Received: 3 January 2019 | Revised: 18 November 2019 | Accepted: 6 June 2020

DOI: 10.1002/prot.25966

PROTEINS WILEY

RESEARCH ARTICLE

cnnAlpha: Protein disordered regions prediction by reduced amino acid alphabets and convolutional neural networks

Mauricio Oberti[1,2] | Iosif I. Vaisman[1]

[1]School of Systems Biology, George Mason University, Manassas, Virginia

[2]Novartis Institutes for BioMedical Research, Cambridge, Massachussets

Correspondence
Iosif I. Vaisman, School of Systems Biology, George Mason University, 10900 University Blvd, MS 5B3, Manassas, VA 20110.
Email: ivaisman@gmu.edu

Abstract

Intrinsically disordered regions (IDR) play an important role in key biological processes and are closely related to human diseases. IDRs have great potential to serve as targets for drug discovery, most notably in disordered binding regions. Accurate prediction of IDRs is challenging because their genome wide occurrence and a low ratio of disordered residues make them difficult targets for traditional classification techniques. Existing computational methods mostly rely on sequence profiles to improve accuracy which is time consuming and computationally expensive. This article describes an ab initio sequence-only prediction method—which tries to overcome the challenge of accurate prediction posed by IDRs—based on reduced amino acid alphabets and convolutional neural networks (CNNs). We experiment with six different 3-letter reduced alphabets. We argue that the dimensional reduction in the input alphabet facilitates the detection of complex patterns within the sequence by the convolutional step. Experimental results show that our proposed IDR predictor performs at the same level or outperforms other state-of-the-art methods in the same class, achieving accuracy levels of 0.76 and AUC of 0.85 on the publicly available Critical Assessment of protein Structure Prediction dataset (CASP10). Therefore, our method is suitable for proteome-wide disorder prediction yielding similar or better accuracy than existing approaches at a faster speed.

KEYWORDS

convolutional neural networks, disordered proteins, machine learning

1 | INTRODUCTION

Intrinsically disordered proteins (IDP) or intrinsically disordered regions (IDR) are segments within a protein chain lacking a stable three-dimensional structure under normal physiological conditions. They have been known to scientists for over 50 years and since then, linked to key biological processes including regulation of transcription, signal transduction, cell cycle control, post-translational modifications, ligand binding, protein interaction, and alternative splicing.[1,2] Disorder regions exist in up to half of the amino acids in eukaryotic proteins.[3] At least 6% of all residues in SwissProt are believed to be within disordered regions.[4]

Experimental structure resolution of IDP/IDRs is complex, lengthy and expensive. DisProt database,[5] a community resource annotating protein sequences for intrinsically disordered regions, currently contains just over 800 proteins. A large number of computational prediction methods have been developed[6,7] because of this inherent complexity. Existing methods can be classified in one of the following categories[8]: (i) Ab initio or sequence based. They rely almost exclusively on amino acid sequence information to make a prediction. Features extracted from the primary sequence, alignment profiles or scoring matrices are used as input for statistical models which then make predictions of disorder regions. Generally, methods that do not rely on complex external sources of information fall into this category and are referenced as sequence-only. (ii) Clustering. This approach generates tertiary structure models from the primary sequence. It then superimposes the different models onto each other with the

assumption that positions in ordered regions will be conserved across models. (iii) Template based. Similar to clustering, template based method predicts disordered regions of proteins by aligning the input sequence to homologous proteins with a known structure. Homologous proteins are found by doing a database search or by fold recognition methods. (iv) Meta or consensus. They combine the output of several disordered predictors into a single average, which tends to have a moderate increase in accuracy. Evolutionary information contained in sequence profiles helps ab initio methods to improve prediction accuracy. However, generating sequence profiles is time consuming and methods relying on them for predictions may not be suitable for large proteome-wide analysis.

This article presents a sequence-only ab initio method for predicting protein disorder based on reduced amino acid alphabets and convolutional neural networks (cnnAlpha). Our method relies solely on the amino acid sequence for determining disorder positions and is aimed to proteome-wide applications where speed and low false positive rate are prioritized over maximum accuracy.[9]

Among the main challenges with sequence based prediction methods are (a) the highly class imbalance nature of the datasets and (b) the difficulty in accurately capturing the interdependency of adjacent residues in determining the transitions between disorder and order states. If not addressed, a class imbalance can severely bias predictions toward the majority class (order state). To solve the imbalance problem, we choose an undersampling technique where we randomly remove examples from the majority class until we have a balanced dataset. Undersampling has been proven to be highly successful yielding a positive performance within the context of convolutional networks and extreme ratio imbalance datasets.[10] In order to capture local sequence context, we use a sliding window approach which feeds into a convolutional neural network that is tasked with learning rich higher-order sequence features.

Convolutional neural networks proved to be very efficient and well performing in the field of computer vision, excelling in tasks such as object detection and image classification.[11] The adaptation of convolutional neural networks architectures for biological problems has been successful in the context of DNA-protein binding prediction[12] and DNA function modeling.[10] Reducing the amino acid alphabet from 20 to 3 letters enables a seamless adaptation of convolutional neural networks for protein models. Instead of analyzing 2-D images with three color channels (R, G, B), fixed length protein sequence windows are mapped to 1-D input vectors with three channels. This translation allows mapping the protein disorder prediction problem to the 2-class image classification problem in the computer vision domain.

2 | METHODS AND MATERIALS

2.1 | Disorder definition and feature extraction

There is no universal agreement on how to define disorder residues from PDB files.[13] In the context of this work, we consider a residue to be in a disorder position if it appears in the sequence records, but its coordinates are missing from the electron density map. We annotated our PDB training and CAMEO validation sets using this definition. The annotation provided by the CASP experiments[14] was created using a similar definition. This is not a perfect definition since there are other reasons why a residue can have missing coordinates (i.e., crystallization artifacts). However, it allows us to use a large number of proteins from PDB without further experimental validation.

The primary sequences from our training set had to be translated to numerical features to be fed into the convolutional network. For that purpose, we implemented a 101-residue length sliding window centered on the target residue. The window length was set after experimenting with different sizes, finding that larger windows were more consistent in capturing disorder information. For each window, residues are represented by letters from the reduced amino acid alphabet and encoded using a one-bit hot encoding scheme. This generates a 3-D input feature matrix per target residue of size $[3 \times 101]$. This process is illustrated in Figure 1.

2.2 | Reduced alphabets

Reduced alphabets cluster residues in ways that prevent the loss of key biochemical information. The 20-letter amino acid alphabet was reduced to a 3-letter alphabet in order to simplify and quicken the network learning process, reducing the number of possible encodings and size of the input feature vectors. The reduced alphabets were selected from the literature (Table 1), where each was designed with a specific structural protein task in mind. In each alphabet, residues are clustered based on various properties, including chemical and genetic properties.

We found that 3-letter alphabets provide a reasonable balance between limiting the complexity of the sequence space and maintaining the model's ability to efficiently predict disordered residues. Higher-order alphabets (in particular between 4 and 10 letters) better characterize the complexity in proteins.[16] In our case, their usage increases the number of trainable parameters, complexity of the network model, and require larger training sets to converge. This is in part supported by the results in Table 4, where the performance of the model without alphabet reduction is consistently below the models using a reduction step. A comprehensive search of published alphabets and groupings is beyond the scope of this work and might be addressed in future studies.

Alphabets 1, 2, and 6 performed better in our specific classification task. Alphabet 1 achieves the reduction by mismatch minimization between the reduced interaction matrix and the Miyazawa and Jernigan (MJ) matrix. Alphabet 2 identifies the reduced alphabet which simplified sequence performs best in the context of protein fold recognition using global sequence alignments with the parent sequence. Alphabet 6 implements an automated reduction protocol using information theory metrics tailored to the prediction of solvent accessibility.

2.3 | Convolutional neural network architectures

The convolutional neural network architectures used in our models are variations of Figure 2. The input is a $3 \times L$ matrix where L is the

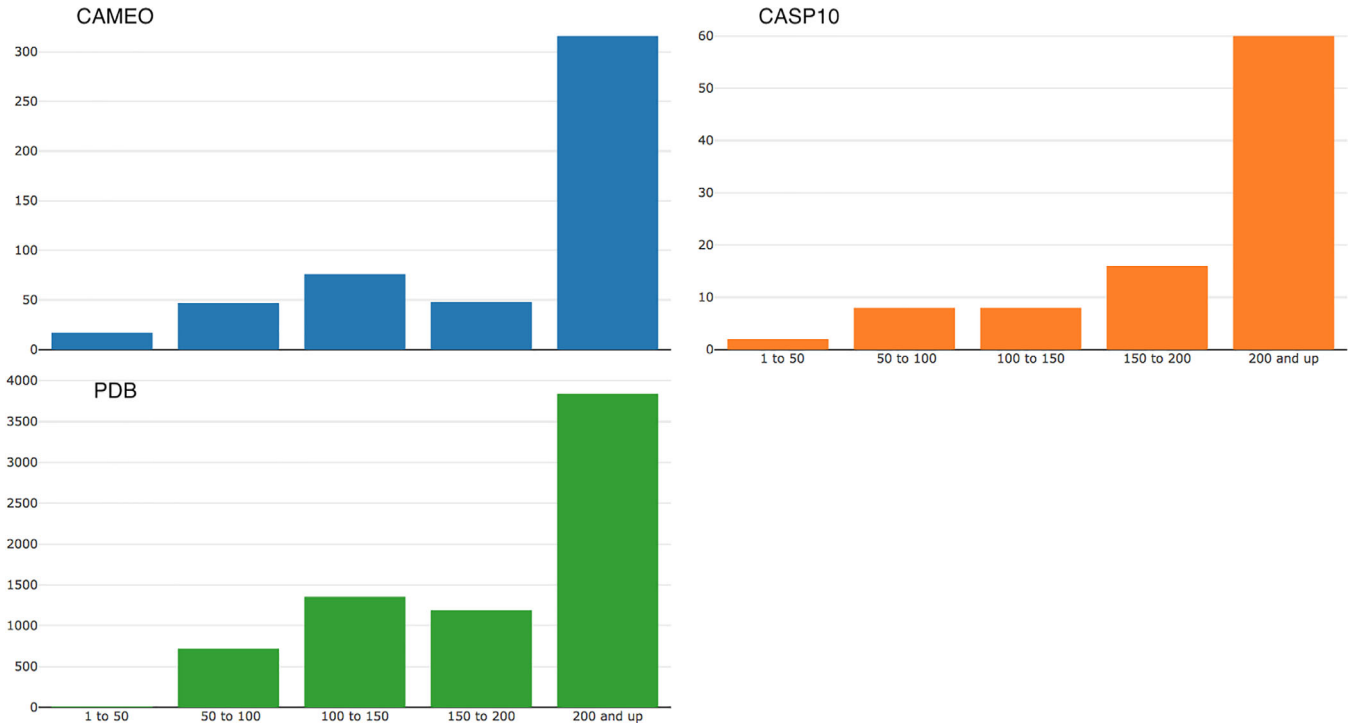

FIGURE 1 Sequence encoding, window generation, and feature extraction steps using sliding window approach

TABLE 1 Six reduced alphabets and their sources

Alphabet reference	Letter 1 (*B*)	Letter 2 (*J*)	Letter 3 (*U*)
a1[15]	CFILMVWY	AGHPRT	DEKNQS
a2[16]	CFILMVWY	AGPST	DEHKNQR
a3[17]	AFGILMPV	DEKR	CHNQSTWY
a4[18]	DHIMNVY	EFKLQ	ACGPRSTW
a5[19]	ACGILMPSTV	EKRDNQH	FYW
a6[20]	CFILMVWY	AGHST	DEKNPQR

Note: Each letter contains a cluster of amino acid residues (one-letter abbreviations). The residue clusters were denoted by the letters "*B*", "*J*", and "*U*".

length of the sequence window (101 residues). Each symbol of the 3-letter reduced alphabet is mapped to one of the three one-hot bit encoded vectors (*B* = [0,0,1], *J* = [0,1,0], *U* = [1,0,0]).

The first layer of our network is a convolutional layer, step size 1 and window size of 32. The output of each neuron on a convolutional layer is the convolution of the kernel matrix. The second layer is a max-pooling layer, one for each convolutional layer. Each of these max-pooling layers only outputs the maximum value (global or local) of its respective convolutional layer outputs. The third layer is a fully connected layer of size 256 where each of its neurons is connected to all of the neurons in the max-pooling layer. We use a dropout layer[21] after the fully connected layer to avoid overfitting. The final output layer consists of two neurons corresponding to the two classification results. These two neurons are fully connected to the previous layer. Table 2 highlights the differences among each of the tested models.

2.4 | Network training details

We train our models using stochastic gradient descent (SGD) with mini batches of size 128. SGD works by utilizing chain ruling which takes the partial derivative of the loss function with respect to each weight vector in the network, and uses the derivative to update the weights. We use a version of SGD with support for momentum and learning rate decay with default parameters and a learning rate set to 1e-3. All models are trained using the same setup and configuration the only difference being the seeds for initializing weights. We use early stopping, based on the validation set in order to pick the optimal set of weights. We train all our neural network models on AWS using G3 instances (NVIDIA Tesla M60 GPU) using python Keras libraries[22] running on top of TensorFlow library to assure model portability.

3 | RESULTS

3.1 | Training, validation, and evaluation datasets

Publicly available datasets are used to train, validate and evaluate the performance of our method. High resolution X-ray crystal structures from the Protein Data Bank (PDB)[23] are used to construct the training and validation data sets while CASP[14] and CAMEO[24] (http://www.cameo3d.org) are used for further validation. Figure 3 and Table 3 show the protein length distribution for training, testing and validation sets.

We use the Pisces protein sequence culling server (http://dunbrack.fccc.edu)[25] to extract sequences from PDB, filter for high

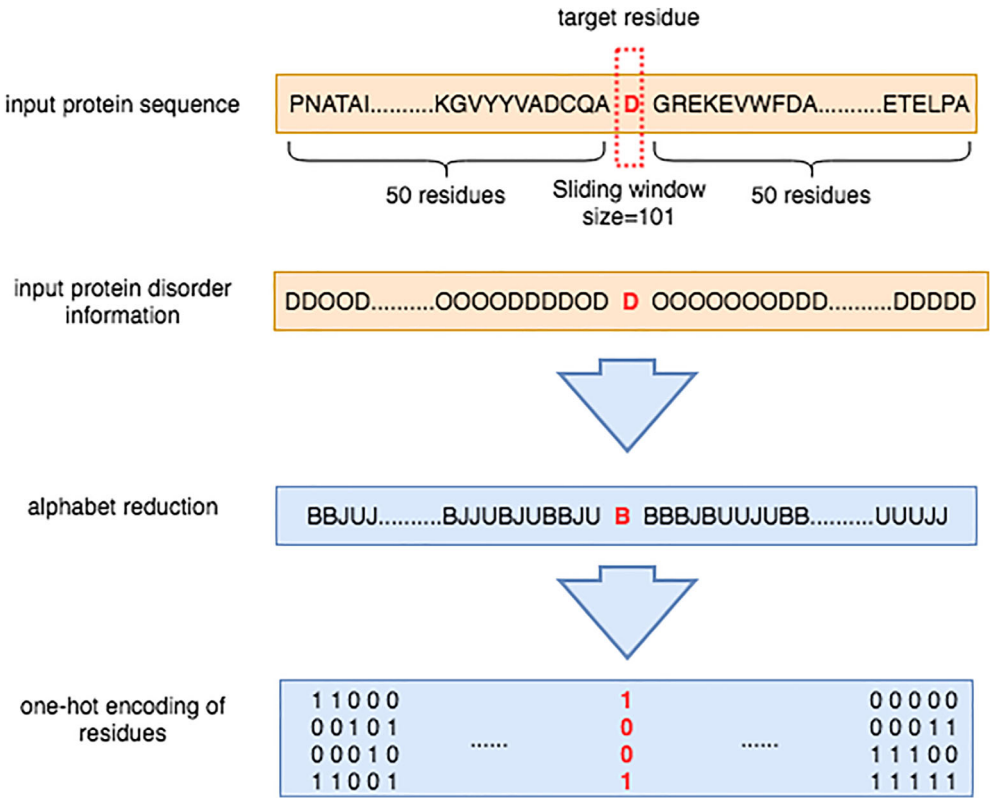

FIGURE 2 Basic 1-layer CNN architecture shared among all models [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 2 Description of the CNN architectures tested

Method	Architecture description
64-ker-local	1-convolutional layer, 64 kernels, local max pooling
128-ker-local	1-convolutional layer, 128 kernels, local max pooling
64-ker-global	1-convolutional layer, 64 kernels, global max pooling
128-ker-global	1-convolutional layer, 128 kernels, global max pooling
2-conv-local	2-convolutional layers, [64, 32] kernels, local max pooling

resolution and reduce redundancy. Parameters selected for culling are (i) proteins sharing less than 25% sequence identity (ii) resolution better than 1.8 Angstroms (iii) *R* value up to 0.30. In total, 7119 proteins are retrieved from PDB with an average length of 349 residues. The original dataset is then undersampled to create a 50/50 class balanced set, containing 181 060 examples. The effect of class imbalance is very detrimental to classification performance. In cases of an extreme ratio of imbalance, undersampling has been shown to perform on a par with oversampling without the risk of overfitting.[26] Undersampling has the additional advantage of reducing training times given that the training set is smaller in size.

The balanced dataset was randomly partitioned into ten equally sized subsets and a ten-fold cross-validation was performed to determine the optimal parameters for (a) convolutional network architecture and (b) encoding reduced protein alphabet (Section 3.4). At each step of the cross validation, one subset is selected and used as validation set while the remaining nine are used as training set. This process is repeated until all subsets are validated, results for each of the parameters tested are shown in Tables 4 and 5.

CASP10 is the latest dataset available from the series experiments, which released specific targets for protein disorder prediction. The 94 available targets are used for initial validation and as an independent benchmark set. Finally, to further assess and compare our method, we tested it against CAMEO 6 months targets released from August 26, 2017 to February 18th, 2018 (504 targets, categorized in three groups). Since CAMEO targets were released after the construction of our PDB training set, there is no sequence overlap between the two set. However, CASP10 targets were already present in PDB at the time of extraction. To prevent any redundancy between sets, we use BLASTClust[27] to filter and remove sequences from the PDB training set sharing at least 25% identity with sequences in the CASP10 set.

3.2 | Metrics and evaluation criteria

Disorder data is characterized by high class imbalance, disordered residues account for less than 5% of the data in the PDB set (training and test). Since disordered residues are relatively rare compared to ordered ones, they are harder to predict. Performance metrics should account for this imbalance and reward correct prediction of disordered residues higher than correct prediction of ordered ones.[28] We selected a subset of the metrics commonly used for the assessment of disorder data[14,29,30] that take into account the nature of the

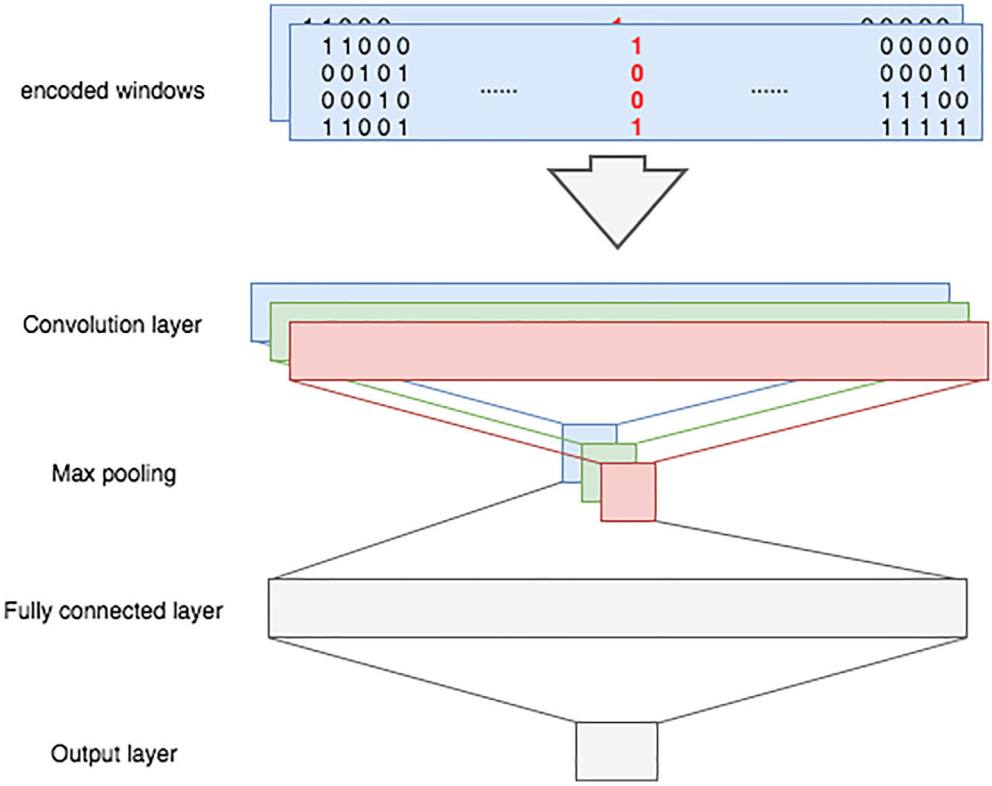

FIGURE 3 Protein length distribution in training, test and validation sets [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 3 Distribution of disordered regions by length on the three main datasets used

	Number of fragments			
Dataset	1-5	6-15	16-25	>25
CASP10	21	41	11	3
CAMEO	143	114	27	11
PDB	768	657	127	37

imbalanced data: (i) specificity (ii) sensitivity (iii) balanced accuracy (iv) Matthews correlation coefficient and (v) AUC.

3.3 | Binary metrics

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (1)$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (2)$$

$$\text{BalanceAcc} = \frac{TP}{TP + FN} + \frac{TN}{TN + FP} \quad (3)$$

$$MCC = \frac{TP \times TN - FP \times FN}{(TP + FP)(TP + FN)(TN + FP)(TN + FN)} \quad (4)$$

True positives (TP) and true negatives (TN) are the numbers of correctly predicted disordered and ordered residues. False positives (FP) and false negatives (FN) are the numbers of incorrectly predicted disordered and ordered residues.

3.4 | Statistical metrics

The Receiver Operating Characteristic (ROC) curve is a plot that compares the true positive rate against the false positive rate under various threshold values for a binary classifier. ROC curve represents a monotonic function describing the balance between the true positive and false positive rates of a predictor.[31] For a set of probability thresholds (from 0 to 1), a residue is considered as a positive example (disordered) if its predicted probability is equal to or greater than the threshold value. The area under the curve (AUC) is used as an aggregate measure of the overall quality of a prediction method. AUC has a minimum value 0, a random value 0.5 and a perfect value 1.

3.5 | Comparison with other methods

To benchmark our method we selected the following methods: Espritz,[32] Disopred3,[33] IUPred,[34] and ngramsAlpha.[35] Given that our predictor is sequence-based, we compared our results with similar methods and we leave out clustering, template and meta based approaches. Espritz is an ensemble of sequence-only and multiple sequence alignments disorder prediction methods. The sequence-only method has three different versions, depending on

TABLE 4 Alphabet cross validation

	AUC value of 10 cross validation batch datasets										
Alphabet	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Mean**
Alphabet 1	87.55%	89.26%	86.47%	88.55%	88.85%	87.23%	87.44%	87.53%	87.54%	87.71	87.84%
Alphabet 2	**87.79%**	88.86%	**87.62%**	**89.31%**	88.50%	87.17%	87.49%	**88.01%**	**88.01%**	87.83%	88.09%
Alphabet 3	83.00%	86.32%	82.87%	86.63%	83.92%	82.89%	83.59%	84.39%	84.54%	84.69%	84.43%
Alphabet 4	81.87%	86.08%	83.93%	86.22%	85.07%	81.42%	84.18%	83.84%	83.87%	85.12%	84.41%
Alphabet 5	85.29%	87.10%	84.07%	87.44%	85.12%	83.85%	86.00%	84.84%	85.32%	85.73%	85.50%
Alphabet 6	87.39%	**89.51**%	87.48%	89.02%	**88.92%**	**87.54%**	**87.55%**	87.66%	87.66%	**87.99%**	**88.15%**
No alphabet	82.15%	85.52%	82.96%	86.01%	83.09%	81.70%	82.94%	82.69%	83.60%	83.54%	83.42%

Note: Bold value highlights the best performant run/method within the column or row.

TABLE 5 Model cross validation

	AUC value of 10 cross validation batch datasets										
Model	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Mean**
64-ker-local	88.18%	**89.59%**	87.64%	**89.63%**	88.42%	87.48%	**87.96%**	88.02%	88.29%	**87.98%**	88.32%
128-ker-local	**88.37%**	89.56%	**87.78%**	89.60%	88.44%	**87.67%**	87.83%	88.25%	**88.40%**	87.97%	**88.39%**
64-ker-global	87.51%	88.24%	85.83%	89.00%	87.47%	86.38%	86.61%	87.02%	87.34%	86.80%	87.22%
128-ker-global	87.40%	88.52%	86.42%	89.13%	87.67%	85.89%	86.75%	87.61%	87.02%	86.82%	87.32%
2-conv-local	87.93%	89.11%	87.33%	89.28%	88.47%	87.40%	87.86%	87.80%	87.89%	87.76%	88.08%

Note: Bold value highlights the best performant run/method within the column or row.

the initial set used for training (X-ray, NMR, Disprot). We used X-ray trained version since it is the one that performs best among the three. Disopred3 runs a PSI-BLAST search for each of the residues in a 15-residue window. The profile is then used as input to a neural network classifier which outputs a probability estimate of the residue being disordered.

IUPred method is based on estimating the capacity of polypeptides to form stabilizing contacts. It has two prediction modes: IUPred (Long) and IUPred (Short). Each mode optimizes predictions for either long or short disordered regions. Finally, ngramsAlpha is our previously published predictor based on n-grams frequencies and reduced protein alphabets.

3.6 | Parameter and model selection

In order to select the best performing model, we experimented with two of the components of our method while leaving the remaining parameters constant. In particular, we tested several network architectures and reduced amino acid alphabets and analyzed their effect on the model predictive value. We performed a ten-fold cross-validation, using the mean AUC across validation batches as the primary metric to compare performance. Values for parameters such as dropout and learning rate, optimizer, and window size have been selected after performing a hyperparameter search across a reduced size training set and are left constant.

3.7 | Alphabet selection

Using reduced alphabets has two main advantages: (i) cluster residues with similar biochemical properties providing additional information to the original sequence and (ii) reduce the amino acid space from 20 to 3 residues, reducing, in turn, the model complexity and amount of data required for training. We tested six different alphabets from the literature and analyzed which performed better in the context of our classification problem. We used the (2-conv-local) network architecture across all runs. A modified version of the network using the full amino acid alphabet as input (no alphabet reduction step) is included for comparison. The effect of alphabet selection is shown in Table 4. Across the ten validation batches, we found that alphabets 1, 2, and 6 achieved better overall performance than alphabets 3, 4, and 5. Results also show that all six alphabets outperformed the model where no alphabet reduction was applied. We selected alphabet 6 for our final model implementation based on the results shown in Table 4.

Disordered regions are characterized by a high content of polar and charged amino acids (disorder-promoting residues) and low content of hydrophobic residues (order-promoting residues).[36] Despite being created with different objectives, alphabets (1, 2, 6) cluster most disorder-promoting residues within the same group (Table 1). Alphabets differ in the composition of the other two groups, which contain a mix of order-promoting and ambiguous residues. The relationship between net charge and hydrophobicity has been explored by other

IDP predictors before.[37] It is the patterns and high-order relationships between residues groups uncovered by the convolutional step that enables our method to achieve its high accuracy.

It may be possible for a neural network to learn the optimal residue groups given sufficient training data. Given our limited training set and network architecture, this wasn't possible to achieve. The adapted model that takes as input the original 20-letter alphabet, did not converge and underperformed when compared to the models using the reduction step (Table 4). These results highlight the benefit of the dimensionality reduction step before training our models.

3.8 | Convolutional network architecture

To test the relationship between network architecture and performance, we trained five different networks models and evaluated their predictive value. We adapted models successfully used in the DNA space to predict DNA-protein binding and function[10,12] hoping they would also perform well in the 3-letter reduced amino acid space. Our models differ in the number of kernels (50, 64, 128), the number of convolutional layers (1, 2) and max-pooling layer implementation (global vs local). We found that the number of convolutional layers does not seem to have a great impact on performance. Models with a higher number of convolution kernels and local pooling implementation achieved better overall classification performance. Based on the results shown in Table 5, we selected 128-ker-local model.

3.9 | Method performance

Figures 4, 5 and Tables 6, 7 compare the performance of our method against Disopred3, Espritz, IUPred, and ngramAlpha. It is worthwhile to mention that—of the listed methods—Disopred is the only to make use of additional evolutionary information through sequence profiles (performing PSI-BLAST[38] searches for each input protein). This added evolutionary information gives the method an extra advantage in performance but comes at the cost of execution time. The other three methods are similar in nature to ours, using sequence-only information to make disorder/order predictions. All methods were downloaded and ran locally in a Linux server using default parameters.

In terms of balanced accuracy (B.Acc), our method outperforms all others on the two independent validation datasets. With respect to area under the ROC curve (AUC) and MCC, our method performs much better than the predictors not using sequence profiles (such as IUpred and Espritz) and nears the performance of Disopred3 for AUC on both validation sets.

The performance of the method was also evaluated on disordered regions of various lengths for the CASP10 dataset and compared with the other top performance methods. The percentage of residues correctly predicted to be disordered is reported in Table 8. While Espritz performs better on short length disorder regions, Disopred3 and cnnAlpha achieve better results on mid and long disordered regions.

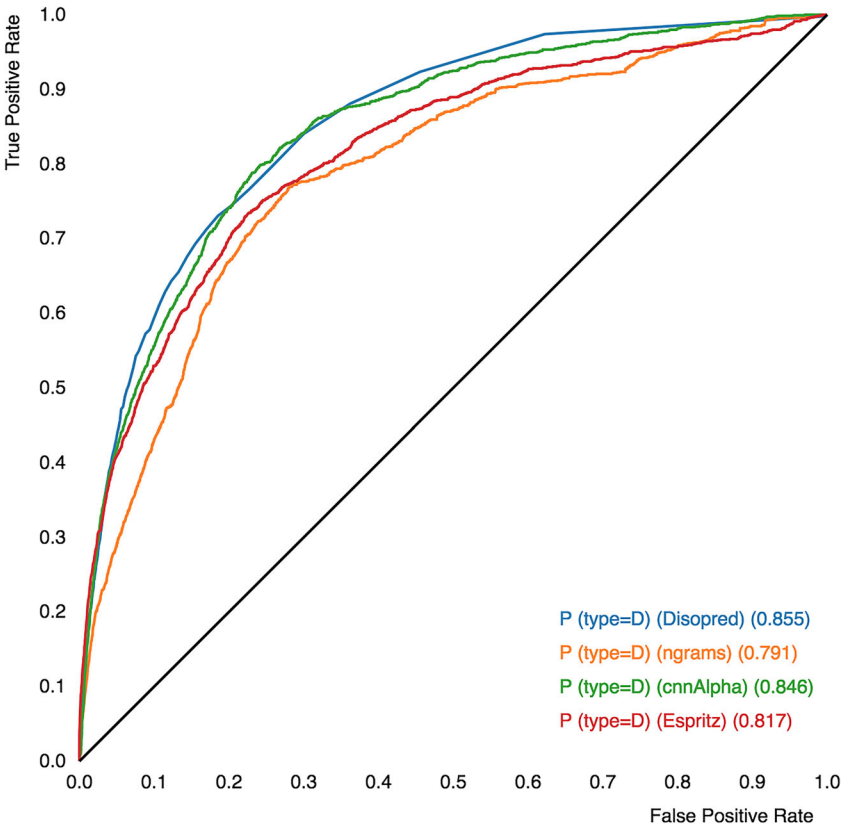

FIGURE 4 ROC curves for the evaluation set targets comparing the performance of the top four models (CASP targets) [Color figure can be viewed at wileyonlinelibrary.com]

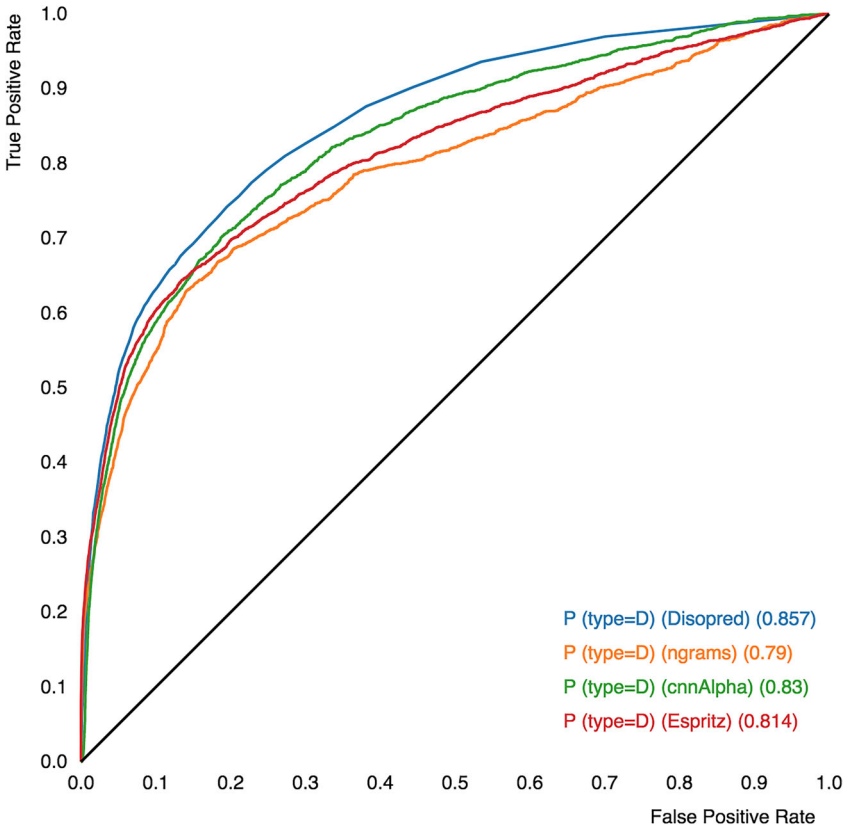

FIGURE 5 ROC curves for the evaluation set targets comparing the performance of the top four models (CAMEO hard targets) [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 6 Performance of predictors on CASP10 dataset

Method	Sequence profile	B.Acc	Sens	Spec	MCC	AUC
Disopred3	Yes	0.64	0.32	0.97	0.32	0.86
cnnAlpha	No	0.75	0.64	0.85	0.31	0.85
Espritz	No	0.72	0.54	0.89	0.30	0.82
ngramAlpha	No	0.72	0.61	0.83	0.26	0.79
IUPred (short)	No	0.63	0.31	0.95	0.26	0.66
IUPred (long)	No	0.57	0.17	0.96	0.15	0.60

Note: Metrics shown: balanced accuracy (B.Acc), sensitivity (Sens), specificity (Spec) Mattehews correlation coefficient (MCC), and area under the ROC curve (AUC).

TABLE 7 Performance of predictors on CAMEO dataset

Method	Sequence profile	B.Acc	Sens	Spec	MCC	AUC
Disopred3	Yes	0.72	0.48	0.96	0.43	0.86
cnnAlpha	No	0.75	0.61	0.88	0.36	0.83
Espritz	No	0.75	0.64	0.88	0.35	0.81
ngramAlpha	No	0.73	0.56	0.89	0.33	0.79
IUPred (short)	No	0.71	0.47	0.94	0.36	0.80
IUPred (long)	No	0.64	0.35	0.93	0.27	0.73

Note: Metrics shown: balanced accuracy (B.Acc), sensitivity (Sens), specificity (Spec) Mattehews correlation coefficient (MCC), and area under the ROC curve (AUC).

3.10 | Large-scale predictions

Finally, we evaluate the speed at which our method performs predictions on a large scale. We compared our method's execution time with Disopred3 since they both ranked on top of our evaluation.

The two applications were installed locally on a standard Linux server (Amazon EC2 m5.xlarge, 4 CPUs/16GB memory). To make predictions, Disopred3 uses PSSM values obtained after three search iterations of PSI-BLAST.[33] BLAST tool and UniRef90 database were installed locally for that purpose. We created a script that takes as input

TABLE 8 Predictors recall by region length in CASP10

Method	<10 AA	10-30 AA	>30 AA
cnnAlpha	0.40	**0.42**	0.46
Espritz	**0.43**	0.39	0.33
Disopred3	0.26	0.32	**0.47**

Note: Bold value highlights the best performant run/method within the column or row.

TABLE 9 Execution times on the CASP10 dataset by predictor

Method	Total time	Average time per protein
Disopred3	39 464 s	424 s
cnnAlpha	34 s	0.37 s

parameters the method name and a list of target proteins FASTA files, performs predictions, and saves the results to an output file. We timed the execution of each script run via the Linux time command.

The execution time needed to perform predictions on the CASP10 dataset (94 proteins, 25 370 residues) is reported in Table 9. The large difference in execution time is explained by the fact that extracting features and performing a forward pass in a previously trained neural network is extremely fast when compared to running multiple PSI-Blast searches. That makes our method several orders of magnitude faster than Disopred3 and still capable of achieving similar accuracy.

4 | DISCUSSION

This paper presents cnnAlpha, a new convolutional neural network-based method for protein disorder prediction using sequence information. We demonstrated that our combination of amino acid alphabet reduction strategy and convolutional neural networks leads to an approach which can successfully compete with more elaborated and computationally expensive sequence based algorithms. The source code for an R/Shiny application with the model implementation of our predictor can be found at https://github.com/mauricioob/shiny-pred.

CNNs are good at learning rich higher-order sequence features, such as secondary motifs and local sequence context. We believe that the reduction in dimension from 20 to 3 letter amino acid alphabet helped the convolutional layer to better detect these relationships and patterns. The reduction in dimensionality and our under sampling approach to the class imbalance problem have the additional advantage of reducing the amount of data required by the training sets. This, in turn, made our models faster to train and allow us further experimentation in parameter setting.

Overall, our method outperforms similar sequence-only algorithms across both evaluation data sets and nears the performance of sequence based methods using additional evolutionary information (sequence profiles). Being several orders of magnitude faster than sequence profile based approaches, our method is suitable for high-throughput predictions at the proteomic scale. The high specificity of cnnAlpha also ensures a low false positive rate on high-throughput contexts, making it even more suitable for this task.

ACKNOWLEDGMENT

The authors are grateful for the computational facilities provided by Novartis Institutes for BioMedical Research.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1002/prot.25966

ORCID

Mauricio Oberti https://orcid.org/0000-0001-7107-2616

REFERENCES

1. Dunker AK, Oldfield CJ, Meng J, et al. The unfoldomics decade: an update on intrinsically disordered proteins. *BMC Genomics*. 2008;9 (Suppl 2):S1.
2. Oldfield CJ, Dunker AK. Intrinsically disordered proteins and intrinsically disordered protein regions. *Annu Rev Biochem*. 2014;83:553-584.
3. Dunker AK, Bondos SE, Huang F, Oldfield CJ. Intrinsically disordered proteins and multicellular organisms. *Semin Cell Dev Biol*. 2015;37:44-55.
4. Di Domenico T, Walsh I, Martin AJM, Tosatto SCE. MobiDB: a comprehensive database of intrinsic protein disorder annotations. *Bioinformatics (Oxford, England)*. 2012;28(15):2080-2081.
5. Sickmeier M, Hamilton JA, LeGall T, et al. DisProt: the Database of Disordered Proteins. *Nucleic Acids Res*. 2007;35(suppl 1):D786-D793. http://nar.oxfordjournals.org/content/35/suppl_1/D786.
6. He B, Wang K, Liu Y, Xue B, Uversky VN, Dunker AK. Predicting intrinsic disorder in proteins: an overview. *Cell Res*. 2009;19(8): 929-949.
7. Deng X, Eickholt J, Cheng J. A comprehensive overview of computational protein disorder prediction methods. *Mol BioSyst*. 2012;8(1):114-121.
8. Atkins JD, Boateng SY, Sorensen T, McGuffin LJ. Disorder prediction methods, their applicability to different protein targets and their usefulness for guiding experimental studies. *Int J Mol Sci*. 2015;16(8): 19040-19054. http://www.mdpi.com/1422-0067/16/8/19040.
9. Sirota FL, Ooi HS, Gattermayer T, Schneider G, Eisenhaber F, Maurer-Stroh S. Parameterization of disorder predictors for large-scale applications requiring high specificity by using an extended benchmark dataset. *BMC Genomics*. 2010;11(1):S15. https://doi.org/10.1186/1471-2164-11-S1-S15.
10. Quang D, Xie X. DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Res*. 2016;44(11):e107. http://nar.oxfordjournals.org/content/early/2016/04/15/nar.gkw226.
11. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;521(7553): 436-444. http://www.nature.com/nature/journal/v521/n7553/full/nature14539.html.
12. Zeng H, Edwards MD, Liu G, Gifford DK. Convolutional neural network architectures for predicting DNA–protein binding. *Bioinformatics*. 2016;32(12):i121-i127. http://bioinformatics.oxfordjournals.org/content/32/12/i121.
13. Linding R, Jensen LJ, Diella F, Bork P, Gibson TJ, Russell RB. Protein disorder prediction: implications for structural proteomics. *Structure (London, England: 1993)*. 2003;11(11):1453-1459.

14. Monastyrskyy B, Kryshtafovych A, Moult J, Tramontano A, Fidelis K. Assessment of protein disorder region predictions in CASP10. *Proteins*. 2014;82(02):127-137. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4406047/.
15. Wang J, Wang W. A computational approach to simplifying the protein folding alphabet. *Nat Struct Mol Biol*. 1999;6(11):1033-1038. http://www.nature.com/nsmb/journal/v6/n11/full/nsb1199_1033.html.
16. Li T, Fan K, Wang J, Wang W. Reduction of protein sequence complexity by residue grouping. *Protein Eng*. 2003;16(5):323-330. http://peds.oxfordjournals.org/content/16/5/323.
17. Branden C. Introduction to protein structure. By C Branden and J Tooze. pp 302. garland publishing, New York. 1991 ISBN 0–8513–0270–3 (pbk). *Biochem Educ*. 1992;20(2):121-122. http://onlinelibrary.wiley.com/doi/10.1016/0307-4412(92)90129-A/abstract.
18. Mekler LB. Specific selective interaction between amino acid residues of polypeptide chains; 1969. OCLC: 26411216.
19. Murphy LR, Wallqvist A, Levy RM. Simplified amino acid alphabets for protein fold recognition and implications for folding. *Protein Eng*. 2000;13(3):149-152. http://peds.oxfordjournals.org/content/13/3/149.
20. Bacardit J, Stout M, Hirst JD, Valencia A, Smith RE, Krasnogor N. Automated alphabet reduction for protein datasets. *BMC Bioinform*. 2009;10(1):6. http://www.biomedcentral.com/1471-2105/10/6/abstract.
21. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res*. 2014;15:1929-1958. http://jmlr.org/papers/v15/srivastava14a.html.
22. Chollet F, *Keras*. San Francisco, CA: GitHub; 2015. Available from: https://github.com/fchollet/keras.
23. Berman HM, Westbrook J, Feng Z, et al. The Protein Data Bank. *Nucleic Acids Res*. 2000;28(1):235-242. http://nar.oxfordjournals.org/content/28/1/235.
24. Haas J, Roth S, Arnold K, et al. The Protein Model Portal—a comprehensive resource for protein structure and model information. *Database (Oxford)*. 2013;2013:bat031.
25. Wang G, Dunbrack RL. PISCES: a protein sequence culling server. *Bioinformatics (Oxford, England)*. 2003;19(12):1589-1591.
26. Buda M, Maki A, Mazurowski MA. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Netw*. 2018;106:249-259. http://arxiv.org/abs/1710.05381,arXiv:1710.05381.
27. NCBI News: Spring 2004|BLASTLab. https://www.ncbi.nlm.nih.gov/Web/Newsltr/Spring04/blastlab.html.
28. Monastyrskyy B, Fidelis K, Moult J, Tramontano A, Kryshtafovych A. Evaluation of disorder predictions in CASP9. *Proteins*. 2011;79(S10):107-118. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3212657/.
29. Huang T, He ZS, Cui WR, et al. A Sequence-based approach for predicting protein disordered regions. *Protein Pept Lett*. 2013;20(3):243-248. http://www.eurekaselect.com/openurl/content.php?genre=article&issn=0929-8665&volume=20&issue=3&spage=243.
30. Wang S, Ma J, Xu J. AUCpreD: proteome-level protein disorder prediction by AUC-maximized deep convolutional neural fields. *Bioinformatics (Oxford, England)*. 2016;32(17):i672-i679.
31. Fawcett T. An introduction to ROC analysis. *Pattern Recogn Lett*. 2006;27(8):861-874. http://www.sciencedirect.com/science/article/pii/S016786550500303X.
32. Walsh I, Martin AJM, Domenico TD, Tosatto SCE. ESpritz: accurate and fast prediction of protein disorder. *Bioinformatics*. 2012;28(4):503-509. http://bioinformatics.oxfordjournals.org/content/28/4/503.
33. Ward JJ, McGuffin LJ, Bryson K, Buxton BF, Jones DT. The DISOPRED server for the prediction of protein disorder. *Bioinformatics*. 2004;20(13):2138-2139. http://bioinformatics.oxfordjournals.org/content/20/13/2138.
34. Dosztányi 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 (Oxford, England)*. 2005;21(16):3433-3434.
35. Oberti M, Vaisman II. Identification and prediction of intrinsically disordered regions in proteins using N-grams. In: Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology,and Health Informatics ACM-BCB '17, New York, NY, USA: ACM; 2017. pp. 67–72. http://doi.acm.org/10.1145/3107411.3107480.
36. Uversky VN. Intrinsically disordered proteins from A to Z. *Int J Biochem Cell Biol*. 2011;43(8):1090-1103. http://www.sciencedirect.com/science/article/pii/S1357272511000914.
37. Prilusky J, Felder CE, Zeev-Ben-Mordehai T, et al. FoldIndexQc: a simple tool to predict whether a given protein sequence is intrinsically unfolded. *Bioinformatics*. 2005;21(16):3435-3438. http://bioinformatics.oxfordjournals.org/content/21/16/3435.
38. Altschul SF, Madden TL, Schäffer AA, et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*. 1997;25(17):3389-3402.

How to cite this article: Oberti M, Vaisman II. cnnAlpha: Protein disordered regions prediction by reduced amino acid alphabets and convolutional neural networks. *Proteins*. 2020; 88:1472–1481. https://doi.org/10.1002/prot.25966