Signal, Image and Video Processing (2023) 17:2055–2061
https://doi.org/10.1007/s11760-022-02419-5

ORIGINAL PAPER

A hybrid deep learning model for classification of plant transcription factor proteins

Ali Burak Öncül[1,2] · Yüksel Çelik[2]

Received: 23 April 2022 / Revised: 15 November 2022 / Accepted: 21 November 2022 / Published online: 1 December 2022

Abstract
Studies on the amino acid sequences, protein structure, and the relationships of amino acids are still a large and challenging problem in biology. Although bioinformatics studies have progressed in solving these problems, the relationship between amino acids and determining the type of protein formed by amino acids are still a problem that has not been fully solved. This problem is why the use of some of the available protein sequences is also limited. This study proposes a hybrid deep learning model to classify amino acid sequences of unknown species using the amino acid sequences in the plant transcription factor database. The model achieved 98.23% success rate in the tests performed. With the hybrid model created, transcription factor proteins in the plant kingdom can be easily classified. The fact that the model is hybrid has made its layers lighter. The training period has decreased, and the success has increased. When tested with a bidirectional LSTM produced with a similar dataset to our dataset and a ResNet-based ProtCNN model, a CNN model, the proposed model was more successful. In addition, we found that the hybrid model we designed by creating vectors with Word2Vec is more successful than other LSTM or CNN-based models. With the model we have prepared, other proteins, especially transcription factor proteins, will be classified, thus enabling species identification to be carried out efficiently and successfully. The use of such a triplet hybrid structure in classifying plant transcription factors stands out as an innovation brought to the literature.

Keywords Protein classification · Deep learning · GRU · CNN · Hybrid models · Word2Vec

1 Introduction

Proteins are macromolecules made up of amino acids that play an essential role in fulfilling many of the functions in biological systems. Amino acids, the smallest building blocks of proteins, peptides, and enzymes, are small molecules with carboxyl on one end and an amino group on the other and bearing neutral, polar, or ionizable groups in their side chains. Many amino acids and their derivatives have various functions in metabolism in living things [1].

Some of the large number of protein sequences found in various databases and studies have been classified or labeled through experiments. However, the amount of unclassified or unlabeled data is still quite large. Biological approaches used to classify these proteins can include categorizing proteins that work with small differences in other living things or species into different classes. However, when the sequences of these proteins are examined, it can be revealed that they show high similarity and are structurally similar. When this is the case, the necessity of protein classification arises; protein classification is necessary for researchers who focus on a particular protein type or a particular function to obtain detailed and successful results [2].

Time-consuming and costly biological experiments used to understand the structure of proteins, which are the most important parts of living things, have been replaced by studies in computer science over time [2]. Studies discovered that statistical models could be used in this field, and some studies were made [3]. Hidden Markov model [4]-based studies [5] and basic local alignment search tool (BLAST) [6] can be given as examples of these studies.

Yüksel Çelik contributed equally to this work.

✉ Ali Burak Öncül
boncul@kastamonu.edu.tr

Yüksel Çelik
yukselcelik@karabuk.edu.tr

[1] Department of Computer Engineering, Faculty of Engineering and Architecture, Kastamonu University, 37150 Kastamonu, Turkey

[2] Department of Computer Engineering, Faculty of Engineering, Karabük University, 78050 Karabük, Turkey

Some models and applications have been proposed to analyze or classify various protein sequences in the literature. These applications have come to the fore with different datasets and different designs. Examples of these studies are EzyPred [7], FFPred [8], ECPred [9], GoFDR [10], Asgari and Mofrad's SVM model [11], Naveenkumar et al.'s deep learning studies [12], UDSMProt [13], LSTM and GRU-based studies by Le et al. [14], LSTM-based studies by Li et al. [15], CNN and LSTM-based studies by Bileshi et al. [16], TAPE [17], Belzen et al. [18], Torissi et al. deep learning study [19] and the work of Gustafsson et al. [20].

With recent developments in computer science, recurrent neural network and convolutional neural network-based deep learning algorithms are also used in classification problems in this field with different variations [12]. Inspired by these models, we have designed a hybrid model based on various deep learning models. Using the hybrid model, we classified plant transcription factor proteins with similar structures but different functions according to their motifs. With the hybrid structure of our model, we took education one step ahead by creating our Word2Vec vocabulary and reducing our training weight. We benefited from the success of feature extraction of CNN-based architectures and the correct management of long short-term dependencies of LSTM-based networks. Using the hybrid structure, instead of dense model architecture, we created both the CNN and the GRU layer lighter, which increased the relative success by shortening the relative training time.

The innovative contributions of our proposed hybrid model can be listed as follows:

- Support biological studies in avoiding human-based errors.
- Provide the basis for future DNA-protein interaction prediction studies.
- The model has eliminated the need to use labeled data as it only needs the sequence to run. In this way, the need for more than one parameter in biological experiments and some statistical-based or machine learning-based models has been prevented, and high performance has been achieved and has increased the success in the literature.
- Thanks to the prepared model, there is no need for a database query like some other models in use, so the time to work and produce results is greatly shortened.
- The dataset and protein word-vector representation we prepared for use in the development process will shed light on other studies in the literature.

2 Materials and methods

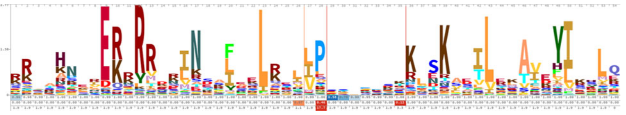


Fig. 1 Basic helix-loop-helix (bHLH) logo [27], [28]

Transcription factors (TFs) are proteins that can bind to a specific sequence or section according to their role in DNA to regulate genes' transcription. They can also be referred to as sequence-specific DNA binding proteins [21]. We obtained amino acid sequences from different plant families used in this study from the Plant Transcription Factor Database (PlantTFDB) [22], [23]. We split the sequences into words during preprocessing, performed proximity analysis with the Word2Vec model, and generated vectors from the sequences. For classes, we used the one-hot encoding method. We prepared an hybrid model used convolutional neural network (CNN) [24] and recurrent neural network (RNN) [24]-based models for classification after preprocessing. In the conclusion part, we presented the models we used comparatively.

2.1 Structure of the protein sequences

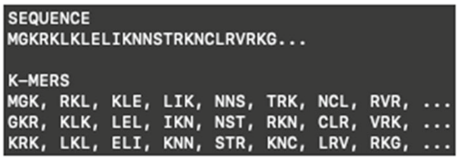
Protein sequences are formed by the sequences of characters expressing amino acids [25]. Each different sequence in these sequences creates a different protein. Even if each sequence creates a different protein, certain sequences, namely motifs, within these sequences reveal that protein type. Protein sequences are expressed in letters. Even though they are different sequences, sequences containing motifs belonging to the same class belong to the same protein class [26]. Figure 1 shows the motifs of the basic Helix-Loop-Helix (bHLH) protein class and the probability of different amino acids in each region of this motif.

The protein class can be found by looking at the motif in the sequence. However, this motif (in Fig. 1) is different in length according to the protein type, and the amino acids in the motif, i.e., the characters, may differ within each protein class.

2.2 Obtaining and converting data to the appropriate file format

The raw sequences that have been obtained must be put into a specific format by preprocessing before being given to deep learning models for education. In this study, we applied a series of operations such as size limitation of sequences, filling in gaps, preparation of embeddings from sequences, and sub-methods of these operations to the dataset.

We downloaded the files of the protein sequences for each of the 63 plant families in fasta format from PlantTFDB [22], [23]. In the downloaded files, we extracted only the sequence and protein class information, and summed them in a single tab-delimited text file in two columns. We obtained sequence



```
SEQUENCE
MGKRKLKLELIKNNSTRKNCLRVRKG...

K-MERS
MGK, RKL, KLE, LIK, NNS, TRK, NCL, RVR, ...
GKR, KLK, LEL, IKN, NST, RKN, CLR, VRK, ...
KRK, LKL, ELI, KNN, STR, KNC, LRV, RKG, ...
```

Fig. 2 Prepare *k*-mers from sequence

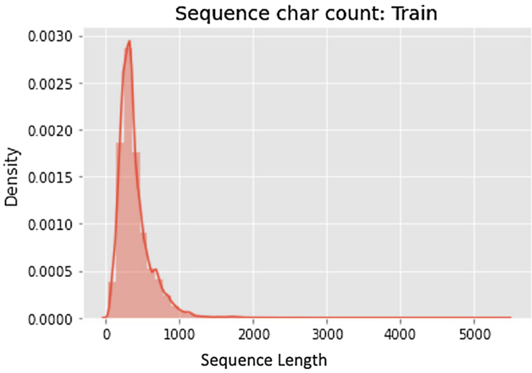


Fig. 3 Density of sequences according to their length

and class information of 58 different protein transcription factor classes in 132,330 lines from the plant genetic source.

2.3 Representation of sequences with *k*-mers and Word2Vec embeddings

The relationship of the words before and after the source or target word is very important in order not to miss the motifs that indicate the protein properties and class in the sequences, and thus to make the correct classification. Like in natural language processing, words are needed here, and strings must be divided into words (*k*-mers). In the literature, the number of *k*-mers generally varies between 3 and 6 [29], [30]. In order to classify sequences more successfully, it is appropriate to know the proximity of these *k*-mers to each other. For this, a vocabulary should be created by dividing existing sequences into three characters (3-mer) words. Getting three separate 3-mer groups from each sequence by shifting each string three times by one character to get more words and not miss possible triple word combinations will provide a more successful vocabulary [11]. This way, almost no combinations are lost when creating 3-mers in the sequence. A brief example of this structure is shown in Fig. 2.

Not all protein sequences are the same length. Some sequences are short, while others can be quite long. For the model to work correctly, a certain fixed length should be determined by examining the lengths and quantities of the sequences according to their lengths. For this, we first checked the number of sequences according to their length. We have given the graph showing the number of sequences according to their lengths in Fig. 3.

Table 1 Comparison of code dictionary versus Word2Vec with proposed hybrid model

Prep. met	Train loss	Train acc	Val. loss	Val. acc
Code Dict	2.8305	0.2519	2.2621	0.3770
Word2Vec	1.1427	0.7017	0.3897	0.8984

When the *k*-mer size is chosen as 4 or more, short relationships and dependencies are overlooked. In addition, when words are longer than 3, the amount of sequence obtained from each sequence increases, word repetition occurs, and model training slows down since the sequences are scrolled by word character length. In addition, since less number of each word will be used, similarity values will have less effect. For this reason, as a result of the research, the size of the words was determined as 3. Since sequences are divided into words, it will be sufficient to limit the length of sequences to 250 words. If it is desired to use sequences with a larger number of words, the end of many short sequences will be filled with too many 0 s, and there will be an unnecessary processing cost. All these choices were determined as a result of experimental tests.

In this study, the CBOW model was preferred in terms of ease of structural estimation, speed, and resource utilization. In the model, the vector size is set to 300. We tested the window size of the Word2Vec model as 4, 5, 7, and 10 in our deep learning model. We determined that the appropriate window size is 4 based on the sequence lengths.

In Table 1, we present the first epoch results of running the proposed hybrid model with classic dictionary and Word2Vec preprocessing. When we experimented with this preprocessing methods, we saw that the preprocessing we did with the Word2Vec model had the highest success in all models.

2.4 Proposed hybrid deep learning model

The success of models based on the recurrent neural network (RNN) [24] in texts is known. Long short-term memory (LSTM) [31] and gated recurrent unit (GRU) [32] models, which are RNN-based models, are particularly effective for long texts. Input, output and forget gates in LSTM; GRU has input and output gates and forget key. Although the achievements of LSTM and GRU are approximately the same, the GRU has a faster training time because one key is less in the GRU. In addition, the feature extraction success of convolutional neural network (CNN) [24] models is high. For this reason, in this study, Word2Vec + CNN + Bidirectional GRU hybrid model, which will contribute to the literature, have been created. One-time working graphics and tenfold cross-validation results were prepared. ADAM [33] is used as an optimization function, and binary cross-entropy function is used as loss function. For the network to have a healthy

Table 2 Structure of the proposed hybrid model

Layer	Output shape	Param#
Embedding	(None, 250, 300)	2,784,600
Convolutional 1D (128)	(None, 250, 128)	115,328
Max pooling 1D (3)	(None, 83, 128)	0
Dropout (0.3)	(None, 83, 128)	0
Convolutional 1D (256)	(None, 83, 256)	98,560
Max pooling 1D (3)	(None, 27, 256)	0
Dropout (0.35)	(None, 27, 256)	0
Convolutional 1D (256)	(None, 27, 256)	196,864
Max pooling 1D (3)	(None, 9, 256)	0
Dropout (0.4)	(None, 9, 256)	0
Bidirectional GRU (384)	(None, 9, 768)	1,479,168
Dropout (0.25)	(None, 9, 768)	0
Flatten	(None, 6912)	0
Dense (128)	(None, 128)	884,864
Dropout (0.45)	(None, 128)	0
Dense (58) (classification)	(None, 58)	7482

training process, an early stop tool was added, and the number of epochs suitable for the hybrid model was determined.

In the hybrid model, high success is achieved using three convolutions, three max-pooling, one LSTM, five dropouts, one flatten, and two dense layers. In the proposed model, the sequences were first divided into three-character words, starting from the zeroth, first and second characters, as shown in Fig. 2, and transformed into the numerical vector equivalents of the pre-prepared vector representations of the words. Then, the sequences were brought to a fixed length of 250 words. Long ones are truncated, and 0 is added to the end of short ones. Then, the prepared fixed-length sequences were given as input to the embedding layer of the learning model. The input and output dimensions of the embedding layer are determined as the shape of the Word2Vec vector representation. For the model to start training with higher success, the initial weights are given as the vectors prepared with Word2Vec, that is, the proximity of the words in the vector space. For model development, the “trainable”parameter of the embedding layer is enabled. In this way, the model started education with the proximity analysis of words and was open to development, so it had high success and rapid development from the first epoch. The data coming out of the embedding layer was first given to the CNN and then to the bidirectional GRU layer of the model detailed in Table 2. Thanks to the Dropout layers in between, overfitting is prevented. The data from the bidirectional GRU layer is transmitted to a fully connected layer and then to the other fully connected layer, where classification will be made. The classification process of the sequences is completed. The design details of the hybrid model are shown in Table 2 and Fig. 4.

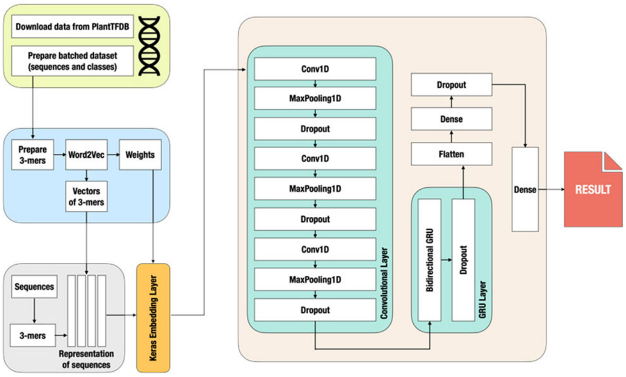


Fig. 4 Proposed hybrid deep learning model

Table 3 Test result of proposed hybrid model

Model	Acc (%)	Precision (%)	Recall (%)	F-score (%)	Time (min)
CNN	97.09	93.02	88.71	90.11	29.14
Bi. GRU	97.51	96.75	93.96	94.92	13.88
Hybrid	**98.23**	95.88	**95.27**	**95.36**	**11.80**

Bold values indicate the most successful models and the percentages in which these models are most successful

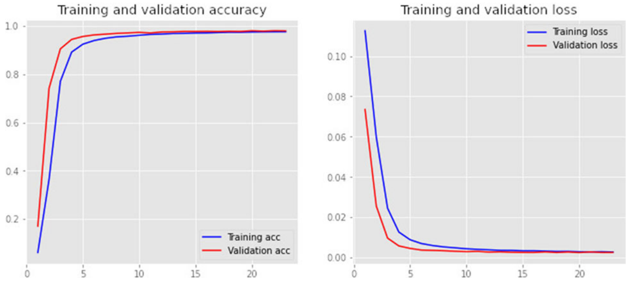


Fig. 5 Accuracy and loss graphs about training and validation of the proposed model

3 Results

In the hybrid model we designed, 80% of the data were used for training, 10% for verification, and 10% for testing. 105,864 sequences were used for training, 13,233 sequences for validation, and 13,233 sequences for testing. The separation of the dataset for training, validation, and testing are done completely randomly with the train_test_split function of Python’s Scikit-Learn library. Learning coefficients were determined as 0.001 in the CNN layers and 0.01 in the bidirectional GRU layers. As a result of the experiments made for the hybrid model, different neuron numbers and batch sizes were determined. Table 3 shows the success and train duration of the proposed hybrid model.

In Fig. 5, graphs are presented with comparisons of training and validation success and error of the proposed model.

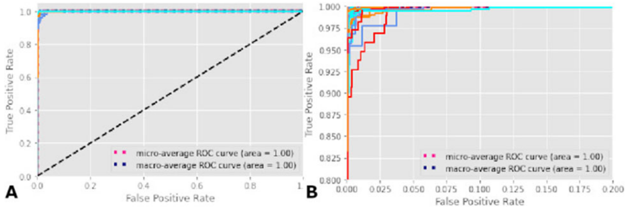


Fig. 6 ROC curve graph (with zoom) of the proposed model

Table 4 Tenfold cross val. result of proposed model

Model	Acc. of tenfold cross-val. (%)
CNN Bidirectional GRU	98.07

Table 5 Methods comparison with our dataset

Methods	Acc (%)	Precision (%)	Recall (%)	F-score (%)
Bidirectional LSTM [16]	85.04	86.21	82.03	84.45
ProtCNN [16]	85.18	94.01	80.33	85.09
Proposed model	**98.23**	**95.88**	**95.27**	**95.36**

Bold values indicate the most successful models and the percentages in which these models are most successful

In Fig. 6, the ROC curve graph of the proposed model (Fig. 6A) and the zoom of the graph (Fig. 6B) are presented. Each curve in the graph represents a class. When the ROC curve graphs of the proposed model are examined, it is seen that the existing 58 classes are also classified with very high success.

The tenfold cross-val. result of the proposed model we designed are listed in Table 4.

In addition, when a bidirectional LSTM model and a CNN-based ResNet model [16], which has one of the highest success rates in the literature and is proposed for a different dataset, are tested with our dataset, the success of these models was found to be lower. The comparison results are shown in Table 5.

In the hybrid model proposed in this study, firstly feature extraction was made from sequences with CNN layers, and then all 58 different classes were successfully classified using the success of GRU in long sequences. As shown in Table 5, our proposed model is more successful than other models in the literature, working with a similar dataset. When the hybrid model proposed within the scope of this thesis is compared with the studies made with similar data sets in the literature, given in Table 6, the model has the highest success and has provided an important innovation to the literature with its triple hybrid structure. Representing the sequences with Word2Vec and recording the weights showing the affinities as a result of this vector creation process and using them in the model increased the success of the model from the first epoch (see Fig. 5 and Table 1).

4 Discussion and conclusion

In this study, we developed a three-stage hybrid deep learning model to classify plant transcription factor proteins that play roles in various tasks from the Plant Transcription Factor Database. As input, our model, which takes only one protein sequence, performs the classification of proteins efficiently and with a high success rate, as opposed to the need for more than one parameter in biological experiments and some statistical-based or machine learning-based models. Our model is the most successful hybrid model in the field of plant transcription factor proteins and has a high success among models working with similar datasets. The data set we prepared and used during the development of our model became a data set that can also be used in follow-up studies. Again, the protein word-vector representation we prepared has been a pre-train model that can be the basis for different protein classification studies. In summary, we used three layers in the design of the model we proposed: The first layer is to divide the sequences into 3-mers to have a quality education, to create vector representations with our vocabulary, and to save the weights showing the similarity; the second layer is feature extraction by using the feature extraction success of CNN networks; the third layer is to determine the long-short term dependencies of the relatively long protein sequences with the bidirectional GRU layer, which works in the LSTM structure but is relatively fast, and the classification success is significantly increased. In this way, researchers will be able to query their protein sequences on the network without the need for any other information, determine which protein family the sequence belongs to, and use them in their fieldwork. In addition, this speed, convenience, and time will allow for more analysis and research in unit time. Likewise, the use of such a triple hybrid structure in the classification of plant transcription factors stands out as an innovation introduced in the literature. In addition, thanks to the model, we aimed to minimize human or solution, machine, or substance-based errors that can be made in biological experiments. Future studies can prepare the three-stage hybrid model to work faster and more efficiently on a more extensive data set or a classification problem and make it suitable for working in different genetic resources. The proposed method will also show high success on different datasets. Because, transcription factors in different kingdoms (plant, animal, etc.) have similar structures, our proposed model will have good success in other databases as well. The maximum length can be changed in the preprocessing part of the method to fine-tune it in order to achieve the highest success in different datasets. In addition, optimization and fine-tuning work on the model will

Table 6 Comparison of the proposed hybrid model and studies with similar data sets

Author	Method	Dataset	Acc (%)
Belzen et al. [18]	3-layer ResNet	CAFA3	93.7
Strodthoff et al. [13]	Pre. Tra. CNN	EC40, EC50	98
Bileschi et al. [16]	Bi. LSTM, ResNet	Pfam Seed	95.8
Proposed model	**CNN Bi. GRU**	**Plant TF**	**98.23**

Bold values indicate the most successful models and the percentages in which these models are most successful

play an essential role in developing success. In addition, this study will provide the basis for the linkage estimation and region detection of DNA regions with a future transcription factor.

Author Contributions All authors have an equal contribution to the article.

Funding Not applicable.

Data availability After the article is published, the data set can be accessed by e-mailing the corresponding author.

Declarations

Conflict of interest The author declares that there is no competing interests related to this paper.

Ethical approval Not applicable.

References

1. Acar, N., Gündeğer, E., Selçuki, C.: Protein yapı analizleri. In: Baloğlu, M.C. (ed.) Biyoinformatik Temelleri Ve Uygulamaları, pp. 85–128. Pegem Akademi Yayıncılık, Kastamonu (2018)
2. Petrey, D., Honig, B.: Is protein classification necessary? towards alternative approaches to function annotation. Curr. Opin. Struct. Biol. **19**(3), 363–368 (2009)
3. Baldi, P., Brunak, S.: Bioinformatics: the machine learning approach. The MIT Press, London (2001)
4. Eddy, S.R.: Hidden markov models. Curr. Opin. Struct. Biol. **6**(3), 361–365 (1996)
5. Gromiha, M.M.: Chapter 2 - protein sequence analysis. In: Protein Bioinformatics. pp. 29–62. Academic Press, Tokyo (2010)
6. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local aligment search tool. J. Mol. Biol. **215**(3), 403–410 (1990)
7. Shen, H.-B., Chou, K.-C.: Ezypred: A top-down approach for predicting enzyme functional classes and subclasses. Biochem. Biophys. Res. Commun. **364**(1), 53–59 (2007)
8. Cozzetto, D., Minneci, F., Currant, H., Jones, D.T.: Ffpred 3: feature-based function prediction for all gene ontology domains. Sci. Rep. **6**, 1–11 (2016)
9. Dalkıran, A., Rifaioğlu, A.S., Martin, M.J., Çetin, A.R., Atalay, V., Doğan, T.: Ecpred: a tool for the prediction of the enzymatic functions of protein sequences based on the ec nomenclature. BMC Bioinf. **19**, 1–13 (2018)
10. Gong, Q., Ning, W., Tian, W.: Gofdr: A sequence alignment based method for predicting protein functions. Methods **93**(2), 3–14 (2016)
11. Asgari, E., Mofrad, M.R.K.: Continuous distributed representation of biological sequences for deep proteomics and genomics. PLoS ONE **10**(11), 1–15 (2015)
12. Naveenkumar, K.S., R., M.H.B., Vinayakumar, R., Soman, K.P.: Protein family classification using deep learning. Preprint at https://www.biorxiv.org/content/10.1101/414128v2 (2018)
13. Strodthoff, N., Wagner, P., Wenzel, M., Samek, W.: Udsmprot: universal deep sequence models for protein classification. Bioinformatics **36**(8), 2401–2409 (2020)
14. Le, N.Q.K., Yapp, E.K.Y., Nagasundaram, N., Chua, M.C.H., Yeh, H.-Y.: Computational identification of vesicular transport proteins from sequences using deep gated recurrent units architecture. Comput. Struct. Biotechnol. J. **17**, 1245–1254 (2009)
15. Li, S., Chen, J., Liu, B.: Protein remote homology detection based on bidirectional long short-term memory. BMC Bioinf. **18**, 1–8 (2017)
16. Bileschi, M.L., Belanger, D., Bryant, D., Sanderson, T., Carter, D.B., Sculley DePristo, M.A., Colwell, L.J.: Using deep learning to annotate the protein universe. Nat. Biotechnol. **40**(6), 932–937 (2022)
17. Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, X., Canny, J., Abbeel, P., Song, Y.S.: Evaluating protein transfer learning with tape. Adv. Neural Inf. Process. Syst. **32**, 9689–9701 (2019)
18. Belzen, J.U.Z., Bürgel, T., Holderbach, S., Bubeck, F., Adam, L., Gandor, C., Klein, M., Mathony, J., Pfuderer, P., Platz, L., Przybilla, M., Schwendemann, M., Heid, D., Hoffmann, M.D., Jendrusch, M., Schmelas, C., Waldhauer, M., Lehmann, I., D., N., Eils, R.: The index of general nonlinear DAES. Nat. Mach. Intell. **1**, 225–235 (2019)
19. Torrisi, M., Pollastri, G., Le, Q.: Deep learning methods in protein structure prediction. Comput. Struct. Biotechnol. Jo. **18**, 1301–1310 (2020)
20. Gustafsson, C., Minshull, J., Govindarajan, S., Ness, J., Villalobos, A., Welch, M.: Engineering genes for predictable protein expression. Protein Expr. Purif. **83**(1), 37–46 (2012)
21. Latchman, D.S.: Transcription factors: An overview. Int. J. Biochem. Cell Biol. **29**(12), 1305–1312 (1997)
22. Jin, J., Zhang, H., Kong, L., Gao, G., Luo, J.: Planttfdb 3.0: a portal for the functional and evolutionary study of plant transcription factors. Nucl. Acids Res. **42**(D1), 1182–1187 (2014)
23. Jin, J., Tian, F., Yang, D.-C., Meng, Y.-Q., Kong, L., Luo, J., Gao, G.: Planttfdb 4.0: toward a central hub for transcription factors and regulatory interactions in plants. Nucl. Acids Res. **45**(D1), 1040–1045 (2017)
24. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–444 (2015)
25. on Biochemical Nomenclature (CBN), I.-I.C.: A one-letter notation for amino acid sequences tentative rules. European J. Biochem. **7**(8), 151–153 (1968)
26. Ofer, D., Brandes, N., Linial, M.: The language of proteins: Nlp, machine learning & protein sequences. Comput. Struct. Biotechnol. J. **19**, 1750–1758 (2021)

27. Pfam: Family: HLH (PF00010). Available at http://pfam.xfam.org/family/PF00010 (Access date: February 2019)
28. Schuster-Böckler, B., Schultz, J., Rahmann, S.: Hmm logos for visualization of protein families. BMC Bioinf. **5**, 1–8 (2004)
29. Vries, J.K., Liu, X., Bahar, I.: The relationship between n-gram patterns and protein secondary structure. Proteins **68**(4), 830–9838 (2007)
30. Vries, J.K., Liu, X.: Subfamily specific conservation profiles for proteins based on n-gram patterns. BMC Bioinf. **9**, 1–13 (2008)
31. Greff, K., Srivastava, R.K., Koutnik, J., Steunebrink, B.R., Schmidhuber, J.: Lstm: a search space odyssey. Trans. Neural Netw. Learn. Syst. **28**(10), 2222–2232 (2017)
32. Gao, Y., Glowacka, D.: Deep gate recurrent neural network. In: JMLR: Workshop and Conference Proceedings 63, 350–365 (2016)
33. Kingma, D.P., Ba, J.L.: ADAM: A Method for Stochastic Optimization. In: Paper presented at International Conference on Learning Representations (ICLR), pp. 7–9 May 2015 (2014)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.