density of points in the RP or in other words, the probability that a similar state recurs to its neighbourhood in phase space,

$$RR = \frac{1}{N_s^2} \sum_{i,j=1}^{N_s} R_{i,j} \tag{A.1}$$

where $R_{i,j}$ refers to a point of the recurrence matrix and N_s is the length of time series.

- **Determinism:** The ratio of the number of recurrence points forming diagonal structures to the total number of recurrence points is regarded as determinism or predictability of the system. Determinism is close to unity in a periodic system and close to zero in systems with no time-dependence,

$$DET = \frac{\sum_{\ell=\ell_{\min}}^{N_s} \ell P(\ell)}{\sum_{\ell=1}^{N_s} \ell P(\ell)} \tag{A.2}$$

where ℓ is the diagonal length, $P(\ell)$ is the probability of a diagonal structure of length ℓ. $\ell_{\min} = 2$ by default.

- **Average diagonal length:** This average length is actually the mean time that we can predict the next recurrence of states from the state we observe now. Intuitively, a diagonal line of length ℓ means that trajectories are co-evolving during ℓ samples but they correspond to different times of the system evolution. These lines indicate how different trajectories diverge during the evolution of the system and as time passes by,

$$L_{mean} = \frac{\sum_{\ell=\ell_{\min}}^{N_s} \ell P(\ell)}{\sum_{\ell=\ell_{\min}}^{N_s} P(\ell)} \tag{A.3}$$

where ℓ is the diagonal length, $P(\ell)$ is the probability of a diagonal structure of length ℓ.

- **Length of longest diagonal/vertical line:** Refers to the maximum length of the diagonal/vertical lines in the recurrence plot that represent the maximum time that the system evolves or remains in a certain state, respectively

$$L_{max} = \max(\{\ell_i\}_{i=1}^{n_\ell}) \tag{A.4}$$
$$V_{max} = \max(\{v_i\}_{i=1}^{n_v}), \tag{A.5}$$

where n_ℓ and n_v is the number of diagonal and vertical lines accordingly.

- **Entropy of diagonal/vertical length:** Indicates the complexity of the recurrence plot in respect of the diagonal/vertical lines. The entropy of vertical lines reflects the distribution of time periods for which the system abides in laminar phases. Signals with no time dependence present diagonal entropy≈0, i.e., the diagonal lines distribution is fully concentrated on very short lines (e.g., single dots),

$$ENTR = -\sum_{\ell=\ell_{\min}}^{N_s} P(\ell) \ln P(\ell) \tag{A.6}$$

- **Laminarity:** Provides information about the occurrence of the laminar states in the system. However, it does not describe the length of the laminar states. The value of laminarity decreases if an increased number of single recurrence points are present in the recurrence plot than the vertical structures,

$$LAM = \frac{\sum_{v=v_{\min}}^{N_s} vP(v)}{\sum_{v=1}^{N_s} vP(v)} \tag{A.7}$$

where $P(v)$ is the frequency distribution of the lengths v of the vertical lines, which have at least a length of $v_{\min}$.

- **Trapping time:** The average length of vertical lines is called trapping time and it is related with laminarity time. This value contains information about the frequency and the length of laminar states,

$$TT = \frac{\sum_{v=v_{\min}}^{N_s} vP(v)}{\sum_{v=v_{\min}}^{N_s} P(v)} \tag{A.8}$$

where v is the diagonal length, $P(v)$ is the probability of a diagonal structure of length v.

- **Size of the phase space:** Refers to the number of states created via time delay embedding.

Appendix B. HVG Features

- **Maximum Degree:** The degree of a node is the number of edges connected to the node. In terms of the adjacency matrix A, the degree for a node w_i in an undirected network is,

$$k_i = \sum_j A_{ij} \tag{B.1}$$

where the sum is over all nodes in the network. Thus, the maximum degree of a network is

$$k_{\max} = \max\{k_i\},\ i = 1, \ldots, W \tag{B.2}$$

- **Average shortest path:** Denote two vertices $v_i, v_j \in V$. The average path length between them is,

$$L_G = \frac{1}{W(W-1)} \sum_{i \neq j} d(v_i, v_j) \tag{B.3}$$

where $d(v_i, v_j)$ is the shortest distance among v_i and v_j with $i \neq j$.

- **Diameter:** The diameter is a measure of the compactness in a network. Practically, is the longest shortest path between any two nodes in the network and is defined as,

$$D = \max\{d(v_i, v_j)\},\ v_i, v_j \in V \text{ and } i \neq j \tag{B.4}$$

- **Clustering coefficient:** The clustering coefficient is a measure of the degree to which nodes in a graph tend to cluster together and it is denoted as,

$$C = \frac{1}{N} \sum_i \frac{e_i}{k_i(k_i - 1)/2} \tag{B.5}$$

where e_i is the actual number of edges between the neighbors of node i. In other words, the clustering coefficient refers to the probability that two neighbors of any node are also neighbors.

- **Energy:** The energy of a network is defined as,

$$E = \sum_{i=1}^{N} |\lambda_i| \tag{B.6}$$

where λ_i is the i-th eigenvalue of the adjacency matrix A.

- **Laplacian Energy:** The Laplacian Energy is denoted as,

$$LE = \sum_{i=1}^{N} \left| \mu_i - \frac{2M}{W} \right| \tag{B.7}$$

where μ_i is the i-the eigenvalue of the Laplacian matrix of the network.

- **Pearson correlation coefficient:** To understand whether an unweighted undirected network is of assortive or disassortive type, the Pearson correlation coefficient of the degrees at either ends of an edge is calculated as,

$$r = \frac{M^{-1} \sum_i j_i k_i - \left[M^{-1} \sum_i 0.5(j_i + k_i) \right]^2}{\sum_i 0.5({j_i}^2 + {k_i}^2) - \left[M^{-1} \sum_i 0.5(j_i + k_i) \right]^2} \tag{B.8}$$

where j_i and k_i are the degrees of the nodes at the two ends of the i-th edge, with $i = 1, 2, \ldots, M$.

- **Average closeness centrality:** The closeness value is the inverse of the average distance between node v_i and a node v_j and is denoted as,

$$CC(i) = \frac{N-1}{\sum_{j \in V, j \neq i} d(v_i, v_j)} \tag{B.9}$$

where d is the average distance.

- **Number of nodes:** The number of nodes is an important feature for the network and it is equal to W.

All the aforementioned metrics are utilized in order to later form a feature matrix for solving the protein secondary structure classification problem.

References

[1] M. E. Noble, J. A. Endicott, L. N. Johnson, Protein kinase inhibitors: insights into drug design from structure, Science 303 (5665) (2004) 1800–1805.

[2] C. B. Anfinsen, Principles that govern the folding of protein chains, Science 181 (4096) (1973) 223–230.

[3] M. Levitt, C. Chothia, Structural patterns in globular proteins, Nature 261 (5561) (1976) 552–558.

[4] H. Nakashima, K. Nishikawa, T. Ooi, The folding type of a protein is relevant to the amino acid composition, The Journal of Biochemistry 99 (1) (1986) 153–162.

[5] K.-C. Chou, Prediction of protein cellular attributes using pseudo-amino acid composition, Proteins: Structure, Function, and Bioinformatics 43 (3) (2001) 246–255.

[6] R.-y. Luo, Z.-p. Feng, J.-k. Liu, Prediction of protein structural class by amino acid and polypeptide composition, European Journal of Biochemistry 269 (17) (2002) 4219–4225.

[7] T. Liu, X. Zheng, J. Wang, Prediction of protein structural class for low-similarity sequences using support vector machine and psi-blast profile, Biochimie 92 (10) (2010) 1330–1334.

[8] T. Liu, C. Jia, A high-accuracy protein structural class prediction algorithm using predicted secondary structural information, Journal of theoretical biology 267 (3) (2010) 272–275.

[9] J.-Y. Yang, Z.-G. Yu, V. Anh, Protein structure classification based on chaos game representation and multifractal analysis, in: 2008 Fourth International Conf. on Natural Computation, Vol. 4, IEEE, 2008, pp. 665–669.

[10] S. Zhang, F. Ye, X. Yuan, Using principal component analysis and support vector machine to predict protein structural class for low-similarity sequences via pssm, Journal of Biomolecular Structure and Dynamics 29 (6) (2012) 1138–1146.

[11] Y. Liang, S. Liu, S. Zhang, Prediction of protein structural classes for low-similarity sequences based on consensus sequence and segmented pssm, Computational and mathematical methods in medicine 2015.

[12] J. Wang, C. Wang, J. Cao, X. Liu, Y. Yao, Q. Dai, Prediction of protein structural classes for low-similarity sequences using reduced pssm and position-based secondary structural features, Gene 554 (2) (2015) 241–248.

[13] B. Yu, L. Lou, S. Li, Y. Zhang, W. Qiu, X. Wu, M. Wang, B. Tian, Prediction of protein structural class for low-similarity sequences using chou's pseudo amino acid composition and wavelet denoising, Journal of Molecular Graphics and Modelling 76 (2017) 260–273.

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

[15] M. Apurva, 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 84 (2020) 107164.

[16] J.-Y. Yang, Z.-L. Peng, Z.-G. Yu, R.-J. Zhang, V. Anh, D. Wang, Prediction of protein structural classes by recurrence quantification analysis based on chaos game representation, Journal of Theoretical Biology 257 (4) (2009) 618–626.

[17] J.-Y. Yang, Z.-L. Peng, X. Chen, Prediction of protein structural classes for low-homology sequences based on predicted secondary structure, BMC bioinformatics 11 (1) (2010) S9.

[18] M. H. Olyaee, A. Yaghoubi, M. Yaghoobi, Predicting protein structural classes based on complex networks and recurrence analysis, Journal of theoretical biology 404 (2016) 375–382.

[19] H. Jiang, A. Zhang, Z. Zhang, Q. Meng, Y. Li, Protein tertiary structure prediction based on multiscale recurrence quantification analysis and horizontal visibility graph, in: International Symposium on Neural Networks, Springer, 2019, pp. 531–539.

[20] J. Eckmann, S. O. Kamphorst, D. Ruelle, Recurrence plots of dynamical systems, Europh. Lett. 4 (9) (1987) 973.

[21] L. Lacasa, B. Luque, F. Ballesteros, J. Luque, J. C. Nuno, From time series to complex networks: The visibility graph, Proceedings of the National Academy of Sciences 105 (13) (2008) 4972–4975.

[22] D. T. Jones, Protein secondary structure prediction based on position-specific scoring matrices, Journal of molecular biology 292 (2) (1999) 195–202.

[23] H. J. Jeffrey, Chaos game representation of gene structure, Nucleic acids research 18 (8) (1990) 2163–2170.

[24] M. A. Zervou, G. Tzagkarakis, P. Tsakalides, Automated screening of dyslexia via dynamical recurrence analysis of wearable sensor data, in: 2019 IEEE 19th International Conf. on Bioinformatics and Bioengineering (BIBE), IEEE, 2019, pp. 770–774.

[25] A. M. Fraser, H. L. Swinney, Independent coordinates for strange attractors from mutual information, Physical review A 33 (2) (1986) 1134.

[26] M. B. Kennel, R. Brown, H. D. Abarbanel, Determining embedding dimension for phase-space reconstruction using a geometrical construction, Physical review A 45 (6) (1992) 3403.

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

[28] N. Marwan, M. C. Romano, M. Thiel, J. Kurths, Recurrence plots for the analysis of complex systems, Physics reports 438 (5-6) (2007) 237–329.

[29] Z.-Q. Zhao, L. Luo, X.-Y. Liu, Low-homology protein structural class prediction from secondary structure based on visibility and horizontal visibility network, American Journal of Biochemistry and Biotechnology 14 (2018) 67–75. doi:10.3844/ajbbsp.2018.67.75.

[30] http://biomine.cs.vcu.edu/datasets/SCPRED/SCPRED.html.

[31] H. Kantz, T. Schreiber, Nonlinear time series analysis, Vol. 7, Cambridge university press, 2004.

[32] S. Wallot, D. Mønster, Calculation of average mutual information (ami) and false-nearest neighbors (fnn) for the estimation of embedding parameters of multidimensional time series in matlab, Frontiers in psychology 9 (2018) 1679.

[33] K. H. Krämer, R. V. Donner, J. Heitzig, N. Marwan, Dimension-scalable recurrence threshold estimation, arXiv preprint arXiv:1802.01605.

[34] https://www.mathworks.com/help/stats/fitcecoc.html.

[35] M. Xu, P. Shang, A. Lin, Multiscale recurrence quantification analysis of order recurrence plots, Physica A: Statistical Mechanics and its Applications 469 (2017) 381–389.