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

Michaela Areti Zervou\textsuperscript{a,b}, Effrosyni Doutsi\textsuperscript{b,*}, Pavlos Pavlidis\textsuperscript{b}, Panagiotis Tsakalides\textsuperscript{a,b}

\textsuperscript{a}Department of Computer Science, University of Crete, Heraklion, Greece
\textsuperscript{b}Institute of Computer Science, Foundation for Research and Technology-Hellas, Heraklion, Greece

Abstract

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\textsuperscript{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\textsuperscript{2}. Based on their folding patterns, proteins can be classified into four structural categories, namely, (i) all-\(\alpha\)-, where the structural domains are mainly composed of \(\alpha\)-helices and only a little amount of \(\beta\)-strands (a.k.a. \(\beta\)-sheets), (ii) all-\(\beta\)-, that is mostly formed by \(\beta\)-strands and a few isolated \(\alpha\)-helices, (iii) \(\alpha + \beta\)-, forming \(\alpha\)-helices and mostly anti-parallel \(\beta\)-strands, and (iv) \(\alpha/\beta\) consisted of \(\alpha\)-helices and almost all parallel \(\beta\)-strands\textsuperscript{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 the protein structural class prediction. Some extensively utilized methods for the representation of protein samples via feature extraction are the amino acid composition\textsuperscript{4}, the pseudo-amino acid composition\textsuperscript{5}, the dipeptide and tripeptide compositions\textsuperscript{6}, the PSI-BLAST profile\textsuperscript{7}, and the predicted secondary structure information\textsuperscript{8}, among others. However, the performance of these techniques is less significant when low-similarity proteins are encountered. Thus, a lot of effort has been made to improve the prediction accuracy for proteins that we cannot obtain sufficient amount of homologous information\textsuperscript{9,10,11,12,13,14,15}. Numerous studies\textsuperscript{16,17,18,19,20} 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)\textsuperscript{20}, Horizontal Visibility Graphs (HVG)\textsuperscript{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 the three secondary structural elements, namely H (helix), E (strand) and C (coil) using the PSI-PRED\textsuperscript{22} tool. Then, by employing the Chaos Game Representation (CGR)\textsuperscript{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 this graph are considered as two
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 the inefficient process- ing of multidimensional data. This is mainly due to the fact that (i) a pair of coordinates has to be processed separately, resulting in 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 proposes (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 gain maximum profit of 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 reduces the number of features by far as it is applied directly to the two-dimensional time series instead of each time series separably, as stated in the literature.

(ii) The data-driven parameter selection employing FNN and AMI algorithms.

(iii) The design of novel feature extraction schema consisted of HVG and a Data-Driven unidimensional RQA (DD-RQA) or a 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 not only regarding the feature selection methods but also the classification process.

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 the 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, PSIPRED 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 not only reduces the dimensionality of our data from 20 amino acids to three structural elements but also 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 unidimensional 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. 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 as helix (H), coils (C), and strands (E), with xy-plane coordinates (0,0), (0.5, √3/2) and (1,0), respectively.

The CGR graph, as shown in Fig. 2a is obtained through the following procedure: Initially, the triangle centroid \((x_0, y_0) = (0.5, \sqrt{3}/6)\) is defined and then, the \((x^{(i)}, y^{(i)})\) 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,

\[
\begin{align*}
    x^{(i)} &= \frac{1}{2}(x^{(i-1)} + x^{(i)}), \quad &i = 1, \ldots, N \\
    y^{(i)} &= \frac{1}{2}(y^{(i-1)} + y^{(i)}), \quad &i = 1, \ldots, N
\end{align*}
\]

where \(x^{(i)}\) and \(y^{(i)}\) are respectively the x and y coordinates of the vertex corresponding to the \(i^{th}\) secondary structure element of a protein consisted 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 x and y coordinates so that \(x^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \ldots, x_N^{(i)}\}\) and \(y^{(i)} = \{y_1^{(i)}, y_2^{(i)}, \ldots, y_N^{(i)}\}\).

2.3. Time Series Analysis

2.3.1. Recurrence Quantification Analysis

The recurrence quantification analysis (RQA) 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,

\[
x_i = [t_i, t_{i+\tau}, \ldots, t_{i+(m-1)\tau}], \quad i = 1, \ldots, N_x,
\]

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

\[
R_{ij} = \Theta(\|x_i - x_j\|_p), \quad i, j = 1, \ldots, N_x,
\]

where \(x_i, x_j \in \mathbb{R}^m\) are the states, \(\varepsilon\) 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

\[
\Theta(w) = \begin{cases} 
1, & \text{if } w \geq 0 \\
0, & \text{if } w < 0
\end{cases}, \quad w \in \mathbb{R}.
\]

The resulting matrix \(R\) exhibits the main diagonal, \(R_{ij} = 1\), \(i = 1, \ldots, N_x\), 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 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,

\[
\text{ns} < \min(n_i, n_j), \quad i < k < j.
\]

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.

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, the all-α, all-β and α + β 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 α + β 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 α + β fold.
3.2. Generalized Multidimensional Recurrence Quantification Analysis

Multidimensional recurrence quantification analysis extracts in general 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 \( \{t^{(d)}_{i,j}\}_{i=1}^{N} \), where \( d \) stands for the data dimensionality, the corresponding phase space representation is reconstructed as follows,

\[
\begin{align*}
\mathbf{u}_i &= (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(D)}) \\
\mathbf{u}_j &= (x_j^{(1)}, x_j^{(2)}, \ldots, x_j^{(D)}) \\
\mathbf{X}_i &= (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(D)}) \\
\mathbf{X}_j &= (x_j^{(1)}, x_j^{(2)}, \ldots, x_j^{(D)})
\end{align*}
\]

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

\[
\mathbf{X}_i = \begin{pmatrix}
\mathbf{x}_i^{(1)} & \mathbf{x}_i^{(2)} & \cdots & \mathbf{x}_i^{(K)} \\
\mathbf{x}_i^{(1)} & \mathbf{x}_i^{(2)} & \cdots & \mathbf{x}_i^{(K)} \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{x}_i^{(1)} & \mathbf{x}_i^{(2)} & \cdots & \mathbf{x}_i^{(K)} \\
\end{pmatrix}
\]

where \( k = \lfloor \sqrt{D} \rfloor \), \( l = \lfloor D/k \rfloor \) and \( i = 1, \ldots, N_t \). Subsequently, the Generalized multidimensional Recurrence Plot (GmdRP) is defined by

\[
\text{GmdR}_{ij} = \Theta (\varepsilon - \| \mathbf{x}_i - \mathbf{x}_j \|)
\]

where \( \varepsilon \) 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, the parameter tuning has been proceeded in a grid search manner which is time demanding. In the herein work, the estimation of the embedding parameters is proposed to happen in a Data-Driven (DD) fashion where the AMI and the FNN methods are utilized to evaluate the optimal time delay \( \tau \) 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 \( \{t_i\}_{i=1}^{N} \) and a time delayed version of this signal by \( \tau \) samples \( \{t_{i+\tau}\}_{i=1}^{N} \) and is expressed as,

\[
I(t_i, t_{i+\tau}) = \sum_{t_j} p(j | t_i) \log \left( \frac{p(j | t_i)}{p(j)} \right)
\]

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

Determining a proper value for \( \tau \) implies that the coordinates of the phase space embedded signal will be maximally independent. As proposed in [26], this is guaranteed by choosing as the optimal value for the time lag \( \tau \) 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 \( \tau \) for which the AMI function descents below the value \( 1/e \approx 2.71 \). Furthermore, in this study, the max lag \( \tau \) 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. 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 neighboured 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 \( \tau \) and two of its coordinate vectors \( x_i \) and \( x_j \) that are adjacent at a time instance, the squared Euclidean distance between them when moving from \( m \) into \( (m + 1) \) dimensions is,

\[
D_{m+1}^2 = D_m^2 + (x_i - x_j)^2
\]

where \( D_m \) is the Euclidean distance between \( x_i \) and \( x_j \) and is defined as,

\[
D_m = \sqrt{\sum_{c=0}^{m-1} (x_i - x_j)^2}
\]

If the one-dimensional time series is already properly embedded in \( m \) dimensions, then the distance \( D \) between \( x_i \) and its nearest neighbor \( x_j \) should not appreciably change by some distance criterion \( D_{tol} \) so that \( D < D_{tol} \). Moreover, the distance of the nearest neighbor when embedded into the next higher dimension should be less than some criterion \( A_{tol} \) such that \( D_{m+1} < A_{tol} \). According to [32] the next settings of the thresholds \( D_{tol} \approx 10 \) 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
4. Results

This section initially describes the experimental evaluation of the optimal set of parameters, $m$, $\tau$, and $\epsilon$, of the herein proposed RQA parameter selection schemes. Thereafter the performance of each proposed architecture is compared to state-of-the-art in terms of overall classification accuracy, feature multiplicity, and running time complexity. It is important to note that in this work the classification procedure is decoupled from 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 without vanishing the 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 $\tau$ 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 $\tau$ values are computed as a consequence of a statistical analysis of the optimal set of parameters. This three parameter selection approaches are employed by RQA and GmdRQA, respectively 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 $\tau$ varies between 1 and 8. In case that both $m$ and $\tau$ are equal to 8, the phase space can not 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-GmdRQA, we generated 10 randomly shuffled datasets and for each dataset the classification accuracy is reported for all combinations of $m$ and $\tau$ 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 $\tau = 1$ and for FDA are $m = 7$ and $\tau = 1$. The implementation of GS-RQA and GS-GmdRQA for the rest of this work employs 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, time delay embedding parameters are evaluated per protein using FNN and AMI algorithms. However, in this work, for proteins of length lower than 45 residues, time delay embedding parameters are set as $\tau = 1$ and $m = 2$, respectively. 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 parameters $m$ and $\tau$ are selected as the most frequent value among the optimal sets of $m$ and $\tau$ that derive when FNN and AMI are employed. Again, for proteins of length lower than 45 residues, time delay embedding parameters are set as $\tau = 1$ and $m = 2$. The core difference between the GS-RQA/GS-GmdRQA and MF-RQA/MF-GmdRQA is that for the grid search process $m$ 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 $T_{\text{tol}} = 2$, or iii) the point before which the number of FNNs starts to increase again.

Figure 4: General proposed RQA time delay embedding parameter selection scheme.
and \(\tau\) are set equal for both time series dimensions, whereas for the most frequent parameter searching \(m\) and \(\tau\) 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 \(t_x\) and \(t_y\) time series, respectively. Consequently, for MF-GmdRQA derives that \((m, \tau) = (2, 6)\) 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 \(\varepsilon\)

Following the empirical rule proposed in [33] the neighborhood threshold \(\varepsilon\) can be given 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 \(\varepsilon\) for both RQA and GmdRQA schemes using FDA and SVM classifier accordingly. As depicted, the variation of the average classification accuracy for GS-RQA and GS-GmdRQA architectures is minor when \(\varepsilon\) lies between 30%-50% for both classifiers. Therefore, \(\varepsilon = 30\%\) is selected as the minimum percentage that provides high classification results. Concerning DD-RQA architecture it performs its best when \(\varepsilon\) lies between 30%-50% for FDA and SVM as well. Thus, \(\varepsilon = 30\%\) is chosen. On the other hand, 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 \(\varepsilon\) lies between 20%-50% according to accordingly. Consequently, we select \(\varepsilon = 20\%\) and \(\varepsilon = 30\%\) when FDA and SVM are employed, respectively.

4.3. Classification

The first step in the classification process is to z-score normalize the feature matrix. In the related work, the process of leave-one-out cross-validation is preferred. However, splitting the data into 70%-30% for training and testing respectively, yields nearly the same overall accuracy in significantly lesser time. Therefore, in order 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...
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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>DD-RQA</th>
<th>DD-GmdRQA</th>
<th>MF-RQA</th>
<th>MF-GmdRQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>Specificity</td>
<td>Sensitivity</td>
<td>Specificity</td>
<td>Sensitivity</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>78.7 ± 3.7%</td>
<td>88.4 ± 1.8%</td>
<td>76.4 ± 4.1%</td>
<td>82.5 ± 2.2%</td>
</tr>
<tr>
<td>$\beta$</td>
<td>62.5 ± 4.1%</td>
<td>90.2 ± 1.7%</td>
<td>65.2 ± 3.9%</td>
<td>89.1 ± 1.7%</td>
</tr>
<tr>
<td>$\alpha/\beta$</td>
<td>59.7 ± 4.6%</td>
<td>93.7 ± 1.2%</td>
<td>67.8 ± 4.6%</td>
<td>90.5 ± 1.6%</td>
</tr>
<tr>
<td>$\alpha + \beta$</td>
<td>64.9 ± 4.5%</td>
<td>83.1 ± 2.1%</td>
<td>61.1 ± 3.8%</td>
<td>94.6 ± 1.7%</td>
</tr>
<tr>
<td>OA</td>
<td>83.4 ± 0.9%</td>
<td>83.8 ± 0.9%</td>
<td>85.4 ± 0.9%</td>
<td>83.3 ± 0.9%</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>DD-RQA</th>
<th>DD-GmdRQA</th>
<th>MF-RQA</th>
<th>MF-GmdRQA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensitivity</td>
<td>Specificity</td>
<td>Sensitivity</td>
<td>Specificity</td>
<td>Sensitivity</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>74.9 ± 4.7%</td>
<td>87.1 ± 2.8%</td>
<td>65.5 ± 4.2%</td>
<td>88.4 ± 1.9%</td>
</tr>
<tr>
<td>$\beta$</td>
<td>49.9 ± 9.7%</td>
<td>93.6 ± 2.3%</td>
<td>65.5 ± 4.2%</td>
<td>88.4 ± 1.9%</td>
</tr>
<tr>
<td>$\alpha/\beta$</td>
<td>63.2 ± 4.6%</td>
<td>91.2 ± 1.5%</td>
<td>73.9 ± 4.6%</td>
<td>85.9 ± 1.9%</td>
</tr>
<tr>
<td>$\alpha + \beta$</td>
<td>68.5 ± 4.1%</td>
<td>80.1 ± 2.4%</td>
<td>57.7 ± 3.6%</td>
<td>100%</td>
</tr>
<tr>
<td>OA</td>
<td>82.1 ± 1.2%</td>
<td>83.6 ± 0.8%</td>
<td>85.4 ± 0.9%</td>
<td>83.5 ± 1%</td>
</tr>
</tbody>
</table>

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 α + β. Concerning SVM classifier, the regularization parameter $C$ and kernel width parameter $\gamma$ can take all positive values log-scaled in the range $[10^{-7}, 10^{7}]$. 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,
  \[ \text{Sensitivity} = \frac{TP}{TP + FN} \]
- **Specificity**, also known as the true negative rate, measures the proportion of actual negatives that are correctly identified as such,
  \[ \text{Specificity} = \frac{TN}{TN + FP} \]
- **Overall accuracy** is the most intuitive and frequently used performance parameter indicating the correctly classified samples against all samples,
  \[ \text{Overall Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \]

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, RQA and GmdRQA architectures are examined for each of the proposed data-driven time delay embedding parameter selection schemes. 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 scheme 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 benefits 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 is extensively utilized in the state-of-the-art. However, none of the corresponding works report the optimal parameter setup they employ. 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 GS-RQA framework with SVM achieves the highest classification accuracy, namely 86.20%, when it is compared to its
rest proposed RQA counterparts. 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 in our case is 20. Moreover, it is important to mention that the range of the grid search parameter values may vary. Nevertheless, the performance of GS-RQA is comparable to MF-RQA with SVM, that reaches an overall classification accuracy of 85.40%. It is clear that GS-RQA improves the overall classification accuracy on a quite small scale with the cost of an extremely high running time. Thus, as highlighted in blue, we conclude that MF-RQA is the most profitable architecture in terms of classification accuracy and running time performance considering the Experiment 1. Finally, it is also worth to be mentioned that the most efficient 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, 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 that is 83.88% for SVM classifier, and running time efficiency. The most important outcome of the previous two experiments though is that the overall classification accuracy remains nearly unaffected for RQA and GmdRQA schemes with the later reducing the computational cost approximately to half in most cases. Therefore, among Experiment 1 and 2 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].

Table 3: Summarized predicted quality results for 25PDB dataset. The minimum computational time as well as the highest achieved accuracy for SVM and FDA per experiment, are indicated by 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of Features</th>
<th>Overall Accuracy</th>
<th>Computational Complexity (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>FDA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSI-PRED</td>
<td>CGR</td>
<td>GS RQA DD MF</td>
<td>GmdRQA GS DD MF</td>
</tr>
<tr>
<td>X X X X</td>
<td>16</td>
<td>65.08</td>
<td>-</td>
</tr>
<tr>
<td>Yang et al. [16]</td>
<td>X X</td>
<td>16</td>
<td>-</td>
</tr>
<tr>
<td>Experiment 1</td>
<td>X X</td>
<td>20</td>
<td>86.20</td>
</tr>
<tr>
<td></td>
<td>X X</td>
<td>20</td>
<td>83.69</td>
</tr>
<tr>
<td></td>
<td>X X</td>
<td>20</td>
<td>85.40</td>
</tr>
<tr>
<td>Experiment 2</td>
<td>X X</td>
<td>10</td>
<td>85.30</td>
</tr>
<tr>
<td></td>
<td>X X</td>
<td>10</td>
<td>83.88</td>
</tr>
<tr>
<td></td>
<td>X X</td>
<td>10</td>
<td>83.32</td>
</tr>
<tr>
<td>Olyaee et al. [18]</td>
<td>X X X X</td>
<td>24</td>
<td>90.08</td>
</tr>
<tr>
<td>Jiang et al. [19]</td>
<td>X X X X</td>
<td>30</td>
<td>95.33</td>
</tr>
<tr>
<td>Experiment 3</td>
<td>X X X</td>
<td>37</td>
<td>86.57</td>
</tr>
<tr>
<td></td>
<td>X X X</td>
<td>37</td>
<td>85.92</td>
</tr>
<tr>
<td></td>
<td>X X X</td>
<td>37</td>
<td>85.64</td>
</tr>
<tr>
<td>Experiment 4</td>
<td>X X X</td>
<td>27</td>
<td>85.93</td>
</tr>
<tr>
<td></td>
<td>X X X</td>
<td>27</td>
<td>89.00</td>
</tr>
<tr>
<td></td>
<td>X X X</td>
<td>27</td>
<td>85.63</td>
</tr>
</tbody>
</table>

The most important outcome of the previous two experiments though is that the overall classification accuracy remains nearly unaffected for RQA and GmdRQA schemes with the later reducing the computational cost approximately to half in most cases. Therefore, among Experiment 1 and 2 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 in the state-of-the-art. Thus DD-GmdRQA not only reduces the time complexity due to its automated parameter estimation, but the memory complexity as well.

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 [19] multiscale coarse-grained RQA is combined with HVG. In this 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 significant 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. HVG and DD-GmdRQA exceeds the rest architectures in terms of average overall classification accuracy and computational complexity compared to the rest architectures. Particularly, HVG and DD-GmdRQA reaches the score of 89.19 ± 0.82% with 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 an insignificantly higher overall accuracy, namely 90.08%, utilizing 24 features that derive from the combination of RQA and Complex Networks. However, the tuning of RQA hyperparameters is performed in a grid search manner which leads to the conclusion that is extremely time consuming based on the experiments presented in this work for GS-RQA. The same as-
sumption can be also made for the work of Jiang et al. [19] as the part of parameter selection and definition is not stated in their work. Specifically, they utilize multiscale coarse-grained RQA [15] 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. implements an extremely expensive classification procedure that requires hours of training. Characteristically, they perform a leave-one-out cross-validation using SVM classifier with a grid search parameter selection of the regularization parameter $C$ and kernel width parameter $\gamma$ 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 set up 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 great impact on the overall classification accuracy and the running time complexity. Therefore, 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 the direct processing of the primary amino acid protein sequence without employing intermediate tools such as PSI-PRED.

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,

$$RR = \frac{1}{N^2} \sum_{i,j=1}^{N} R_{i,j}$$  

(A.1)

where $R_{i,j}$ refers to a point of the recurrence matrix and $N$ 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,

$$DET = \frac{\sum_{\ell=t_{\text{min}}}^{N_i} \ell \mathbf{P}(\ell)}{\sum_{\ell=t_{\text{min}}}^{N_i} \ell \mathbf{P}(\ell)}$$  

(A.2)

where $\ell$ is the diagonal length, $P(\ell)$ is the probability of a diagonal structure of length $\ell$, $t_{\text{min}} = 2$ 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 $\ell$ means that trajectories are co-evolving during $\ell$ 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,

$$L_{\text{mean}} = \frac{\sum_{\ell=t_{\text{min}}}^{N_i} \ell P(\ell)}{\sum_{\ell=t_{\text{min}}}^{N_i} P(\ell)}$$  

(A.3)

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

- **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

$$L_{\text{max}} = \max(\{|\ell|_{\ell \leq 1}\})$$  

(A.4)

$$V_{\text{max}} = \max(\{|v|_{v \leq 1}\})$$  

(A.5)

where $n_\ell$ 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),

$$\text{ENTR} = - \sum_{\ell=t_{\text{min}}}^{N_i} P(\ell) \ln P(\ell)$$  

(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
Appendix B. HVG Features

- **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,
  \[
  TT = \frac{\sum_{i=1}^{v_{\text{max}}} v P(v)}{\sum_{i=1}^{v_{\text{max}}} P(v)} \tag{A.8}
  \]
  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.

### Maximum Degree

- **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
  \[
  k_{\text{max}} = \max\{k_i\}, \quad i = 1, \ldots, W \tag{B.2}
  \]

- **Average shortest path**: Denote two vertices \(v_i, v_j \in V\). The average path length between them is,
  \[
  L_{av} = \frac{1}{W(W-1)} \sum_{i \neq j} d(v_i, v_j) \tag{B.3}
  \]
  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, it is the longest shortest path between any two nodes in the network and is defined as,
  \[
  D = \max\{d(v_i, v_j)\}, \quad v_i, v_j \in V \quad \text{and} \quad i \neq j \tag{B.4}
  \]

- **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,
  \[
  C = \frac{1}{N} \sum_i \frac{e_i}{k(k - 1)/2} \tag{B.5}
  \]
  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 \(\lambda_i\) 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 \(\mu_i\) 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 dis-assortive type, the Pearson correlation coefficient of the degrees at either ends of an edge is calculated as,
  \[
  r = \frac{M^{-1} \sum_i j_i k_i - \left[ M^{-1} \sum_j 0.5(j_i + k_i) \right]^2}{\sqrt{\sum_j 0.5(j_i^2 + k_i^2) - \left[ M^{-1} \sum_j 0.5(j_i + k_i) \right]^2}} \tag{B.8}
  \]
  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,
  \[
  CC(i) = \frac{N - 1}{\sum_{j \in V \setminus \{i\}} d(v_i, v_j)} \tag{B.9}
  \]
  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

M. A. Zervou, G. Tzagkarakis, P. Tsakalides, Automated screening of M. B. Kennel, R. Brown, H. D. Abarbanel, Determining embedding di-
A. M. Fraser, H. L. Swinney, Independent coordinates for strange attrac-
N. Marwan, M. C. Romano, M. Thiel, J. Kurths, Recurrence plots for the
J. Zbilut, C. Webber, Embeddings and delays as derived from quantifica-
M. Apurva, H. Mazumdar, Predicting protein structural classes based on complex networks and recurrence analysis, Journal of theo-
D. T. Jones, Protein secondary structure prediction based on position-
H. J. Jeffrey, Chaos game representation of gene structure, Nucleic acids
research 18 (8) (1990) 2163–2170.
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 Bioengineer-
ging (BIBE), IEEE, 2019, pp. 665–669.
S. Zhang, F. Ye, X. Yuan, Using principal component analysis and sup-70s vector machine to predict protein structural class for low-similarity
sequences based on consensus sequence and segmented pssm, Computational and mathematical methods in medicine 2015.
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.
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.
D. T. Jones, Protein secondary structure prediction based on position-
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 Bioengineer-
ging (BIBE), IEEE, 2019, pp. 770–774.
A. M. Fraser, H. L. Swanney, Independent coordinates for strange attrac-
tors from mutual information, Physical review A 33 (2) (1986) 1134.
M. B. Kennel, R. Brown, H. D. Abarbanel, Determining embedding di-
J. Zbilut, C. Webber, Embeddings and delays as derived from quantifica-
Z.-Q. Zhao, L. Luo, X.-Y. Liu, Low-homology protein structural class prediction from secondary structure based on visibility and horizontal vis-
http://biomine.cs.vcu.edu/datasets/SOPRED/SOPRED.html
S. Wallot, D. Monaster, Calculation of average mutual information (ami) and false-nearest neighbors (fnn) for the estimation of embedding param-
M. Xu, P. Shang, A. Lin, Multiscale recurrence quantification analysis of order recurrence plots, Physica A: Statistical Mechanics and its Applica-