Received: 7 January 2021 | Revised: 21 April 2021 | Accepted: 11 May 2021

DOI: 10.1002/prot.26149

RESEARCH ARTICLE

PROTEINS WILEY

IGPRED: Combination of convolutional neural and graph convolutional networks for protein secondary structure prediction

Yasin Görmez[1] | Mostafa Sabzekar[2] | Zafer Aydın[3]

[1]Faculty of Economics and Administrative Sciences, Management Information Systems, Sivas Cumhuriyet University, Sivas, Turkey

[2]Department of Computer Engineering, Birjand University of Technology, Birjand, Iran

[3]Engineering Faculty, Computer Engineering Department, Abdullah Gül University, Kayseri, Turkey

Correspondence
Yasin Görmez, Faculty of Economics and Administrative Sciences, Management Information Systems, Sivas Cumhuriyet University, Sivas, Turkey.
Email: yasin.gormez@agu.edu.tr; yasingormez@cumhuriyet.edu.tr

Abstract

There is a close relationship between the tertiary structure and the function of a protein. One of the important steps to determine the tertiary structure is protein secondary structure prediction (PSSP). For this reason, predicting secondary structure with higher accuracy will give valuable information about the tertiary structure. Recently, deep learning techniques have obtained promising improvements in several machine learning applications including PSSP. In this article, a novel deep learning model, based on convolutional neural network and graph convolutional network is proposed. PSIBLAST PSSM, HHMAKE PSSM, physico-chemical properties of amino acids are combined with structural profiles to generate a rich feature set. Furthermore, the hyper-parameters of the proposed network are optimized using Bayesian optimization. The proposed model IGPRED obtained 89.19%, 86.34%, 87.87%, 85.76%, and 86.54% Q3 accuracies for CullPDB, EVAset, CASP10, CASP11, and CASP12 datasets, respectively.

KEYWORDS

Bayesian optimization, convolutional neural network, deep learning, graph convolutional network, protein secondary structure prediction

1 | INTRODUCTION

Proteins have crucial importance for the living organisms. Experimental methods that are used to determine the protein structure are time consuming and costly. Protein tertiary (3D) structure prediction is still one of the unsolved problems of computational biology and it is an alternative to experimental methods. Prediction of 3D structure is also useful for enzyme and drug design and helps to elucidate the function of the target protein.[1] Prediction of 3D structure from amino acid sequence benefits from the prediction of other structural properties, such as secondary structure, torsion angle, solvent accessibility. and contact maps.[2] Accurate prediction of these elements provides significant information about the 3D structure of a protein.

The first protein secondary structure prediction (PSSP) algorithms were based on the tendency of each amino acid to form leaves or helices, and rules to predict the formation of secondary structural elements. These algorithms were able to reach an overall accuracy of 60%.[3] Subsequently, a significant improvement in accuracy was achieved reaching to 77%-80% using information contained in multiple sequence alignments, which was derived by the PSI-BLAST algorithm.[4] Thereafter, in most of the studies, PSI-BLAST profiles, expressed as position specific scoring matrices (PSSMs), have been used as input features of machine learning methods.[3,5] However, using PSI-BLAST PSSMs alone as input features may not be the best solution due to noise contained in PSI-BLAST alignments and the availability of other more sensitive alignment methods. In order to enrich the feature set, Aydin et al developed a new PSSM using the HHblits alignment algorithm, which was shown to improve the prediction accuracy by 2% to 3%.[3] Another type of input feature is the structural profile, which is derived by aligning the target protein to template proteins with known structures. With the incorporation of structural profiles to the feature set, the accuracy of PSSP further

increased to 84%-85%.[6,7] Aydin et al obtained a prediction accuracy of 83% to 84% when distant templates are used only to compute structural profiles and 91% to 93% when close templates are also used.[8] These progresses show that using informative features helps to improve the accuracy of predicting secondary structure information.[3]

It should be noted that the efforts for improving the prediction accuracy is not limited to deriving better feature representations only. To date, many machine learning models have been developed for predicting secondary structure. Hua and Sun obtained a 73.5% accuracy with support vector machines (SVM) model trained on CB513 and RS126 datasets.[9] Aydin et al achieved an 80.3% accuracy score with the model trained using dynamic Bayesian network (DBN) and SVM.[3] Huang and Chen used four physicochemical features (net charges, side chain mass, hydrophobic and conformation parameters) as an additional feature and obtained a 79.52% accuracy rate with an SVM model.[10] In another study, Wang et al used genetic algorithm and grid search to optimize the parameters of SVM and reached a 76.11% accuracy.[11] Recurrent neural network is employed by Pollastri et al and obtained a 78% prediction accuracy.[7] Yao et al obtained a 78.1% accuracy by a method called DBNN that used dynamic Bayesian network and neural network.[12] Aydin et al compared SVM with four different algorithms, such as extreme learning machines, k-nearest neighbors, random forest, and artificial neural networks. Based on the obtained results, SVM reported the best performance on EVAset with an 83.83% accuracy rate.[13] Jian-wei et al proposed a neural network model for secondary structure prediction and compared it with traditional back propagation algorithm. As a result, they obtained a 9% improvement.[14] Mirabello and Pollastri developed methods using bidirectional recurrent neural networks called Porter 4.0 and Paleale 4.0. Porter 4.0 obtained an 82.2% accuracy and Paleale 4.0 received an 80.0% accuracy.[5] In another study, Yaseen and Li applied a neural network on a dataset that was obtained by statistical-context based scores and achieved a 82.74% accuracy.[15] Yang et al proposed a novel nearest neighbor method that used both non-homologous and homologous characteristics of protein secondary structure and obtained 87.51% accuracy.[16]

In addition to standard machine learning methods, deep learning approaches have also been used for PSSP with increasing frequency in the last decade. Wang et al employed deep convolutional neural fields on five different datasets and obtained significantly better accuracy rate for all datasets.[17] Heffernan et al proposed bidirectional long short-term memory (LSTM) to capture nonlocal interactions and achieved an 83.9% accuracy for PSSP.[18] Fang et al proposed deep inception-inside-inception model called MUFOLD-SS and reached an 85.98% accuracy.[19] Ma et al applied a method called data partition and semi random subspace on six different datasets, such as 25PDB, CB513, CASP10, CASP11, CASP12, and T100 and achieved accuracy rates of 86.38%, 84.53%, 85.51%, 85.89%, 85.55%, and 85.09%, respectively.[20] Kumar et al applied a model based on convolutional neural network (CNN) and bidirectional recurrent neural network (BRNN) on four different datasets, such as CB6113, CB513, CASP10, and CASP11 and achieved Q3 accuracy rates of 85.4%, 85.4%, 83.7%, and 81.5%, respectively.[21] Hanson et al proposed a new method based on LSTM-BRNN and ResNet and achieved an 87.5% overall accuracy on a test set of 1213 proteins, selected from a larger set of 12 450 proteins derived using the PISCES server.[22] Xu et al showed that their proposed deep learning model based on CNN, LSTM, transformers layer, and multitask learning increases the accuracy of predicting structural properties of proteins.[23]

In this study, a novel deep learning model is introduced for PSSP that includes CNN and graph convolutional network (GCN). The hyper-parameters of the proposed model are optimized using the Bayesian optimization technique. The input features of this model include PSI-BLAST PSSMs, HHblits PSSMs, structural profile matrices, an extended set of physico-chemical properties and a noseq label. Utilizing structural profiles and new physico-chemical properties together with the proposed network architecture are among the novelties of this article.

2 | MATERIALS AND METHODS

2.1 | Problem definition

PSSP aims to assign a secondary structure class (loop, helix, or beta strand) to each amino acid of a given protein. Typically, supervised learning methods are used to predict secondary structure. For this purpose, a machine learning model is trained using a dataset, which contains proteins with known secondary structure labels. Protein secondary structure can be represented by eight classes (ie, 8-state notation) or three classes (ie, 3-state notation). In 8-state representation, the secondary structure labels are H, G, I, E, B, T, S and " ". In this study, 3-state prediction of secondary structure is performed. For this purpose, H, G and I are assigned to H, E and B are assigned to E, S, T, and " " are assigned to L.

2.2 | Datasets

2.2.1 | CullPDB

The CullPDB dataset of 9090 proteins dated as 18 October 2018 was downloaded from the PISCES server (http://dunbrack.fccc.edu/Guoli/pisces_download.php#culledpdb) with percentage of identity cut-off equal to 20%, maximum resolution to 2.5, maximum *R*-value to 1.0. Then, a maximum *R*-free cut-off of 1.0 was applied. The 3D coordinate information files of the CullPDB dataset were downloaded from the PDB database using the get_pdb.py script of the Rosetta software (https://www.rosettacommons.org) and the secondary structure labels were assigned using the DSSP program (https://swift.cmbi.umcn.nl/gv/dssp/DSSP_3.html). Torsion angles were also derived using a script called phipsi_linux, and the proteins for which the number of amino acids in the output of phipsi_linux was less than the original number of amino acids were eliminated. This resulted in 8552 proteins. A nonredundant hard test set of 22 proteins with 5229 amino acids was selected as CullPDB-test by performing pair-wise

blast alignments among 8552 proteins with a stringent *E*-value cut-off of 0.05 and the remaining set of 8530 proteins was taken as the train set (CullPDB-train). In other words, the proteins in the test set had *E*-value greater than 0.05 when aligned with proteins in train set.

2.2.2 | EVAset

Being one of the benchmark datasets in PSSP, EVAset originally consists of 3074 proteins selected from the PDB.[24] It was developed for assessing the accuracy of prediction methods in structure prediction tasks. In this work, proteins shorter than 30 amino acids were removed from the EVAset and a set of 2876 targets including 584 595 amino acids remained, which is used as a test set.

2.2.3 | CASP Datasets

The CASPs datasets were commonly used for assessing the accuracy of structure prediction methods including secondary structure prediction. In this article, CASP10, CASP11, and CASP12 data sets downloaded from the official web site http://predictioncenter.org/ are used as test sets. The secondary structure labels were assigned using the DSSP program. Some of the proteins were eliminated due to lack of PDB tags. The protein sequences that were too short (ie, those containing less than 30 amino acids) were also eliminated. As a result, 75 protein sequences for CASP10, 67 protein sequences for CASP11 and 39 protein sequences for CASP12 remained. The number of amino acids were obtained as 18 231 for CASP10, 17 179 for CASP11, and 11 246 for CASP12.

2.2.4 | Train sets for EVAset and CASP datasets

Separate train sets for EVAset and CASP datasets were obtained from the original CullPDB dataset. For this purpose, pairwise BLAST alignments with a stringent *E*-value cut-off of 0.05 between 8552 CullPDB proteins and each of the EVAset and CASP datasets were performed. Then, CullPDB proteins for which the E-value of the alignment is less than 0.05 were eliminated and the remaining set of 6068 proteins for EVAset, 7147 proteins for CASP10, 7156 proteins for CASP11, and 7164 proteins for CASP12 were taken as the train sets for the EVAset and CASP datasets, respectively. These are denoted as CullPDB-train-EVAset, CullPDB-train-CASP10, CullPDB-train-CASP11, and CullPDB-train-CASP12.

2.3 | Feature extraction

2.3.1 | PSI-BLAST position-specific scoring matrices

PSSMs are commonly used to represent patterns (ie, input features) in proteins. In this study, each target protein was aligned with the NR database using the PSI-BLAST algorithm. The number of iterations was set to three, *E*-value was set to 10 and inclusion *E*-value was set to 0.001. As a consequence, 20 scores were obtained for each amino acid (as the output of PSI-BLAST), which formed the PSSM with a dimension of 20 by *N*, where *N* is the number of amino acids. These values were normalized to the interval between 0 to 1 by applying a sigmoidal transformation as in Aydin et al.[3]

2.3.2 | HHMAKE position-specific scoring matrices

In this article, the HHBlits software[25] was used to compute HHMAKE PSSM features. Each protein was aligned with the NR20 database and the number of iterations was set to two, which is the default setting for HHblits. As a result, an HMM-profile model was obtained for each protein, which contains scores for emission probability, background probability, and transition probability distributions. First, a set of 20 PSSM scores was computed for each amino acid as log-odds ratios, which can be expressed as $\log_2\left(\frac{P_e(i,j)}{P_b(j)}\right)$ where $P_e(i, j)$ is the emission probability for the j^{th} amino acid at the i^{th} match state with $1 \le j \le 20$, $1 \le i \le N$, N being the number of amino acids in the target, $P_b(j)$ is the background probability of emitting the j^{th} amino acid. These scores were then converted to the interval [0,1] by applying a sigmoidal transformation as in Aydin et al.[3] As the second set of features, seven transition probability values of the HMM-profile are taken directly without applying any normalization. Finally, the three local diversity scores denoted as *Neff*, $Neff_1$, $Neff_D$ parameters of the HMM-profile[25] are taken and normalized by sigmoidal transformation. As a result, a PSSM with dimension 30 by *N* was obtained for each target. Special care was taken for the file format of the HMM-profiles. For instance, a value of a star character (ie, "*") represents zero for the emission probability of match states (ie, $P_e(i, j)$) and for transition probability scores. In the case of emission probabilities these values would be mapped first to minus infinity (when $\log_2$ transform is applied for computing log-odds score) and then to zero by the sigmoidal transformation. Therefore * values are directly taken as zeros in the final feature representation without computing the transformations explicitly. Furthermore, the values for *Neff*, $Neff_1$, $Neff_D$ parameters were divided by 1000 to obtain the actual values since these are expressed in units of 0.001 according to the file format. The values for emission probabilities, background probabilities and transition probabilities in HMM-profile files were divided by -1000 to obtain the $\log_2$ transform of the corresponding probability scores. These $\log_2$-transformed scores are later used directly to compute log-odds ratios as explained above or inverted to compute the transition probability scores.

2.3.3 | Structural profiles

A structural profile matrix (SPM) is a collection of probability distributions, in which each distribution shows the probability of a given amino acid to belong to one of the three secondary structure classes. The size of an SPM is three by *N*, where *N* is the number of amino

acids in the target protein. Aydin et al showed that using SPMs increased the accuracy of PSSP.[8,26] In this study, SPMs were computed using the HHblits method. In the first step, the target protein was aligned against the NR20 database. In the second step, HMM-profile of the target was aligned with the HMM-profiles of the proteins in the PDB99 database, which is an in-house (ie, customized) database derived by Aydin lab using the scripts and commands available in the user guide of HHblits starting from the PDB99 dataset obtained from the PISCES server. Finally, the SPM was computed as the weighted average of label frequencies resulting from the alignment between the target and templates in PDB99. Only distant templates were used to construct SPMs. For this purpose, templates having a percentage of sequence identity score above 20% were eliminated to minimize the impact of template similarity on accuracy rate. Details on computing the structural profiles can be found in the paper by Aydin et al[8] Once the SPMs are computed, each amino acid is represented by three scores, which are taken as the structural profile features.

2.3.4 | Physico-chemical properties

Amino acids in a protein may have different types of physico-chemical properties. Fang et al showed that summarizing these properties may improve the accuracy of prediction methods.[19] In this article, seven physico-chemical properties, including volume of side chains, polarity, polarizability, hydrophilicity, hydrophobicity, net charge index of side chains, and solvent accessible surface area, are used as the first set of physico-chemical input features for each amino acid.

In addition to using the standard set of seven physico-chemical properties, 35 features from the AAindex database (https://www.genome.jp/aaindex/) are also added forming the second set of 42 physico-chemical features. AAindex is a database of amino acid indices proposed by Kawashima et al.[27] It contains numerical values for more than 500 indices (representing various properties of amino acids), amino acid substitution matrices and amino acid contact potential matrices. In this article, a set of 35 indices listed in Table 1 are selected and added to the feature set.

TABLE 1 Selected amino acid indices from AAindex

AAindex accession number	Description of index
BHAR880101	Average flexibility indices[28]
BIGC670101	Residue volume[29]
PONJ960101	Average volumes of residues[30]
CHAM820102	Free energy of solution in water, kcal/mol[31]
CIDH920105	Normalized average hydrophobicity scales[32]
BASU050101	Interactivity scale obtained from the contact matrix[33]
BASU050102	Interactivity scale obtained by maximizing the mean of correlation[33] coefficient over single-domain globular proteins
ZHOH040101	The stability scale from the knowledge-based atom-atom potential[34]
ZHOH040102	The relative stability scale extracted from mutation experiments[34]
ZHOH040103	Buriability[34]
WOLR790101	Hydrophobicity index[35]
KIDA850101	Hydrophobicity-related index[36]
FASG890101	Hydrophobicity index[37]
KRIW710101	Side chain interaction parameter[38]
SIMZ760101	Transfer free energy[39]
ROBB790101	Hydration free energy[40]
RADA880108	Mean polarity[41]
ROSM880102	Side chain hydropathy, corrected for solvation[42]
VELV850101	Electronion interaction potential[43]
WARP780101	Average interactions per side chain atom[44]
WOLR810101	Hydration potential[45]
HOPA770101	Hydration number[46]
ZIMJ680101	Hydrophobicity[47]
ZIMJ680102	Bulkiness[47]
GRAR740102	Polarity[48]
ZIMJ680104	Isoelectric point[47]
ZIMJ680105	RF rank[47]
TAKK010101	Side-chain contribution to protein stability (kJ/mol)[49]
MEIH800103	Average side chain orientation angle[50]
MCMT640101	Refractivity[51]
HUTJ700102	Absolute entropy[52]
HUTJ700103	Entropy of formation[52]
FAUJ880103	Normalized van der Waals volume[53]
FASG760103	Optical rotation[54]
FASG760101	Molecular weight[54]

2.4 | Summary of input features

In this article, three different feature sets are computed. The first one includes 58 real-valued features for each amino acid: 20 features from the corresponding column of PSI-BLAST profile (ie, PSI-BLAST PSSM), 30 features from the corresponding column of HHBlits profile (ie, HHMAKE PSSM), seven features for physico-chemical properties of amino acids, and one feature for the no sequence (NoSeq) label (explained below). Note that, this is the same feature representation as in the MUFOLD-SS paper.[19] The second set includes 61 features for each amino acid, in which three structural profile features are added to the first feature set. The third representation extends the physico-chemical features by adding 35 features from the AAindex database and contains 96 features in total for each amino acid. Figure 1 shows the pipeline of the classification model for the second and third feature representations. Different from the literature, structural profile features and the new physico-chemical features are

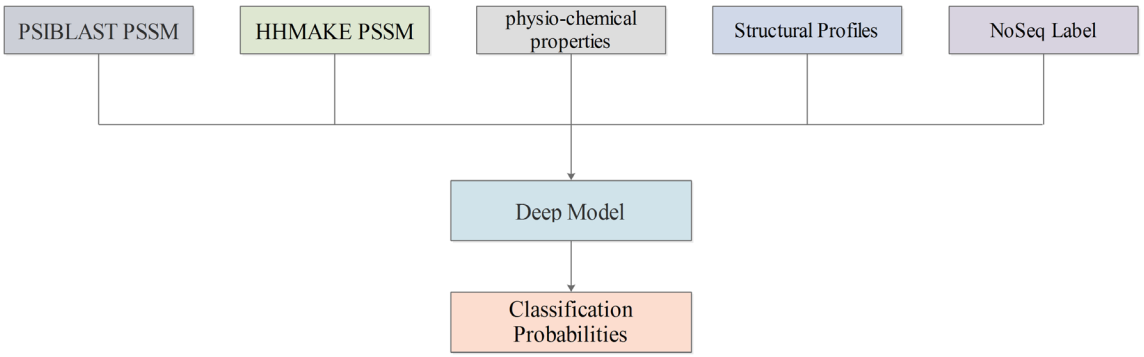

FIGURE 1 Pipeline of the classification model

added to the feature set and employed by the newly proposed network architecture.

It should be noted that each protein contains multiple amino acids, which represent the time dimension. Therefore, input feature data for each protein can be represented as a 2D array with rows representing amino acids and columns representing features.

The NoSeq label is included to the feature set because the proposed deep neural networks are designed to take a fixed sized input in time dimension (ie, they expect the number of amino acids to be the same). If the number of amino acids in a target protein is less than the sequence length parameter, zero padding is applied for the remaining time steps and NoSeq label will be set to 1. Otherwise, NoSeq label will be set to 0. In this article, the sequence length parameter of the network model is set to 700. For example, if the length of the target protein is 500, then the first 500 rows of the data matrix contain features computed as explained above (ie, including PSI-BLAST PSSM, HHMAKE PSSM, structural profiles and physico-chemical properties) and the NoSeq label is set to zero. The last 200 rows have zeros as the values of all features and the NoSeq label is set to one. This representation is also the same as in the MUFOLD-SS paper.[19] If the number of amino acids in a target protein is equal to the sequence length parameter, no zero padding will be applied and the NoSeq label will be set to zero in the data array of the protein. If the number of amino acids in a target is more than the sequence length parameter, the protein's data will be divided into multiple parts. For example, if the protein contains 1000 amino acids, then its data will contain two parts. The first part will include features for the first 700 amino acids (with NoSeq label set to zero). The second part will contain features for the remaining 300 amino acids (with NoSeq label set to zero) as well as zero-valued features for 400 time steps (with NoSeq label set to one). There are 125 proteins in CullPDB train set, 49 proteins in EVAset and one protein in CASP12, which have more than 700 amino acids. Other datasets do not have any proteins that are longer than 700.

The value ranges of the features are as follows: PSI-BLAST PSSMs and HHMAKE PSSMs range from 0 to 1, physicochemical properties from -1 to 1 and NoSeq label can take a value of 0 or 1.

2.5 | Deep learning model architecture

In this study, a new deep learning model called IGPRED is proposed, which consists of several CNN and GCN modules concatenated in different ways and followed by fully connected layers. Figure 2 shows the architecture of the proposed model.

Each CNN module is generated through six different convolutional layers with kernel sizes (1, *M*), (3, *M*), (5, *M*), (9, *M*), (11, *M*), and (15, *M*) that are connected in parallel as in inception module. Here, *M* represents the number of features because 1-D convolutional layers are used. These layers are connected using the inception architecture, which obtained significant improvements in many areas such as computer vision and bioinformatics.[55-57] Within each CNN module, the number of filters of the convolutional layers are identical and is a hyper-parameter that can be set for each module. As a result, different CNN modules can contain different number of filters. Each convolution layer consists of four operations in sequential order: Convolution, batch normalization, activation, and dropout. Figure 3 shows the details of each CNN module.

A GCN module is generated using a multigraph convolutional layer (mGCN),[58] which consists of two operations in sequential order: A multigraph convolution and dropout. An mGCN layer needs a graph as an extra input for each protein. As mentioned above, the sequence length parameter is set to 700. Therefore, a graph with 700 nodes is generated for each protein where each node represents an amino acid and edges represent interactions between amino acids, which can be summarized by an adjacency matrix of size 700 by 700. This graph is unweighted for which the adjacency matrix contains ones or zeros only. If there is an edge between nodes *m* and *n*, the value at row *m* and column *n* of the adjacency matrix is one, which represents an interaction between amino acids at positions *m* and *n*. This graph is also undirected with a symmetric adjacency matrix. This means that if there is an edge between nodes *m* and *n*, there will also be an edge

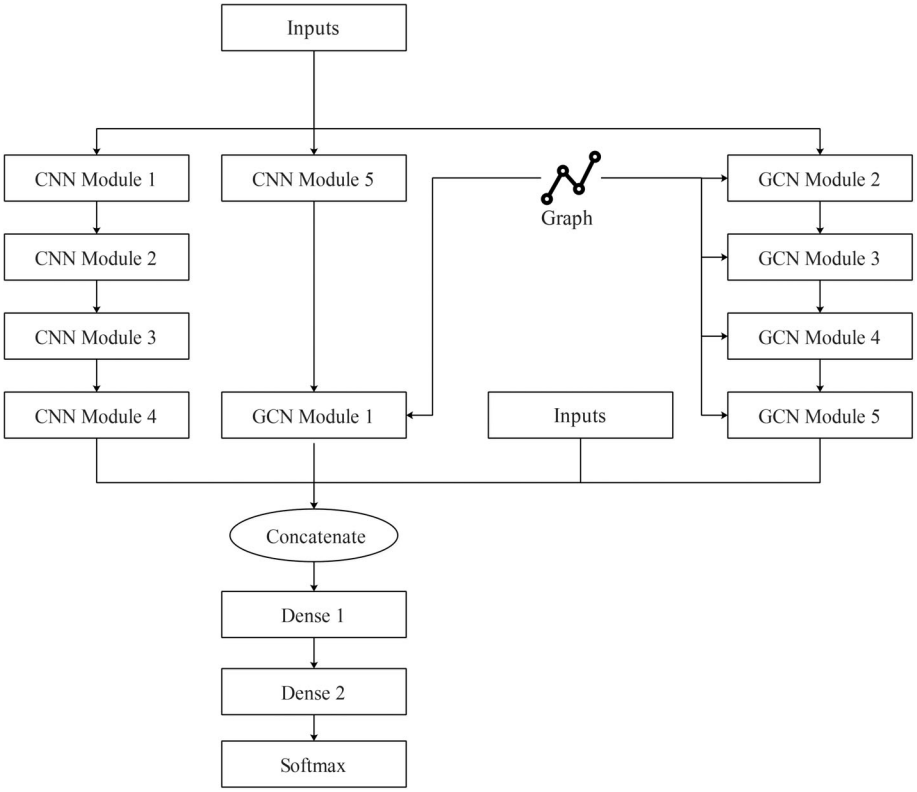

FIGURE 2 Architecture of IGPRED

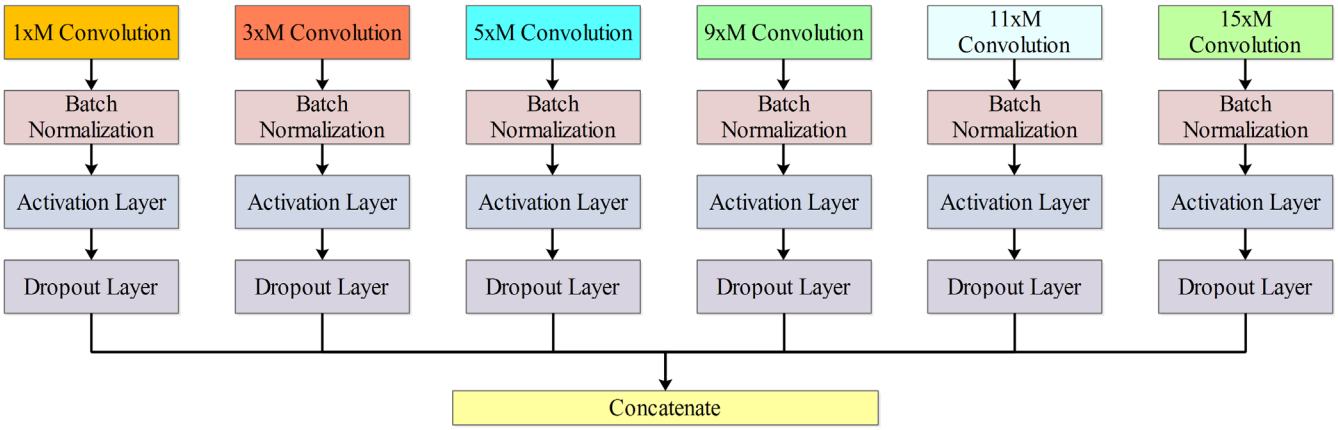

FIGURE 3 The layers inside each convolutional neural network (CNN) module

between nodes *n* and *m*. A total of *N* disconnected graphs each with an adjacency matrix of size 700 by 700 is generated where *N* is the number of proteins in the dataset. This type of graph representation enables to capture short-range as well as long-range interactions by defining connections between interacting amino acids. In this article, short-range interactions are modeled only by selecting a two-sided symmetric window around each amino acid to define the interaction graph of proteins. Based on this constraint, a hyper-parameter called number of connections is introduced, which is equal to the number of interactions an amino acid makes on one side of the window. This value should be smaller than half of the number of amino acids in a protein. For example, if the number of connections is 10, then there will be an edge between a given amino acid and 10 amino acids that come before as well as 10 amino acids that come after according to the sequence representation of the protein.

2.6 | Hyper-parameter optimization

Hyper-parameters of a machine learning model are among the crucial factors for obtaining satisfactory accuracy rates. Although hyper-parameter optimization can affect the model performance positively, it can take a long time especially if the number of parameters that are going to be explored is high. Deep learning models contain many

hyper-parameters, such as learning rate, number of epochs, number of hidden units, regularization coefficients, and layer specific parameters. On the other hand, if the hyper-parameter space is selected as narrow, the optimum parameter combination can be missed resulting in suboptimal models.

In this article, the proposed model was implemented by using the Keras[59] library of Python and learning rate, the number of filters in convolution layers (n_filters_conv), batch size, the number of epochs, dropout rate, the number of hidden units (ie, neurons) in dense layers (n_dense), the number of connections in graphs (nconn) and the number of outputs in multi-graph layers (out_dim_gcn) are optimized using the Bayesian optimization technique, which can perform faster optimization than the traditional grid-search method in a wide parameter space. Note that a separate number of filters parameter is defined for each CNN module (ie, at the module level), a separate output dimension parameter is defined for each GCN module and a separate number of hidden units parameter for each dense layer. As there are five CNN modules, five GCN modules and two dense layers, considering the other hyper-parameters as well, a total of 17 hyper-parameters were defined and optimized. Table 2 shows the lowest and highest values of these hyper-parameters. In addition, the following hyper-parameter settings were used without optimization. Activation function for all layers except for the classification (ie, output) layer is ReLU, activation function for classification layer is softmax, loss function is cross entropy, weight optimizer is Adam, beta1 parameter of the Adam optimizer is 0.95, beta2 parameter of the Adam optimizer is 0.99 and the number of filters for each graph convolutional network module is 2.

Bayesian optimization algorithm was implemented using scikit-optimize library of Python.[60] Gp_minimize method was implemented using the following parameters settings: *acq_func* = "EI" and *n_calls* = 100.[61] Here the expected improvement (EI) is the function to minimize over the Gaussian prior and *n_calls* is the number of calls to the function. All the other parameters of *gp_minimize* are set to their default values. This method uses a Gaussian process with two aims: Modeling the surrogate function and optimizing the expected probability which is based on improving the existing best solutions through new trials. It assumes that the function values track a multivariate Gaussian distribution. A Gaussian kernel designates the covariance of the function values among the parameters. In each iteration, the next value of a parameter is selected through the acquisition function over the Gaussian prior.

TABLE 2 Hyper-parameter ranges of deep learning model used for optimization

Parameter	Lowest	Highest
Learning rate	10^{-6}	10^{-1}
n_filters_conv	50	150
Batch size	2^2	2^7
Epoch	10	200
Dropout rate	0	0.6
n_dense	400	1400
nconn	5	50
out_dim_gcn	20	200

3 | EXPERIMENTAL RESULTS

In order to assess the prediction performance of the proposed deep learning model, five different datasets were used as test sets including CullPDB-test, EVAset, CASP10, CASP11, and CASP12. For each benchmark, separate train sets were re-generated from the original CullPDB dataset by performing pairwise BLAST alignments as explained in Sections 2.2.1 and 2.2.4. Each dataset had two versions: The first one included structural profiles and the other one did not include structural profiles in the feature set.

3.1 | Hyper-parameter optimization

In the first step, the optimum hyper-parameters were found using CullPDB-train dataset. To achieve this, 10% of the proteins in CullPDB-train were randomly selected as the test set for optimization (ie, validation set called CullPDB-val) and the remaining proteins were selected as the train set for optimization denoted as CullPDB-train-opt. As explained in Section 2.6, 17 hyper-parameters were optimized by constraining them to take values in specified ranges and the remaining hyper-parameters were assigned to fixed values. Table 3 shows the optimized values of the hyper-parameters for both versions of the datasets (with and without using structural profiles in feature sets). In this table, n_filters_conv, n_dense and out_dim_gcn rows show the hyper-parameters for each module in the order shown in Figure 2.

3.2 | Adding structural profiles

After hyper-parameter optimization, a total of 10 different models were trained using the two versions of the five train sets (ie, with and

TABLE 3 Optimum hyper-parameters for the proposed deep learning model

Parameter	Dataset without structural profiles	Dataset with structural profiles
Learning rate	0.00011778	1.501×10^{-5}
n_filters_conv	103, 107, 96, 115, 98	119, 113, 115, 101, 112
Batch size	4	4
Epoch	96	78
Dropout rate	0.5	0.3
n_dense	547, 537	564, 473
Nconn	11	17
out_dim_gcn	123, 189, 87, 63, 71	101, 123, 116, 75, 25

TABLE 4 Accuracy measures of IGPRED without using structural profiles

Parameter	Accuracy	SOV	MCC 'H'	MCC 'E'	MCC 'L'	Recall 'H'	Recall 'E'	Recall 'L'	Precision 'H'	Precision 'E'	Precision 'L'
CullPDB-test	87.80%	83.12%	0.86	0.67	0.76	92.94%	69.43%	87.18%	90.29%	69.72%	89.15%
EVAset	86.12%	76.15%	0.82	0.64	0.72	89.53%	69.25%	86.32%	87.45%	64.27%	88.70%
CASP10	87.24%	75.95%	0.83	0.64	0.73	89.55%	55.62%	91.32%	86.95%	79.46%	88.24%
CASP11	85.26%	74.10%	0.80	0.59	0.68	89.15%	51.74%	88.90%	84.78%	79.98%	86.01%
CASP12	86.04%	78.01%	0.83	0.63	0.72	91.93%	65.92%	85.53%	86.13%	65.76%	89.67%

TABLE 5 Accuracy measures of IGPRED with structural profiles

Parameter	Accuracy	SOV	MCC 'H'	MCC 'E'	MCC 'L'	Recall 'H'	Recall 'E'	Recall 'L'	Precision 'H'	Precision 'E'	Precision 'L'
CullPDB-test	89.19%	87.15%	0.88	0.68	0.79	93.13%	67.73%	89.91%	92.29%	73.84%	89.24%
EVAset	86.34%	76.35%	0.82	0.61	0.73	89.50%	58.37%	88.31%	87.47%	70.32%	87.49%
CASP10	87.87%	75.19%	0.84	0.68	0.75	89.31%	68.28%	90.39%	88.54%	73.29%	89.76%
CASP11	85.76%	76.48%	0.82	0.61	0.71	88.73%	59.59%	88.64%	87.01%	70.40%	87.28%
CASP12	86.54%	77.50%	0.84	0.59	0.73	91.96%	50.66%	89.29%	87.00%	76.27%	87.37%

without structural profiles) and predictions are computed on the corresponding test sets. The same hyper-parameter configuration found for CullPDB-train is used in these experiments. Table 4 shows results for the datasets that did not use structural profiles and Table 5 includes the results when structural profiles are added to feature set. In these tables, Accuracy represents the overall accuracy (ie, Q3 measure), SOV[62] denotes the segment overlap measure, MCC[63] is the Matthew's correlation coefficient. MCC, recall and precision metrics are computed for each secondary structure class in a one vs all setting.

According to these results, the models that used structural profiles obtained better accuracy rates than models that did not use structural profiles. Using structural profiles increased the overall accuracy rate by 1.39% for CullPDB-test, 0.22% for EVAset, 0.63% for CASP10, 0.5% for CASP11, 0.50% for CASP12 compared with the models that did not use structural profiles.

When the MCC values of secondary structure classes are compared the highest values are obtained for helices, followed by loops, followed by strands. When the recall values are compared, except for CASP10, the same ordering is obtained: helices come first, followed by loops and then strands. For CASP10, the ordering is loops, followed by helices and then strands. Finally, when the precision values are compared, except for CullPDB-test, loops come first, followed by helices and then strands. For CullPDB-test, helix performance is the best, followed by loops and then strands. In all cases, the accuracy of beta-strands is lowest as compared to helices and loops. This is reasonable because the proposed model architecture can only capture local correlations between amino acids, which is characteristic of helices and loops. The lower accuracy rates for beta-strands can be due to non-local (ie, distant or long-range) interactions present between amino acids of beta-strand segments. It can be anticipated that if such interactions are predicted a priori (eg, in the form of contact maps) with sufficient accuracy and used as input to graph convolutional networks, the accuracy rates of beta-strands may improve.

TABLE 6 Q3 accuracy of IGPRED when new physico-chemical properties are added to feature set

CullPDB-test	EVAset	CASP10	CASP11	CASP12
89.17%	86.39%	87.66%	85.56%	86.24%

3.3 | Adding physico-chemical features from AAindex database

In addition to the effect of using structural profile features, adding new physico-chemical properties of the amino acids to the feature set (ie, using the third feature representation) is also explored. For this purpose, 35 AAindex values were selected and added to the feature set that includes structural profiles (among with other feature categories) and train/test experiments are repeated for the five benchmarks. Table 6 shows the overall Q3 accuracies of the proposed method on five benchmarks.

These results are comparable to those presented in Table 5. Therefore, using AAindex values as additional physico-chemical features did not improve the overall prediction accuracy.

3.4 | Comparison with the state-of-the-art

In the next step, our model, IGPRED, is compared with the state-of-the-art methods in the literature. Table 7 shows the overall accuracy of MUFOLD-SS,[19] OPUS-TASS[23] and our model on CASP datasets.

TABLE 7 Q3 accuracy comparison between IGPRED and state-of-the-art methods on CASP datasets

Method	CASP10	CASP11	CASP12
MUFOLD-SS	86.49%	85.20%	83.36%
OPUS-TASS	–	–	85.47%
IGPRED without SP	87.24%	85.26%	85.76%
IGPRED with SP	87.87%	85.76%	86.54%

Based on these results, our model outperforms the state-of-the-art methods on all of the CASP datasets. The improvements obtained when structural profiles are included to the feature set are statistically significant according to a two-tailed Z-test[64] with *P*-values less than .01 for all CASP datasets. Note that the results of MUFOLD-SS and OPUS-TASS are taken from the Reference papers 19 and23, respectively, while our results are obtained on subsets of the CASP datasets (ie, on CASP proteins that have PDB IDs and that are not short as described in Section 2.2.3). The reason for this difference is because the CASP datasets (including the label assignments) used by MUFOLD-SS are not shared publicly. Based on this, the results presented in Table 7 contain variance components due to slightly different versions of the datasets being used. Nonetheless, obtaining improvements consistently on all datasets is promising.

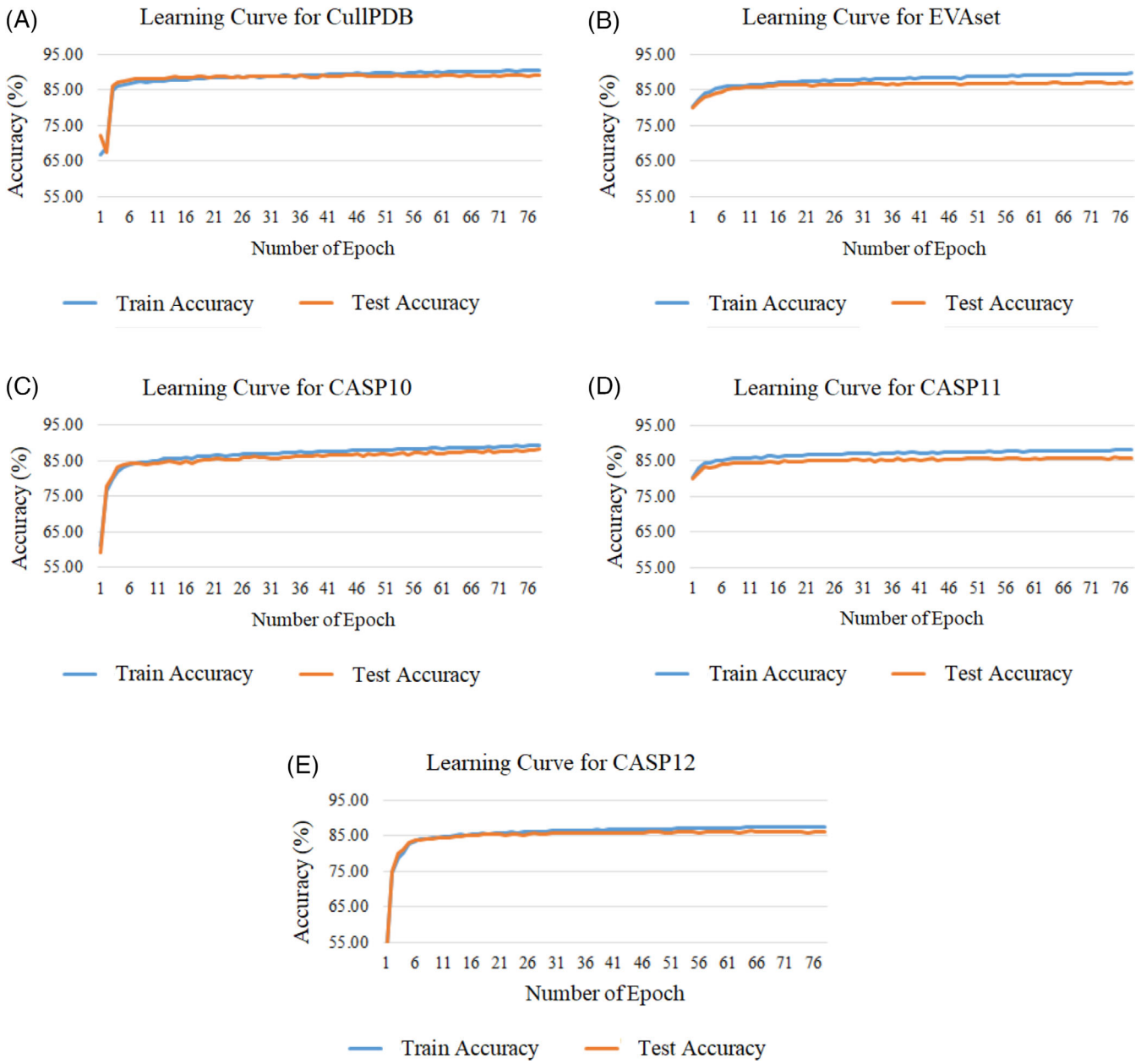

FIGURE 4 Learning curves for models that use structural profiles: A, CullPDB; B, EVAset; C, CASP10; D, CASP11; and E, CASP12

3.5 | Learning curves

Another factor that can be analyzed is the overfitting behavior of the models. Deep learning models with many layers can be prone to overfitting due to large number of weight coefficients learned during training. In the proposed model architecture, batch normalization and dropout are used as regularization techniques to prevent overfitting. In order to understand whether a model has learned the patterns in train set well enough and is able to generalize to new examples, learning curves can be used. Figure 4 shows the learning curves of the model for different datasets. In all of these experiments, structural profiles were used in the feature vector. Each subfigure shows the accuracy of the model with respect to number of epochs, which is directly proportional to model complexity. These curves show that our model did not suffer from significant amount of overfitting since the test curves follow the train curves closely.

3.6 | Accuracy with respect to length of protein

In this article, long proteins were split naively using blocks of 700 amino acids as explained in Section 2.4. This was preferred for computational reasons. In order to analyze whether this type of splitting degrades performance, the overall accuracy values were computed for proteins belonging to different length intervals (including those that are longer than 700 amino acids) in EVAset, which is the only benchmark that has sufficient number of long proteins. The results of this experiment are given in Table 8.

Based on these results, as the length of the protein increases, there is only a minor decrease in the overall prediction accuracy. For example, the accuracy obtained for the fourth length interval is around 0.2% lower than the accuracy of the third length interval both of which contain proteins with no splitting applied. This shows that as the protein length increases, the prediction accuracy may decrease slightly even if there is no splitting due to missed correlations caused by long-range interactions (the convolutional networks developed in this study only model short-range interactions between the amino acids). On the other hand, the accuracy of the last length interval (that contains proteins longer than 700 amino acids) is only 0.1% lower than the accuracy of the fourth interval and only 0.3% lower than the top performing second and third intervals. Based on this result, although the naïve splitting approach ignores domain boundaries and any interactions between the split regions, it can be anticipated that the splitting process has little contribution to the decrease in accuracy. This can be because the proposed model already ignores long-range interactions and the decrease in performance for long proteins can be mainly due to this restriction. If there is an extra degrade in performance due to splitting, that may happen due to missed correlations/interactions around the boundaries of the feature matrix (ie, at positions close to every 700th amino acid) which may affect a few amino acids only. As a result, the naïve splitting approach can be preferred due to its computational simplicity without degrading performance considerably.

TABLE 8 Q3 accuracy of IGPRED on EVAset at different length intervals

Length intervals	Q3 accuracy
1 to 175 amino acids	86.39%
176 to 350 amino acids	86.44%
351 to 525 amino acids	86.41%
526 to 700 amino acids	86.23%
> 700 amino acids	86.11%

4 | CONCLUSIONS

In this study, a deep inception model architecture is combined with graph convolutional network to predict secondary structure of proteins with high accuracy rates. As the graph input and in inception networks only short-range interactions between amino acids are considered. As a future work, long-range interactions will also be included by feeding contact maps or distance maps as inputs to graph convolutional networks. Furthermore, several deep learning layers, such as bidirectional recurrent networks, will be added to the proposed model, which will be trained for predicting secondary structure, solvent accessibility and torsion angle information of proteins.

ACKNOWLEDGMENTS

The experiments reported in this article were partially performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources), the National Center for High Performance Computing of Turkey (UHeM) under project no 5004062016, and AGÜ HPC.

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1002/prot.26149.

DATA AVAILABILITY STATEMENT

Data available on request from the authors.

ORCID

Yasin Görmez https://orcid.org/0000-0001-8276-2030

REFERENCES

1. Klebe G. Protein modeling and structure-based drug design. In: Klebe G, ed. *Drug Design: Methodology, Concepts, and Mode-of-Action*. Berlin, Germany: Springer; 2013:429-448. https://doi.org/10.1007/978-3-642-17907-5_20.
2. Deng H, Jia Y, Zhang Y. Protein structure prediction. *Int J Mod Phys B*. 2018;32(18). https://www.worldscientific.com/doi/10.1142/S021797921840009X.
3. Aydin Z, Singh A, Bilmes J, Noble WS. Learning sparse models for a dynamic Bayesian network classifier of protein secondary structure. *BMC Bioinformatics*. 2011;12(1):154. https://doi.org/10.1186/1471-2105-12-154.

4. McGuffin LJ, Bryson K, Jones DT. The PSIPRED protein structure prediction server. *Bioinformatics*. 2000;16(4):404-405. https://doi.org/10.1093/bioinformatics/16.4.404.
5. Mirabello C, Pollastri G. Porter, PaleAle 4.0: high-accuracy prediction of protein secondary structure and relative solvent accessibility. *Bioinformatics*. 2013;29(16):2056-2058. https://doi.org/10.1093/bioinformatics/btt344.
6. Li D, Li T, Cong P, Xiong W, Sun J. A novel structural position-specific scoring matrix for the prediction of protein secondary structures. *Bioinformatics*. 2012;28(1):32-39. https://doi.org/10.1093/bioinformatics/btr611.
7. Pollastri G, Martin AJ, Mooney C, Vullo A. Accurate prediction of protein secondary structure and solvent accessibility by consensus combiners of sequence and structure information. *BMC Bioinformatics*. 2007;8(1):201. https://doi.org/10.1186/1471-2105-8-201.
8. Aydin Z, Azginoglu N, Bilgin HI, Celik M. Developing structural profile matrices for protein secondary structure and solvent accessibility prediction. *Bioinformatics*. 2019;35(20):4004-4010. https://doi.org/10.1093/bioinformatics/btz238.
9. Hua S, Sun Z. A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. *J Mol Biol*. 2001;308(2):397-407. https://doi.org/10.1006/jmbi.2001.4580.
10. Huang YF, Chen SY. Protein secondary structure prediction based on physicochemical features and PSSM by SVM. Paper presented at: 2013 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB). 2013:9-15 doi:https://doi.org/10.1109/CIBCB.2013.6595382
11. Wang Y, Cheng J, Liu Y, Chen Y. Prediction of protein secondary structure using support vector machine with PSSM profiles. Paper presented at: 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference; 2016:502–505. doi: https://doi.org/10.1109/ITNEC.2016.7560411
12. Yao X-Q, Zhu H, She Z-S. A dynamic Bayesian network approach to protein secondary structure prediction. *BMC Bioinformatics*. 2008;9(1):49. https://doi.org/10.1186/1471-2105-9-49.
13. Aydin Z, Kaynar O, Görmez Y, Işik YE. Comparison of machine learning classifiers for protein secondary structure prediction. Paper presented at: 2018 26th Signal Processing and Communications Applications Conference (SIU); 2018:1–4. doi:https://doi.org/10.1109/SIU.2018.8404547
14. Jian-wei L, Guang-hui C, Hai-en L, Yuan L, Xiong-lin L. Prediction of protein secondary structure using multilayer feed-forward neural networks. Paper presented at: 2013 25th Chinese Control and Decision Conference (CCDC); 2013:1346–1351. doi:https://doi.org/10.1109/CCDC.2013.6561135
15. Yaseen A, Li Y. Context-based features enhance Protein secondary structure prediction accuracy. *J Chem Inf Model*. 2014;54(3):992-1002. https://doi.org/10.1021/ci400647u.
16. Wei Yang, Kuanquan Wang, Wangmeng Zuo. A fast and efficient nearest neighbor method for protein secondary structure prediction. Paper presented at: 2011 3rd International Conference on Advanced Computer Control; 2011:224–227. doi:https://doi.org/10.1109/ICACC.2011.6016402
17. Wang S, Peng J, Ma J, Xu J. Protein secondary structure prediction using deep convolutional neural fields. *Sci Rep*. 2016;6:18962. https://doi.org/10.1038/srep18962.
18. Heffernan R, Yang Y, Paliwal K, Zhou Y. Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility. *Bioinformatics*. 2017;33(18):2842-2849. https://doi.org/10.1093/bioinformatics/btx218.
19. Fang C, Shang Y, Xu D. MUFOLD-SS: new deep inception-inside-inception networks for protein secondary structure prediction. *Proteins Struct Funct Bioinform*. 2018;86(5):592-598. https://doi.org/10.1002/prot.25487.
20. Ma Y, Liu Y, Cheng J. Protein secondary structure prediction based on data partition and semi-random subspace method. *Sci Rep*. 2018;8(1):1-10. https://doi.org/10.1038/s41598-018-28084-8.
21. Kumar P, Bankapur S, Patil N. An enhanced protein secondary structure prediction using deep learning framework on hybrid profile based features. *Appl Soft Comput*. 2020;86:105926. https://doi.org/10.1016/j.asoc.2019.105926.
22. Hanson J, Paliwal K, Litfin T, Yang Y, Zhou Y. Improving prediction of protein secondary structure, backbone angles, solvent accessibility and contact numbers by using predicted contact maps and an ensemble of recurrent and residual convolutional neural networks. *Bioinformatics*. 2019;35(14):2403-2410. https://doi.org/10.1093/bioinformatics/bty1006.
23. Xu G, Wang Q, Ma J. OPUS-TASS: a protein backbone torsion angles and secondary structure predictor based on ensemble neural networks. *Bioinformatics*. 2020;36(20):5021-5026. https://doi.org/10.1093/bioinformatics/btaa629.
24. Koh IYY, Eyrich VA, Marti-Renom MA, et al. EVA: evaluation of protein structure prediction servers. *Nucleic Acids Res*. 2003;31(13):3311-3315. https://doi.org/10.1093/nar/gkg619.
25. Remmert M, Biegert A, Hauser A, Söding J. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nat Methods*. 2012;9(2):173-175. https://doi.org/10.1038/nmeth.1818.
26. Aydin Z, Baker D, Noble WS. Constructing structural profiles for protein torsion angle Prediction: SciTePress; 2015. Accessed December 3, 2017. https://iths.pure.elsevier.com/en/publications/constructing-structural-profiles-for-protein-torsion-angle-predic
27. Kawashima S, Pokarowski P, Pokarowska M, Kolinski A, Katayama T, Kanehisa M. AAindex: amino acid index database, progress report 2008. *Nucleic Acids Res*. 2008;36(suppl 1):D202-D205. https://doi.org/10.1093/nar/gkm998.
28. Bhaskaran R, Ponnuswamy PK. Positional flexibilities of amino acid residues in globular proteins. *Int J Pept Protein Res*. 1988;32(4):241-255. https://doi.org/10.1111/j.1399-3011.1988.tb01258.x.
29. Bigelow CC. On the average hydrophobicity of proteins and the relation between it and protein structure. *J Theor Biol*. 1967;16(2):187-211. https://doi.org/10.1016/0022-5193(67)90004-5.
30. Pontius J, Richelle J, Wodak SJ. Deviations from standard atomic volumes as a quality measure for Protein crystal structures. *J Mol Biol*. 1996;264(1):121-136. https://doi.org/10.1006/jmbi.1996.0628.
31. Charton M, Charton BI. The structural dependence of amino acid hydrophobicity parameters. *J Theor Biol*. 1982;99(4):629-644. https://doi.org/10.1016/0022-5193(82)90191-6.
32. Cid H, Bunster M, Canales M, Gazitúa F. Hydrophobicity and structural classes in proteins. *Protein Eng des Sel*. 1992;5(5):373-375. https://doi.org/10.1093/protein/5.5.373.
33. Bastolla U, Porto M, Roman HE, Vendruscolo M. Principal eigenvector of contact matrices and hydrophobicity profiles in proteins. *Proteins Struct Funct Bioinform*. 2005;58(1):22-30. https://doi.org/10.1002/prot.20240.
34. Zhou H, Zhou Y. Quantifying the effect of burial of amino acid residues on protein stability. *Proteins Struct Funct Bioinform*. 2004;54(2):315-322. https://doi.org/10.1002/prot.10584.
35. Wolfenden RV, Cullis PM, Southgate CC. Water, protein folding, and the genetic code. *Science*. 1979;206(4418):575-577. https://doi.org/10.1126/science.493962.
36. Kidera A, Konishi Y, Oka M, Ooi T, Scheraga HA. Statistical analysis of the physical properties of the 20 naturally occurring amino acids. *J Protein Chem*. 1985;4(1):23-55. https://doi.org/10.1007/BF01025492.
37. Fasman GD. *Prediction of Protein Structure and the Principles of Protein Conformation*. New York: Springer Science & Business Media; 2012.

38. Krigbaum WR, Rubin BH. Local interactions as a structure determinant for globular proteins. *Biochim Biophys Acta BBA Protein Struct*. 1971;229(2):368-383. https://doi.org/10.1016/0005-2795(71)90196-6.
39. Perutz MF, Kilmartin JV, Nagai K, Szabo A, Simon SR. Influence of globin structures on the state of the heme. IV. Ferrous low spin derivatives. *Biochemistry*. 1976;15(2):378-387. https://doi.org/10.1021/bi00647a022.
40. Robson B, Osguthorpe DJ. Refined models for computer simulation of protein folding: applications to the study of conserved secondary structure and flexible hinge points during the folding of pancreatic trypsin inhibitor. *J Mol Biol*. 1979;132(1):19-51. https://doi.org/10.1016/0022-2836(79)90494-7.
41. Lee AW, Karplus M, Poyart C, et al. Comparing the polarities of the amino acids: side-chain distribution coefficients between the vapor phase, cyclohexane, 1 -0ctano1, and neutral aqueous solution.
42. Roseman MA. Hydrophilicity of polar amino acid side-chains is markedly reduced by flanking peptide bonds. *J Mol Biol*. 1988;200(3):513-522. https://doi.org/10.1016/0022-2836(88)90540-2.
43. Veljkovic V, Cosic I, Dimitrijevic LD. Is it possible to analyze DNA and Protein sequences by the methods of digital signal processing? *IEEE Trans Biomed Eng*. 1985;BME-32(5):337-341. https://doi.org/10.1109/TBME.1985.325549.
44. Warme PK, Morgan RS. A survey of amino acid side-chain interactions in 21 proteins. *J Mol Biol*. 1978;118(3):289-304. https://doi.org/10.1016/0022-2836(78)90229-2.
45. Wolfenden R, Andersson L, Cullis PM, Southgate CCB. Affinities of amino acid side chains for solvent water. *Biochemistry*. 1981;20(4):849-855. https://doi.org/10.1021/bi00507a030.
46. Kjær J, Høj L, Fox Z, Lundgren JD. Prediction of phenotypic susceptibility to antiretroviral drugs using physiochemical properties of the primary enzymatic structure combined with artificial neural networks. *HIV Med*. 2008;9(8):642-652. https://doi.org/10.1111/j.1468-1293.2008.00612.x.
47. Zimmerman JM, Eliezer N, Simha R. The characterization of amino acid sequences in proteins by statistical methods. *J Theor Biol*. 1968;21(2):170-201. https://doi.org/10.1016/0022-5193(68)90069-6.
48. Grantham R. Amino Acid difference formula to help explain protein evolution. *Science*. 1974;185(4154):862-864. https://doi.org/10.1126/science.185.4154.862.
49. Takano K, Yutani K. A new scale for side-chain contribution to protein stability based on the empirical stability analysis of mutant proteins. *Protein Eng Des Sel*. 2001;14(8):525-528. https://doi.org/10.1093/protein/14.8.525.
50. Meirovitch H, Rackovsky S, Scheraga HA. Empirical studies of hydrophobicity. 1. Effect of Protein size on the hydrophobic behavior of amino acids. *Macromolecules*. 1980;13(6):1398-1405. https://doi.org/10.1021/ma60078a013.
51. Stekol JA. *Amino Acids and Serum Proteins*. Washington: AMERICAN CHEMICAL SOCIETY; 1964. https://doi.org/10.1021/ba-1964-0044
52. Acid L, Citrulline D, Hci D. Heat capacities absolute entropies and entropies of formation of amino acids and related compounds. *Handbook of Biochemistry and Molecular Biology*. Boca Raton, FL: CRC Press; 1984.
53. Fauchère J-L, Charton M, Kier LB, Verloop A, Pliska V. Amino acid side chain parameters for correlation studies in biology and pharmacology. *Int J Pept Protein Res*. 1988;32(4):269-278. https://doi.org/10.1111/j.1399-3011.1988.tb01261.x.
54. Fasman GD. *Handbook of Biochemistry: Section D Physical Chemical Data*. Boca Raton, FL: CRC Press; 2018.
55. Muralikrishnan M, Anitha R. In: Hemanth DJ, Kumar VDA, Malathi S, Castillo O, Patrut B, eds. Comparison of breast cancer multi-class classification accuracy based on inception and InceptionResNet architecture. Paper presented at:*Emerging Trends in Computing and Expert Technology*. Lecture Notes on Data Engineering and Communications Technologies. Springer International Publishing; 2020:1155–1162. doi:https://doi.org/10.1007/978-3-030-32150-5_118
56. Walker EY, Sinz FH, Cobos E, et al. Inception loops discover what excites neurons most using deep predictive models. *Nat Neurosci*. 2019;4:1-6. https://doi.org/10.1038/s41593-019-0517-x.
57. Wang J, Wang J, Wang J, et al. Deep learning for quality assessment of retinal OCT images. *Biomed Opt Express*. 2019;10(12):6057-6072. https://doi.org/10.1364/BOE.10.006057.
58. Keras deep learning on graphs. Published 2020. https://vermamachinelearning.github.io/keras-deep-graph-learning/
59. Keras: the python deep learning library. Published 2019. https://keras.io/
60. Snoek J, Larochelle H, Adams RP. Practical Bayesian optimization of machine learning algorithms. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, eds. *Advances in Neural Information Processing Systems 25*. New York, NY: Curran Associates Inc.; 2012:2951-2959. Accessed December 11, 2019. http://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf.
61. Skopt module. Published 2019. https://scikit-optimize.github.io/
62. Zemla A, Venclovas Č, Fidelis K, Rost B. A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment. *Proteins Struct Funct Bioinform*. 1999;34(2):220-223.
63. Matthews BW. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim Biophys Acta BBA Protein Struct*. 1975;405(2):442-451. https://doi.org/10.1016/0005-2795(75)90109-9.
64. Z score calculator for 2 population proportions. Published October 2, 2018 https://www.socscistatistics.com/tests/ztest/Default2.aspx

How to cite this article: Görmez Y, Sabzekar M, Aydın Z. IGPRED: Combination of convolutional neural and graph convolutional networks for protein secondary structure prediction. *Proteins*. 2021;89(10):1277–1288. https://doi.org/10.1002/prot.26149