A Mixed Convolution Neural Network for Identifying RNA Pseudouridine sites

Abu Zahid Bin Aziz
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology
Rajshahi, Bangladesh
Email: abuzahid.cse@gmail.com

Md. Al Mehedi Hasan
Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology
Rajshahi, Bangladesh
Email: mehedi_ru@yahoo.com

***Abstract*—Pseudouridine is widely popular among various RNA modifications which has been confirmed to occur in rRNA, mRNA, tRNA, and nuclear/nucleolar RNA. Hence, identifying them has vital significance in academic research, drug development and gene therapies. Several laboratory techniques for Ψ identification have been introduced over the years. Although these techniques produce satisfactory results, they are costly, time consuming and requires skilled experience. As the lengths of RNA sequences are getting longer day by day, an efficient method for identifying pseudouridine sites using computational approach is very important. In this paper, we proposed a mixed convolution neural network using "one-hot" encoding. We employed k-fold cross-validation and grid search to tune the hyperparameters. Our model took care of the feature selection and extraction process automatically. We evaluated its performance in the independent datasets and found promising results. The results proved that our method can be used to identify pseudouridine sites for associated purposes. Our work also projects the increased performance of applying CNN for biological sequences.**

***Index Terms*—RNA, pseudouridine site, CNN, binary classification, k-fold cross-validation**

I. Introduction

Pseudouridine (Ψ) is the most common RNA modification observed in both prokaryotes and eukaryotes [1]. It is formed by the Ψ synthase enzyme which leads to the proof of its occurrence in various kinds of RNAs [2]. This enzyme separates the uridine residue's base from its sugar and rotates it 180° along the N3-C6 axis. The separation is completed by the subsequent reattachment of the base's 5-carbon to the 1'-carbon of the sugar which results in the formation of an isomer of uridine, Pseudouridine [3]. Psudouridines play a vital role in both biological and genetic aspects of RNAs, especially for tRNA and rRNA. In case of rRNA, RNP proteins are proved to be needed for pseudouridylation [4]. They also work as a powerful mechanism for stabilizing tRNAs in both single and double-stranded regions [5]. Besides, different species present different prospects due to pseudouridines such as U6 snRNA mutants pseudouridylate at Ψ28 contributing to the filamentation growth program [6]. Furthermore, mRNAs incorporated with Ψ increase translation efficiency and restrict innate immune response [7].

Some laboratory techniques have been introduced over the years producing promising results. Carlile et al. introduced a transcriptome-wide pseudouridine-seq approach where Lovejoy et al. used induced termination of reverse transcription in their work [8], [9]. Furthermore, Schwartz et al. developed a transcriptome-wide quantitative mapping system to identify pseudouridine [10]. All of these systems are not only expensive but also time consuming. Moreover, skilled and experienced people are required to maintain these systems. That is why a more user-friendly method is required for identifying pseudouridine sites.

Despite the necessity, there are not many computational methods to identify Ψ sites from RNA sequences. Li et al. introduced an SVM based web server which is, to the best of our knowledge, the first computational method to identify pseudouridine synthase (PUS) specific Ψ sites [11]. They used nucleotides around the Ψ sites as features in their work which provided good results for human and yeast samples. Later, their performance was improved by taking account of the chemical properties and the occurrence frequency density distributions of nucleotides by iRNA-Pseu, proposed by Chen et al. Their work also covered another species (*M. musculus*) [12]. He et al. proposed another web server named PseUI by using SVM [13]. First, they generated five different types of features and selected one by using the sequential forward feature selection approach. Recently, Tahir et al. implemented both machine learning and deep learning methods in their work [14]. They extracted features using n-gram and MMI in their SVM classifier and adopted a convolutional neural network (CNN) in their deep learning method, where the CNN classifier produced better performance.

Many of the recent works used PseKNC for feature extraction [15]–[17]. That is why we wanted to adopt a CNN model which extracts and selects features automatically. CNN has already proven to be useful in computer vision problems. So we wanted to see whether CNN can be used for sequence inputs. In this work, we employed a CNN model where multiple convolution layers with different sized filters are added separately. Each of these convolution layers is then added to a max-pooling layer and concatenated. Our model yielded satisfactory results in the training and test datasets.

978-1-7281-7366-5/20/$31.00 ©2020 IEEE

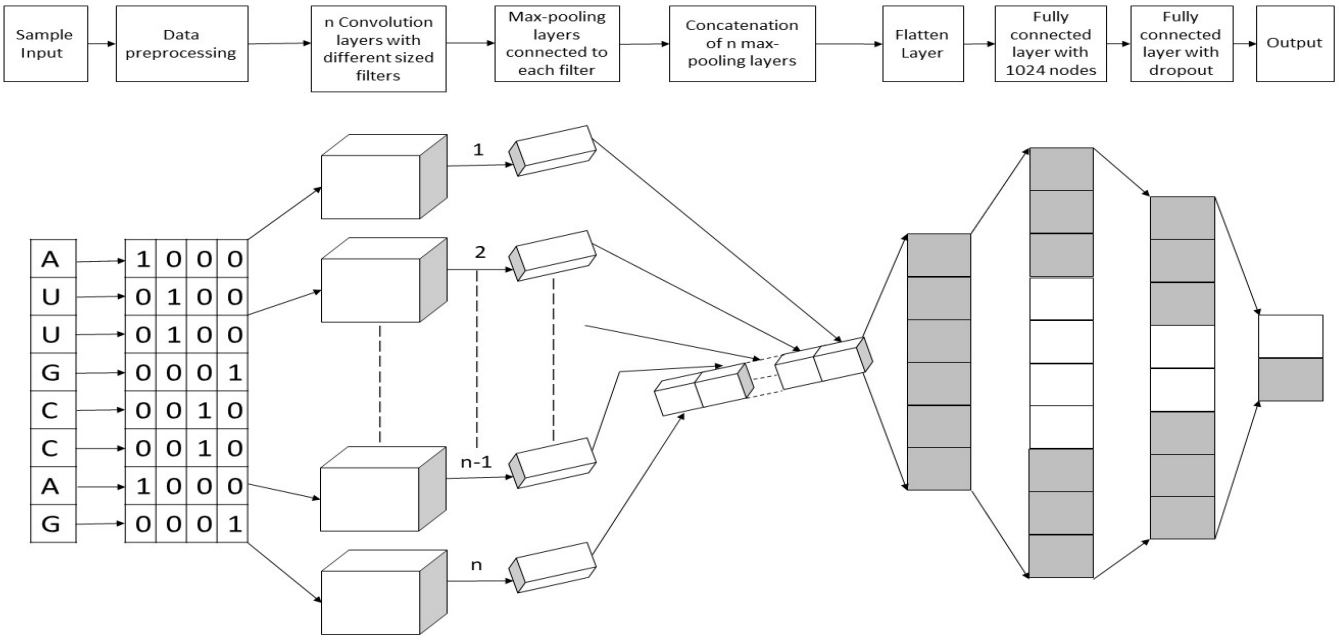

Fig. 1. The architecture of our mixed CNN model.

II. Materials and Methods

A. Training and test datasets

In this work, data were collected for three different species which are *H. sapience*, *S. cerevis*iae and *M. musculus* represented by H, S and M respectively. There were three benchmark datasets, H_990, S_628, and M_944, one for each species for training purposes. Each of these datasets had an equal number of positive and negative samples. These are the same datasets used in Chen et al's work where they downloaded the RNA sequences from RMBase [12], [18]. In addition to these training datasets Chen et al. also provided two independent datasets, H_200 and S_200 for testing purposes which were for *H. sapience* and *S. cerevisiae* but not for *M. musculus*. In both H_200 and S_200, the number of positive and negative samples was equal. In the datasets, RNA sequences were formulated as shown:

$$R_\xi(U) = N_{-\xi}N_{-(\xi-1)}\cdots N_{-1}UN_1\cdots N_{+(\xi-1)}N_\xi \quad (1)$$

Here, U indicates "uridine", $N_{-\xi}$ denotes the ξ-th nucleotide towards the 5' end and $N_{+\xi}$ denotes the ξ-th downstream nucleotide towards the 3' end from the central uridine. The value of ξ in H_990 and M_944 was 10 and 15 in S_628.

B. Data preprocessing

Before applying the RNA sequences to our model, we needed to preprocess it first. There was only one step involved in the preprocessing step, which was binary "one-hot" encoding to convert our inputs into a 2 -dimensional vector. Each of the nucleotides of an input sequence was represented as a row vector where all the values are zero except for one value. The length of these row vectors was four which is the number of nucleotides found in RNA. Therefore, a sequence with N nucleotides would be a (N x 4) vector. The 1D vectors we chose for the nucleotides were: ("A"=[1,0,0,0], "U"=[0,1,0,0], "C"=[0,0,1,0], "G"=[0,0,0,1]). In our *H. Sapience* and *M. musculus* datasets, the length of the sequences was 31 which would be preprocessed into (31 x 4) vector. And the length was 21 for the *S. cerevisiae* dataset which would be converted into (21 x 4) vector.

C. CNN architecture

After preprocessing ("one-hot" encoding), the converted 2D vectors were fed to a convolutional neural network. Generally, in a CNN model, the inputs are connected to some convolution and max-pooling layers, followed by a couple of fully connected layers that are connected to the output layer. But we made some changes in the general structure, hence, the term mixed CNN is used.

A basic structure of mixed model is shown in Figure 1. In our model, the inputs were connected to multiple convolution layers separately, each having different sized filters. The width of the filters was the same where the height varied from 3 to 9. Each of these convolution layers was then connected to a max-pooling layer. Then, the max-pooling layers were concatenated together to combine the features extracted by the convolution and max-pooling layers. Next, the max-pooling layers are connected to the first fully connected layer which

had 320 nodes. After that, we employed dropout regularization to reduce the number of parameters. Then, the final layer was connected which gave a probability distribution of the classes. From the probability distribution, the final output was predicted.

The number of convolution layers was selected by applying k-fold cross-validation and grid search. Cross-validation also helped us to select the learning rate,dropout probability and height of the filters. Relu activation function was employed in every layer except for the last layer where the softmax activation function is used.

D. Method evaluation

Four evaluation metrics have been frequently used to evaluate the quality of a method in recent studies [19], [20]. To calculate them, we required four parameters: true positive (tp), true negative (tn), false positive (ft) and false negative (fn). The equations for the evaluation metrics are given below:

- Sensitivity (Recall):

$$\text{Sensitivity} = \text{tp}/(\text{tp} + \text{fn}) \tag{2}$$

- Specificity:

$$\text{Specificity} = \text{tn}/(\text{tn} + \text{fp}) \tag{3}$$

- Mathews Correlation Coefficient (MCC):

$$\text{MCC} = (\text{tp} * \text{tn} - \text{fp} * \text{fn})/[(\text{tp} + \text{fp}) * (\text{tp} + \text{fn}) * (\text{tn} + \text{fp}) * (\text{tn} + \text{fn})]^{1/2} \tag{4}$$

- Accuracy:

$$\text{Accuracy} = (\text{tp} + \text{tn})/(\text{tp} + \text{tn} + \text{fp} + \text{fn}) \tag{5}$$

III. Results and discussions

A. Hyperparameter tuning

Hyperparameter tuning is vital to maximize a model's predictive performance. On the training dataset, we tuned a number of hyperparameters to fine-tune our model. We did it in two separate steps using k-fold (k=10) cross-validation and grid search. First, we tuned the number of epochs and batch size. Then, we tuned the number of convolution layers, height of filters, learning rate and dropout probability using the values from the first step. The number of convolution layers was tuned to investigate how many of them can be separately connected to the input layer to produce the maximum accuracy. Grid search was adopted to select the values that produced the best result. The considered values for the hyperparameters are given in the second column in Table 1. We calculated accuracies for every possible combination of values of these hyperparameters and selected the ones that provided the highest accuracy. The selected values are given in the third and fourth columns in Table 1. As the shape of the inputs were different in the datasets, the selected values were not the same. They were used to train our model in the training dataset and were evaluated by the test data.

TABLE I
SELECTED VALUES OF THE HYPERPARAMETERS FOR THE TRAINING DATASETS.

Hyperparameter	Ranges	Selected value	
		H_990, M_944	S_628
Batch size	[10,20,30,40]	10	10
No. of epochs	[10,50,100,200]	100	100
No. of conv. layers	[5,7,9,10,11]	11	9
Filter height	[3,5,7,9]	9	7
Learning rate	[0.001,0.0005,0.0001]	0.0005	0.0001
Dropout probability	[0.4,0.5,0.6]	0.6	0.4

TABLE II
COMPARISON OF THE EVALUATION METRICS IN THE TESTING DATASET BETWEEN OUR MODEL AND EXISTING METHODS

Testing Dataset	Models	AC(%)	SN(%)	SP(%)	MCC
H_200	Ours	**72.5**	**80.0**	**65.0**	**0.44**
	iPseU-CNN	69.0	77.72	60.81	0.40
	PseUI	65.50	63.00	68.00	0.31
	iRNA-PseU	61.50	58.00	65.00	0.23
S_200	Ours	**75.0**	67.0	**83.0**	**0.50**
	iPseU-CNN	73.50	68.76	77.82	0.47
	PseUI	68.50	65.00	72.00	0.37
	iRNA-PseU	60.00	63.00	57.00	0.20

**AC= Accuracy, SN= Sensitivity, SP= Specificity

B. Training and testing

Since the performance of CNN in computer vision and NLP tasks is well established, we wanted to use its classification success for biological sequence inputs. In case of feature extraction, the convolution and max-pooling layers did that automatically. To avoid overfitting, we employed dropout regularization in the first fully-connected layer. After tuning the hyperparameters, we used the selected values to train our model in the training dataset. The validation and training process were done in a core i5 laptop having NVIDIA 940m as GPU. Because of the grid search, the validation process took almost an hour to complete and the training process took about 2-3 minutes.

C. Analysis

Compared to the most recent existing classifier, our model had increased performance in sensitivity, specificity, accuracy and MCC. More specifically, the H_200 dataset had improved performance in sensitivity by 2.93%, specificity by 6.89%, accuracy by 5% and MCC by 10%. And in the S_200 dataset specificity, accuracy, MCC was increased by 6.65%, 2%, 6.38% respectively. Among the evaluation metrics, only in the S_200 dataset, sensitivity decreased by 2.6%. The comparison is shown in Table 2. We also plotted the receiver operating characteristic (ROC) curve on the testing datasets to have a

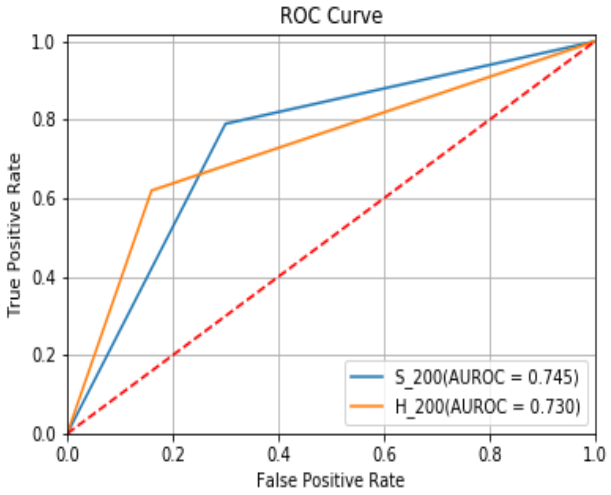

Fig. 2. Illustration of the performance of our classifier through ROC curve in the testing datasets.

better understanding of our model. The plot is shown in Figure 2.

The discussed classifier is already implemented and can be taken to the next stage by providing a user-friendly web server. In this work, we tried to tune only those hyperparameters that can impact the performance of our classifier positively. Nevertheless, tuning other hyperparameters may result in improved performance. We can also look for other encoding techniques of RNA sequences other than "one-hot" encoding in the future. Moreover, compared to the existing methods, our model produced the maximum accuracy in both H_200 and S_200 dataset.

IV. Conclusion

The purpose of our work was to identify pseudouridine sites from RNA sequences using computational methods, in our case, a mixed convolutional neural network. One of the main reasons behind employing a CNN model was its automatic feature extraction capability. After preprocessing our data using "one-hot" encoding, we adopted a CNN model having multiple convolution and max-pooling layers connected to the input layer individually, which was followed by a couple of fully-connected layers and an output layer. We applied k-fold cross-validation and grid search for hyperparameter tuning. We trained our model by using the selected values from tuning. Then we tested the performance of our model using the independent datasets and found 72.5% accuracy in the H_200 dataset and 75% accuracy in the S_200 dataset. It is projected that our classifier can become a helpful tool for identifying Ψ sites. We can also say that CNN can be used as an important method for classifying biological data.

References

[1] G. A. Hudson, R. J. Bloomingdale, and B. M. Znosko, "Thermodynamic contribution and nearest-neighbor parameters of pseudouridine-adenosine base pairs in oligoribonucleotides," *Rna*, vol. 19, no. 11, pp. 1474–1482, 2013.

[2] J. Ge and Y.-T. Yu, "Rna pseudouridylation: new insights into an old modification," *Trends in biochemical sciences*, vol. 38, no. 4, pp. 210–218, 2013.

[3] M. Charette and M. W. Gray, "Pseudouridine in rna: what, where, how, and why," *IUBMB life*, vol. 49, no. 5, pp. 341–351, 2000.

[4] C. Bousquet-Antonelli, Y. Henry, J.-P. Gélugne, M. Caizergues-Ferrer, and T. Kiss, "A small nucleolar rnp protein is required for pseudouridylation of eukaryotic ribosomal rnas," *The EMBO journal*, vol. 16, no. 15, pp. 4770–4776, 1997.

[5] D. R. Davis, C. A. Veltri, and L. Nielsen, "An rna model system for investigation of pseudouridine stabilization of the codon-anticodon interaction in trnalys, trnahis and trnatyr," *Journal of Biomolecular Structure and Dynamics*, vol. 15, no. 6, pp. 1121–1132, 1998.

[6] A. Basak and C. C. Query, "A pseudouridine residue in the spliceosome core is part of the filamentous growth program in yeast," *Cell reports*, vol. 8, no. 4, pp. 966–973, 2014.

[7] J. Karijolich and Y.-T. Yu, "The new era of rna modification," *RNA*, vol. 21, no. 4, pp. 659–660, 2015.

[8] T. M. Carlile, M. F. Rojas-Duran, B. Zinshteyn, H. Shin, K. M. Bartoli, and W. V. Gilbert, "Pseudouridine profiling reveals regulated mrna pseudouridylation in yeast and human cells," *Nature*, vol. 515, no. 7525, pp. 143–146, 2014.

[9] A. F. Lovejoy, D. P. Riordan, and P. O. Brown, "Transcriptome-wide mapping of pseudouridines: pseudouridine synthases modify specific mrnas in s. cerevisiae," *PLoS One*, vol. 9, no. 10, 2014.

[10] S. Schwartz, D. A. Bernstein, M. R. Mumbach, M. Jovanovic, R. H. Herbst, B. X. León-Ricardo, J. M. Engreitz, M. Guttman, R. Satija, E. S. Lander *et al.*, "Transcriptome-wide mapping reveals widespread dynamic-regulated pseudouridylation of ncrna and mrna," *Cell*, vol. 159, no. 1, pp. 148–162, 2014.

[11] Y.-H. Li, G. Zhang, and Q. Cui, "Ppus: a web server to predict pus-specific pseudouridine sites," *Bioinformatics*, vol. 31, no. 20, pp. 3362–3364, 2015.

[12] W. Chen, H. Tang, J. Ye, H. Lin, and K.-C. Chou, "irna-pseu: Identifying rna pseudouridine sites," *Molecular Therapy-Nucleic Acids*, vol. 5, p. e332, 2016.

[13] J. He, T. Fang, Z. Zhang, B. Huang, X. Zhu, and Y. Xiong, "Pseui: pseudouridine sites identification based on rna sequence information," *BMC bioinformatics*, vol. 19, no. 1, p. 306, 2018.

[14] M. Tahir, H. Tayara, and K. T. Chong, "ipseu-cnn: Identifying rna pseudouridine sites using convolutional neural networks," *Molecular Therapy-Nucleic Acids*, vol. 16, pp. 463–470, 2019.

[15] S.-H. Guo, E.-Z. Deng, L.-Q. Xu, H. Ding, H. Lin, W. Chen, and K.-C. Chou, "inuc-pseknc: a sequence-based predictor for predicting nucleosome positioning in genomes with pseudo k-tuple nucleotide composition," *Bioinformatics*, vol. 30, no. 11, pp. 1522–1529, 2014.

[16] P. Feng, H. Yang, H. Ding, H. Lin, W. Chen, and K.-C. Chou, "idna6ma-pseknc: Identifying dna n6-methyladenosine sites by incorporating nucleotide physicochemical properties into pseknc," *Genomics*, vol. 111, no. 1, pp. 96–102, 2019.

[17] H. Yang, W.-R. Qiu, G. Liu, F.-B. Guo, W. Chen, K.-C. Chou, and H. Lin, "irspot-pse6nc: Identifying recombination spots in saccharomyces cerevisiae by incorporating hexamer composition into general pseknc," *International journal of biological sciences*, vol. 14, no. 8, p. 883, 2018.

[18] W.-J. Sun, J.-H. Li, S. Liu, J. Wu, H. Zhou, L.-H. Qu, and J.-H. Yang, "Rmbase: a resource for decoding the landscape of rna modifications from high-throughput sequencing data," *Nucleic acids research*, vol. 44, no. D1, pp. D259–D265, 2016.

[19] W. Chen, H. Ding, X. Zhou, H. Lin, and K.-C. Chou, "irna (m6a)-psednc: identifying n6-methyladenosine sites using pseudo dinucleotide composition," *Analytical biochemistry*, vol. 561, pp. 59–65, 2018.

[20] W.-R. Qiu, B.-Q. Sun, X. Xiao, Z.-C. Xu, J.-H. Jia, and K.-C. Chou, "ikcr-pseens: Identify lysine crotonylation sites in histone proteins with pseudo components and ensemble classifier," *Genomics*, vol. 110, no. 5, pp. 239–246, 2018.