life

MDPI

Article

Identifying Intrinsically Disordered Protein Regions through a Deep Neural Network with Three Novel Sequence Features

Jiaxiang Zhao * and Zengke Wang

College of Electronic Information and Optical Engineering, Nankai University, Tianjin 300350, China; wangzk@mail.nankai.edu.cn

* Correspondence: zhaojx@nankai.edu.cn

Abstract: The fast, reliable, and accurate identification of IDPRs is essential, as in recent years it has come to be recognized more and more that IDPRs have a wide impact on many important physiological processes, such as molecular recognition and molecular assembly, the regulation of transcription and translation, protein phosphorylation, cellular signal transduction, etc. For the sake of cost-effectiveness, it is imperative to develop computational approaches for identifying IDPRs. In this study, a deep neural structure where a variant VGG19 is situated between two MLP networks is developed for identifying IDPRs. Furthermore, for the first time, three novel sequence features—i.e., persistent entropy and the probabilities associated with two and three consecutive amino acids of the protein sequence—are introduced for identifying IDPRs. The simulation results show that our neural structure either performs considerably better than other known methods or, when relying on a much smaller training set, attains a similar performance. Our deep neural structure, which exploits the VGG19 structure, is effective for identifying IDPRs. Furthermore, three novel sequence features—i.e., the persistent entropy and the probabilities associated with two and three consecutive amino acids of the protein sequence—could be used as valuable sequence features in the further development of identifying IDPRs.

Keywords: intrinsically disordered proteins; the persistent entropy; the probabilities associated with two and three consecutive amino acids; VGG19

Citation: Zhao, J.; Wang, Z. Identifying Intrinsically Disordered Protein Regions through a Deep Neural Network with Three Novel Sequence Features. *Life* **2022**, *12*, 345. https://doi.org/10.3390/life12030345

Academic Editor: Sonia Longhi

Received: 10 January 2022
Accepted: 23 February 2022
Published: 26 February 2022

1. Introduction

Protein regions which lack stable three-dimensional structures are referred to as intrinsically disordered regions (IDPRs) [1]. In recent years, it has come to be recognized more and more that IDPRs have a huge impact on many important physiological processes [2,3], such as molecular recognition and molecular assembly, the regulation of transcription and translation, protein phosphorylation, cellular signal transduction, etc. [4–6]. Furthermore, some human diseases, such as certain types of cancer, Parkinson's disease, and cardiovascular disease [7–9], have been found to be linked with IDPRs. However, the experimental methods used to identify IDPRs are usually expensive and time-consuming [10]. Thus, the fast, reliable, and accurate identification of IDPRs by computational methods is a valuable complement to experimental studies.

There are many computational methods for identifying IDPRs. These methods can be divided into three categories: (1) Physicochemical-based methods, such as FoldIndex [11], GlobPlot [12], IUPred [13], FoldUnfold [14], and IsUnstruct [15], which rely on the amino acid physiochemical properties for identifying disorder. (2) Machine learning-based methods—for instance, DISvgg [16], RFPR-IDP [17], IDP-Seq2Seq [18], SPOT-Disorder [19], SPOT-Disorder2 [20], DISOPRED3 [21], SPINE-D [22], ESpritz [23], BVDEA [10], POODLE-S [24], RONN [25], and PONDRs [26]—which treat the identification of IDRs as labeling each amino acid of a protein sequence or as a classification problem. (3) Meta methods, including MFDp [27], MetaPrDOS [28], and Meta-Disorder predictor [29], which fuse multiple predictors to yield the final prediction for IDPRs.

While all of the above methods have contributed to the development of the field, there are still some new features that have not been discovered. Because of the interaction between amino acids, the question of how to describe them is key to improving predictions based on protein sequences.

In this paper, we develop a deep neural structure composed of a variant VGG19 [30], where the variant VGG19 is situated between two multilayer perceptron (MLP) networks for identifying IDPRs. In the variant VGG19, we erase the fully connected (FC) layers of VGG19 but preserve the other parts of the VGG19 structure and related parameters. In comparison with ResNet, the parameters of VGGNet could be easily manipulated. The MLP network consists of an input layer, hidden layers, and an output layer. The MLPs are employed for transforming the features into the formats suitable for serving as the inputs of the variant VGG19 and classification network, respectively. Compared with our previous DISpre algorithm [31] and DISvgg algorithm [16], we introduce VGG19 as a part of the network instead of as a single MLP network, and additionally use one VGG19 instead of ten VGG16. Moreover, to further improve the performance of prediction, we introduce new features for prediction. For the first time, three sequence features, which are the persistent entropy based on the persistent homology and the probabilities associated with two and three consecutive amino acids of the protein sequence (PCAA2, PCAA3), are introduced for identifying IDPRs. These three novel sequence features together with those used in [32]—i.e., two sequence features, seven physicochemical propensities, and three propensities of amino acids, as well as twenty evolutionary features—are used as the inputs for our neural structure. The simulation results obtained for two blind testing sets, R80 [25] and MXD494 [33], show that our neural structure either performs considerably better than other well-known methods [17,20] or, when relying on a much smaller training set (DIS1616) compared to the one used in [18], attains a similar performance.

2. Datasets and Input Features

In this section, the datasets used in this paper for training and blind testing are presented. The features extracted from the training dataset are depicted. In particular, we introduce three novel features, which are used for the first time for identifying IDPRs. These three novel features are persistent entropy based on persistent homology, PCAA2, and PCAA3.

2.1. Datasets

The dataset DIS1616 from the DisProt [34] (accessed on June 2020) is employed for training and cross validating, while the datasets R80 [25] and MXD494 [33] are used for blind testing. The training dataset DIS1616 consists of 1616 protein sequences which contain 182,316 disordered and 706,362 ordered amino acids. The dataset DIS1616 is randomly split into two subsets: DIS1450 and DIS166. They contain 1450 protein sequences and 166 protein sequences and are used for training and testing, respectively. The blind testing dataset R80 has 78 protein sequences, in which there are 3566 disordered and 29,243 ordered amino acids. There are 494 protein sequences in the blind testing dataset, MXD494, among which 44,087 disordered and 152,414 ordered amino acids are presented.

2.2. Input Features Used for the Identification of IDPRs

The features fed to our neural structure for identifying IDPRs can be summarized as five sequence features, seven physiochemical propensities, and three propensities of amino acids, as well as twenty evolutionary features of the given protein sequence. Of these five sequence features, persistent entropy based on persistent homology, PCAA2, and PCAA3 are, for the first time, introduced for identifying IDPRs. The remaining two sequence features are the Shannon entropy and topological entropy [32]. Topological entropy is used to depict the complexity of the protein sequence. The seven physiochemical properties of the amino acids are steric parameter, polarizability, volume, hydrophobicity, isoelectric point, helix, and sheet probability, as illustrated in the reference [35]. Three propensities

of the amino acids are Remark 465, Deleage/Roux, and Bfactor(2STD), which are derived from the GlobPlot NAR paper [12]. Twenty evolutionary features can be determined through the Position-Specific Substitution Matrix (PSSM) [36], which is computed using the Position-Specific Iterative Basic Local Alignment Search Tool (PSI-BLAST) [37].

2.2.1. The Computation of Persistent Entropy

In this section, we will to briefly illustrate the procedure used for computing the persistent homology as well as its persistent entropy from the given protein sequence. More information related to the computation of the persistent homology and its persistent entropy can be found in [38,39].

Given a protein sequence $\hat{w} = w_1 \cdots w_L$ of length L, we choose a sliding window of odd length N ($N < L$) to extract N consecutive amino acids from $\hat{w}$. For simplicity, we first transform $\hat{w}$ into a sequence of size $L + N - 1$ through appending $(N-1)/2$ amino acids to both ends of $\hat{w}$. The $(N-1)/2$ appended amino acids at both ends are identical to either the first or last amino acid of the protein sequence $\hat{w}$. Thus, utilizing a sliding window of size N, we can slice the transformed $\hat{w}$ of size $L + N - 1$ into L amino acid subsequences $\beta_j = w_j \cdots w_{j+(N-1)}$ with $1 \le j \le L$. To compute the persistent entropy of β_j, we need to map each amino acid w_m in β_j to a set of points, which leads us to define

$$I_1(m) = \sum_{\Phi_1(k) \in \mathrm{Y}_1} k\delta(w_m - \Phi_1(k)) \quad \text{for } 1 \le m \le N \tag{1}$$

where the value for k is $1 \le k \le 20$ and $\delta(\cdot)$ is the delta function. We use a one to one correspondence to represent the set of amino acid symbols as:

$$\begin{aligned} \mathrm{Y}_1 &\triangleq \{\Phi_1(1), \ldots, \Phi_1(20)\} \\ &= \{A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V\}. \end{aligned} \tag{2}$$

Thus, each amino acid symbol w_m with $1 \le m \le N$ in $\beta_j = w_j \cdots w_{j+(N-1)}$ ($1 \le j \le L$) is mapped to $(x_m^{\beta_j}, y_m^{\beta_j}, z_m^{\beta_j}) \in \mathcal{R}^3$, where we have:

$$\begin{array}{rcl} x_m^{\beta_j} & = & \cos\frac{2\pi}{20} I_1(m) \\ y_m^{\beta_j} & = & \sin\frac{2\pi}{20} I_1(m) \\ z_m^{\beta_j} & = & \frac{1}{N-1}(m-1) \end{array}. \tag{3}$$

We use $x_m^{\beta_j} = \cos\frac{2\pi}{20} I_1(m)$ and $y_m^{\beta_j} = \sin\frac{2\pi}{20} I_1(m)$ to project different amino acids to different positions on the axis. $x_m^{\beta_j}$ and $y_m^{\beta_j}$ are combined with $z_m^{\beta_j}$; then, all amino acids in β_j are projected to different positions on the axis. Thus, we can map each amino acid w_m for $1 \le m \le N$ in $\beta_j = w_j \cdots w_{j+(N-1)}$ ($1 \le j \le L$) to a unique element in the set of $\{(x_m^{\beta_j}, y_m^{\beta_j}, z_m^{\beta_j})\}_{m=1}^N$ in $\mathcal{R}^3$ through Equations (1)–(3).

The persistent entropy of $\mathcal{V}$ associated with $\{(x_m^{\beta_j}, y_m^{\beta_j}, z_m^{\beta_j})\}_{m=1}^N$ can be computed as

$$E(\mathcal{V}) = -\sum_{i=1}^{n} p_i \log_2 p_i, \quad p_i = \frac{\varepsilon_{e_i} - \varepsilon_{s_i}}{S_L}, \quad S_L = \sum_{i=1}^{n} l_i, \tag{4}$$

where $\mathcal{V}$ denotes a filtration with its associated persistence diagram $dgm(\mathcal{V}) = \{(\varepsilon_{s_i}, \varepsilon_{e_i}) : 1 \le i \le n\}$ (we assume $\varepsilon_{s_i} < \varepsilon_{e_i}$ for all $1 \le i \le n$). We have $L = \{l_i = \varepsilon_{e_i} - \varepsilon_{s_i}\}_{i=1}^n$. A filtration $\mathcal{V}$ of the simplicial complex $VR(\beta_j, K)$ ($K > 0$) associated with $\{(x_m^{\beta_j}, y_m^{\beta_j}, z_m^{\beta_j})\}_{m=1}^N$ is obtained through increasing the parameter values $0 \le \varepsilon \le K$—i.e.,

$$\phi = VR(\beta_j, 0) \subseteq VR(\beta_j, \varepsilon_1) \subseteq \cdots \subseteq VR(\beta_j, \varepsilon_l) = VR(\beta_j, K) \tag{5}$$

with $0 < \varepsilon_1 < \cdots < \varepsilon_l = K$. In Equation (5), the simplicial complex $VR(\beta_j, K)$ $(K > 0)$ is chosen to be the Vietoris Rips complex of β_j, which is defined as:

$$\begin{aligned} VR(\beta_j, \varepsilon) = \{\sigma \subset \{(x_m^{\beta_j}, y_m^{\beta_j}, z_m^{\beta_j})\}_{m=1}^N : \\ B_\varepsilon((x_m^{\beta_j}, y_m^{\beta_j}, z_m^{\beta_j})) \cap B_\varepsilon((x_l^{\beta_j}, y_l^{\beta_j}, z_l^{\beta_j})) \neq \phi \\ \forall\ (x_m^{\beta_j}, y_m^{\beta_j}, z_m^{\beta_j}),\ (x_l^{\beta_j}, y_l^{\beta_j}, z_l^{\beta_j}) \in \sigma\} \end{aligned} \tag{6}$$

where $B_\varepsilon((x_m^{\beta_j}, y_m^{\beta_j}, z_m^{\beta_j}))$ is the ball centered at $(x_m^{\beta_j}, y_m^{\beta_j}, z_m^{\beta_j})$ with the radius $\varepsilon \geq 0$. Given a filtration $\mathcal{V}$ defined by (5), a barcode in the k-dimensional persistence with endpoints $[\varepsilon_s, \varepsilon_e)$ corresponds to a k-dimensional hole that appears at filtration time ε_s and remains until filtration time ε_e. The set of bars $[\varepsilon_s, \varepsilon_e)$, representing the birth and death times of homology classes, is called the persistence barcode $\mathcal{B}(\mathcal{V})$ for the filtration $\mathcal{V}$ of (5). Analogously, the set of points $(\varepsilon_s, \varepsilon_e) \in \mathcal{R}^2$ is called the persistence diagram $dgm(\mathcal{V})$ of the filtration $\mathcal{V}$ of (5). The persistent entropy of each amino acid w_m for $1 \leq m \leq N$ in $\beta_j = w_j \cdots w_{j+(N-1)}$ $(1 \leq j \leq L)$ is therefore equal to the persistent entropy $E(\mathcal{V})$ of $\mathcal{V}$ associated with $\{(x_m^{\beta_j}, y_m^{\beta_j}, z_m^{\beta_j})\}_{m=1}^N$.

2.2.2. The Computation of the Features Using the Probabilities Associated with the Protein Sequence

The probability associated with two and three consecutive amino acids of the protein sequence depends on the probability of each amino acid occurring in the observed protein, which depends on the protein sequence length and the number of each individual amino acids in the protein sequence. We put all amino acids from all proteins in DIS1616 together and, based on this set, calculate the probabilities associated with two and three consecutive amino acids of the protein sequence. Consider the given protein sequence $\hat{w} = w_1 \cdots w_L$. For convenience, we define two sets:

$$\begin{aligned} \mathrm{Y}_2 &\triangleq \{\Phi_2(1), \ldots, \Phi_2(400)\} \\ &\triangleq \{AA, AR, AN, AD, AC, \ldots, VS, VT, VW, VY, VV\} \end{aligned} \tag{7}$$

$$\begin{aligned} \mathrm{Y}_3 &\triangleq \{\Phi_3(1), \ldots, \Phi_3(8000)\} \\ &\triangleq \{AAA, AAR, AAN, AAD, AAC, \ldots, VVS, VVT, VVW, VVY, VVV\} \end{aligned} \tag{8}$$

which represent all the possible combinations of two or three consecutive amino acids in this protein sequence. Two novel features introduced in this paper are:

$$\mathbf{H}_2 = [H_2(1), \ldots, H_2(L)] \tag{9}$$

$$\mathbf{H}_3 = [H_3(1), \ldots, H_3(L)] \tag{10}$$

which can be derived from the probability features $\mathbf{P}_2$ and $\mathbf{P}_3$, respectively, associated with two or three consecutive amino acids of the protein sequence $\hat{w} = w_1 \cdots w_L$. Using the notation $\delta(\cdot)$ function, $H_2(j)$ and $H_3(j)$ in Equations (9) and (10) for $1 \leq j \leq L$ can, respectively, be computed using:

$$H_2(j) = \begin{cases} N_2(I_2(j)) & j = 1 \\ \frac{1}{2}(N_2(I_2(j-1)) + N_2(I_2(j))) & 2 \le j \le L-1, \\ N_2(I_2(j-1)) & j = L \end{cases} \tag{11}$$

$$H_3(j) = \begin{cases} N_3(I_3(j)) & j = 1 \\ \frac{1}{2}(N_3(I_3(j-1)) + N_3(I_3(j))) & j = 2 \\ \frac{1}{3}(N_3(I_3(j-2)) + N_3(I_3(j-1)) + N_3(I_3(j))) & 3 \le j \le L-2 \\ \frac{1}{2}(N_3(I_3(j-2)) + N_3(I_3(j-1))) & j = L-1 \\ N_3(I_3(j-2)) & j = L\,. \end{cases} \tag{12}$$

In view of (7) and (8), functions $I_2(j)$ and $I_3(j)$ in (11) and (12) are defined as:

$$I_2(j) = \sum_{\Phi_2(k) \in \Upsilon_2} k\delta(\mathbf{w}_j - \Phi_2(k)) \tag{13}$$

$$I_3(j) = \sum_{\Phi_3(k) \in \Upsilon_3} k\delta(\bar{\mathbf{w}}_j - \Phi_3(k)) \tag{14}$$

where $\mathbf{w}_j$ and $\bar{\mathbf{w}}_j$, respectively, represent $w_j w_{j+1} (1 \le j \le L-1)$ and $w_j w_{j+1} w_{j+2} (1 \le j \le L-2)$. It is easy to verify $1 \le I_2(j) \le 400$ and $1 \le I_2(l) \le 8000$ for $1 \le j \le L-1$ and $1 \le l \le L-2$. Functions $N_2(\cdot)$ and $N_3(\cdot)$ in (11) and (12) defined over the sets $\{1, \ldots, 400\}$ and $\{1, \ldots, 8000\}$, respectively, are scaled probability features $\mathbf{P}_2$ and $\mathbf{P}_3$, with

$$N_2(k) = \begin{cases} \frac{P_2(k) - \min_{P_2(k) \in \mathbf{P}_2}\{P_2(k)\}}{\Delta_2} & \text{if} \quad \Delta_2 \ne 0 \\ 1 & \text{if} \quad \Delta_2 = 0 \end{cases} \qquad 1 \le k \le 400 \tag{15}$$

$$N_3(k) = \begin{cases} \frac{P_3(k) - \min_{P_3(k) \in \mathbf{P}_3}\{P_3(k)\}}{\Delta_3} & \text{if} \quad \Delta_3 \ne 0 \\ 1 & \text{if} \quad \Delta_3 = 0 \end{cases} \qquad 1 \le k \le 8000 \tag{16}$$

where we have $\Delta_2 = \max_{P_2(k) \in \mathbf{P}_2}\{P_2(k)\} - \min_{P_2(k) \in \mathbf{P}_2}\{P_2(k)\}$ and $\Delta_3 = \max_{P_3(k) \in \mathbf{P}_3}\{P_3(k)\} - \min_{P_3(k) \in \mathbf{P}_3}\{P_3(k)\}$. The probability features $\mathbf{P}_2$ and $\mathbf{P}_3$ associated with two and three consecutive amino acids of the protein sequence, respectively, are equal to:

$$\mathbf{P}_2 \triangleq \{P_2(1), \ldots, P_2(400)\} \triangleq \{\frac{S_2(1)}{\sum_{k=1}^{400} S_2(k)}, \ldots, \frac{S_2(400)}{\sum_{k=1}^{400} S_2(k)}\} \tag{17}$$

$$\mathbf{P}_3 \triangleq \{P_3(1), \ldots, P_3(8000)\} \triangleq \{\frac{S_3(1)}{\sum_{k=1}^{8000} S_3(k)}, \ldots, \frac{S_3(8000)}{\sum_{k=1}^{8000} S_3(k)}\} \tag{18}$$

In (17) and (18), we have:

$$S_2(k) = \sum_{\hat{w} \in \Omega} M_{\Phi_2(k)}^{\hat{w}} \qquad 1 \le k \le 400 \tag{19}$$

$$S_3(k) = \sum_{\hat{w} \in \Omega} M_{\Phi_3(k)}^{\hat{w}} \qquad 1 \le k \le 8000 \tag{20}$$

where we assume that the set of the protein sequences is denoted by Ω (in this paper, we have Ω = DIS1616). For a given protein sequence $\hat{w} = w_1 \cdots w_L$, the functions $M_{\Phi_2(k)}^{\hat{w}}$

and $M^{\hat{w}}_{\Phi_3(k)}$, which, respectively, count the total number of occurrences of a particular combination of two or three consecutive amino acids in $\hat{w}$, are equal to:

$$M^{\hat{w}}_{\Phi_2(k)} = \sum_{j=1}^{L-1} \delta(\mathbf{w}_j - \Phi_2(k)) \quad 1 \le k \le 400 \tag{21}$$

$$M^{\hat{w}}_{\Phi_3(k)} = \sum_{j=1}^{L-2} \delta(\bar{\mathbf{w}}_j - \Phi_3(k)) \quad 1 \le k \le 8000 \tag{22}$$

where $\mathbf{w}_j$ and $\bar{\mathbf{w}}_j$, respectively, represent $w_j w_{j+1} (1 \le j \le L-1)$ for $1 \le j \le L-1$ and $w_j w_{j+1} w_{j+2} (1 \le j \le L-2)$ for $1 \le j \le L-2$.

2.2.3. Pre-Processing the Data Extracted from the Protein Sequences

In this section, we illustrate how to compute the input of our deep neural network, which is composed of 35 features derived from a protein sequence. Of these 35 features, there are twenty evolutionary features which are determined through the PSSM [36] computed through the PSI-BLAST [37]. Seven physiochemical properties of the amino acids are steric parameter, polarizability, volume, hydrophobicity, isoelectric point, helix, and sheet probability, which can be obtained from the paper [35]. Three propensities of the amino acids are Remark 465, Deleage/Roux, and Bfactor (2STD), as detailed in the GlobPlot NAR paper [12]. The other two features used to measure the complexity of the protein sequence are Shannon entropy and topological entropy [32].

Given a protein sequence $\hat{w} = w_1 \cdots w_L$ of length L, we choose a sliding window of odd size N $(N < L)$ to extract N consecutive amino acids. Then, for these amino acids in the sliding window, we compute the evolutionary features, physiochemical properties, and propensities, as defined in the previous paragraph. These thirty computed feature values of amino acids in the sliding window are averaged and the averaged results are used to represent the feature values of the amino acid in the center of the sliding window. For simplicity, we first transform $\hat{w}$ into a sequence of size $L + N - 1$ through appending $(N-1)/2$ zeros to the both ends of the protein sequence. With this sliding window of size N, we also compute the Shannon and topological entropy through the procedure from Equations (1)–(14), as described in the paper [32], as well as the persistent entropy defined in (4). Thus, for each w_j with $1 \le j \le L$ in the protein sequence $\hat{w} = w_1 \cdots w_L$, we can combine it with a $33 \times L$ feature matrix

$$\mathbf{v}_j = [m_{1,j} \quad m_{2,j} \quad \ldots \quad m_{33,j}] \tag{23}$$

where $m_{k,j}$ for $1 \le k \le 20$, $21 \le k \le 27$, $28 \le k \le 30$ and $31 \le k \le 33$, respectively, align to a 20-dimensional PSSM of the evolutionary information [36,37], seven physiochemical properties [35], three propensities of amino acids from the paper [12], and three entropies (Shannon, topological [32], and persistent entropy). We also use Equations (11) and (12) to compute two novel features $H_2(j)$ and $H_3(j)$ $(1 \le j \le L)$ that are associated with two or three consecutive amino acids of the protein sequence and set $m_{34,j} = H_2(j)$ and $m_{35,j} = H_3(j)$. Finally, we modify the $33 \times L$ feature matrix defined in (23) to a $35 \times L$ feature matrix:

$$\mathbf{F} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \ldots \quad \mathbf{x}_L] \tag{24}$$

with

$$\mathbf{x}_j = [\mathbf{v}_j \quad m_{34,j} \quad m_{35,j}]^{\mathrm{T}} \quad 1 \le j \le L \tag{25}$$

where $\mathbf{v}_j$ $(1 \le j \le L)$ is defined in (23). The input to our deep neural network is $\mathbf{x}_j$ $(1 \le j \le L)$, as defined in (25).

3. The Structure of Our Neural Network and Training Procedure

In this section, we develop a deep neural structure composed of a variant VGG19, where the variant VGG19 is situated between two MLP networks used for identifying IDPRs. Then, we introduce the process of training the deep neural network.

3.1. The Structure of Our Deep Neural Network

The overall architecture of our model, as shown in Figure 1, is based on a variant VGG19 in cascade with two MLP networks, with the variant VGG19 being situated between two MLP networks. In the variant VGG19, we erase the fully connected (FC) layers of VGG19 but preserve the remaining VGG19 structure and its associated weights and biases.

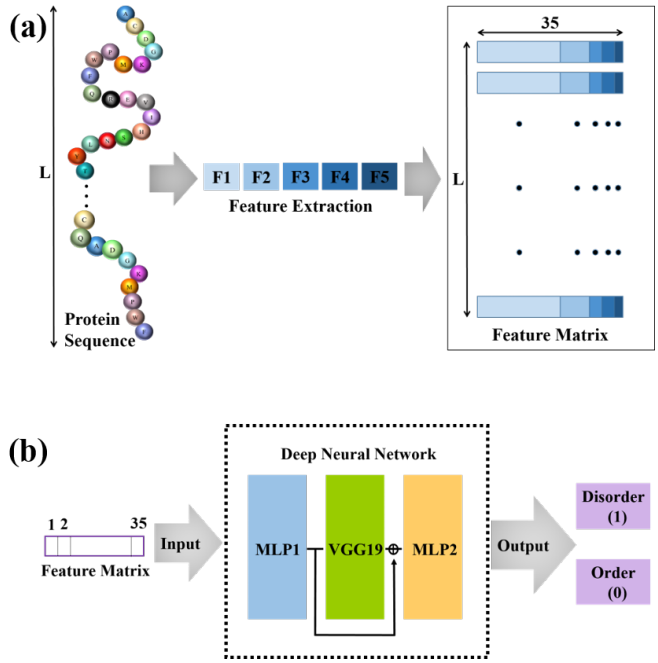

Figure 1. The overall framework for the prediction of intrinsically disordered proteins. (**a**) We extract five types of features from the protein sequence and obtain the feature matrix with 35 features for each amino acid. (**b**) The obtained feature matrix is input into the deep neural network. The output can be used to predict IDPRs.

Figure 2a depicts the structure of the MLP network whose outputs are fed as the inputs to the variant VGG19 . This MLP network with two hidden layers takes each column (i.e., 35×1 features) defined in (25) as its input and yields a 1×3675 vector as its output. The 1×3675 output vector of this MLP network is then mapped to a $35 \times 35 \times 3$ matrix through the reshape function of Keras, and this $35 \times 35 \times 3$ matrix is fed as the input to the variant VGG19. The two hidden layers contain 35 and 3675 neurons, respectively. The activation functions of neurons in this MLP are the rectified linear unit (ReLU).

The output of the variant VGG19 is a 1×3675 vector. As shown in Figure 2b, the skip connection is employed, where the sum of the output from the variant VGG19 and the output from the MLP network connecting to the features defined in (25) is fed as the input to a novel MLP network. This MLP network contains one hidden layer with 3675 neurons, whose activation functions are chosen to be the ReLU. The output layer has only 1 neuron with the sigmoid function as its activation function—i.e.,

$$\sigma(z_i) = \frac{1}{1 + e^{-z_i}} \tag{26}$$

where z_i $(1 \leq i \leq L)$ is the output of this sigmoid function and the index i is the i-th amino acid in the protein sequence $\hat{w} = w_1 \cdots w_L$. The dropout algorithm [40] with a dropout percentage of 50% is employed for this MLP network.

The total loss function of our model for a package of size m (i.e., the number of amino acids used in each iteration during the training) is therefore defined as:

$$L = \frac{1}{m} \sum_{i=1}^{m} -[y_i \ln(a_i) + (1 - y_i) \ln(1 - a_i)] . \tag{27}$$

In Equation (27), the predicted probability a_i of the output $y_i = 1$ is equal to:

$$a_i \triangleq \sigma(z_i) = \frac{1}{1 + e^{-z_i}} \tag{28}$$

where y_i is equal to either 1, suggesting that the i-th amino acid is disordered, or to 0, implying that it is ordered.

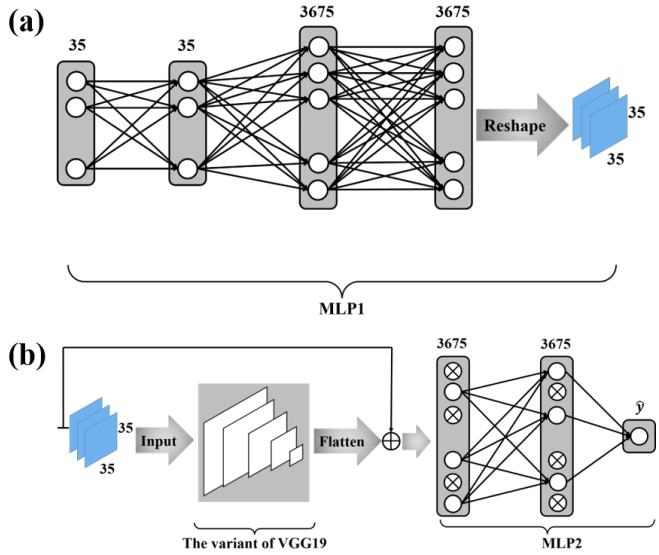

Figure 2. The deep neural network configuration. (**a**) is the first part of the deep neural network configuration. The function of MLP1 is to convert the protein sequence features into a mode suitable for VGG19 input. (**b**) is the second part of the deep neural network configuration. We use a variant of VGG19 for further feature extraction and MLP2 for classification. In MLP2, a dropout algorithm is used.

3.2. Training Procedure

In this section, we present the process of training the deep neural network developed in the previous section. The training dataset we use in this paper is DIS1450 from the DisProt [34]. We put all amino acids from all proteins in DIS1450 together and, based on this set, randomly divide them into packages of 128 amino acids. The training procedure is as follows: For each amino acid in a given package, we use the deep neural network constructed above to calculate the predicted probability defined in the Equation (28). When we have calculated all predicted probabilities for this given package, we can use the Equation (27) to estimate the average loss for the package. This computed averaged loss of the package is used to update the weights and biases of our network via a stochastic gradient descent (SGD) algorithm [41], where the learning rate $\eta = 0.0001$. We repeat the above process until all the packages have completed. We refer to this process as an epoch. Then, we repeat the above process until the loss function stops converging or reaches the maximum number of epochs.

3.3. Performance Evaluation

Four metrics were used to evaluate the performance of IDPR prediction [42]. These were sensitivity (*Sens*), specificity (*Spec*), balanced accuracy (*BACC*), and Matthews correlation coefficient (*MCC*). The related formulas are as follows:

$$Sens = \frac{TP}{TP + FN} \tag{29}$$

$$Spec = \frac{TN}{TN + FP} \tag{30}$$

$$BACC = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{31}$$

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \tag{32}$$

We use TP, FP, TN, and FN to represent the number of true positives, false positives, true negatives, and false negatives, respectively. The values of MCC can be any number between -1 and 1. The prediction accuracy for both ordered and disordered residue increases as the MCC value becomes closer and closer to 1.

4. Experimental Results

In this section, we will demonstrate the performance of our deep neural network on the different test sets: DIS166 [34], R80 [25], and MXD494 [33]. As a comparison, we also present the simulation results of the best known predictors for these datasets, such as RFPR-IDP (available at http://bliulab.net/RFPR-IDP/server (accessed on 26 March 2021)), SPOT-Disorder2 (available at https://sparks-lab.org/server/spot-disorder2/ (accessed on 26 March 2021)), DISvgg [16], and IDP-Seq2Seq [18]. For convenience, we refer to our method as MLP-VGG19-MLP. A ten-fold cross validation was performed on the training dataset DIS1450. The results of MLP-VGG19-MLP with different window sizes are shown in Table 1. In addition, the values achieved for $BACC$ and MCC with different sliding window sizes are shown in Figure 3. When the sliding window size was larger than 33, the values tended to be smooth. Thus, we used the sliding window size of $N = 33$ in subsequent simulations.

Table 1. Performance on dataset DIS1450 with different sliding window sizes.

Sliding Window Sizes	*Sens*	*Spec*	*BACC*	*MCC*
3	0.7471	0.5813	0.6642	0.2519
9	0.7972	0.6536	0.7164	0.3339
15	0.8192	0.6447	0.7319	0.3583
21	0.8183	0.6675	0.7492	0.3772
27	0.8233	0.6717	0.7475	0.3848
31	0.8183	0.6872	0.7527	0.3949
33	0.8125	0.7010	0.7568	0.4033
35	0.8100	0.7069	0.7585	0.4070
37	0.8679	0.6515	0.7597	0.4009
39	0.8266	0.6788	0.7527	0.3936
45	0.8214	0.6910	0.7562	0.4008

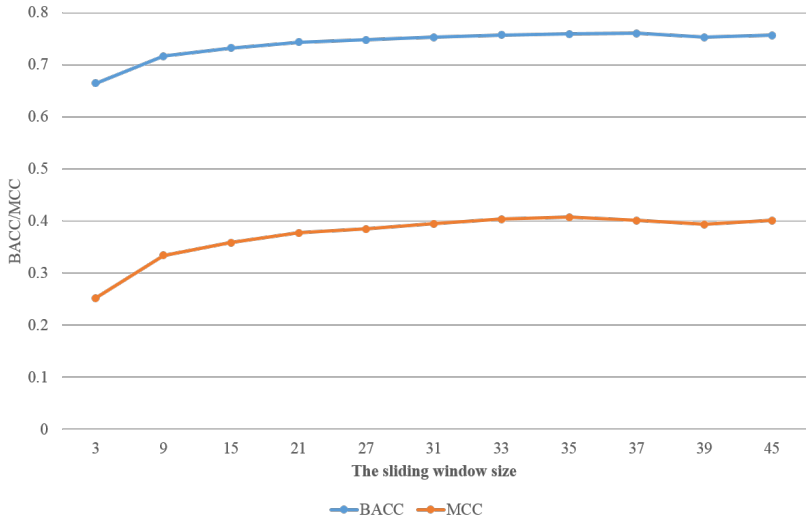

Figure 3. The performance with different sliding window sizes on *BACC* and *MCC*.

On the test sets DIS166, R80, and MXD494, the performance of MLP-VGG19-MLP was superior to that of RFPR-IDP, SPOT-Disorder2, and DISvgg. The *MCC* value of MLP-VGG19-MLP is 0.5674 on the test set DIS166, 0.5775 on the blind test set R80, and 0.4737 on the blind test set MXD494. The simulation results show that MLP-VGG19-MLP either considerably outperforms these methods or, when relying on a much smaller training dataset compared to the one used in [18], attains a performance similar to that of IDP-Seq2Seq [18]. Tables 2–4, respectively, present the performances of all these methods on test sets DIS166, R80, and MXD494.

Table 2. Performance of various methods on dataset DIS166.

Methods	*Sens*	*Spec*	*BACC*	*MCC*
MLP-VGG19-MLP	0.8351	0.8338	0.8345	0.5674
DISvgg	0.6713	0.8828	0.7710	0.5132
RFPR-IDP	0.7557	0.7817	0.7687	0.4406
SPOT-Disorder2	0.7103	0.8084	0.7594	0.4952
IDP-Seq2Seq	0.7890	0.8212	0.8051	0.5475

Table 3. Performance of various methods on blind test dataset R80.

Methods	*Sens*	*Spec*	*BACC*	*MCC*
MLP-VGG19-MLP	0.7269	0.9261	0.8265	0.5775
DISvgg	0.5993	0.9429	0.7711	0.5270
RFPR-IDP	0.5464	0.9546	0.7505	0.5139
SPOT-Disorder2	0.4941	0.9439	0.7190	0.4486
IDP-Seq2Seq	0.7787	0.9124	0.8456	0.5884

Table 4. Performance of various methods on blind test dataset MXD494.

Methods	*Sens*	*Spec*	*BACC*	*MCC*
MLP-VGG19-MLP	0.7169	0.8081	0.7625	0.4737
DISvgg	0.7160	0.7956	0.7558	0.4577
RFPR-IDP	0.7490	0.7580	0.7540	0.4420
SPOT-Disorder2	0.6380	0.8200	0.7290	0.4482
IDP-Seq2Seq	0.7430	0.7910	0.7670	0.4750

5. Conclusions

In this study, a deep neural structure is developed for identifying IDPRs, where a variant VGG19 is situated between two MLP networks. Furthermore, for the first time, three novel sequence features—i.e., persistent entropy, PCAA2, and PCAA3—are introduced for identifying IDPRs. In comparison with our previous DISvgg algorithm, the prediction performance of MLP-VGG19-MLP exceeded it. Furthermore, only one VGG19 was used in this paper, while ten VGG16nets were employed in the previous paper. In comparison with RFPR-IDP, SPOT-Disorder2, and IDP-Seq2Seq, MLP-VGG19-MLP relies on a much smaller training set to achieve a performance that is better or similar to that achieved using other methods. The simulation results show that our neural structure either considerably outperforms other known methods or, when relying on a much smaller training set, attains a similar performance. Three novel sequence features could be used as valuable sequence features in the further development of identifying IDPRs.

Author Contributions: Z.W. designed the study, carried out the data analysis, and drafted the manuscript. J.Z. participated in the design of the study and revised the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets DIS1616, DIS1450, and DIS166; models; and code can be found at the website https://github.com/ZakeWang/MLP_VGG19_MLP.git accessed on 9 January 2022.

Acknowledgments: We would like to thank the DisProt database (http://www.disprot.org/ accessed on 9 January 2022), which is the basis of our research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dyson, H.J.; Wright, P.E. Intrinsically unstructured proteins and their functions. *Nat. Rev. Mol. Cell Biol.* **2005**, *6*, 197–208. [CrossRef] [PubMed]
2. Iakoucheva, L.M.; Brown, C.J.; Lawson, J.D.; Obradović, Z.; Dunker, A.K. Intrinsic disorder in cell-signaling and cancer-associated proteins. *J. Mol. Biol.* **2002**, *323*, 573–584. [CrossRef]
3. Piovesan, D.; Tabaro, F.; Mičetić, I.; Necci, M.; Quaglia, F.; Oldfield, C.J.; Aspromonte, M.C.; Davey, N.E.; Davidović, R.; Dosztányi, Z. DisProt 7.0: A major update of the database of disordered proteins. *Nucleic Acids Res.* **2017**, *45*, D219–D227. [CrossRef] [PubMed]
4. Uversky, V.N. Functional roles of transiently and intrinsically disordered regions within proteins. *FEBS J.* **2015**, *282*, 1182–1189. [CrossRef]
5. Holmstrom, E.D.; Liu, Z.; Nettels, D.; Best, R.B.; Schule, R.B. Disordered rna chaperones can enhance nucleic acid folding via local charge screening. *Nat. Commun.* **2019**, *10*, 1–11. [CrossRef]
6. Sun, X.L.; Jones, W.T.; Rikkerink, E.H.A. Gras proteins: The versatile roles of intrinsically disordered proteins in plant signalling. *Biochem. J.* **2012**, *442*, 1–12. [CrossRef]
7. Uversky, V.N.; Oldfield, C.J.; Dunker, A.K. Intrinsically disordered proteins in human diseases: Introducing the D2 concept. *Annu. Rev. Biophys.* **2008**, *37*, 215–246. [CrossRef]
8. Uversky, V.N.; Oldfield, C.J.; Midic, U.; Xie, H.; Xue, B.; Vucetic, S.; Iakoucheva, L.M.; Obradovic, Z.; Dunker, A.K. Unfoldomics of human diseases: Linking protein intrinsic disorder with diseases. *BMC Genom.* **2009**, *10*, S7. [CrossRef]
9. Kulkarni, V.; Kulkarni, P. Intrinsically disordered proteins and phenotypic switching: Implications in cancer. *Prog. Mol. Biol. Transl. Sci.* **2019**, *166*, 63–84.
10. Kaya, I.E.; Ibrikci, T.; Ersoy, O.K. Prediction of disorder with new computational tool: BVDEA. *Expert Syst. Appl.* **2011**, *38*, 14451–14459. [CrossRef]
11. Prilusky, J.; Felder, C.E.; Zeev-Ben-Mordehai, T.; Rydberg, E.H.; Man, O.; Beckmann, J.S.; Silman, I.; Sussman, J.L. FoldIndex: A simple tool to predict whether a given protein sequence is intrinsically unfolded. *Bioinformatics* **2005**, *13*, 3435–3438. [CrossRef] [PubMed]
12. Linding, R.; Russell, R.B.; Neduva, V.; Gibson, T.J. Globplot: Exploring Protein Sequences for Globularity and Disorder. *Nucleic Acids Res.* **2003**, *31*, 3701–3708. [CrossRef] [PubMed]

13. Dosztanyi, Z.; Csizmok, V.; Tompa, P.; Simon, I. IUPred: Web server for the prediction of intrinsically unstructured regions of proteins based on estimated energy content. *Bioinformatics* **2005**, *21*, 3433–3434. [CrossRef] [PubMed]
14. Galzitskaya, O.V.; Garbuzynskiy, S.O.; Lobanov, M.Y. FoldUnfold: Web server for the prediction of disordered regions in protein chain. *Bioinformatics* **2006**, *22*, 2948–2949. [CrossRef]
15. Lobanov, M.Y.; Galzitskaya, O.V. The Ising model for prediction of disordered residues from protein sequence alone. *Phys. Biol.* **2011**, *8*, 35004. [CrossRef]
16. Xu, P.; Zhao, J.; Zhang, J. Identification of Intrinsically Disordered Protein Regions Based on Deep Neural Network-VGG16. *Algorithms* **2021**, *14*, 107. [CrossRef]
17. Liu, Y.; Wang, X.; Liu, B. RFPR-IDP: Reduce the false positive rates for intrinsically disordered protein and region prediction by incorporating both fully ordered proteins and disordered proteins. *Briefings Bioinform.* **2021**, *22*, 2000–2011. [CrossRef]
18. Tang, Y.J.; Pang, Y.H.; Liu, B. IDP-Seq2Seq: Identification of intrinsically disordered regions based on sequence to sequence learning. *Bioinformatics* **2020**, *36*, 5177–5186. [CrossRef]
19. Hanson, J.; Yang, Y.; Paliwal, K.; Zhou, Y. Improving protein disorder prediction by deep bidirectional long short-term memory recurrent neural networks. *Bioinformatics* **2017**, *33*, 685–692. [CrossRef]
20. Hanson, J.; Paliwal, K.K.; Litfin, T.; Zhou, Y. SPOT-Disorder2: Improved Protein Intrinsic Disorder Prediction by Ensembled Deep Learning. *Genom. Bioinform.* **2019**, *17*, 645–656. [CrossRef]
21. Jones, D.T.; Cozzetto, D. DISOPRED3: Precise disordered region predictions with annotated protein-binding activity. *Bioinformatics* **2015**, *31*, 857–863. [CrossRef] [PubMed]
22. Zhang, T.; Faraggi, E.; Xue, B.; Dunker, A.K.; Uversky, V.N.; Zhou, Y. SPINE-D: Accurate prediction of short and long disordered regions by a single neural-network based method. *J. Biomol. Struct. Dyn.* **2012**, *29*, 799–813. [CrossRef] [PubMed]
23. Walsh, I.; Martin, A.J.; Domenico, T.D.; Tosatto, S.C. ESpritz: Accurate and fast prediction of protein disorder. *Bioinformatics* **2012**, *28*, 503–509. [CrossRef]
24. Shimizu, K.; Hirose, S.; Noguchi, T. POODLE-S: Web application for predicting protein disorder by using physicochemical features and reduced amino acid set of a position-specific scoring matrix. *Bioinformatics* **2007**, *23*, 2337–2338. [CrossRef] [PubMed]
25. Yang, Z.R.; Thomson, R.; McNeil, P.; Esnouf, R.M. RONN: The bio-basis function neural network technique applied to the detection of natively disordered regions in proteins. *Bioinformatics* **2005**, *21*, 3369–3376. [CrossRef] [PubMed]
26. Peng, K.; Vucetic, S.; Radivojac, P.; Brown, C.J.; Dunker, A.K.; Obradovic, Z. Optimizing long intrinsic disorder predictors with protein evolutionary information. *J. Bioinform. Comput. Biol.* **2005**, *3*, 35–60. [CrossRef] [PubMed]
27. Mizianty, M.J.; Stach, W.; Chen, K.; Kedarisetti, K.D.; Disfani, F.M.; Kurgan, L. Improved sequence-based prediction of disordered regions with multilayer fusion of multiple information sources. *Bioinformatics* **2010**, *26*, 489–496. [CrossRef]
28. Kozlowski, L.P.; Bujnicki, J.M. MetaDisorder: A meta-server for the prediction of intrinsic disorder in proteins. *BMC Bioinform.* **2012**, *13*, 111. [CrossRef]
29. Schlessinger, A.; Punta, M.; Yachdav, G.; Kajan, L.; Rost, B. Improved disorder prediction by combination of orthogonal approaches. *PLoS ONE* **2009**, *4*, e4433. [CrossRef]
30. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2015**, arXiv:1409.1556.
31. He, H.; Zhao, J.; Sun, G. The Prediction of Intrinsically Disordered Proteins Based on Feature Selection. *Algorithms* **2019**, *12*, 46. [CrossRef]
32. He, H.; Zhao, J.X. A Low Computational Complexity Scheme for the Prediction of Intrinsically Disordered Protein Regions. *Math. Probl. Eng.* **2018**, *2018*, 8087391. [CrossRef]
33. Peng, Z.L.; Kurgan, L. Comprehensive comparative assessment of in-silico predictors of disordered regions. *Curr. Protein Pept. Sci.* **2012**, *13*, 6–18. [CrossRef] [PubMed]
34. Hatos, A.; Hajdu-Soltész, B.; Monzon, A.M.; Palopoli, N.; Álvarez, L.; Aykac-Fas, B.; Bassot, C.; Benítez, G.I.; Bevilacqua, M.; Chasapi, A.; et al. DisProt: Intrinsic protein disorder annotation in 2020. *Nucleic Acids Res.* **2020**, *48*, D269–D276. [CrossRef]
35. Meiler, J.; Muller, M.; Zeidler, A.; Schmaschke, F. Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks. *J. Mol. Model.* **2001**, *7*, 360–369. [CrossRef]
36. Jones, D.T.; Ward, J.J. Prediction of disordered regions in proteins from position specific score matrices. *Proteins Struct. Funct. Genet.* **2003**, *53*, 573–578. [CrossRef] [PubMed]
37. Pruitt, K.D.; Tatusova, T.; Klimke, W.; Maglott, D.R. NCBI Reference Sequences: Current status, policy and new initiatives. *Nucleic Acids Res.* **2009**, *37*, D32–D36. [CrossRef]
38. Atienza, N.; Gonzalez-Diaz, R.; Rucco, M. Persistent entropy for separating topological features from noise in vietoris-rips complexes. *J. Intell. Inf. Syst.* **2019**, *52*, 637–655. [CrossRef]
39. Edelsbrunner, H.; Harer, J. Persistent homology—A survey. *Contemp. Math.* **2008**, *453*, 257–282.
40. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Ournal Mach. Learn. Res.* **2014**, *15*, 1929–1958.
41. Bottou, L.; Curtis, F.E.; Nocedal, J. Optimization Methods for Large-Scale Machine Learning. *Siam Rev.* **2018**, *60*, 223–311. [CrossRef]
42. Monastyrskyy, B.; Fidelis, K.; Moult, J.; Tramontano, A.; Kryshtafovych, A. Evaluation of disorder predictions in CASP9. *Proteins* **2011**, *79*, 107–118. [CrossRef] [PubMed]