A Novel Machine-Learing Approach for the Prediction of Disorder Proteins

Bi Zhao, Chengbin Hu, Yi Li, and Yu Liang

Abstract

Many proteins or partial regions of proteins are lacking of stable and well-defined three-dimensional structures in vitro [1]. Understanding intrinsically disorder proteins (IDPs) or intrinsically disorder regions (IDRs) are significant for interpreting biological function as well as associating many diseases [2]. To predict intrinsically disordered amino acids, many computational tools have been developed [3]. Although around 70 disorder predictors have been invented, many existing predictors are limited on the characteristics of proteins or are not effective on the N- and C- terminal [4]. As the increasing of the datasets, the accuracy of predictors are challenged [5, 6]. Therefore, formulating new strategies on disorder protein prediction are critical [7]. Here, we propose studies to improve the predicted accuracy of disordered proteins and disordered regions. Several strategies, including Decision-tree based algorithm, Naive Bayes, multilayer preceptron, and Neural Network, were applied to generate a novel meta-strategy. By applying different strategies, the results suggested Random forest can improve the predicted accuracy significantly. The application of Convolution Neural Network can improve the predicted result up to 99 percentage. The convolutional neural network (CNN) can improve the performance of disordered protein prediction significantly.

1 Introduction

Around 70 percentage of Protein Data Bank (PDB) structures have some disordered residues, as well as about 25 percentage have IDRs more than 10 residues in length. The system studies of IDPs, not only revealed the abundance of IDRs but also linked IDRs with many diseases, including cancer, cardiovascular diseases, amyloidosis, neurodegenerative diseases, diabetes, etc [8]. Computational methods were widely used in the study of IDPs and IDRs because they are more efficient and economy compared with traditional experimental methods [9]. More than 70 disorder protein predictors have been invented so far [10]. However, they are challenged by limitation of protein characteristics and the increasing number of protein datasets [11]. One possible strategy to solve this problem is to use meta-strategy based methods to combine current exist individual predictors, because meta-strategy is able to combine those "orthogonal true predictions" and improve the final true prediction rate [12]. In addition, meta-strategies have been used in multiple field, such as protein fold recognition, protein secondary structure prediction, protein interaction, protein subcellular locations, post-translational modification, promoter prediction, and many others [13]. Therefore, we plan to develop a novel meta-strategy by integrating eight different individual predictors, including IUPred, DisEMBL, Espritz, RONN, PONDR@VXLT, PONDR-VSL2, Globplot, and Dispro [14, 15, 16]. The PONDR-VSL2, DisEMBL, and Globplot are based on artificial neural network [14, 15]; PONDR@VLXT is based on support vector machines [14]; Dispro used a combination of neural networks and Bayesian methods [14]; IUPred, Espritz and RONN have been developed by both physics-based methods and neural network [15, 16]. We choose these eight predictors is because two reasons: (1)These predictors are widely used in scientific research; (2) These predictors have well-maintained software packages or webservers. In addition to differences in the computational method being used, various predictors have different outputs from the prediction, including score for amino acids and score for composition of disordered amino acids. Hence, we will apply algorithms in weka to predict the disordered amino acid and use CNN to predict the classifier of proteins.

Many popular machine learning techniques are now used in protein prediction, Min et al. [17] proposed an alternating decision tree algorithm for assessing protein interaction reliability. Geng et al. [18] proposed the method using Naive Bayes to predict Protein-Protein interaction sites. Qi et al. [19] present a new method by constructing random forest to compute the similarities of protein to protein to classify pairs of proteins as interacting or not. Leo et al. [20] proposed the multilayer perceptron approach to predict protein secondary structures using different set of input features and network parameters in distributed computing environment.

Constitutional Neural Network (CNN) is widely used as image recognition classifier because of its high performance for image data. Recently, scientist also use CNN on protein prediction such as DNA–protein binding. Haoyang Zeng et al. [21] identified the best CNN architectures by varying CNN parameters, depth

and pooling designs. Because biological data can be very huge, CNN could be a good solution for classifying these data. In David R. Kelley et al.'s study [22], they trained a compendium of accessible genomic sites with DNase-seq data which yields a high accuracy than previous methods. To use CNN to process protein data, we can consider a protein data sequence as an image. Instead of processing image pixel data, we process a sequence of metrics of protein. We also can arrange the 1-D sequence of protein metrics to 2-D matrix data for CNN's processing.

We test the accuracy of different classifiers for single amino acid. We also rearranged the protein data to a matrix for CNN classification. Our main contributions are summarized here:

- We test different single amino acid classifier and select Random Forest as the best one which has 93.3497% accuracy.
- We deploy CNN for the whole protein data classification. We got 99.25% on classifying protein to four disorder protein categories.

The rest of this paper is organized as follows. Section 2 provides the methodology for solving the classification problem. Section 3 shows the experimental results. Section 4 concludes our study with future work.

2 Methods

2.1 Datasets Preparation

The quality of datasets are essential for performance of disorder predictors. Usually, it is hard to decide which datasets will be used in developing predictor because the dataset should include as many as existing data and cause as little as noise. In addition, each dataset should keep independent with others. In this study, we generate three independent datasets, of which, one dataset was used as training set and the other two datasets were used as testing. The training set is composed from the Database of Protein Disorder (DisProt) (http://www.disprot.org), version 7 v0.3 of September 26, 2016. The sequences were clustered by CD-HIT with 30 percentage sequence identity [23]. The reason we use 30 percentage identity as cutoff value is because usually researchers treat sequences less than 30 percentage identity as functional and evolutionary unrelated. 802 protein sequences with true disordered amino acid residues compared with all residues 92360/408691 was obtained. The first independent training set is obtained from PDB (http://www.rcsb.org/pdb/) database of February 28, 2017 by choosing single chain and more than 15 models structures. Structures having ligands or disulfide bonds were removed from dataset. The same sequences with training set were removed, then remaining sequences were clustered by CD-HIT with 30 percentage sequences identity. By applying the distribution of RMSD value of each residue, threshold value 5 was selected as cutoff to determine whether this residue is disordered or structured. From the calculation, 3165 chains have been obtained and the number of true disordered residues compared with all residues in this dataset is 37189/327443. The second independent testing set is comprised by RCSB Protein Data Bank (PDB) (http://www.rcsb.org/pdb/) X-ray dataset prepared by Dunbrack et al, which was updated on March 2, 2017. After removing the same sequences with both training set and first testing set, the remaining dataset contains 10525 chains and 2579528 amino acid residues. We treated missing residues in PDB file as experimental validated disorder residues because it is hard to obtain the structure of disorder residues. The number of true disordered amino acids in this dataset is 143860. All of the disordered amino acids were labeled by "D" and the structured amino acids were labeled by "O". This is also the predicted target for single amino acid classifier. All of these three datasets are predicted by eight individual predictors—IUPred, Disembl, Espritz, RONN, PONDR@VLXT, PONDR-VSL2, Globplot, and Dispro [14, 15, 16]. The predicted values of amino acids from these eight predictors were used as eight different attributes.

2.2 Single Amino Acid Classifier

Protein contains many amino acids which were labeled by "D" and "O". Therefore, single amino acid approaches were first proposed using Weka with 10-fold cross-validation. Here, each amino acid was treated as an instance. There are 408,691 instances in training set, containing 8 attributes. In our single amino acid approach, we use 4 different classifiers in Weka with 10 fold cross validation. The process of our single amino acid approach is shown in Figure 1.

In the single amino acid approach, we use 4 different classifiers, which are Decision Tree-J48, Naive Bayes, Random Forest and Multilayer Perceptron. Here we will just give a brief description of each classifiers.

We first use Decision tree-J48, it is the implementation of algorithm ID3 (Iterative Dichotomiser 3) developed by the WEKA project team. J48 is a popular algorithm used to generate a decision tree developed by Ross

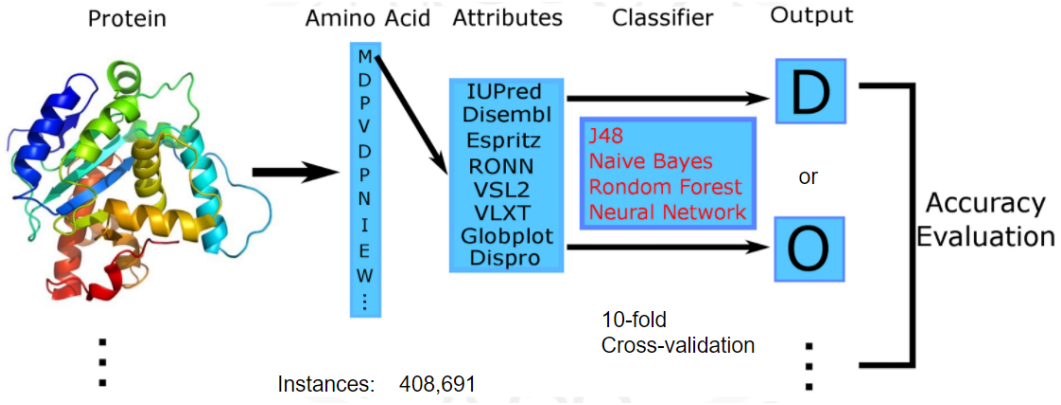

Figure 1: Workflow for Single Amino Acid Approach

Quinlan. Decision trees require relatively little effort from users for data preparation. In order to classify a new item, it first creates a decision tree based on the attributes from the training dataset and predicts the value of a target variable.

The Naive Bayes classifier works well for independent attributes, which is based on the Bayes rule of conditional probability. It will consider each the attributes separately when classifying a new instance.

Random Forest is an ensemble machine learning method for classification, which constructs multiple decision trees at training stage. It is fast and accurate and is better compared with J48 on a single decision tree. Random forest also corrects the overfitting problem of decision tree. It can used some trick to perform better, such as, using bagging to create training set each time and randomly pick some features and then use information gain to pick the best.

Multilayer Perceptron is a type of neural network that usually consists of at least three layers of node. The node in each layer uses nonlinear activation function. It is different from linear perceptron, in that between the input and the output layer, there can be one or more non-linear layers, called hidden layers. It uses back-propagation in the training process.

2.3 Protein Convolutional Neural Network

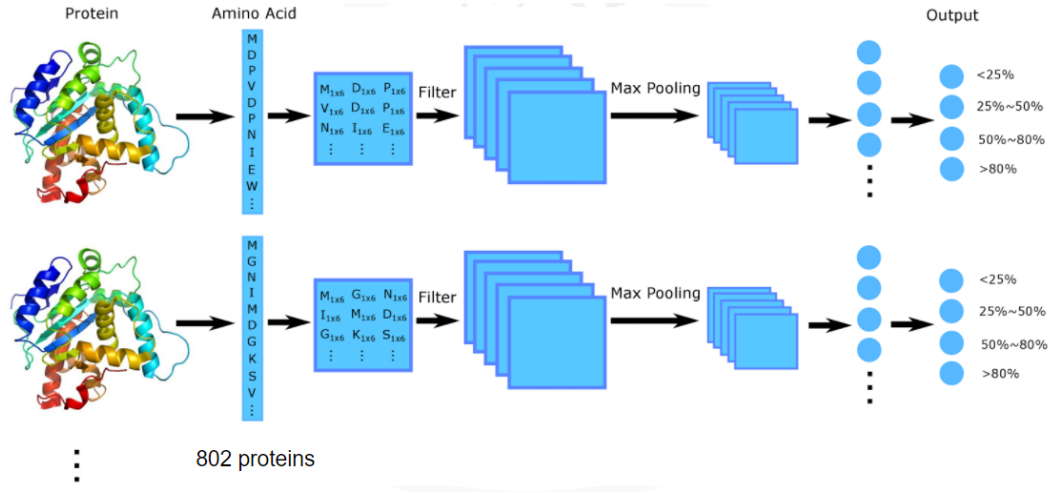

Figure 2: Workflow for Convolutional Neural Network Approach

To classify the whole protein, we choose 4 labels for each protein based on the ratio of discorded region as shown in Table 1. We use a 2-D CNN to take the input of the protein data. For each protein, we extract the eight attributes of every amino acid and fit them into a vector. Because the length of the protein ranges from 50 to more than 2000. We choose the vector size as 12000 which is enough to contain 1500 amino acids' data. If the protein is shorter than 1500, we use padding to make the data vector at 12000. If the protein is longer than 1500, only first 1500 amino acids are included into the data vector. After the data vector is constructed,

we reshape the data to a 2-D 120×100 matrix for CNN input. The process for whole protein CNN is shown in Figure 2. In CNN, we first use several filters to extract feature maps of the original data. Then we use max pooling layer to reduce the data dimension. After one or several CNN/max pooling layers the data go through classify layer which is fully connected neural network. The final class can be read from the classifier output. During our experiments, we test different architectures of CNN by tuning the convolutional layers, filters, hidden layers. We report accuracy and training time for each models. All the CNN models are built with Tensorflow flame work on GAIVI gpu cluster with GTX TITAN X as our training hardware.

Table 1: Classes of disorder protein

Class	1	2	3	4
Disorder region	0∼25%	25∼50%	50∼80%	80∼100%

3 Results

3.1 The Performances of individual predictors and meta-predictors

Each individual predictor was further analyzed by performing the distribution of true samples and false samples. Figure 3 display the fraction of true positive rate and the fraction of true negative rate. The distribution of these predictors suggested the predictors have their limitation in the range 0-0.1 and 0.9-1.0. This table presented the performance of individual predictors and meta-predictors. The accuracy of all of these predictors are around 0.65-0.77. Most importantly, all of these predictors perform the low MCC and F-measure value. Hence, it requires better algorithms used in disordered protein predictor.

Table 2: Comparison of current predictors

Predictors	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	Accuracy
IUPred	0.627	0.239	0.434	0.627	0.513	0.222	0.759	72.3%
Disemble	0.390	0.146	0.439	0.390	0.413	0.205	0.704	72.2%
Espritz	0.460	0.108	0.554	0.460	0.503	0.310	0.826	76.8%
RONN	0.747	0.356	0.380	0.747	0.504	0.158	0.750	67.3%
VSL2	0.796	0.401	0.367	0.796	0.502	0.137	0.769	65.5%
VLXT	0.614	0.386	0.651	0.317	0.425	0.227	0.684	64.5%
Globplot	0.564	0.312	0.345	0.564	0.428	0.132	0.670	65.2%
Dispro	0.353	0.104	0.497	0.353	0.413	0.241	0.734	74.0%
PONDRFIT	0.662	0.263	0.424	0.662	0.517	0.211	0.714	71.4%
MFDp	0.737	0.284	0.431	0.737	0.544	0.211	0.726	72.2%
Dismeta	0.629	0.229	0.445	0.629	0.521	0.232	0.733	73.6%

3.2 The performances of Single Amino Acid Classifier

We run 4 classifier in Weka with 10 fold cross-validation. We first evaluate 4 different single amino acid classifiers separately, then we will compare between them.

The accuracy we get from J48 is 87.6631%, the confusion matrix table is shown in Table 3. The first row is the classified label and the first column is the real label. See from this table, the diagnal denotes the correct classifications. We can also get the true positive, false positive, true negative, and false negative information out of this table.

Table 3: Confusion Matrix of J48.

	D	O
D	58372	33988
O	16432	299899

The next is Naive Bayes, the accuracy is 78.0744%. The confusion matrix is shown in table 4.

Then we run Random Forest, the accuracy we get is 93.3497%. Table 5 shows the confusion matrix of random forest classifier.

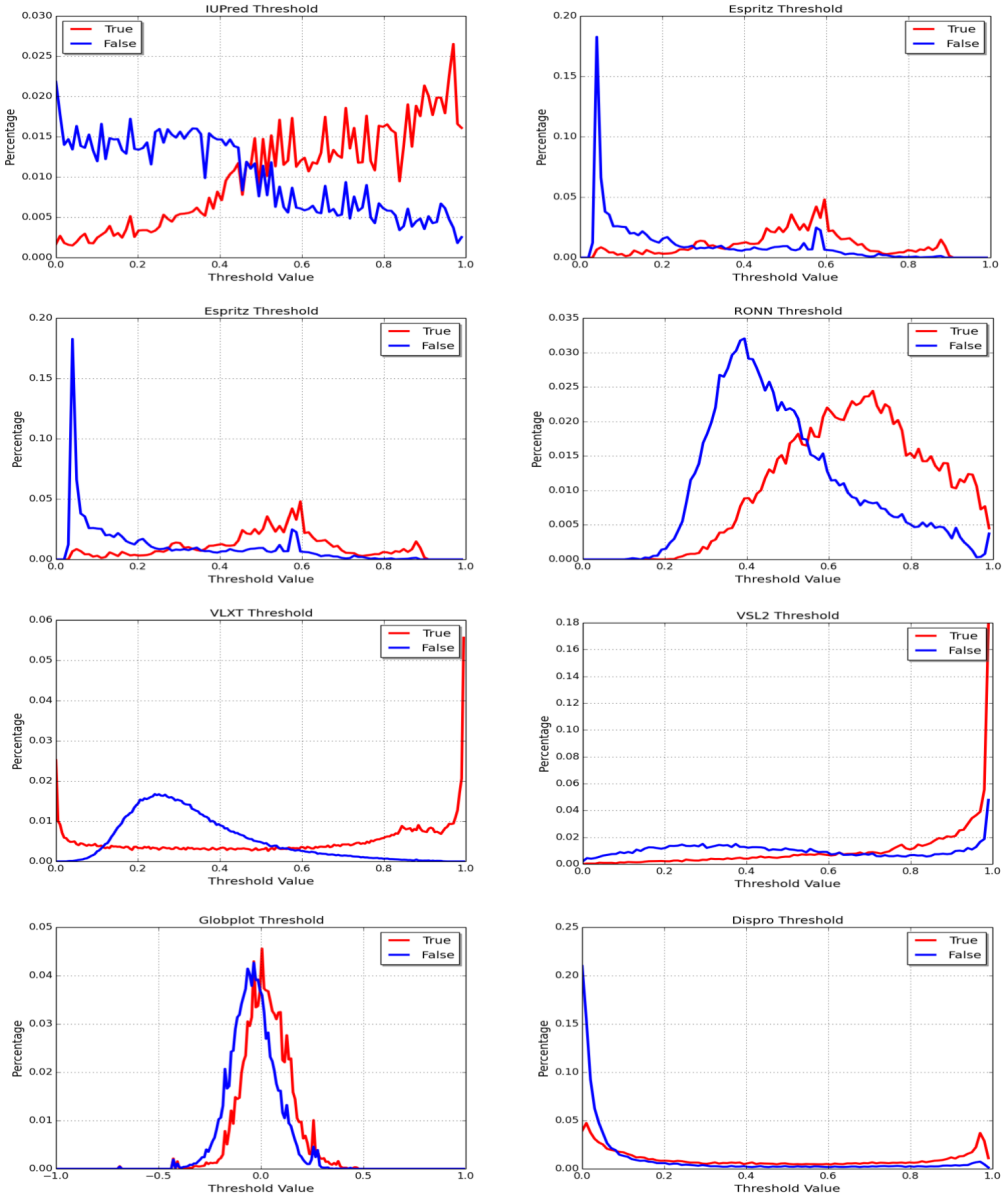

Figure 3: The performance of individual predictors and meta-predictors

Table 4: Confusion Matrix of Naive Bayes.

	D	O
D	60111	32249
O	57359	258972

Table 5: Confusion Matrix of Random Forest.

	D	O
D	71199	21161
O	6018	310313

The last classifier we use is Multilayer Perceptron, the accuracy we get is 82.1334%. The confusion matrix is shown in Table 6

To evaluate these four classifiers together, we first compare the accuracy alone. Figure 4 show the accuracy of all four classifiers. We can see that as the size of training data set increases, the accuracy also increases. Among the four classifiers, Random Forest has the highest accuracy, then is J48 Decision Tree, then is Multilayer

Table 6: Confusion Matrix of Multilayer Perceptron.

	D	O
D	43698	48662
O	24357	291974

Perceptron(NN), and Naive Bayes has the lowest accuracy. The reason Naive Bayers is the worst performance among these four classifiers could be that the Naive Bayes is good for handling independent attributes, whereas in our dataset, the attribute are somehow correlated.

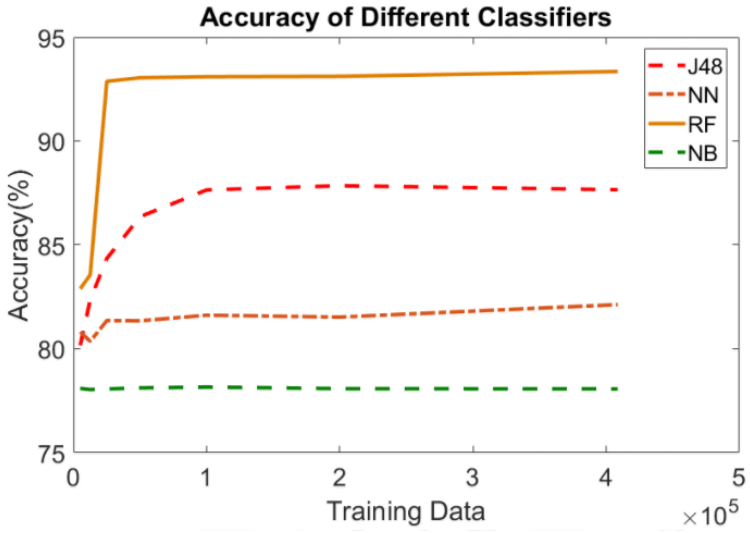

Figure 4: Accuracy Comparison Among Four Classifiers.

Aside from just compare the accuracy, we also use other metrics to compare them, such as false positive rate, F-Measure, MCC and ROC area. The comparison result is shown in Table 7. The FP rate and TP rate are common measures in machine learning, the TP rate is the higher the better and the FP rate is the lower the better. Precision is defined as the number of true positives(TP) over the number of true positives(TP) plus the number of false positives (FP). Recall is defined as the number of true positives (TP) over the number of true positives (TP) plus the number of false negatives (FN). The F-Measure is a measure of a test's accuracy and is defined as the weighted harmonic mean of the precision and recall of the test. The Matthews correlation coefficient (MCC) is used in machine learning as a measure of the quality of binary (two-class) classifications, the value are more approximate to positive 1 represents a perfect prediction. The ROC curve is created by plotting the true positive rate against the false positive rate at various threshold settings. We can see that Random Forest has the highest value of MCC, ROC, F-Measure, etc. This demonstrates that Random Forest has the best performance among all four classifiers.

Table 7: Comparison.

Classifier	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	Accuracy
J48	0.877	0.297	0.872	0.877	0.872	0.627	0.873	87.67%
NB	0.781	0.311	0.804	0.781	0.789	0.434	0.814	78.07%
RF	0.933	0.182	0.933	0.933	0.931	0.803	0.973	93.35%
NN	0.821	0.425	0.809	0.821	0.811	0.445	0.843	82.13%

3.3 The Performances of Whole Protein CNN classifier

We constructed a series of different CNN architectures by changing three parameters: the number of filters, the number of layers and the unit number of fully connected hidden layers. The results are shown in Table 8.

We can see that we got highest accuracy of 99.252% when there is 1 convolutional layer with 8 filters, 512 units in hidden layer

Table 8: The accuracy and training time of different CNN models

Convolution Layers	Number of Filters	Fully connected units	Training Time	Accuracy
1	8	512	111.2018s	99.252%
1	16	512	133.1798s	99.012%
1	8	1024	146.574	99.127%
2	8,16	512	99.7634s	98.628%
2	8,16	1024	132.1231s	98.201%
2	2,4	512	56.3512	66.708%
3	8,16,32	512	142.351	92.808%

4 Conclusions

Currently, there are more than 70 predictors created to predict the IDP. However, all of them have the limitation on N- and C- terminal prediction. In addition to the differences of outputs from predictors, the outputs include both single amino acid classifier and fraction of disorder amino acid in whole proteins. Different outputs are used in different research directions based on whether the goal is to study the biological systems and protein functions. In this project, we first proposed single amino acid approach. In this approach, we treat each amino acid as an instance and use four different classifiers, J48, Naive Bayes, Random Forest, and Multilayer Perceptron, to run in Weka with 10 fold cross-validation. We evaluated which classifier has the best performance for single amino acid classification. From our results, we conclude that Random forest has the best performance in single amino acid classification, which has accuracy of 93.35% . Then we proposed Convolutional Neural Network. We choose four labels for each protein to do whole protein classification. When using 1 convolutional layer with 8 filters and 1 fully connected layer with 0.5 dropout, we get the accuracy of 99.25%, which is the best result we get. In the future, we want to test with more different classifiers and develop a Web API that integrates all the classifiers we test.

References

[1] Vladimir N Uversky, Joel R Gillespie, and Anthony L Fink. Why are "natively unfolded" proteins unstructured under physiologic conditions? *Proteins: structure, function, and bioinformatics*, 41(3):415–427, 2000.

[2] A Keith Dunker, Celeste J Brown, J David Lawson, Lilia M Iakoucheva, and Zoran Obradović. Intrinsic disorder and protein function. *Biochemistry*, 41(21):6573–6582, 2002.

[3] Zoran Obradovic, Kang Peng, Slobodan Vucetic, Predrag Radivojac, Celeste J Brown, and A Keith Dunker. Predicting intrinsic disorder from amino acid sequence. *Proteins: Structure, Function, and Bioinformatics*, 53(S6):566–572, 2003.

[4] Jianzong Li, Yu Feng, Xiaoyun Wang, Jing Li, Wen Liu, Li Rong, and Jinku Bao. An overview of predictors for intrinsically disordered proteins over 2010–2014. *International journal of molecular sciences*, 16(10):23446–23462, 2015.

[5] Zsuzsanna Dosztányi, Veronika Csizmok, Peter Tompa, and István Simon. Iupred: web server for the prediction of intrinsically unstructured regions of proteins based on estimated energy content. *Bioinformatics*, 21(16):3433–3434, 2005.

[6] Avner Schlessinger, Jinfeng Liu, and Burkhard Rost. Natively unstructured loops differ from other loops. *PLoS computational biology*, 3(7):e140, 2007.

[7] Rune Linding, Robert B Russell, Victor Neduva, and Toby J Gibson. Globplot: exploring protein sequences for globularity and disorder. *Nucleic acids research*, 31(13):3701–3708, 2003.

[8] Zoltán Gáspári, Dániel Süveges, András Perczel, László Nyitray, and Gábor Tóth. Charged single alpha-helices in proteomes revealed by a consensus prediction approach. *Biochimica et Biophysica Acta (BBA)-Proteins and Proteomics*, 1824(4):637–646, 2012.

[9] Daniel Fischer. 3d-shotgun: a novel, cooperative, fold-recognition meta-predictor. *Proteins: Structure, Function, and Bioinformatics*, 51(3):434–441, 2003.

[10] Masahito Ohue, Yuri Matsuzaki, Takehiro Shimoda, Takashi Ishida, and Yutaka Akiyama. Highly precise protein-protein interaction prediction based on consensus between template-based and de novo docking methods. In *BMC proceedings*, volume 7, page S6. BioMed Central, 2013.

[11] Ji Wan, Shuli Kang, Chuanning Tang, Jianhua Yan, Yongliang Ren, Jie Liu, Xiaolian Gao, Arindam Banerjee, Lynda BM Ellis, and Tongbin Li. Meta-prediction of phosphorylation sites with weighted voting and restricted grid search parameter selection. *Nucleic acids research*, 36(4):e22–e22, 2008.

[12] Jaime Prilusky, Clifford E Felder, Tzviya Zeev-Ben-Mordehai, Edwin H Rydberg, Orna Man, Jacques S Beckmann, Israel Silman, and Joel L Sussman. Foldindex©: a simple tool to predict whether a given protein sequence is intrinsically unfolded. *Bioinformatics*, 21(16):3435–3438, 2005.

[13] Rune Linding, Lars Juhl Jensen, Francesca Diella, Peer Bork, Toby J Gibson, and Robert B Russell. Protein disorder prediction: implications for structural proteomics. *Structure*, 11(11):1453–1459, 2003.

[14] Nicolas Guex and Manuel C Peitsch. Swiss-model and the swiss-pdb viewer: an environment for comparative protein modeling. *electrophoresis*, 18(15):2714–2723, 1997.

[15] Zheng Rong Yang, Rebecca Thomson, Philip McNeil, and Robert M Esnouf. Ronn: the bio-basis function neural network technique applied to the detection of natively disordered regions in proteins. *Bioinformatics*, 21(16):3369–3376, 2005.

[16] Ian Walsh, Alberto JM Martin, Tomàs Di Domenico, and Silvio CE Tosatto. Espritz: accurate and fast prediction of protein disorder. *Bioinformatics*, 28(4):503–509, 2011.

[17] Min Su Lee and Sangyoon Oh. Alternating decision tree algorithm for assessing protein interaction reliability. *Vietnam Journal of Computer Science*, 1(3):169–178, 2014.

[18] Haijiang Geng, Tao Lu, Xiao Lin, Yu Liu, and Fangrong Yan. Prediction of protein-protein interaction sites based on naive bayes classifier. *Biochemistry research international*, 2015, 2015.

[19] Yanjun Qi, Judith Klein-Seetharaman, and Ziv Bar-Joseph. Random forest similarity for protein-protein interaction prediction from multiple sources. In *Biocomputing 2005*, pages 531–542. World Scientific, 2005.

[20] T Ramkumar et al. Analysis of multilayer perceptron machine learning approach in classifying protein secondary structures. *Biomedical Research*, 2016.

[21] Haoyang Zeng, Matthew D Edwards, Ge Liu, and David K Gifford. Convolutional neural network architectures for predicting dna–protein binding. *Bioinformatics*, 32(12):i121–i127, 2016.

[22] David R Kelley, Jasper Snoek, and John L Rinn. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome research*, 26(7):990–999, 2016.

[23] Ying Huang, Beifang Niu, Ying Gao, Limin Fu, and Weizhong Li. Cd-hit suite: a web server for clustering and comparing biological sequences. *Bioinformatics*, 26(5):680–682, 2010.