Interdisciplinary Sciences: Computational Life Sciences (2020) 12:193–203
https://doi.org/10.1007/s12539-020-00362-y

ORIGINAL RESEARCH ARTICLE

iPseU-Layer: Identifying RNA Pseudouridine Sites Using Layered Ensemble Model

Yashuang Mu[1,2] · Ruijun Zhang[3] · Lidong Wang[3] · Xiaodong Liu[4]

Received: 9 October 2019 / Revised: 16 February 2020 / Accepted: 19 February 2020 / Published online: 13 March 2020

Abstract

Pseudouridine represents one of the most prevalent post-transcriptional RNA modifications. The identification of pseudouridine sites is an essential step toward understanding RNA functions, RNA structure stabilization, translation process, and RNA stability; however, high-throughput experimental techniques remain expensive and time-consuming in lab explorations and biochemical processes. Thus, how to develop an efficient pseudouridine site identification method based on machine learning is very important both in academic research and drug development. Motived by this, we present an effective layered ensemble model designated as iPseU-Layer for identification of RNA pseudouridine sites. The proposed iPseU-Layer approach is essentially based on three different machine learning layers including: feature selection layer, feature extraction and fusion layer, and prediction layer. The feature selection layer reduces the dimensionality, which can be regarded as a data pre-processing stage. The feature extraction and fusion layer utilizes an ensemble method which is implemented through various machine learning algorithms to generate some outputs. The prediction layer applies classic random forest to identify the final results. Furthermore, we systematically conduct the validation experiments using cross-validation tests and independent test with the current state-of-the-art models. The proposed iPseU-Layer provides a promising predictive performance in terms of sensitivity, specificity, accuracy and Matthews correlation coefficient. Collectively, these findings indicate that the framework of iPseU-Layer is a feasible and effective strategy for the prediction of RNA pseudouridine sites.

Keywords Pseudouridine · Feature extraction · Ensemble model · Prediction

✉ Yashuang Mu
muyashuang324@126.com

Ruijun Zhang
1584764709@qq.com

Lidong Wang
ldwang@hotmail.com

Xiaodong Liu
xdliuros@dlut.edu.cn

1 Key Laboratory of Grain Information Processing and Control, Ministry of Education, Henan University of Technology, Zhengzhou 450001, People's Republic of China

2 College of Information Science and Engineering, Henan University of Technology, Zhengzhou 450001, People's Republic of China

3 School of Science, Dalian Maritime University, Dalian 116026, People's Republic of China

4 School of Control Science and Engineering, Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, People's Republic of China

1 Introduction

Pseudouridine represents the most abundant of the post-transcriptional modifications, and is frequently referred to as "the fifth ribonucleoside". Pseudouridine, the C5-glycoside isomer of uridine, is derived from uridine via base-specific isomerization and contains an extra imino group (>C=NH), which serves as an additional hydrogen-bond donor and a carbon–carbon (C–C) glycosidic linkage. These two chemical changes confer rigidity to the sugarphosphate backbone and enhance local base stacking [1]. Pseudouridine has been found in eukaryotes and prokaryotes [2]. Given its abundance, widespread localization, and highly conserved properties, the pseudouridine modification has been identified as an important process in molecular mechanism; meanwhile, it is also extremely important for gene regulation machinery [3]. Owing to its unique structural and chemical properties and its proven biological relevance, pseudouridine has gained significant attention. For decades, pseudouridine is often clustered in important regions of a variety of cellular

RNAs, including small nuclear RNA [4–6], ribosomal RNA (rRNA) [7, 8], transfer RNA (tRNA) [9–11] and messenger RNA (mRNA) [12, 13]. It is believed that pseudouridine contributes to RNA functions, RNA structure stabilization, translation process, and RNA stability. Although a variety of methods have unveiled the functional significance of pseudouridine modification, the functional role for most of these modifications has not yet been elucidated. Therefore, it becomes imperative to identify and functionally characterize the pseudouridine modification sites in diverse biological contexts [3]. Besides, accurate identification of pseudouridine sites in RNA will be of immense importance for understanding these cellular processes.

In this context, several laboratories have made substantial progress in developing lab exploratory techniques; however, they are highly expensive and labor-intensive [14–16] since it takes up a lot of time and energy during the actual situation. More recently, with the avalanche of genomics and proteomics technology in the post-genomics era, exceptional complements to experimental techniques have been increasingly utilized to identify and predict pseudouridine modification sites. These kinds of methods develop some rapid, robust, and cost-effective computational models via machine learning based on the large-scale data yielded from these high-throughput sequencing technologies. Many researchers study the topic through various machine-learning-based methods and have resulted in several important discoveries, which is also an interesting direction to identify pseudouridine sites [17, 18].

In general, these machine-learning-based computational methods predominantly employ machine learning to extract the utilized features for original data or to predict pseudouridine sites in terms of the utilized features. Li et al. [19] treated nucleotides around pseudouridine as the features and employed support vector machine (SVM) to identify some specific pseudouridine sites. Chen et al. [20] combined the occurrence frequency density distributions of nucleotides and their chemical properties into pseudo K-tuple nucleotide composition to achieve the purpose of identifying pseudouridine sites. Using different types of feature-extraction techniques through sequential forward-feature-selection strategy, He et al. [21] selected a combination of relevant features, employing the classical SVM as a classifier for the identification of pseudouridylation sites. Furthermore, Muhammad et al. [3] developed a deep learning technique to automatically extract the important features directly from the sequence itself for classification, where two simple feature-extraction techniques were used as baselines, and SVM was used to design a classifier. Meanwhile, a convolution neural network model was employed to enhance the identifying performance. Recently, Liu et al. [22] proposed an eXtreme Gradient Boosting-based method, called XG-PseU, to identify pseudouridine sites, which was based on some optimal features obtained using the forward feature selection together with increment feature selection method. For the similar problems, Dou et al. [23] applied a bi-profile Bayes model to extract some RNA sequence features and these features were used to build a predictor on the base of some machine learning methods.

Currently, most of the available computational approaches might vary from one to another but are based on some extracted features and machine learning algorithms. Despite each approach has its own advantages and shows some effective results, developing a robustly efficient methodology for the identification of novel putative pseudouridine sites still remains an essential area of research and drug development. Thus, our goal is not to find a replacement for these approaches but to develop a new machine-learning-based method to enrich the researches. As the proposed method is based on a layered ensemble model to identify the novel pseudouridine sites in transcriptome, we simply name it as iPseU-Layer in this study. It mainly includes three type layers with different functions: (1) the first-type layer is a data pre-processing method to reduce the dimensionality of data; (2) the second-type layer is an ensemble method implemented through various machine learning algorithms, such as native bayes (NB), bayes network (BN), linear logistic regression (LR), sequential minimal optimization (SMO), C4.5 decision tree (C4.5), and random forest (RF). Some middle outputs are generated in this stage; (3) the third-type layer is also an ensemble learning model by applying the classic random forest to identify the final prediction results. In the experimental studies, the established prediction model is evaluated on three training benchmark datasets and two independent testing benchmark datasets. The main contribution of the proposed iPseU-Layer model is providing a promising predictive performance in terms of sensitivity, specificity, accuracy and Matthews correlation coefficient through comparing with the current state-of-the-art models.

2 Materials and Methods

In this section, the layered ensemble model iPseU-Layer is designed for the identification of pseudouridine sites. We firstly describe the overall architecture of iPseU-Layer model. Then, according to the main steps of identifying pseudouridine sites in many recent publications [24–34], the related benchmark datasets and the details of iPseU-Layer are introduced one by one.

2.1 The Overall Architecture of the Proposed iPseU-Layer Model

The architecture of iPseU-Layer is developed using five machine learning layers: one feature selection layer, three

features extraction and fusion layers, and one prediction layer. The main flowchart can be illustrated in Fig. 1.

As show in Fig. 1, the feature selection layer is used to reduce the dimension of dataset; the three feature extraction and fusion layers aim to extract and combine the features to generate a new dataset, where each layer contains six ensemble models (EMs) including NB-based EM, BN-based EM, LR-based EM, SMO-based EM, C4.5-based EM and RF-based EM; the prediction layer is utilized to obtain the final predicted category, which is a RF-based EM. Suppose $X = \{x_i\}_{i=1}^{n}$ denotes a training dataset, where each sample contains some condition attributes and one decision attribute. In Algorithm 1, we simply introduce the pseudocode about how to train an iPseU-Layer model.

Algorithm 1: The learning algorithm of iPseU-Layer model.

Input: Let $X = \{x_i\}_{i=1}^{n}$ be a training dataset.
Output: A trained iPseU-Layer model.
Apply feature selection method to the original dataset X and get a reduced dataset $\overline{X}$ (feature selection layer).
for *the i-th feature extraction and fusion layer* $(i = 1, 2, 3)$ **do**
  Use the dataset $\overline{X}$ to train six ensemble models (EMs).
  Apply the six trained EMs to $\overline{X}$ and each EM generates a voting feature.
  The six voting features and the original features in $\overline{X}$ are fused to form a new dataset $\widehat{X}$.
  $\overline{X} = \widehat{X}$.
end
Use $\overline{X}$ to train a random forest for prediction (prediction layer).
return A trained iPseU-Layer model.

2.2 The Benchmark Datasets Introduction

To provide a comprehensive and unbiased comparison, three different benchmark datasets and two independent testing datasets are employed in this paper. The three benchmark datasets are obtained from the additional materials in [20], and are considered as the training datasets. In [3], the three benchmark datasets are simply described as *M*_944, *S*_628, and *H*_990, where *M*, *S*, and *H* denoted *M. musculus*, *S. cerevisiae*, and *H. sapiens*, respectively, and each digit presents the number of samples in the corresponding benchmark dataset. In additional, the two independent testing datasets for *S. cerevisiae* and *H. sapiens* are also introduced in [20], which are denoted as *S*_200 and *H*_200, respectively. In this study, we follow the above-mentioned symbols about these datasets.

Both the training datasets and the independent testing datasets consist of some positive samples and some negative samples, where each positive RNA sample has a uridine at the center position, which can be pseudouridylated, and each negative RNA sample has a uridine at the center position, but it cannot be pseudouridylated. All the benchmark datasets are balanced. The feature vectors in these datasets are transformed from the biology sequences by five kinds of feature extraction methods which are Nucleic Acid Composition (NAC), Di-Nucleotide Composition (DNC), Tri-Nucleotide Composition (TNC), Position-specific trinucleotide propensity based on single-strand (PSTNPss) and Nucleotide Chemical Property (NCP) [35]. The detailed information about the three training datasets and the two independent testing datasets are summarized in Table 1, where the numbers in 2–6 columns represent the feature indexes extracted by the corresponding feature extraction methods and the last two columns denote the numbers of positive samples and negative samples.

As shown in Table 1, if the sequence length is L, then the dimensions of feature vector obtained by the five feature extraction methods are 4, 4^2, 4^3, $(L-2)$ and $(L \times 3)$, respectively. Since the length of each RNA sequence in *M*_944 and *H*_990 is 21, and the length of each RNA sequence in *S*_628 is 31, thus the number of features is 166 both in *M*_944 and *H*_990, and the number of features in *S*_628 is 206. Similarly, the dataset *S*_200 and the dataset *H*_200 have 206 and 166 features, respectively.

2.3 The Feature Selection Layer

As depicted in Table 1, the numbers of features in these benchmark datasets are 166 and 206, respectively. For a dataset, some redundant features not only take up more storage space and computational cost, but also weaken the generalization ability of machine learning algorithms. To detect the redundant features, we employ a feature selection technique to preprocess the benchmark datasets, which is considered as the feature selection layer.

In the feature selection layer, we employ the Correlation-based Feature Selection (CFS) algorithm by Hall [36] to

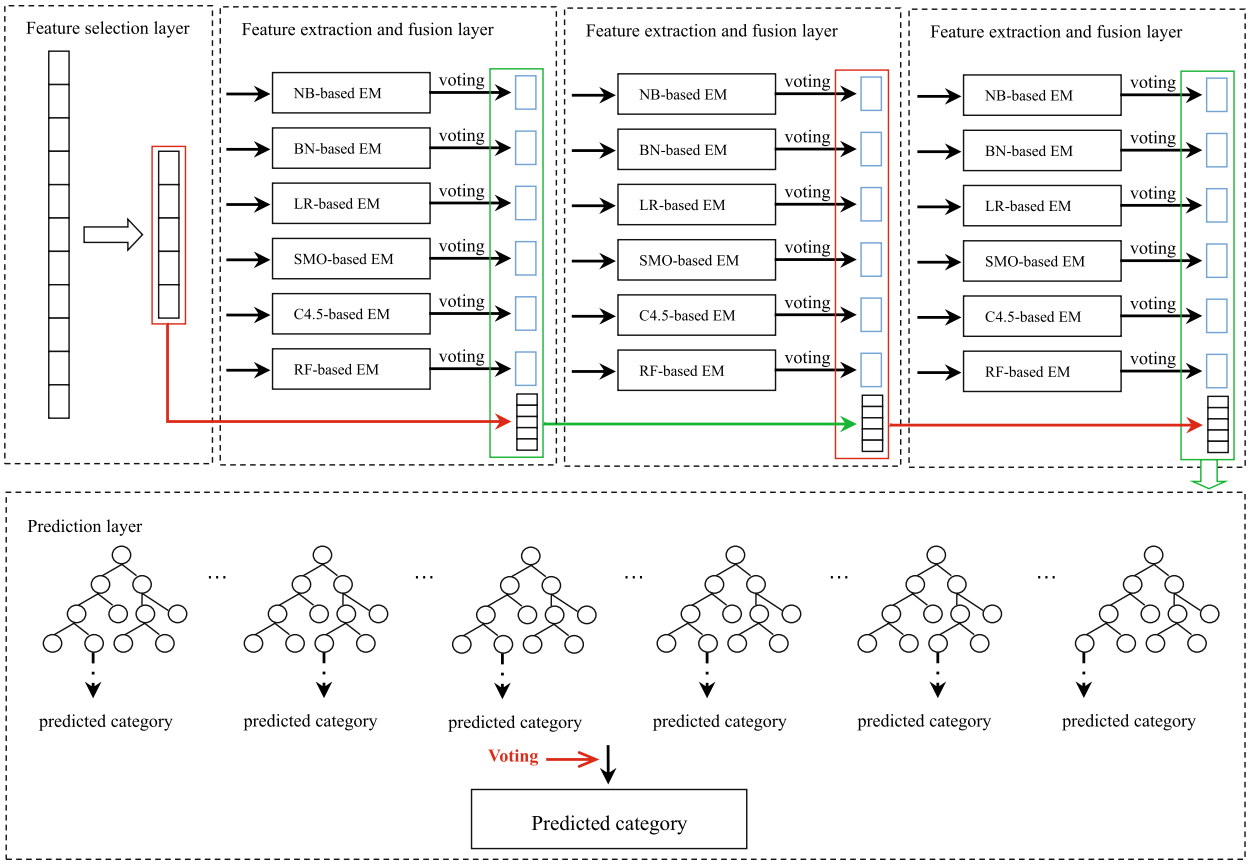

Fig. 1 The flowchart of the designed iPseU-Layer model. The sample is firstly preprocessed by a feature selection layer. Then, the reduced sample is delivered to the first feature extraction and fusion layer, where there are six different base-classifier-based ensemble models, and each model can generate a vote for the prediction category. After that, the votes and the original features are fused to form a new sample. Similarly, the new sample is delivered to the next feature extraction and fusion layer. Finally, through the processing of three feature extraction and fusion layer, a new sample is produced and its predicting category is output in the last prediction layer through the voting strategy

Table 1 The details of the employed benchmark datasets

Datasets	Feature extraction methods					Positive	Negative
	NAC	NDC	TNC	PSTNPss	NCP		
*M*_944	1–4	5–20	21–84	85–103	104–166	472	472
*H*_990	1–4	5–20	21–84	85–103	104–166	495	495
*S*_628	1–4	5–20	21–84	85–113	114–206	314	314
*S*_200	1–4	5–20	21–84	85–113	114–206	100	100
*H*_200	1–4	5–20	21–84	85–103	104–166	100	100

search the space of feature subsets by greedy hill-climbing augmented with a backtracking facility. Each feature subset is evaluated by considering the individual predictive ability of each feature together with the degree of redundancy between them. The CFS algorithm prefers the feature subsets which are highly associated with the class, while having low inter correlation. In this study, the feature selection method is implemented with the class *CfsSubsetEval.java* in Weka [37] software (the version is Weka 3.6.9). Besides, all the parameters related to the feature selection method are typically set to their default values. The indexes of remaining features after using the feature selection on three training datasets are listed in Table 2.

As presented in Table 2, through applying the feature selection technique to the three benchmark training datasets, the original features of these datasets are markedly reduced. The number of features in *M*_944 dataset is reduced from 166 to 22; the number of features in *S*_628 dataset is reduced

Table 2 The results of feature selection on training datasets

Datasets	The indexes of remaining features after reduction
*M*_944	15, 45, 85, 86, 87, 88, 89, 90, 91, 92, 94, 95, 96, 97, 98, 99, 100
	101, 102, 103, 139, 140
*S*_628	4, 29, 61, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98
	101, 102, 103, 104, 105, 106, 107, 108, 110, 111, 112, 113
*H*_990	14, 48, 85, 86, 87, 88, 89, 90, 91, 92, 94, 96, 97, 98, 99, 100, 101
	102, 103

from 206 to 29, and the number of features in *H*_990 dataset is reduced from 206 to 19.

attribute. For an ensemble learning model, two parameters are needed to preset in advance: the number of base-classifiers and the proportion of the randomly selected samples. In this layer, we consider six base-classifiers, that is, NB, BN, LR, SMO, C4.5 and RF. Each base-classifier can generate an ensemble model. We use the symbol δ and the symbol N to denote the proportion of the randomly selected samples and the number of base-classifiers in an ensemble model, respectively. For a sample, each ensemble model can give a predicting category through the voting strategy, where the vote is treated as a new feature. Algorithm 2 summarizes the extracting process of the voting feature in detail, where *bc* represents the name of base-classifier.

Algorithm 2: The extraction of the voting feature.

Input: Let $X = \{x_i\}_{i=1}^n$ be a dataset; δ describes the proportion of randomly selected samples; N denotes the number of base-classifier *bc*.
Output: A voting feature f_{bc} extracted by the base-classifier *bc*.
1 Construct an ensemble model (*bc*-based EM) on X via parameters δ and N.
2 **for** *each sample* x_i *in* X **do**
3 | Obtain the vote v_i of predicting category by applying *bc*-based EM to x_i.
4 **end**
5 **return** $f_{bc} = \{v_i\}_{i=1}^n$.

2.4 The Feature Extraction and Fusion Layer

The second-type layer designs an ensemble model based on various machine learning algorithms for the extraction and fusion of features. Thus, we regard it as the feature extraction and fusion layer. In what follows, we first introduce how to extract some new features, and describe how to generate a new dataset.

Let $X = \{x_i\}_{i=1}^n$ be a set of samples, where each sample contains some condition attributes and one decision

From the above Algorithm 2, we can find that each base-classifier can generate a voting feature. As there are six base-classifiers, six voting features (the respective votes of six base-classifiers) can be generated in this layer. In the sequel, the original dataset and the six voting features are fused together to produce a new dataset, which is further processed in the next stage. The detailed fusing process of features can be described by Algorithm 3. In this study, all the base-classifiers are also implemented by the corresponding classes in Weka. Meanwhile, all parameters related to these machine learning algorithms are preset by their default values.

Algorithm 3: The generation of new dataset.

Input: Let $X = \{x_i\}_{i=1}^n$ be a dataset; δ describes the proportion of randomly selected samples; N denotes the number of base-classifier *bc*.
Output: A new dataset $\overline{X}$.
1 Suppose $U = \{NB, BN, LR, SMO, C4.5, RF\}$.
2 **for** *each base-classifier bc in* U **do**
3 | Compute the voting feature f_{bc} by applying *bc* to Algorithm 2.
4 | Conduct a new dataset $\overline{X}$ through adding f_{bc} to dataset X.
5 | $X = \overline{X}$;
6 **end**
7 **return** $\overline{X}$.

2.5 The Prediction Layer

In this layer, we take RF as the base-classifier to construct an ensemble model for the prediction of an unknown sample. For a training dataset $X = \{x_i\}_{i=1}^{n}$, the symbol δ and the symbol N represent the proportion of randomly selected samples and the number of base-classifiers, respectively. According to the main constructing steps of an ensemble model, there are N subsets of samples through a simple random sampling with replacement, and each subset covers $\delta * n$ samples. Finally, an ensemble model including N base-classifiers are constructed.

Unlike the previous layers, this layer aims to predict the category, and thus the output is not a voting feature but the predicting category. The labeling rule is according to the majority voting, which represents the majority of the class labels predicted by each individual classifier. In additional, the base-classifier RF is implemented by the class *RandomForest.java* in Weka, and the parameters are preset to their default values. As the base-classifier RF is also an ensemble model based on random trees, we mark it using random trees in Fig. 1.

2.6 The Evaluation of Performance

From the introduction about the benchmark datasets, the proposed iPseU-Layer model is mainly focused on the binary (two classes) classification problem. There are four widely used measurements such as accuracy (Acc), sensitivity (Sn), specificity (Sp), and Matthews correlation coefficient (MCC) to measure the prediction quality of the designed iPseU-Layer model. These measurements provide some more intuitive and simple understandings about the performance, which have been described in various publications [20, 24, 25, 27, 38–54]. This study follows the symbols introduced by [55, 56] in the prediction of signal peptides. Suppose N^+ denotes the total number of pseudouridine modification sequences, N_-^+ represents the number of sequences incorrectly predicted as non-pseudouridine modification sequences, N^- is the total number of non-pseudouridine modification sequences, and N_+^- is the number of non-pseudouridine sequences incorrectly predicted as pseudouridine modification sequences. For the datasets used in this study, if the uridine in an RNA sample can be pseudouridylated, the sample is designated to positive; otherwise, it is negative. The formulas of these measures are summarized as follows:

$$Acc = 1 - \frac{N_-^+ + N_+^-}{N^+ + N^-}. \tag{1}$$

$$Sn = 1 - \frac{N_-^+}{N^+}. \tag{2}$$

$$Sp = 1 - \frac{N_+^-}{N^-}. \tag{3}$$

$$MCC = \frac{1 - \frac{N_-^+ + N_+^-}{N^+ + N^-}}{\sqrt{\left(1 + \frac{N_+^- - N_-^+}{N^+}\right)\left(1 + \frac{N_-^+ - N_+^-}{N^-}\right)}}. \tag{4}$$

3 Results and Discussions

Several experiments are performed to highlight the feasibility and effectiveness of the proposed iPseU-Layer model. In the experiments, the proposed method is implemented in Java language, and the data structures of Weka are applied. All experiments are conducted on a machine with Intel Core i7-8550U 1.99 GHZ, 8 GB RAM, and Windows 10 (64 Bit) OS. Based on the two independent testing datasets, we first investigate the effects of the parameters δ and N on the prediction quality, respectively. Furthermore, a comparative analysis is conducted with some current state-of-the-art models.

3.1 The Impacts of Parameter δ on Performance

This experiment mainly studies the impacts of parameter δ including Sn, Sp, Acc and MCC for identification of RNA pseudouridine sites. As δ represents the proportion of randomly selected samples in original data, the parameter δ varies from 2 to 100% with the step length 1% during the experiment. Meanwhile, the parameter N is preset to 50 during the experiment. The used training data sets are S_628 and H_990. The datasets S_200 and H_200 are considered as the testing datasets, respectively. The experiments are conducted twenty times. The detailed impacts of parameter δ on the average values can be illustrated in Fig. 2, in which X-axis represents the changes of parameter δ, and Y-axis denotes the values of evaluation measures including Sn, Sp, Acc and MCC.

From the curves in Fig. 2, the following observations can be summarized: (1) when the parameter δ is very small, there are not enough samples to train the models, and thus the evaluation indexes are not competitive; (2) with the increase in parameter δ, the performances are gradually improved as the whole with the increase in number of training samples; (3) for a range value of parameter δ, the evaluation indexes are not significantly improved; (4) when the parameter δ is large enough, the performances are gradually impaired for the dataset S_200 attributed to the over-fitting problem.

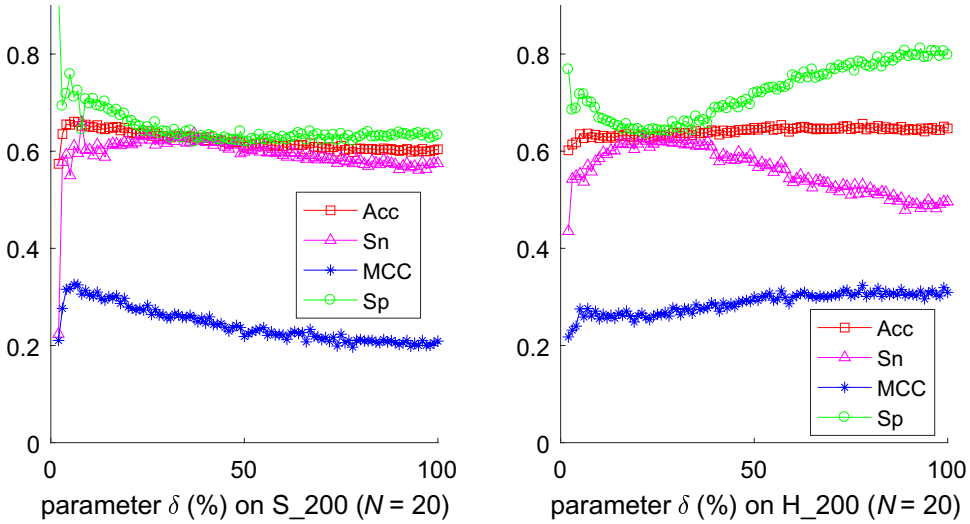

Fig. 2 The impacts of parameter δ on performance

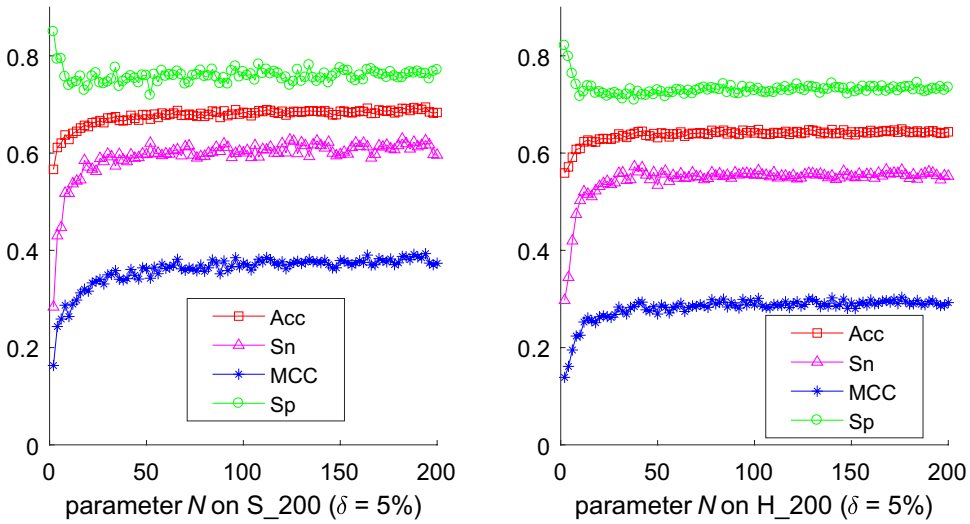

Fig. 3 The impacts of parameter N on performance

3.2 The Impacts of Parameter *N* on Performance

Similarly, we also investigate the impacts of parameter N on the performance in terms of Sn, Sp, Acc and MCC. The dataset S_628 and the dataset H_990 are utilized to train the models, where the parameter δ is fixed on 5% and the parameter N is varied from 2 to 200 with a step length of 2. These trained models are applied to the two independent testing datasets S_200 and H_200, respectively. After repetition for twenty times experiments, the average values are recorded. The detailed changes of the average values on the two independent testing datasets are depicted in Fig. 3.

As illustrated in Fig. 3, the evaluation indexes are not exhibiting a competitive result for the first several parameter N. As described in the previous introduction, parameter N determines the number of base-classifiers. A smaller parameter N can only try to construct a smaller number of base-classifiers, which limits the ability of ensemble learning. With the increase of the parameter N, the number of base-classifiers also increased, and the values of Acc, Sn, MCC and Sp gradually present a competitive tendency. This could be attributed to the fact that the proposed model is an ensemble solution which effectively improves the predicted performance. Nevertheless, a limited number of base-classifiers are always better. Besides, for different datasets, the limitations are different. When the parameter reaches some reasonable limits, the evaluation indexes are not significantly changed.

3.3 Comparisons with State-of-the-Art Models

To verify the effectiveness of iPseU-Layer in terms of Acc, Sn, Sp and MCC, this section makes some comparisons with some state-of-the-art models: iPseU-CNN [3],

iRNA-PseU[20] and PseUI [21]. The comparative analysis is mainly conducted from two aspects: cross-validation tests and independent test. The cross-validation tests are focused on the comparative analysis on the three training datasets including S_628, H_990 and M_944. The independent test is aimed to evaluate the performance through two independent testing datasets S_200 and H_200.

For the three training datasets, we first apply the technique of five folds cross-validation test to evaluate the four evaluation indexes. Apparently, the technique divides a dataset into five folds for cross-validation. Onefold is reserved for the testing purpose, while the remaining four folds are used to train a particular model. This is a five-time recursive process in which every fold is tested once. The average values of Acc, Sn, Sp and MCC are recorded and presented in Table 3. During the experiments, the two parameters are simply preset as $N = 50$ and $\delta = 5\%$, respectively. From the comparison results presented in Table 3, we can obverse that: (1) for dataset S_628, the accuracy, sensitivity, specificity, and MCC of iPseU-Layer are improved by 21.19%, 20.32%, 23.31%, and 0.42, respectively; (2) for dataset H_990, the proposed iPseU-Layer model can improve the evaluation indexes by 13.02%, 6.18%, 19.44%, and 0.26 in terms of accuracy, sensitivity, specificity, and MCC, respectively; (3) the improvements of the proposed iPseU-Layer model on dataset M_944 are 8.27%, 11.48%, and 0.16 in terms of accuracy, specificity, and MCC, respectively.

Furthermore, as the state-of-the-art models PseUI and iRNA-PseU also use the Jackknife cross-validation test to evaluate their performance in terms of the four evaluation indexes, we also make a comparative analysis with PseUI and iRNA-PseU through the Jackknife cross-validation technique. During the jackknife test, each sample in the benchmark datasets is in turn singled out as an independent test sample and the rest samples are used to train the models.

Table 3 The comparisons on three training datasets via fivefold cross-validation test

Datasets	Predictors	Acc(%)	Sn(%)	Sp(%)	MCC
S_628	iPseU-CNN	68.15	66.36	70.45	0.37
	PseUI	65.13	62.74	67.52	0.30
	iRNA-PseU	64.49	64.65	64.33	0.29
	iPseU-Layer	89.34	84.68	93.76	0.79
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	61.01	59.80	0.21
	iPseU-Layer	79.70	71.18	88.22	0.60
M_944	iPseU-CNN	71.81	74.79	69.11	0.44
	PseUI	70.44	79.87	70.34	0.41
	iRNA-PseU	69.07	73.31	64.83	0.38
	iPseU-Layer	80.08	77.92	81.82	0.60

The values of the four evaluation indexes can be found in Table 4, where the two parameters in iPseU-Layer are simply preset as $N = 50$ and $\delta = 5\%$. The results show that the proposed iPseU-Layer model is superior on these training datasets.

For the two independent testing datasets H_200 and S_200, we use H_990 and S_628 to train the proposed iPseU-Layer models, respectively. The values of two parameters N and δ are determined using the technique of grid search. Through applying the trained model to the two independent testing datasets, the four evaluation indexes are calculated. The detailed comparison results can be found in Table 5. As shown in Table 5, it can be observed that: (1) the proposed iPseU-Layer model can obtain the four evaluation indexes for H_200 by 71.00%, 63.00%, 79.00% and 0.43 in terms of accuracy, sensitivity, specificity, and MCC, respectively; (2) for dataset S_200, the accuracy, sensitivity, specificity, and MCC of the proposed iPseU-Layer model are 72.50%, 68.00%, 77.00% and 0.45, respectively. The proposed iPseU-Layer model is not performed better than the state-of-the-art iPseU-CNN model on all evaluation indexes. However, it should be pointed out that the results are similar; meanwhile, the proposed iPseU-Layer model has less hyper-parameters than iPseU-CNN model, since the iPseU-CNN model is a

Table 4 The comparisons on three training datasets via Jackknife cross-validation test

Datasets	Predictors	Acc(%)	Sn(%)	Sp(%)	MCC
S_628	PseUI	66.56	62.10	71.02	0.33
	iRNA-PseU	64.49	64.65	64.33	0.29
	iPseU-Layer	89.80	84.71	94.90	0.80
H_990	PseUI	64.24	64.85	63.64	0.28
	iRNA-PseU	60.40	61.01	59.80	0.21
	iPseU-Layer	78.79	69.70	87.88	0.59
M_944	PseUI	70.44	74.58	66.31	0.41
	iRNA-PseU	69.07	73.31	64.83	0.38
	iPseU-Layer	81.88	77.54	86.23	0.64

Table 5 The comparisons on independent testing datasets

Datasets	Predictors	Acc(%)	Sn(%)	Sp(%)	MCC
S_200	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
	iPseU-Layer	72.50	68.00	77.00	0.45
H_200	iPseU-CNN	69.00	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
	iPseU-Layer	71.00	63.00	79.00	0.43

deep-learning-based solution, where there are more hyper-parameters needed to determine.

From the above descriptions, we can get two conclusions: (1) for the cross-validation tests, the proposed iPseU-Layer model makes an improvement on all the training datasets in terms of the evaluation indexes; (2) for the independent test, the proposed iPseU-Layer model is also achieved a similar result with some deep-learning-based model.

4 Conclusions

This study proposes an iPseU-Layer model to identify RNA pseudouridine sites. The proposed iPseU-Layer model is a layered ensemble model, which predominantly consists of three different machine learning layers including: feature selection layer, feature extraction and fusion layer, and prediction layer. Each layer has its own function. Based on these machine learning components, the layered ensemble model iPseU-Layer is established to identify pseudouridine sites. To verify our model, the experimental studies are conducted from two aspects: cross-validation tests and independent test. For the cross-validation tests, the proposed model achieves an improvement prediction performance comparing with the current state-of-the-art models in terms of sensitivity, specificity, accuracy and Matthews correlation coefficient for the prediction of RNA Pseudouridine. For the independent test, the proposed iPseU-Layer model can also achieve a competitive result. In conclusion, the findings of this study highlight that the proposed iPseU-Layer model can serve as a putative potential tool for the pseudouridine site prediction of RNA.

In our future work, it deserves to highlight three aspects regarding the proposed iPseU-Layer model. (1) there are two parameters N and δ. During the experiments, we use the technique of grid search to confirm the values of the two parameters. How to quickly and effectively determine the optimal parameters is a meaningful topic. (2) there may be some over-fitting problems in iPseU-Layer model. The performance on independent testing datasets can be further improved, when the problems are resolved or weaken. (3) through a user-friendly and publicly accessible web-server to develop a useful prediction method or computational tool is a prevalent research direction. We anticipate that a web-server for the proposed iPseU-Layer model will be provided in the future.

Acknowledgements This work was supported by the Research Foundation for Advanced Talents (Nos. 2019BS007, 31401204) of Henan University of Technology and the National Natural Science Foundation of China under Grants (Nos. 61673082, 61773352).

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Ge J, Yu YT (2013) RNA pseudouridylation: new insights into an old modification. Trends Biochem Sci 38(4):210–218. https://doi.org/10.1016/j.tibs.2013.01.002
2. Hudson GA, Bloomingdale RJ, Znosko BM (2013) Thermodynamic contribution and nearest-neighbor parameters of pseudouridine-adenosine base pairs in oligoribonucleotides. Rna 19(11):1474–1482. https://doi.org/10.1261/rna.039610.113
3. Tahir M, Tayara H, Chong KT (2019) iPseU-CNN: identifying RNA pseudouridine sites using convolutional neural networks. Mol Ther Nucl Acids 16:463–470. https://doi.org/10.1016/j.omtn.2019.03.010
4. Reddy R, Busch H (1998) Small nuclear RNAs: RNA sequences, structure, and modifications. Structure and function of major and minor small nuclear ribonucleoprotein particles. Springer, Berlin, pp 1–37
5. Andrew TY, Ge J, Yu YT (2011) Pseudouridines in spliceosomal snRNAs. Protein Cell 2(9):712–725. https://doi.org/10.1007/s13238-011-1087-1
6. Wu G, Yu AT, Kantartzis A et al (2011) Functions and mechanisms of spliceosomal small nuclear RNA pseudouridylation. Wires Rna 2(4):571–581. https://doi.org/10.1002/wrna.77
7. Maden BEH (1990) The numerous modified nucleotides in eukaryotic ribosomal RNA. Prog Nucl Acid Res 39:241–303. https://doi.org/10.1016/S0079-6603(08)60629-7
8. Schattner P, Barberan-soler S, Lowe TM (2006) A computational screen for mammalian pseudouridylation guide H/ACA RNAs. Rna 12(1):15–25. https://doi.org/10.1261/rna.2210406
9. Grosjean H, Sprinzl M, Steinberg S (1995) Posttranscriptionally modified nucleosides in transfer RNA: their locations and frequencies. Biochimie 77(1–2):139–141. https://doi.org/10.1016/0300-9084(96)88117-X
10. Sprinzl M, Horn C, Brown M et al (1998) Compilation of tRNA sequences and sequences of tRNA genes. Nucleic Acids Res 26(1):148–153. https://doi.org/10.1093/nar/26.1.148
11. Hopper AK, Phizicky EM (2003) tRNA transfers to the limelight. Genes Dev 17(2):162–180. https://doi.org/10.1101/gad.1049103
12. Karijolich J, Yu YT (2015) The new era of RNA modification. Rna 21(4):659–660. https://doi.org/10.1261/rna.049650.115
13. Karijolich J, Yu YT (2011) Converting nonsense codons into sense codons by targeted pseudouridylation. Nature 474(7351):395–398. https://doi.org/10.1038/nature10165
14. Carlile TM, Rojas-Duran MF, Zinshteyn B et al (2014) Pseudouridine profiling reveals regulated mRNA pseudouridylation in yeast and human cells. Nature 515(7525):143–146. https://doi.org/10.1038/nature13802
15. Lovejoy AF, Riordan DP, Brown PO (2014) Transcriptome-wide mapping of pseudouridines: pseudouridine synthases modify specific mRNAs in *S.* cerevisiae. PLoS One 9(10):e110799. https://doi.org/10.1371/journal.pone.0110799
16. Schwartz S, Bernstein DA, Mumbach MR et al (2014) Transcriptome-wide mapping reveals widespread dynamic-regulated pseudouridylation of ncRNA and mRNA. Cell 159(1):148–162. https://doi.org/10.1016/j.cell.2014.08.028
17. Chen W, Feng P, Tang H et al (2016) Identifying 2'-O-methylationation sites by integrating nucleotide chemical properties and

nucleotide compositions. Genomics 107(6):255–258. https://doi.org/10.1016/j.ygeno.2016.05.003
18. Sun WJ, Li JH, Liu S et al (2016) RMBase: a resource for decoding the landscape of RNA modifications from high-throughput sequencing data. Nucleic Acids Res 44(D1):D259–D265. https://doi.org/10.1093/nar/gkv1036
19. Li YH, Zhang G, Cui Q (2015) PPUS: a web server to predict PUS-specific pseudouridine sites. Bioinformatics 31(20):3362–3364. https://doi.org/10.1093/bioinformatics/btv366
20. Chen W, Tang H, Ye J et al (2016) iRNA-PseU: identifying RNA pseudouridine sites. Mol Ther Nucl Acids 5:e332. https://doi.org/10.1038/mtna.2016.37
21. He J, Fang T, Zhang Z et al (2018) PseUI: pseudouridine sites identification based on RNA sequence information. BMC Bioinform 19(1):306. https://doi.org/10.1186/s12859-018-2321-0
22. Liu K, Chen W, Lin H (2020) XG-PseU: an eXtreme gradient boosting based method for identifying pseudouridine sites. Mol Genet Genomics 295(1):13–21. https://doi.org/10.1007/s00438-019-01600-9
23. Dou L, Li X, Ding H et al (2020) Is there any sequence feature in the RNA pseudouridine modification prediction problem? Mol Ther Nucl Acids 19:293–303. https://doi.org/10.1016/j.omtn.2019.11.014
24. Jia J, Liu Z, Xiao X et al (2015) iPPI-Esml: an ensemble classifier for identifying the interactions of proteins by incorporating their physicochemical properties and wavelet transforms into PseAAC. J Theor Biol 377:47–56. https://doi.org/10.1016/j.jtbi.2015.04.011
25. Jia J, Liu Z, Xiao X et al (2016) pSuc-Lys: predict lysine succinylation sites in proteins with PseAAC and ensemble random forest approach. J Theor Biol 394:223–230. https://doi.org/10.1016/j.jtbi.2016.01.020
26. Jia C, Zuo Y (2017) S-SulfPred: a sensitive predictor to capture S-sulfenylation sites based on a resampling one-sided selection undersampling-synthetic minority oversampling technique. J Theor Biol 422:84–89. https://doi.org/10.1016/j.jtbi.2017.03.031
27. Chen W, Feng P, Yang H et al (2018) iRNA-3typeA: identifying three types of modification at RNA's adenosine sites. Mol Ther Nucl Acids 11:468–474. https://doi.org/10.1016/j.omtn.2018.03.012
28. Cheng X, Xiao X, Chou KC (2018) pLoc-mEuk: predict subcellular localization of multi-label eukaryotic proteins by extracting the key GO information into general PseAAC. Genomics 110(1):50–58. https://doi.org/10.1016/j.ygeno.2017.08.005
29. Cheng X, Lin WZ, Xiao X et al (2019) pLoc_bal-mAnimal: predict subcellular localization of animal proteins by balancing training dataset and PseAAC. Bioinformatics 35(3):398–406. https://doi.org/10.1093/bioinformatics/bty628
30. Feng P, Yang H, Ding H et al (2019) iDNA6mA-PseKNC: identifying DNA N6-methyladenosine sites by incorporating nucleotide physicochemical properties into PseKNC. Genomics 111(1):96–102. https://doi.org/10.1016/j.ygeno.2018.01.005
31. Cheng X, Xiao X, Chou KC (2018) pLoc-mGneg: predict subcellular localization of gram-negative bacterial proteins by deep gene ontology learning via general PseAAC. Genomics 110(4):231–239. https://doi.org/10.1016/j.ygeno.2017.10.002
32. Liu B, Li K, Huang DS et al (2018) iEnhancer-EL: identifying enhancers and their strength with ensemble learning approach. Bioinformatics 34(22):3835–3842. https://doi.org/10.1093/bioinformatics/bty458
33. Liu B, Weng F, Huang DS et al (2018) iRO-3wPseKNC: identify DNA replication origins by three-window-based PseKNC. Bioinformatics 34(18):3086–3093. https://doi.org/10.1093/bioinformatics/bty312
34. Su ZD, Huang Y, Zhang ZY et al (2018) iLoc-lncRNA: predict the subcellular location of lncRNAs by incorporating octamer composition into general PseKNC. Bioinformatics 34(24):4196–4204. https://doi.org/10.1093/bioinformatics/bty508
35. Chen Z, Zhao P, Li F et al (2019) iLearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA, RNA and protein sequence data. Brief Bioinform. https://doi.org/10.1093/bib/bbz041
36. Hall MA (1998) Correlation-based feature subset selection for machine learning. University of Waikato, Hamilton
37. Shi H (2007) Best-first decision tree learning. The University of Waikato, Hamilton
38. Jia J, Liu Z, Xiao X et al (2016) iCar-PseCp: identify carbonylation sites in proteins by Monte Carlo sampling and incorporating sequence coupled effects into general PseAAC. Oncotarget 7(23):34558. https://doi.org/10.18632/oncotarget.9148
39. Jia J, Liu Z, Xiao X et al (2016) Identification of protein-protein binding sites by incorporating the physicochemical properties and stationary wavelet transforms into pseudo amino acid composition. J Biomol Struct Dyn 34(9):1946–1961. https://doi.org/10.1080/07391102.2015.1095116
40. Jia J, Liu Z, Xiao X et al (2016) iPPBS-Opt: a sequence-based ensemble classifier for identifying protein-protein binding sites by optimizing imbalanced training datasets. Molecules 21(1):95. https://doi.org/10.3390/molecules21010095
41. Jia J, Liu Z, Xiao X et al (2016) iSuc-PseOpt: identifying lysine succinylation sites in proteins by incorporating sequence-coupling effects into pseudo components and optimizing imbalanced training dataset. Anal Biochem 497:48–56. https://doi.org/10.1016/j.ab.2015.12.009
42. Jia J, Zhang L, Liu Z et al (2016) pSumo-CD: predicting sumoylation sites in proteins with covariance discriminant algorithm by incorporating sequence-coupled effects into general PseAAC. Bioinformatics 32(20):3133–3141. https://doi.org/10.1093/bioinformatics/btw387
43. Chen W, Feng PM, Lin H et al (2013) iRSpot-PseDNC: identify recombination spots with pseudo dinucleotide composition. Nucleic Acids Res 41(6):e68–e68. https://doi.org/10.1093/nar/gks1450
44. Lin H, Deng EZ, Ding H et al (2014) iPro54-PseKNC: a sequence-based predictor for identifying sigma-54 promoters in prokaryote with pseudo k-tuple nucleotide composition. Nucleic Acids Res 42(21):12961–12972. https://doi.org/10.1093/nar/gku1019
45. Liu B, Liu F, Wang X et al (2015) Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. Nucleic Acids Res 43(W1):W65–W71. https://doi.org/10.1038/mtna.2016.37
46. Liu B, Wang S, Long R et al (2017) iRSpot-EL: identify recombination spots with an ensemble learning approach. Bioinformatics 33(1):35–41. https://doi.org/10.1093/bioinformatics/btw539
47. Liu B, Wu H, Chou KC (2017) Pse-in-one 2.0: an improved package of web servers for generating various modes of pseudo components of DNA, RNA, and protein sequences. Nat Sci 9(04):67. https://doi.org/10.4236/ns.2017.94007
48. Liu B, Yang F, Chou KC (2017) 2L-piRNA: a two-layer ensemble classifier for identifying piwi-interacting RNAs and their function. Mol Ther Nucl Acids 7:267–277. https://doi.org/10.1016/j.omtn.2017.04.008
49. Qiu WR, Xiao X, Chou KC (2014) iRSpot-TNCPseAAC: identify recombination spots with trinucleotide composition and pseudo amino acid components. Int J Mol Sci 15(2):1746–1766. https://doi.org/10.3390/ijms15021746
50. Chou KC (2015) Impacts of bioinformatics to medicinal chemistry. Med Chem 11(3):218–234. https://doi.org/10.2174/1573406411666141229162834
51. Xiao X, Ye HX, Liu Z et al (2016) iROS-gPseKNC: predicting replication origin sites in DNA by incorporating dinucleotide position-specific propensity into general pseudo nucleotide

composition. Oncotarget 7(23):34180. https://doi.org/10.18632/oncotarget.9057
52. Feng P, Ding H, Yang H et al (2017) iRNA-PseColl: identifying the occurrence sites of different RNA modifications by incorporating collective effects of nucleotides into PseKNC. Mol Ther Nucl Acids 7:155–163. https://doi.org/10.1016/j.omtn.2017.03.006
53. Yang H, Qiu WR, Liu G et al (2018) iRSpot-Pse6NC: identifying recombination spots in *S*accharomyces cerevisiae by incorporating hexamer composition into general PseKNC. Int J Biol Sci 14(8):883. https://doi.org/10.7150/ijbs.24616
54. Song J, Wang Y, Li F et al (2019) iProt-Sub: a comprehensive package for accurately mapping and predicting protease-specific substrates and cleavage sites. Brief Bioinform 20(2):638–658. https://doi.org/10.1093/bib/bby028
55. Chou KC (2001) Prediction of signal peptides using scaled window. Peptides 22(12):1973–1979. https://doi.org/10.1016/S0196-9781(01)00540-X
56. Chou KC (2001) Using subsite coupling to predict signal peptides. Protein Eng 14(2):75–79. https://doi.org/10.1093/protein/14.2.75