CLASSIFICATION OF SIGNALING PROTEINS USING COMPUTER-BASED METHODS

Çağdaş KÜÇÜK

Dokuz Eylül University, Graduate School of Natural and Applied Sciences, Department of Computer Science, 35160 Tinaztepe Campus, Izmir, TURKEY

Çağın KANDEMİR ÇAVAŞ*

[2]Dokuz Eylül University, Faculty of Sciences, Department of Computer Science, 35160 Tinaztepe Campus, Izmir, TURKEY

*Corresponding Author: Prof. Dr. Cagin Kandemir-Cavas

Abstract

Cells send signals between them to start or end biological activities. These signals can be molecules like proteins, hormones. Receptors outside and inside of the cell capture these molecules and help the signal transduction process. There are 3 types of receptors that are on the surface of the cell: G-protein linked receptors, enzymic receptors and chemically gated ion-channels.

The function of a protein depends on its structure. Signaling proteins have a crucial role in many biological activities such as functioning of the brain, tongue …etc. Signaling proteins are important for drug discovery. For these reasons it is important to define whether a protein is a signaling or not. Since experimental studies are very expensive and time consuming, machine learning techniques are used for this purpose.

Aside from the machine learning techniques that are used the protein encoding scheme is also important for getting efficient results. Proteins are made of amino acids. Amino acids in a protein written next to each other is called an amino acid sequence. To use sequences in machine learning there are protein encoding schemes. Amino acid composition is one of them. Amino acid composition encodes amino acid frequencies as percentages but the sequence information is gone in the process. A solution for this was proposed by Kuo-Chen Chou in 2001 called pseudo amino acid composition. The aim of this study is to classify signaling proteins encoded by pseudo amino acid composition.

Protein sequences in Fasta format were downloaded and encoded using pseudo amino acid composition to create a dataset. Training and test sets were separated by 75% and 25% respectively. Dataset had 1867 signaling and 3317 non-signaling proteins. Random forest, support vector machine and deep neural network models were applied to classify the signaling proteins.

Random forest, support vector machine and deep neural network give the accuracy values as 0.749, 0.748 and 0.763, respectively. The AUROC values were similar to each other but Random Forest algorithm has the highest value of 0.745.

The accuracy rate of random forest, support vector machine and artificial neural network algorithms is higher than 0.70, which shows that these models are effective in the classification of signal proteins encoded by pseudo amino acid composition.

Keywords: PseAAC, support vector machine, neural network, random forest, signaling proteins

1. Introduction

Cells can send and receive signals to start or end biological processes. There are four types of cell signaling: through direct contact, paracrine (short distance), endocrine (long distance) and synaptic (neurons). A cell can also send a signal to itself and it is called autocrine signaling [1].

Cells have receptors that capture the signal and start the signal process. Receptors can be inside of the cell (intracellular) or on the surface on the cell. Intracellular receptors are for smaller molecules that can pass the membrane of the cell. Receptors that are on the surface on the cell are for larger molecules like hormones that are too big to pass through the membrane. There are 3 types of receptors that are on the surface of the cell: G-protein linked receptors, enzymic receptors and chemically gated ion-channels. These receptors capture the outer signal and their parts in the cell convert it to an internal one. Some receptors use smaller molecules called second messenger to send the signal inside the cell. Cyclic adenosine monophosphate (cAMP) and calcium are examples of second messengers [1].

Signaling proteins have a role in protein transportation to cell membranes, protein secretion and several biochemical activities [2-3]. The identification of signal sequence is very crucial. Because of its importance for drug discovery [4]. Therefore, it would be helpful to create methods by computational techniques. At this point of view, machine learning techniques are the one of the best solutions. The basic aim in machine learning is to obtain low-cost solutions by exploiting the tolerance of imprecision, approximate reasoning and partial truth [5]. So there are many prediction methods used for protein structure classification in the literature [6-11].

Besides the algorithm used, protein encoding scheme is also very important in terms of obtaining satisfactory prediction accuracy rate. Amino acid composition (AAC) is the basic method to identify a protein sequence in which each kind of amino acid is represented by the percentage [12]. Several studies exist in the literature that used AAC to encode the proteins [13-16] In order to increase the information beyond amino acid composition, the concept of pseudo-amino acid composition (PseAAC) was proposed by Chou [17]. There are also many studies that utilize PseAAC exists in the literature [17-24]. The purpose of this study is to classify signal proteins encoded by PseAAC.

2. Materials and Methods

2.1. PseAAC

Normal amino acid composition has only the frequencies of amino acids as attributes but not the sequence in which they are connected. PseAAC adds sequence information of amino acids aside from their frequencies [17]. They are calculated with the Eq. 1 where λ is the value that

decides how many extra attributes are added and w is the weight factor which decides how much it is going to affect the x_u.

$$x_u = \begin{cases} \frac{f_u}{\sum_{i=1}^{20} f_i + w\sum_{j=1}^{\lambda}\theta_j}, (1 \le u \le 20) \\ \frac{w\theta_{u-20}}{\sum_{i=1}^{20} f_i + w\sum_{j=1}^{\lambda}\theta_j}, (20+1 \le u \le 20+\lambda) \end{cases} \quad (1)$$

The i-th tier correlation factor θ_i has the sequence order correlations of all the amino acids in the protein sequence and is calculated in Eq. 2 where L is the number of amino acids in the protein chain and R_i is the i-th amino acid in a sequence like $R_1, R_2, \dots, R_L$.

$$\theta_\lambda = \frac{1}{L-\lambda}\sum_{i=1}^{L-\lambda}\Theta(R_i, R_{i+\lambda})\ (\lambda < L) \quad (2)$$

$\Theta(R_i, R_j)$ is the correlation function which is calculated using the standardized hydrophobicity (H_1), hydrophilicity (H_2) and side-mass chain (M) values in Eq. 3.

$$\Theta(R_i, R_j) = \frac{1}{3}\{[H_1(R_j) - H_1(R_i)]^2 + [H_2(R_j) - H_2(R_i)]^2 + [M(R_j) - M(R_i)]^2\} (3)$$

Names, 1-letter symbols, hydrophobicity (H_1), hydrophilicity (H_2) and side-mass chain (M) values for the amino acids are given in Table 1 [25-27].

Table 1. Hydrophobicity, hydrophilicity and side-mass chain values of amino acids

Amino Acid Names	1 Letter Symbols	Hydrophobicity (H_1)	Hydrophilicity (H_2)	Side-Chain Mass (M)
Alanine	A	0.62	-0.5	15.0
Cysteine	C	0.29	-1.0	47.0
Aspartic acid	D	-0.90	3.0	59.0
Glutamic acid	E	-0.74	3.0	73.0
Phenylalanine	F	1.19	-2.5	91.0
Glycine	G	0.48	0.0	1.0
Histidine	H	-0.40	-0.5	82.0
Isoleucine	I	1.38	-1.8	57.0
Lysine	K	-1.50	3.0	73.0

Leucine	L	1.06	-1.8	57.0
Methionine	M	0.64	-1.3	75.0
Asparagine	N	-0.78	0.2	58.0
Proline	P	0.12	0.0	42.0
Glutamine	Q	-0.85	0.2	72.0
Arginine	R	-2.53	3.0	101.0
Serine	S	-0.18	0.3	31.0
Threonine	T	-0.05	-0.4	45.0
Valine	V	1.08	-1.5	43.0
Tryptophan	W	0.81	-3.4	130.0
Tyrosine	Y	0.26	-2.3	107.0

H_1, H_2 and M values need to be standardized using the formulas in Eq. 4 where H_1^0, H_2^0 and M^0 are the original values before they can be used in the PseAAC.

$$\begin{cases} H_1(i) = \dfrac{H_1^0(i) - \sum_{i=1}^{20} \frac{H_1^0(i)}{20}}{\sqrt{\frac{\sum_{i=1}^{20}\left[H_1^0(i) - \sum_{i=1}^{20} \frac{H_1^0(i)}{20}\right]^2}{20}}} \\ H_2(i) = \dfrac{H_2^0(i) - \sum_{i=1}^{20} \frac{H_2^0(i)}{20}}{\sqrt{\frac{\sum_{i=1}^{20}\left[H_2^0(i) - \sum_{i=1}^{20} \frac{H_2^0(i)}{20}\right]^2}{20}}} \\ M(i) = \dfrac{M^0(i) - \sum_{i=1}^{20} \frac{M^0(i)}{20}}{\sqrt{\frac{\sum_{i=1}^{20}\left[M^0(i) - \sum_{i=1}^{20} \frac{M^0(i)}{20}\right]^2}{20}}} \end{cases} \quad (4)$$

2.2. Dataset

A FASTA format contains amino acid sequences of proteins. Each amino acid is denoted by their 1-letter symbol. First line starts with a ">" symbol and the descriptions of the proteins. After that comes the sequence of amino acids. A FASTA sequence example is in Figure 1.

```
>1FOE:B|PDBID|CHAIN|SEQUENCE
MQAIKCVVVGDGAVGKTCLLISYTTNAFPGEYIPTV
FDNYSANVMVDGKPVNLGLWDTAGQEDYDRLRP
LSYPQTDVFLICFSLVSPASFENVRAKWYPEVRHHC
PNTPIILVGTKLDLRDDKDTIEKLKEKKLTPITYPQGL
AMAKEIGAVKYLECSAL
```

Figure 1. A FASTA format of a sequence example

The process of gathering signaling and non-signaling proteins are the similar to the work of Fernandez-Lozano et al. [4]. For signaling, 1867 proteins were downloaded from Protein Databank(PDB) by choosing "Advanced Search", then "Biological Process Browser" and inputting "signaling (GO ID:23052)" as a criteria and after that, by selecting "Protein" and "Representative Structures 30%" [28]. For non-signaling, 3404 proteins were downloaded from PISCES CulledPDB by choosing "percent identity cutoff" less than 20%, resolution 1.6 Â and R-factor 0.25 [29]. Non-signaling protein set was checked to remove any signaling protein and the resulting dataset had 1867 signaling and 3317 non-signaling proteins. Then they were put together and PseAAC was used with $\lambda = 3$ and $w = 0.05$ values to turn the data into a 23 attribute and 1 class attribute dataset. First 20 attributes are 20 amino acids' frequencies, the other 3 are about the sequence order and the last attribute showing whether that protein is a signaling protein or not. A scheme of our process is in Figure 2.

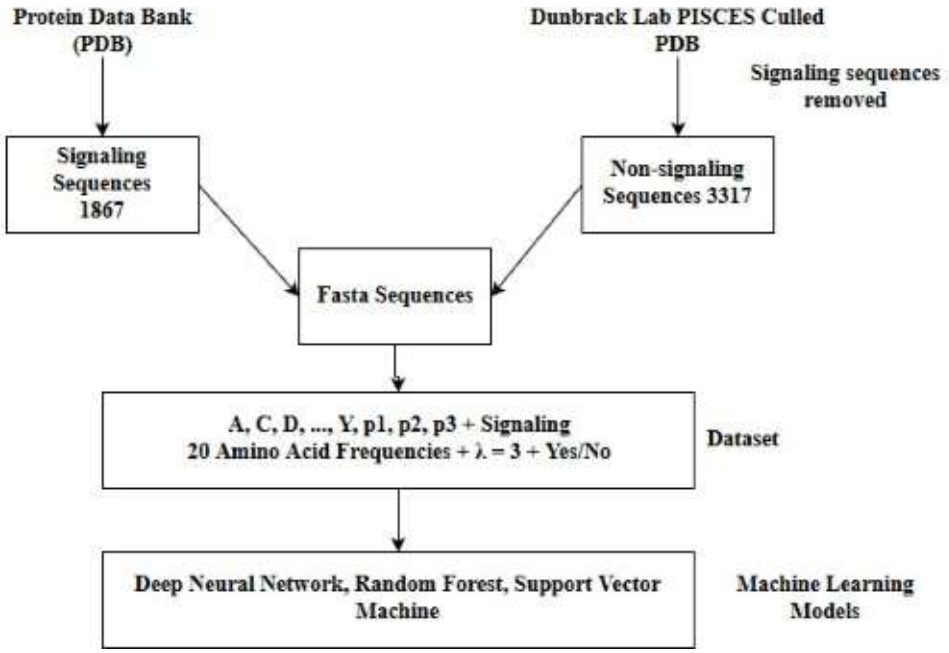

Figure 2. Scheme of dataset creation process

2.3. Deep Neural Network

A Deep Neural Network (DNN) is made of input, output and hidden layers. There are nodes in each layer and every node is connected to the nodes in the previous and next layers [30]. The connections have weights so only a part of the input passes through them. In each node except the input nodes are activation functions like sigmoid, tanh, ReLU. Every node takes the inputs multiplied with the weights and adds them together, then putting them through the activation function and passes the result through to the next layer. The network calculates the output by taking the inputs and moving them through the whole network, passing through nodes in each layer and going out at the output layer giving a result. From that result the error is calculated with the help of a loss function and then the weights are adjusted using backpropagation [31]. This process is called an iteration. The network trains by doing this iteration on the whole dataset and adjusting the weights in order to minimize the loss. Doing the process on the whole dataset once is called epoch and it is repeated until a stopping criteria is reached. A scheme of a neural network can be seen in Figure 3.

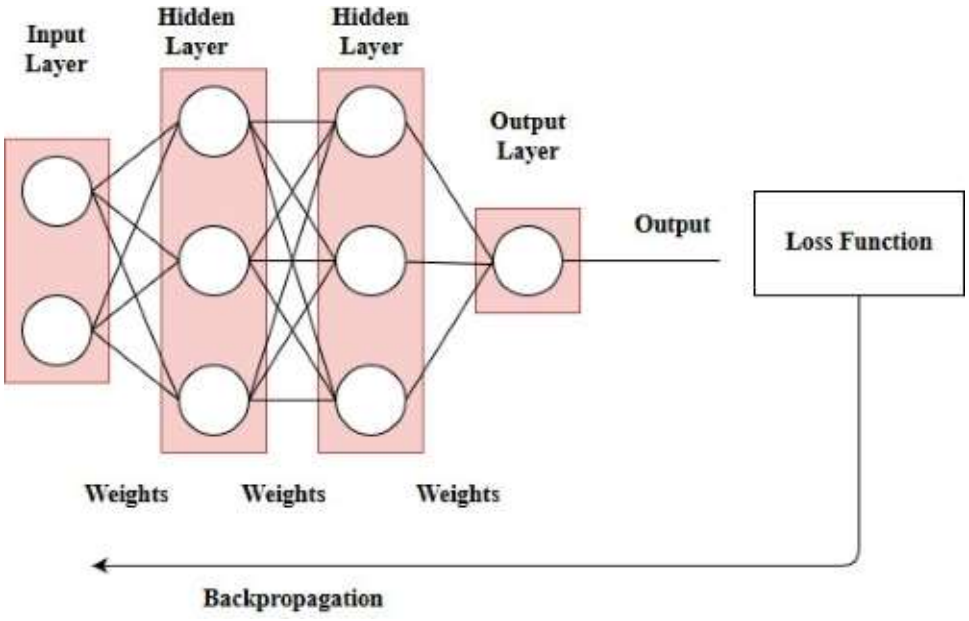

Figure 3. A deep neural network model architecture.

2.4. Random Forest

Random Forest (RF) algorithm is an ensemble of decision trees [32]. The algorithm creates samples with replacement for each tree [33]. A tree is created by selecting the best attribute from a randomly selected collection of attributes for each split. After the trees are formed, the prediction occurs by each tree giving their own result which are treated as votes and they are added together to get the actual result. The class with the highest votes is chosen as predicted.

2.5. Support Vector Machine

Support Vector Machine (SVM) algorithm takes data points from each class that are closer to each other as support vectors and tries to find the optimal hyperplane that separates each class. The length between the closest point in each class to the hyperplane is called margin and the algorithm tries to maximize it [34].

Some data points might be on the wrong side of the hyperplane. To choose how much of that error is acceptable, there is a hyper-parameter c. When c is higher the acceptance of error is lower.

Not all data are linearly separable and to find a hyperplane that separates each class we need to use something called a "kernel trick". By using a kernel function we can change the dimension of the data so that the data can be linearly separable. Polynomial function, sigmoid function, radial basis function are some examples of kernel functions.

2.6. Application

Dataset was split into two sets, training set and test set. Training set contained the 75% of the proteins while the test set contained the remaining 25%. Initial attempts at training a classifier showed that the imbalance between classes made the classifier lean on the majority class. To solve this problem a technique called Synthetic Minority Over-Sampling Technique (SMOTE) was used [35]. Training set which was originally made of 1400 signaling and 2487 non-signaling samples with the help of SMOTE turned into 2800 signaling 3500 non-signaling samples. Test set contains 467 signaling and 830 non-signaling samples.

RF has number of trees (ntree) and number of attributes selected randomly at each split (mtry) hyper-parameters while the SVM has γ and c that needed tuning to achieve a better result. After tuning it was found that mtry = 5, ntree = 500, $\gamma = 0.04347826$ and c = 1 were the best resulting ones. Kernel for SVM was radial basis function.

DNN model for this work has 5 layers with each layer having 256 nodes and 0.5 dropout optimization is used after each layer. For hidden layers, ReLU and for output layer, sigmoid activation functions were used. Binary cross entropy was chosen for the loss function. For learning, Adam optimizer was used with the default parameters. Batch size was set to 512 and epoch was set to 100.

3. Results and Discussion

Different hyper-parameters and settings were tried for each model and the ones that gave the best results are selected as stated in the previous section. The resulting performance metrics are in Table 2.

Table 2. Results, matthews correlation coefficient (MCC), area under the receiver operating characteristic curve (AUROC)

Models	Accuracy	Precision	Recall	MCC	Specificity	F1-Score	AUROC
Support Vector Machine	0.749	0.636	0.713	0.472	0.770	0.672	0.742
Random Forest	0.748	0.629	0.732	0.476	0.757	0.677	0.745
Deep Neural Network	0.763	0.673	0.664	0.483	0.818	0.668	0.741

Accuracy, precision, Matthews Correlation Coefficient (MCC) and specificity values of RF and SVM are very close yet DNN has higher values. Yet on recall and f-1 score DNN has the lowest values. Since RF has the higher Area under the receiver operating characteristic curve (AUROC) value, it is concluded that RF gives the best classification performance among the models.

Detection of signaling proteins are important. PseAAC is used in many proteins related problems and given good results. For that reason, this study was conducted to see its effect on classification of signaling proteins.

The results of the models used in this study are very similar. Even though the datasets are not the same because of the dates that they were created, resulting in two datasets with different amount of data, still the performance of the two approaches may be compared with their AUROC values. Although the performances of the models are decent, the best AUROC value of 0.745 for RF. Therefore, the best classification performance can be obtained from RF for proteins encoded by PseAAC.

4. Conclusion

A total of 5184 proteins were used to create a dataset and 1297 of them were used to create the test set. RF, SVM and DNN classification models were chosen and trained for detecting signaling proteins using PseAAC encoding. The results were similar for each model. In order to classify signaling proteins, RF, SVM and DNN can be referred as satisfying and robust algorithms.

References

1. Johnson GB, Raven PH. Cell-Cell Interactions. Biology. 6th ed. McGraw-Hill, 2002:123-140.

2. Nakai K. Protein sorting signals and prediction of subcellular localization. Advances in protein chemistry 2000;54:277–344.

3. Parker J. Encyclopedia of Genetics. USA: Academic Press, 2001.

4. Fernandez-Lozano C, Cuiñas RF, Seoane JA, Fernandez-Blanco E, Dorado J, Munteanu CR. Classification of signaling proteins based on molecular star graph descriptors using Machine Learning models. Journal of theoretical biology 2015;384:50-58.

5. Mitra S, Hayashi Y. Bioinformatics with soft computing. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 2006;36(5):616-635.

6. Claros M, Brunak S, von Heijne G. Prediction of N-terminal protein sorting signals. Current opinion in structural biology 1997;7(3):394–398.

7. Jain P, Garibaldi JM, Hirst JD. Supervised machine learning algorithms for protein structure classification. Computational biology and chemistry 2009;33(3):216-223.

8. Najibi SM, Maadooliat M, Zhou L, Huang JZ, Gao X. Protein structure classification and loop modeling using multiple Ramachandran distributions. Computational and structural biotechnology journal 2017;15:243-254.

9. Shu JJ, Yong KY. Fourier-based classification of protein secondary structures. Biochemical and biophysical research communications 2017;485(4):731-735.

10. Kathuria C, Mehrotra D, Misra NK. Predicting the protein structure using random forest approach. Procedia computer science 2018;132:1654-1662.

11. Moreira CA, Philot EA, Lima AN, Scott AL. Predicting regions prone to protein aggregation based on SVM algorithm. Applied Mathematics and Computation 2019;359:502-511.

12. Hormoz S. Amino acid composition of proteins reduces deleterious impact of mutations. Scientific reports 2013;3:2919.

13. Dumontier M, Yao R, Feldman HJ, Hogue CWV. Armadillo: domain boundary prediction by amino acid composition. Journal of molecular biology 2005;350(5), 1061-1073.

14. Nasibov E, Kandemir-Cavas C. Protein subcellular location prediction using optimally weighted fuzzy k-NN algorithm. Computational biology and chemistry 2008;32(6):448-451.

15. Nasibov E, Kandemir-Cavas C. Efficiency analysis of KNN and minimum distance-based classifiers in enzyme family prediction. Computational biology and chemistry 2009;33(6):461-464.

16. Shatnawi M, Zaki N. Inter-domain linker prediction using amino acid compositional index. Computational biology and chemistry 2015;55:23-30.

17. Chou KC. Prediction of protein cellular attributes using pseudo-amino acid composition. Proteins: Structure, Function, and Bioinformatics 2001;43(3):246-255.

18. Chen C, Zhou X, Tian Y, Zou X, Cai P. Predicting protein structural class with pseudo-amino acid composition and support vector machine fusion network. Analytical biochemistry 2006;357(1):116-121.

19. Shen HB, Yang J, Chou, KC. Fuzzy KNN for predicting membrane protein types from pseudo-amino acid composition. Journal of theoretical biology 2006;240:9-13.

20. Chen YL,Li QZ. Prediction of apoptosis protein subcellular location using improved hybrid approach and pseudo-amino acid composition. Journal of Theoretical Biology 2007;248(2):377-381.

21. Krajewski Z, Tkacz E. Protein structural classification based on pseudo amino acid composition using SVM classifier. Biocybernetics and Biomedical Engineering 2013;33(2):77-87.

22. Mondal S, Pai PP. Chou′ s pseudo amino acid composition improves sequence-based antifreeze protein prediction. Journal of theoretical biology 2014;356:30-35.

23. Kumar R, Srivastava A, Kumari B, Kumar M. Prediction of β-lactamase and its class by Chou's pseudo-amino acid composition and support vector machine. Journal of Theoretical Biology 2015;365:96-103.

24. Qiu W, Li S, Cui X, Yu Z, Wang M, Du J, Peng Y, Yu, B. Predicting protein submitochondrial locations by incorporating the pseudo-position specific scoring matrix into the general Chou's pseudo-amino acid composition. Journal of theoretical biology 2018;450:86-103.

25. Hopp TP, Woods KR. Prediction of protein antigenic determinants from amino acid sequences. Proceedings of the National Academy of Sciences 1981;78(6):3824-3828.

26. Eisenberg D. THREE-DIMENSIONAL STRUCTURE OF MEMBRANE AND SURFACE PROTEINS. Annual Review of Biochemistry 1984;53(1):595-623.

27. Lide DR. Biochemistry. CRC handbook of chemistry and physics. 85th ed. Boca Raton: CRC press, 2005.

28. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, et al. The protein data bank. Nucleic acids research 2000;28(1):235-242.

29. Wang G, Dunbrack Jr RL. PISCES: a protein sequence culling server. Bioinformatics 2003;19(12):1589-1591.

30. Goodfellow I, Bengio Y, Courville A. Deep Feedforward Networks. Deep Learning. 1st ed. The MIT Press, 2016:163-217.

31. Rumelhart DE, Hinton GE; Williams RJ. Learning representations by back-propagating errors. Nature 1986;323(6088):533-536.

32. Breiman L. Random forests. Machine learning 2001;45(1):5-32.

33. Hastie T, Tibshirani R, Friedman J. Random Forests. The Elements of Statistical Learning. 2nd ed. New York: Springer, 2009:587-604.

34. Aggarwal CC. Support Vector Machines. Data Classification: Algorithms and Applications. 1st ed. Chapman & Hall/CRC, 2014:187-202.

35. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research 2002;16:321-357.