iPseU-CNN: Identifying RNA Pseudouridine Sites Using Convolutional Neural Networks

Muhammad Tahir,1,2 Hilal Tayara,1 and Kil To Chong3

Pseudouridine (Ψ) is the most prevalent RNA modification and has been found in both eukaryotes and prokaryotes. Currently, pseudouridine has been demonstrated in several kinds of RNAs, such as small nuclear RNA, rRNA, tRNA, mRNA, and small nucleolar RNA. Therefore, its significance to academic research and drug development is understandable. Through biochemical experiments, the pseudouridine site identification has produced good outcomes, but these lab exploratory methods and biochemical processes are expensive and time consuming. Therefore, it is important to introduce efficient methods for identification of pseudouridine sites. In this study, an intelligent method for pseudouridine sites using the deep-learning approach was developed. The proposed prediction model is called iPseU-CNN (identifying pseudouridine by convolutional neural networks). The existing methods used handcrafted features and machine-learning approaches to identify pseudouridine sites. However, the proposed predictor extracts the features of the pseudouridine sites automatically using a convolution neural network model. The iPseU-CNN model yields better outcomes than the current state-of-the-art models in all evaluation parameters. It is thus highly projected that the iPseU-CNN predictor will become a helpful tool for academic research on pseudouridine site prediction of RNA, as well as in drug discovery.

INTRODUCTION

Pseudouridine (Ψ) is a common RNA modification that has been found in both eukaryotes and prokaryotes.1 Currently, Ψ has been demonstrated in various categories of RNAs.2 The Ψ synthase enzyme catalyzes Ψ, the isomer of uridine, by removing uridine residue base from its sugar followed by the isomer of uridine, rotating it 180° along the N3–C6 axis, and ultimately, again linking the base’s 5-carbon to the 1’-carbon of the sugar, as shown in Figure 1.3 Currently, Ψ modification is considered to be an important process in the molecular mechanism, including stabilization of the RNA structure,3,4 and is important for gene regulation machinery, i.e., in the spliceosome. The presence of Ψ modifications in regions involved with RNA-protein or RNA-RNA interaction enhances the reaction and assembly of the spliceosome that is responsible for producing a functional mRNA, i.e., in AU/AC intron splicing.5 Furthermore, incorporation of Ψ into mRNA may inhibit the RNA-elicited innate immune response and enhance the translation efficiency of that mRNA.5 Although many researchers have unveiled the role of Ψ modification in most RNA systems, its biological functions and action mechanisms have yet to be identified. Therefore, it is important to highlight the Ψ modification sites in the transcriptome that govern the related biological principle.

Although some lab exploratory techniques have been introduced to identify Ψ sites, they are costly and labor intensive.6,7 Because of the increasing availability of genomics and proteomics samples produced in the post-genomics era, it is necessary to develop robust, fast, low-cost computational models to predict Ψ sites on the RNA sequence. In previous works, several machine-learning-based computational methods or statistical-learning techniques have been introduced to identify Ψ sites.10–12 Li et al.13 introduced a computational method, PPUS, for the identification of Ψ-synthesase (PUS)-specific Ψ sites in Saccharomyces cerevisiae and Homo sapiens. The method used the support vector machine (SVM) for classification and nucleotides around Ψ as the features. Similarly, the identifying RNA Ψ (iRNA-PseU) method was introduced by Chen et al.,14 for the identification of Ψ sites in Mus musculus, S. cerevisiae, and H. sapiens. This method combines the occurrence frequency density distributions of the nucleotides and their chemical properties into pseudo K-tuple nucleotide composition (PseKNC). Most recently, the Ψ identification (PseUI) model was developed by He et al.,15 for identification of Ψ sites from RNA samples in M. musculus, S. cerevisiae, and H. sapiens. This model used five types of feature-extraction technique, including dinucleotide composition (DC), nucleotide composition (NC), position-specific dinucleotide propensity (PSDP), position-specific nucleotide propensity (PSNP), and pseudodinucleotide composition (PseDNC). Then, a sequential forward-feature-selection strategy was used to select a relevant feature combination and a support vector machine as a classifier.16,17

More recently, PseKNC has been effectively and widely used in the predation of several RNA/DNA regulatory elements, such as the
nucleosome-positioning sequence, RNA modification sites, DNA recombination spots, translation initiation site, promoter, and origin of replication. Although the above studies have illustrated that PseKNC is one of the most often used feature-extraction techniques to formulate RNA/DNA sequences, all of them used type-I PseKNC, which mixes various physicochemical properties. Because various properties may play various roles, the type-II PseKNC could handle these variances and improve the description of sequences. Recently, type II PseKNC was used in various DNA element identification and achieved good results. On the other hand, the main focus of our work was use of a deep-learning technique for automatically extracting the important features directly from the sequence itself for classification.

The performance of the above predictors and methods can be further improved by proposing other robust machine-learning or deep-learning methods. The existing methods use hand-designed input features based on domain knowledge. However, the proposed system can automatically learn the features from RNA sequences by using a deep-learning technique. Deep learning has produced better outcomes in natural language processing, information retrieval, speech recognition, and image recognition. Recently, a large number of genomics methods and techniques have been introduced based on deep-learning mechanisms—for example, CNNclust, BiRen, iDeepS, RNA branch point prediction, alternative splicing site prediction, and iRNA-PseKNC(2methyl).

We introduce an efficient computational architecture for prediction of Ψ sites, using machine-learning and deep-learning approaches. In machine learning, two simple feature-extraction techniques were used as baselines—n-gram and multivariate mutual information (MMI)—and SVM was used as the classifier. In deep learning, we used a convolution neural network (CNN) model. As shown in the result and discussion sections, the deep-learning method produced better outcomes than the machine-learning ones. The proposed prediction iPseU-CNN (identifying Ψ by convolutional neural networks) model is based on a CNN. It is an efficient and simple architecture for Ψ site prediction and is evaluated on three various training benchmark datasets and two independent testing benchmark datasets. The proposed model achieves a more efficient outcome than the current state-of-the-art methods published recently in the literature. To the best of our knowledge, the proposed iPseU-CNN prediction model is the first model, automatically capture important features from RNA sequences using CNN for identification of Ψ sites.

RESULTS AND DISCUSSION

In recent studies, four statistical parameters, Matthews’s correlation coefficient (MCC), sensitivity (Sen), specificity (Sp), and accuracy (Acc), have been used to define the effectiveness and performance of the computational methods. These parameters are expressed as:

\[
MCC = \frac{(TN \times TP) - (FN \times FP)}{\sqrt{(FP + TP)(FP + TN)(FN + TN)(TN + FP)}}
\]  

(Equation 1)

\[
Sen = \frac{TP}{TP + FN}
\]  

(Equation 2)

\[
Sp = \frac{TN}{TN + FP}
\]  

(Equation 3)

\[
Acc = \frac{TP + TN}{TP + TN + FN + FP}
\]  

(Equation 4)

In this work, we implemented two simple machine-learning baselines. These methods are based on using n-gram and MMI for feature extraction and SVM as a classifier. The n-gram and MMI feature-extraction techniques are simple and are used widely in many applications. Table 1 shows the success rate of n-gram, MMI, and the proposed iPseU-CNN. It can be seen that the n-gram-based method outperformed the MMI-based one in the H. sapiens (H)_990, S. cerevisiae (S)_628, and M. musculus (M)_944 datasets. However, the CNN-based method markedly outperformed both machine-learning-based techniques. More specifically, iPseU-CNN improved accuracy by 6.68%, sensitivity by 13.49%, and MCC by 0.14 in the H_990 dataset. On the other hand, iPseU-CNN improved the performance of the S_628 dataset by 5.42%, 9.63%, and 0.12 in terms of accuracy, specificity, and MCC, respectively. Furthermore, iPseU-CNN improved the performance of the M_944 dataset by

| Training Dataset | Methods | Accuracy (%) | Sensitivity (%) | Specificity (%) | MCC |
|------------------|---------|--------------|----------------|----------------|-----|
| H_990            | n-gram  | 60.00        | 51.51          | 68.48          | 0.20|
|                  | MMI     | 58.78        | 47.47          | 70.10          | 0.18|
|                  | CNN     | 66.68        | 63.00          | 68.78          | 0.34|
| S_628            | n-gram  | 62.73        | 64.64          | 60.82          | 0.25|
|                  | MMI     | 60.19        | 67.51          | 52.86          | 0.20|
|                  | CNN     | 68.15        | 66.36          | 70.45          | 0.37|
| M_944            | n-gram  | 62.71        | 65.04          | 60.38          | 0.25|
|                  | MMI     | 58.26        | 63.13          | 53.38          | 0.16|
|                  | CNN     | 71.81        | 74.79          | 69.11          | 0.44|

Table 1. The Success Rates of iPseU-CNN and the Baseline Methods with the Training Datasets

Figure 1. Illustration of the Pseudouridine Modification

![Figure 1. Illustration of the Pseudouridine Modification](image-url)
9.1%, 9.75%, 8.73%, and 0.19 in terms of accuracy, sensitivity, specificity, and MCC, respectively. Thus, it is clear that the proposed iPseU-CNN predictor outperforms the baseline machine-learning methods.

The prediction outcomes of the iPseU-CNN model were measured on two independent datasets, i.e., S_200 and H_200, and are illustrated in Table 2. We showed experimentally that the success rate of our iPseU-CNN model based on deep learning was better than that of the machine-learning baseline methods. More specifically, iPseU-CNN method improved the accuracy, sensitivity, and MCC on H_200 dataset by 2%, 19.72%, and 0.05, respectively. On the other hand, the success rates of the S_200 dataset were improved by 3%, 6.82%, and 0.06 in terms of accuracy, specificity, and MCC, respectively.

It is clear that the CNN-based approach outperforms the machine-learning-based approaches with a big margin in the different evaluation metrics as shown in Tables 1 and 2 and Figure 2.

Finally, the prediction performance comparison of the iPseU-CNN model with the existing methods, such as iRNA-PseU14 and PseU15 is shown in Table 3. iRNA-PseU14 combines the occurrence frequency density distributions of the nucleotides and their chemical properties into PseKNC for feature extraction to identify Ψ sites. PseUI15 uses five feature-extraction techniques to identify Ψ sites.

The results in Table 3 show that the iPseU-CNN model improved all evaluation metrics for the H_990 dataset by 2.44%, 0.15%, 5.14%, and 0.06 in terms of accuracy, sensitivity, specificity, and MCC, respectively.

In addition, iPseU-CNN improved all evaluation metrics for the S_628 dataset by 1.71%, 3.02%, 2.93%, and 0.07 in terms of specificity, sensitivity, accuracy, and MCC, respectively, and it improved accuracy and MCC for the M_944 dataset by 1.37% and 0.03, respectively.

Furthermore, the performance of iPseU-CNN on independent datasets has been compared with those of iRNA-Pse and PseUI, as given in Table 4. It can be observed that the iPseU-CNN model improved all evaluation metrics for the S_200 dataset by 5.82%, 3.76%, 5%, and 0.1 in terms of specificity, sensitivity, accuracy, and MCC, respectively, and it improved accuracy, sensitivity, and MCC for the H_200 dataset by 3.5%, 14.72%, and 0.09, respectively.

It is clear that the CNN-based approach outperforms the current predictors in different evaluation metrics, as displayed in Tables 3 and 4 and Figure 3.

Recently, the main direction of bioinformatics applications is in preparing databases48,49 and establishing efficient web servers.22,50 Therefore, our future work is to improve the performance and build a user-friendly web server for our developed tools.

To conclude, we developed a deep-learning mechanism to identify Ψ sites from RNA samples—namely, iPseU-CNN. Machine-learning and deep-learning mechanisms were used; however, the performance of the deep-learning approach outperformed the machine-learning

Table 2. The Success Rates of iPseU-CNN and the Baseline Methods with Two Independent Testing Datasets

| Testing Dataset | Methods  | Accuracy (%) | Sensitivity (%) | Specificity (%) | MCC   |
|-----------------|----------|--------------|----------------|----------------|-------|
| H_200           | n-gram   | 67.00        | 57.00          | 78.00          | 0.35  |
|                 | MMI      | 63.50        | 58.00          | 69.00          | 0.27  |
|                 | CNN      | 69.00        | 77.72          | 60.81          | 0.40  |
| S_200           | n-gram   | 70.50        | 72.00          | 67.00          | 0.39  |
|                 | MMI      | 69.50        | 70.00          | 71.00          | 0.41  |
|                 | CNN      | 73.50        | 68.76          | 77.82          | 0.47  |

Figure 2. The Success Rates of the iPseU-CNN and Baseline Methods
ones. We applied n-gram and MMI to extract the features in the machine-learning approach and SVM for classification. The deep-learning approach used a CNN model. The iPseU-CNN model automatically learned the features from RNA sequences compared with previous works that employ handcrafted features for classification. The proposed iPseU-CNN prediction model is the first model to fully automatically capture important feature from RNA sequences using CNNs for identification of Ψ sites. The success rate indicates that the proposed prediction model is more stable and accurate than the current methods in terms of evaluation parameters. It is highly expected that the iPseU-CNN prediction model may be helpful in drug-related applications and academia.

MATERIALS AND METHODS
We introduce the proposed model and benchmark datasets used for training and testing.

The Proposed Model
We introduce an efficient computational architecture for prediction of Ψ sites using machine-learning and deep-learning approaches.

In machine-learning approaches, we used two different feature spaces, MMI and n-gram, to extract the numerical features from RNA samples and SVM as an operation engine. Second, a deep-learning approach uses CNNs to identify Ψ sites from RNA/DNA samples directly. The CNN model automatically captures the key features from the input samples during training.

Machine-Learning Approach
We selected simple feature-extraction methods to work as baselines for comparison with the proposed deep-learning method.

Table 3. The Success Rates of iPseU-CNN and State-of-the-Art Methods with the Training Datasets

| Training Dataset | Models         | Accuracy (%) | Sensitivity (%) | Specificity (%) | MCC  |
|------------------|----------------|--------------|-----------------|-----------------|------|
| H_990            | iPseU-CNN      | 66.68        | 65.00           | 68.78           | 0.34 |
|                  | PseUI          | 64.24        | 64.85           | 63.64           | 0.28 |
|                  | iRNA-PseU      | 60.40        | 61.91           | 59.80           | 0.21 |
|                  | iPseU-CNN      | 68.15        | 66.36           | 70.45           | 0.37 |
| S_628            | PseUI          | 65.13        | 62.74           | 67.52           | 0.30 |
|                  | iRNA-PseU      | 64.49        | 64.65           | 64.33           | 0.29 |
|                  | iPseU-CNN      | 71.81        | 74.79           | 69.11           | 0.44 |
| M_944            | PseUI          | 70.44        | 79.87           | 70.34           | 0.41 |
|                  | iRNA-PseU      | 69.07        | 73.31           | 64.83           | 0.38 |

Table 4. The Success Rates of the iPseU-CNN and State-of-the-Art Methods with Two Independent Testing Datasets

| Testing Dataset | Models         | Accuracy (%) | Sensitivity (%) | Specificity (%) | MCC  |
|-----------------|----------------|--------------|-----------------|-----------------|------|
| H_200           | iPseU-CNN      | 69.00        | 77.72           | 60.81           | 0.40 |
|                  | PseUI          | 65.50        | 63.00           | 68.00           | 0.31 |
|                  | RNA-PseU       | 61.50        | 58.00           | 65.00           | 0.23 |
|                  | iPseU-CNN      | 73.50        | 68.76           | 77.82           | 0.47 |
| S_200           | PseUI          | 68.50        | 65.00           | 72.00           | 0.37 |
|                  | iRNA-PseU      | 60.00        | 63.00           | 57.00           | 0.20 |

n-gram
In this feature-extraction technique, n-gram is expressed as (v_i, c_i), where v_i represents the feature and c_i represents the total number of this feature in the protein or DNA/RNA sample. For instance, in the case of 3-g, v represents the three-nucleotide combination and c represents the total number of combination occurrences inside the complete sequence. In this work, we constructed a feature vector containing from 1-g to 3-g. The n-gram can be mathematically expressed as:

\[ S = S_1 \cup S_2 \cup S_3 = \{ N_i \} \cup \{ N_iN_j \} \cup \{ N_iN_jN_l \} \]

(Equation 5)

where \( S \) represents the combination list of nucleotides, \( S_1, S_2, \) and \( S_3, \) with the 4\(^1\), 4\(^2\), and 4\(^3\) features, respectively, and \( N_i, N_j, N_l \in \{ A, C, G, U \} \) generates an 84-dimensional vector.

MMI
In prior work, MMI has been widely adopted in protein samples to extract features. In the same manner, the nucleotide samples in RNA/DNA can be represented using the MMI feature-extraction technique. In this method, the RNA/DNA samples are represented by 2-tuple and 3-tuple as follows:

\[ K_2 = \{ AA, AC, AU, AG, CC, CU, CG, UU, UG, GG \} \]
\[ K_3 = \{ AAA, AAC, AAU, AAG, ACC, ACU, ACG, AUU, AUG, AGG, CCC, CCA, CCG, CUU, CUG, CGG, UUU, UUG, UGG, GGG \} \]

(Equation 6)

There is no relationship with the order of the nucleotides for the MMI in a tuple. The K2 has 10 elements and K3 has 20 elements.

The 2-tuple mutual information (MI) for the nucleotide pair in K2 can be defined as below:

\[ I(M_1M_2) = f(M_1, M_2) \ln \frac{f(M_1, M_2)}{f(M_1)f(M_2)} \]

(Equation 7)
The 3-tuple MI for the nucleotide pair in K3 can be defined as below:

\[
I(M_1M_2M_3) = f(M_1, M_2) \ln \frac{f(M_1, M_2)}{f(M_1)f(M_2)} + f(M_1, M_3) \ln \frac{f(M_1, M_3)}{f(M_1)f(M_3)} + f(M_2, M_3) \ln \frac{f(M_2, M_3)}{f(M_2)f(M_3)}
\]

(Equation 8)

where \( f(M_i) \) is a fraction of each nucleotide in the sequence and \( f(M_i, M_j) \) and \( f(M_i, M_j, M_k) \) are the occurrence frequency of 2-tuple and 3-tuple, respectively.

**SVM**

SVM is a learning tool for regression, classification, and pattern recognition. It has achieved more efficient results than other machine-learning methods or techniques. In the current study, the LIBSVM package was used for implementing the SVM model, in which the radial basis function (RBF) was used as the kernel function. The kernel of RBF includes two parameters, \( \alpha \) and \( \gamma \), that are set to 5.5 and 0.0035, respectively. The concrete values of these parameters are determined through the optimization procedure called a grid-search algorithm on the benchmark dataset.

**Deep-Learning Approach**

We used a CNN to predict \( \Psi \) sites from RNA/DNA samples, and during training, it automatically searched the key features in the input samples. The CNN model took a single RNA sequence as an input (\( n = 21 \) for the M_944 and H_990 datasets and \( n = 31 \) for the S_628 dataset) and produced a real value. The input is represented by a one-hot vector with four channels (A, C, G, and U) and \( n \) length.

The best parameters were selected based on validation loss. The sigmoid function outputs normalized class probabilities for a given input. The convolution layer is mathematically represented and computed as

\[
\text{Conv}(R)_j = \text{ReLU} \left( \sum_{i=0}^{S-1} \sum_{n=0}^{N-1} W^f_{i,n} R_{j,i,n} \right)
\]

(Equation 9)

where \( R \) represents the input of the RNA sample, \( f \) denotes the index of the filter, and \( j \) denotes the index of the output position. Each filter \( W^f \) is an \( S \times N \) weight matrix of size \( S \) channels of \( N \). The rectified linear function (ReLU) is expressed as:

\[
\text{ReLU}(z) = \begin{cases} 
  z & \text{if } z \geq 0 \\
  0 & \text{if } z < 0
\end{cases}
\]

(Equation 10)

The output layer is transformed to [0, 1] by a sigmoid function that is used for \( \Psi \) sites predictions.

\[
\text{Sigmoid}(y) = \frac{1}{1 + e^{-y}}
\]

(Equation 11)

In this study, the Keras framework was used to implement the iPseU-CNN model. The Adam optimizer with a learning rate of 0.001 was used, epochs were set to 50, and the batch size was set to 10.

**Benchmark Datasets**

In this study, three different benchmark datasets—M_944, S_628, and H_990—were used for training, where M, S, and H denoted \( M. \text{musculus}, S. \text{cerevisiae}, \) and \( H. \text{sapiens} \), respectively, and each
dataset contained 944, 628, and 990 samples, respectively. These three benchmark datasets of pseudouridylation sites were taken from the additional materials of Chen et al.\textsuperscript{14}, who also introduced two various independent testing datasets for \textit{S. cerevisiae} and \textit{H. sapiens} denoted S\_200 and H\_200, respectively. The H\_990, M\_944, and S\_628 datasets consisted of 495, 472, and 314 positive subsets of RNA samples, and every RNA sample had a uridine at the center position that could be pseudouridylated. Similarly, H\_990, M\_944, and S\_628 datasets contained 495, 472, and 314 negative subsets of RNA samples, and each RNA sample had a uridine at the center position, but it could not be pseudouridylated. The RNA sample of these three datasets can be mathematically formulated as:

\[
S_x(U) = N_{-\xi}N_{-1}...UN_{\xi}...N_{\xi+1}N_{\xi}
\]

(Equation 12)

where \(S_x(U)\) represents the RNA sample, the center U denotes uridine, \(N_{-\xi}\) denotes the upstream, and \(N_{\xi}\) denotes the downstream of the central uridine for all \(\xi\)-th elements.

Cross-Validation

The error rate used in the machine- and deep-learning methods to evaluate the performance of the operation engine. In this regard, the dataset was divided into different mutually exclusive folds. In this work, we used a \(k\)-fold cross-validation test where a particular dataset can be divided into \(k\)-fold for cross-validation.\textsuperscript{61,62,67,68} In this type of validation test, for the testing purpose, 1-fold was reserved, whereas for training a particular model, the remaining \(k - 1\) folds were used. This is a \(k\)-time recursive process where every fold is tested once.\textsuperscript{62,69} We applied a 5-fold cross-validation test to measure the four performance parameters.

**Table 5. The Ranges of the Tuned Hyper-Parameters**

| Hyper-Parameter  | Range             |
|------------------|-------------------|
| Convolution layers | [1,2]            |
| Filters          | [5,7,9]           |
| Filter size      | [3,5,7]           |
| Stride           | [1,2]             |
| Dropout          | [0.25, 0.50]      |

**SUPPLEMENTAL INFORMATION**

Supplemental Information can be found online at https://doi.org/10.1016/j.omtn.2019.03.010.

**AUTHOR CONTRIBUTIONS**

Conceptualization, M.T. and H.T.; Methodology, M.T. and H.T.; Investigation, M.T., H.T., and K.T.C.; Writing – Original Draft, M.T. and H.T.; Writing – Review & Editing, M.T., H.T., and K.T.C.; Visualization, M.T., and H.T.; Supervision, K.T.C.

**CONFLICTS OF INTEREST**

The authors declare no competing interests.

**ACKNOWLEDGMENTS**

This work was supported by the Brain Research Program of the National Research Foundation (NRF) funded by the Korean government (MSIT) (NRF-2017M3C7A1044815).
REFERENCES

1. Hudson, G.A., Bloomingdale, R.L., and Znosko, B.M. (2013). Thermodynamic contribution and nearest-neighbor parameters of pseudouridine-adenosine base pairs in oligoribonucleotides. RNA 19, 1474–1482.

2. Ge, J., and Yu, Y.-T. (2013). RNA pseudouridyllation: new insights into an old modification. Trends Biochem. Sci. 38, 210–218.

3. Charette, M., and Gray, M.W. (2000). Pseudouridine in RNA: what, where, how, and why. IUBMB Life 49, 341–351.

4. Davis, D.R., Veltri, C.A., and Nielsen, L. (1998). An RNA model system for investigation of pseudouridine stabilization of the codon-anticodon interaction in tRNA{\text{\textit{Lys}}} and tRNA{\text{\textit{Try}}}. J. Biomol. Struct. Dyn. 15, 1121–1132.

5. Basak, A., and Querry, C.C. (2014). A pseudouridine residue in the spliceosome core is part of the filamentous growth program in yeast. Cell Rep. 8, 966–973.

6. Karioljtech, J., and Yu, Y.-T. (2015). The new era of RNA modification. RNA 21, 659–660.

7. Carlile, T.M., Rojas-Duran, M.F., Zinshteyn, B., Shin, H., Bartoli, K.M., and Gilbert, J. (2013). Upstream pseudouridylation: new insights into an old modification. Chemometrics and Intelligent Laboratory Systems 141, 100–106.

8. Lovejoy, A.F., Riordan, D.P., and Brown, P.O. (2014). Transcriptome-wide mapping of pseudouridines: pseudouridine synthases modify specific mRNAs in S. cerevisiae. PLoS ONE 9, e110799.

9. Schwartz, S., Bernstein, D.A., Mumbach, M.R., Jovanesic, M., Herbst, R.H., Leon-Ricardo, B.K., Engeertz, J.M., Guttmann, M., Satija, R., Lander, E.S., et al. (2014). Transcriptome-wide mapping reveals widespread dynamic-regulated pseudouridylation of tRNA and mRNA. Cell 159, 148–162.

10. Chen, W., Feng, P., Yang, H., Ding, H., and Lin, H. (2018). A sequence-based approach for identifying translation initiation sites in human genes using pseudo trimucleotide composition. Anal. Biochem. 462, 76–83.

11. Feng, P., Yang, H., Ding, H., Chen, W., and Chou, K.-C. (2019). iDNA6mA-PseKNC: Identifying DNA N6-methyladenosine sites by incorporating nucleotide physicochemical properties into PseKNC. Genomics 111, 96–102.
55. Caragea, C., Silvescu, A., and Mitra, P. (2012). Protein sequence classification using pseudo dinucleotide composition. Anal. Biochem. 561–562, 59–65.

47. Molecular Therapy: Nucleic Acids Vol. 16 June 2019

49. Zhang, T., Tan, P., Wang, L., Jin, N., Li, Y., Zhang, L., Yang, H., Hu, Z., Zhang, L., Hu, C., Liang, Z.-Y., Lai, H.-Y., Yang, H., Zhang, C.-J., Yang, H., Wei, H.-H., Chen, X.X., et al. (2017). RNALocate: a resource for RNA subcellular localizations. Nucleic Acids Res. 45(Suppl 1), D129–D135.

48. Li, B., Li, K., Huang, D.-S., and Chou, K.-C. (2018). A novel computational method for prediction of nucleosome positioning in genomes by extending the concept of Chou’s PseAAC and Chou’s PseKNC. Mol. Biol. 12, 2587–2593.

53. Nanni, L. (2005). Hyperplanes for predicting protein-protein interactions. Neurocomputing 69, 257–263.

54. Cao, J., and Xiong, L. (2014). Protein sequence classification with improved extreme learning machine algorithms. BioMed Res. Int. 2014, 103054.

55. Caragea, C., Silvescu, A., and Mitra, P. (2012). Protein sequence classification using pseudo dinucleotide composition. Anal. Biochem. 561–562, 59–65.

51. Qiu, W.-R., Sun, B.-Q., Xiao, X., Xu, Z.-C., Jia, J.-H., and Chou, K.-C. (2018). iKcr-PseEns: Identify lysine crotonylation sites in histone proteins with pseudo component and ensemble classifier. Genomics 110, 239–246.

52. Pan, G., Jiang, L., Tang, J., and Guo, F. (2018). A novel computational method for prediction of nucleosome positioning by extending the concept of Chou’s PseAAC and Chou’s PseKNC. Mol. Biol. 12, 2587–2593.

45. Liu, B., Li, K., Huang, D.-S., and Chou, K.-C. (2018). iEnhancer-EL: identifying enhancers and their strength with ensemble learning approach. Bioinformatics 34, 3835–3842.

53. Nanni, L. (2005). Hyperplanes for predicting protein-protein interactions. Bioinformatics 22, 1207–1210.

58. Chen, W., Yang, H., Feng, P., Ding, H., and Lin, H. (2017). iDNA4mC: identifying DNA N4-methylcytosine sites based on nucleotide chemical properties. Bioinformatics 33, 3518–3523.

59. Tahir, M., and Hayat, M. (2017). Machine learning based identification of protein-protein interactions using derived features of physicochemical properties and evolutionary profiles. Artif. Intell. Med. 78, 61–71.

60. Tahir, M., Hayat, M., and Kabir, M. (2017). Sequence based predictor for discrimination of enhancer and their types by applying general form of Chou’s trinucleotide composition. Comput. Methods Programs Biomed. 146, 69–75.

61. Hayat, M., and Iqbal, N. (2014). Discriminating protein structure classes by incorporating pseudo average chemical shift to Chou’s PseAAC and support vector machine. Comput. Methods Programs Biomed. 116, 184–192.

62. Hayat, M., and Khan, A. (2011). Predicting membrane protein types by fusing composite protein sequence features into pseudo amino acid composition. J. Theor. Biol. 271, 10–17.

63. Hayat, M., and Tahir, M. (2015). PSOFuzzySVM-TMH: identification of transmembrane helix segments using ensemble feature space by incorporated fuzzy support vector machine. Mol. Biosyst. 11, 2255–2262.

64. Kabir, M., and Hayat, M. (2016). iRSpot-GAEnsC: identifying recombination spots via ensemble classifier and extending the concept of Chou’s PseAAC to formulate DNA samples. Mol. Genet. Genomics 291, 285–296.

65. Tahir, M., Hayat, M., and Khan, S.A. (2018). iNuc-ext-PseTNC: an efficient ensemble model for identification of nucleosome positioning by extending the concept of Chou’s PseAAC to pseudo-tri-nucleotide composition. Mol. Genet. Genomics 294, 199–210.

66. Keras. (2015). Keras: Deep learning library for theano and tensorflow. https://keras.io.

67. Feng, P., Ding, H., Yang, H., Chen, W., Lin, H., and Chou, K.-C. (2017). iRNA-PseColl: identifying the occurrence sites of different RNA modifications by incorporating collective effects of nucleotides into PseKNC. Mol. Ther. Nucleic Acids 7, 155–163.

68. Hayat, M., and Khan, A. (2012). Discriminating outer membrane proteins with Fuzzy K-nearest Neighbor algorithms based on the general form of Chou’s PseAAC. Protein Pept. Lett. 19, 411–421.

69. Tahir, M., Hayat, M., and Khan, S.A. (2018). A Two-Layer Computational Model for Discrimination of Enhancer and Their Types Using Hybrid Features Pace of Pseudo K-Tuple Nucleotide Composition. Arab. J. Sci. Eng. 43, 6719–6727.