PLOS ONE

RESEARCH ARTICLE

iterb-PPse: Identification of transcriptional terminators in bacterial by incorporating nucleotide properties into PseKNC

Yongxian Fan*, Wanru Wang, Qingqi Zhu

School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China

* yongxian.fan@gmail.com

Citation: Fan Y, Wang W, Zhu Q (2020) iterb-PPse: Identification of transcriptional terminators in bacterial by incorporating nucleotide properties into PseKNC. PLoS ONE 15(5): e0228479. https://doi.org/10.1371/journal.pone.0228479

Editor: Y-h. Taguchi, Chuo University, JAPAN

Received: January 15, 2020

Accepted: May 1, 2020

Published: May 15, 2020

Data Availability Statement: All relevant data are within the manuscript and its Supporting Information files.

Funding: This work was supported by the National Natural Science Foundation of China (Grant NO. 61762026, 61462018) to YXF, the Guangxi Natural Science Foundation (Grant NO. 2017GXNSFAA198278, 2016GXNSFAA380043) to YXF, the Innovation Project of GUET Graduate Education (Grant NO. 2018YJCX47, 2019YCXS056) to YXF, the Guangxi Colleges and Universities Key Laboratory of Intelligent Processing of Computer Images and Graphics

Abstract

Terminator is a DNA sequence that gives the RNA polymerase the transcriptional termination signal. Identifying terminators correctly can optimize the genome annotation, more importantly, it has considerable application value in disease diagnosis and therapies. However, accurate prediction methods are deficient and in urgent need. Therefore, we proposed a prediction method "iterb-PPse" for terminators by incorporating 47 nucleotide properties into PseKNC-I and PseKNC-II and utilizing Extreme Gradient Boosting to predict terminators based on *Escherichia coli* and *Bacillus subtilis*. Combing with the preceding methods, we employed three new feature extraction methods K-pwm, Base-content, Nucleotidepro to formulate raw samples. The two-step method was applied to select features. When identifying terminators based on optimized features, we compared five single models as well as 16 ensemble models. As a result, the accuracy of our method on benchmark dataset achieved 99.88%, higher than the existing state-of-the-art predictor iTerm-PseKNC in 100 times five-fold cross-validation test. Its prediction accuracy for two independent datasets reached 94.24% and 99.45% respectively. For the convenience of users, we developed a software on the basis of "iterb-PPse" with the same name. The open software and source code of "iterb-PPse" are available at https://github.com/Sarahyouzi/iterb-PPse.

1 Introduction

DNA transcription is an important step in the inheritance of genetic information and terminators control the termination of transcription which exists in sequences that have been transcribed. When transcription, the terminator will give the RNA polymerase the transcriptional termination signal. Identifying terminators accurately can optimize the genome annotation, more importantly, it has great application value in disease diagnosis and therapies, so it is crucial to identify terminators. Whereas, using traditional biological experiments to identify terminators is extremely time consuming and labor intensive. Therefore, a more effective and convenient began to be applied in researches, that is, adopting machine learning to identify gene sequences.

(Grant NO. GIIP201502) to YXF and Guangxi Key Laboratory of Trusted Software (Grant NO. kx201403) to YXF. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. There was no additional external funding received for this study.

Competing interests: The authors have declared that no competing interests exist.

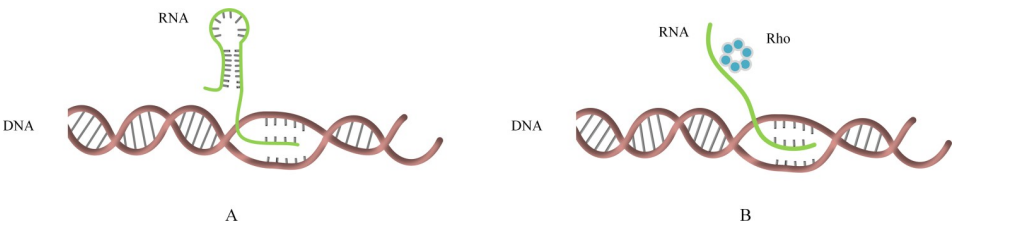

Fig 1. Transcriptional termination process. (A) The termination doesn't require Rho. The transcription stops when the RNA forms the stem loop structure. (B) The termination dependents on Rho.

https://doi.org/10.1371/journal.pone.0228479.g001

Previous research found there are two types of terminators in prokaryotes, namely Rho-dependent and Rho-independent [1], as shown in Fig 1. Although there have been a lot of studies on the prediction of terminators, most of them only focused on one kind of them. In 2004, Wan XF, Xu D et al. proposed a prediction method for Rho-independent terminators with an accuracy of 92.25%. In 2005, Michiel J. L. de Hoon et al. studied the sequence of Rho-independent terminators in *B. subtilis* [2], and the final prediction accuracy was 94%. In 2011, Magali Naville et al. conducted a research on Rho-dependent transcriptional terminators [3]. They used two published algorithms, Erpin and RNA motif, to predict terminators. The specificity and sensitivity of the final results were 95.3% and 87.8%, respectively. In 2019, Macro Di Simore et al. utilized the secondary structure of the sequence as a feature [4], the classification accuracy of the Rho-independent terminators was 67.5%. Not like the above experiments Lin Hao et al. studied the prediction of two kinds of terminators in bacterial [5],they developed a prediction tool for terminators with an accuracy of 95% in 2018.

To further improve the prediction accuracy, we obtained 503 terminator sequences, 719 non-terminator sequences of *Escherichia coli* (*E. coli*), and 425 terminator sequences, 122 non-terminator sequence of *Bacillus subtilis* (*B. subtilis*) to construct the benchmark dataset and two independent sets. Furthermore, we proposed three new feature extraction methods (K-pwm, Base-content, Nucleotidepro) to combine them with PseKNC—I [6] and PseKNC—II [5], then applied the two-step method to select effective features. In addition, we compared five single models (Support Vector Machine (SVM), Naive Bayes, Logistic Regression (LR), Decision Tree, Multi-layer Perceptron (MLP), K-Nearest Neighbor (KNN)) as well as 16 ensemble models based on AdaBoost, Bagging, Extreme Gradient Boosting (XGBoost) and Gradient Boosting Method (GBM). Finally, we proposed a prediction method "iterb-PPse" for terminators.

2 Materials and methods

As shown in the Fig 2, our study is mainly divided into the following steps [7]: (1) data collection, (2) feature extraction, (3) feature combination, (4) feature selection, (5) classification, (6) result evaluation, (7) prediction method.

2.1 Data collection

In our study, the initial datasets were obtained from http://lin-group.cn/server/iTerm-PseKNC [2], which includes 427 terminator sequences, 560 non-terminator sequences of E. coli, and 425 terminator sequences of B. subtilis. To generate reliable benchmark dataset and independent dataset, we collected another 76 terminator sequences, 159 non-terminator sequences from *E. coli K-12* genome in the database RegulonDB [8], and 122 non-terminator sequences of *B. subtilis* were gathered from database DBTBS [2, 9]. The non-terminator sequences of *E. coli* were intercepted from -100 bp to -20 bp upstream and 20 bp to 100 bp of

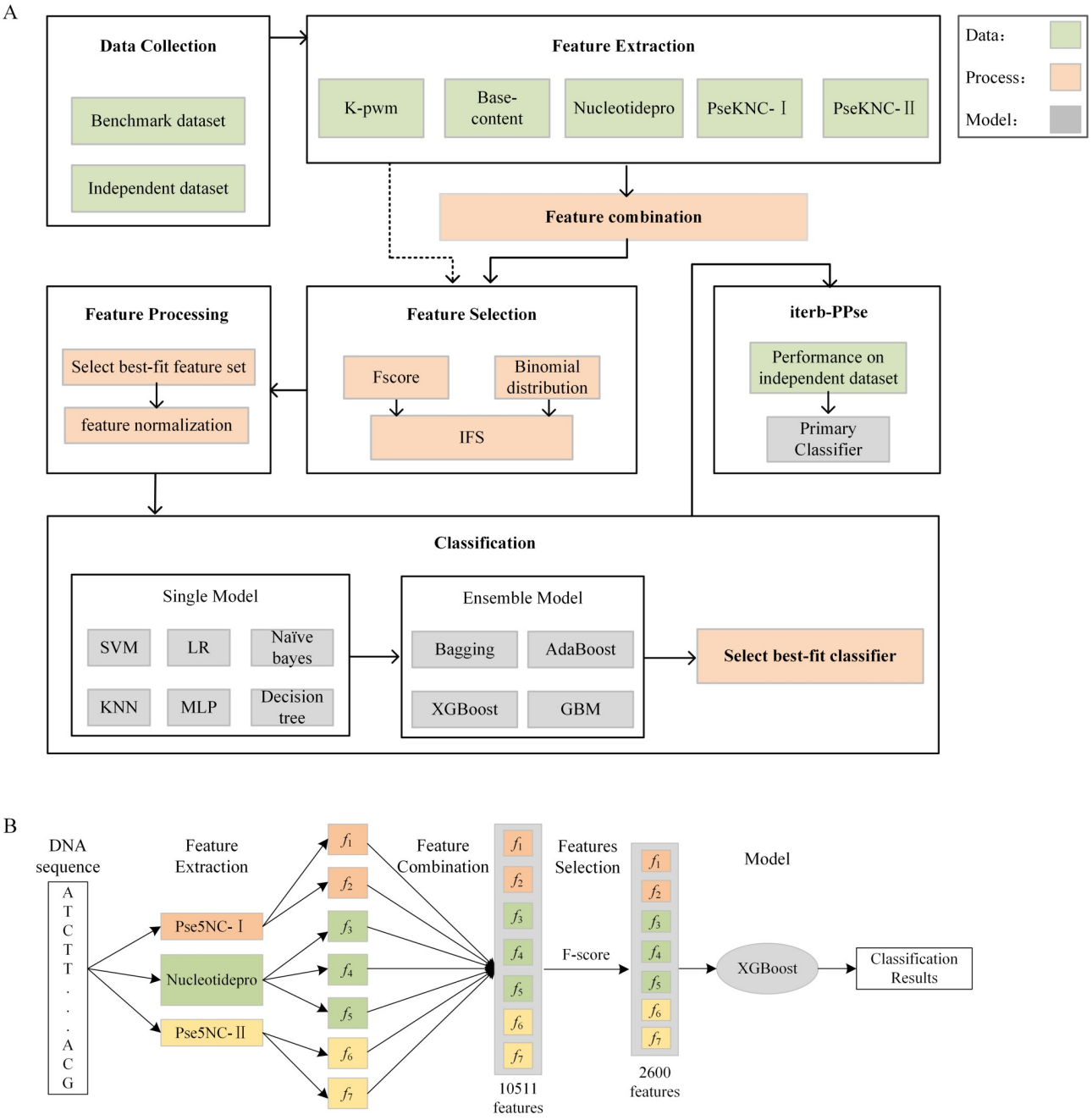

Fig 2. The overall framework. A shows main steps of our study. First step is using five extraction methods to deal datasets, then select more important features by two-step feature selection method, finally compared different models using the selected features. The "iterb-PPse" is the method we proposed to predict terminators. B illustrates the prediction process of "iterm-PPse". It extracts three features from gene sequences at first, namely Pse5NC-I, Pse5NC-II, 47 nucleotide properties. Then sort all features using F-score and select the best feature set by IFS. Finally utilizes trained XGBoost to determine whether these sequences are terminators.

https://doi.org/10.1371/journal.pone.0228479.g002

positive samples not used in the benchmark dataset. The non-terminator sequences of *B. subtilis* were intercepted from -102 bp to -20 bp upstream and 20 bp to 102 bp of positive samples. At last, we divided the collected sequences into the benchmark set and the independent dataset at a ratio of 8: 2. In order to accurately evaluate the identification accuracy of our method to different bacteria, we divided the independent test set into two. Details of the benchmark

Table 1. Benchmark dataset.

Species	Category	Number	Length
E. coli	Rho-dependent terminator	18	~50 bp
	Rho-independent terminator	385	~50 bp
	non-terminator	575	80 bp
B. subtilis	Rho-independent terminator	340	~50 bp
	non-terminator	98	82 bp

"~" represents approximately equal.

https://doi.org/10.1371/journal.pone.0228479.t001

dataset and independent sets are shown in Tables 1 and 2 of respectively. All sequences of *E. coli* and *B. subtilis* could be found in S1–S7 Tables.

2.2 Feature extraction

How to extract effective features from DNA sequences is a particularly important step. At present, the input of most machine learning methods must be numerical values rather than character sequences [10], such as decision tree, logistic regression etc. Thus, it is essential to make use of proper feature extraction methods to represent sequences.

2.2.1 K-pwm. The new feature extraction method "K-pwm" mainly employed the Position Weight Matrix [11–14], where K represents *k*-tuple nucleotides. Considering that the length of negative samples is different from that of the positive samples in the benchmark set. We made a little modification to the calculation of the final sequence score to eliminate the negative impact of sequence length. A total of 6 feature sets were obtained by using this method, namely the position weight features corresponding to $k = 1, 2, 3, 4, 5, 6$. The calculation steps are shown below.

$$p_0 = \frac{1}{4^k}, \tag{1}$$

where p_0 represents the background probability of the occurrence of *k*-tuple nucleotides.

$$p_{xi} = \frac{n_{xi}}{N_i}, \tag{2}$$

where p_{xi} indicates the probability of *k*-tuple nucleotide *x* appearing at site *i*.

$$W_{xi} = \ln\left(\frac{p_{xi}}{p_0}\right), \tag{3}$$

where W_{xi} is the element in the position weight matrix.

$$F = \frac{1}{L}\sum_i W_{xi}, \tag{4}$$

where *L* is the length of the corresponding sequence.

Table 2. Independent dataset.

Species	Category	Number	Length
E. coli	Rho-independent terminator	100	~50 bp
	non-terminator	143	80 bp
B. subtilis	Rho-independent terminator	85	~50 bp
	non-terminator	24	82 bp

https://doi.org/10.1371/journal.pone.0228479.t002

2.2.2 Base-content. Given that the rho-independent terminators are rich in GC base pairs, we extracted a set of features and collectively referred to as Base-content [15, 16]. Specifically, we mainly obtained the content features of the single nucleotide (A, C, G, T) in each DNA sequence [17, 18]. In this paper, 5 kinds of base content features (atContent, gcContent, gcSkew, atSkew, atgcRatio) [15, 16, 19–21] were took into account.

$$p_i^{A+T} = \frac{m_i^{A+T}}{m_i^{A+T+G+C}}; \tag{5}$$

$$p_i^{G+C} = \frac{m_i^{G+C}}{m_i^{A+T+G+C}}; \tag{6}$$

$$p_i^{atgRatio} = \frac{m_i^{A+T}}{m_i^{G+C}}; \tag{7}$$

$$p_i^{gcSkew} = \frac{m_i^{G-C}}{m_i^{G+C}}; \tag{8}$$

$$p_i^{atSkew} = \frac{m_i^{A-T}}{m_i^{A+T}}; \tag{9}$$

where m_i^G, m_i^C are the contents of G and C in the i-th sequence, respectively. m_i^{A+T}, m_i^{G+C}, $m_i^{A+T+G+C}$ are the contents of "A+T", "G+C" and "A+T+G+C", respectively. m_i^{A-T}, m_i^{G-C} represent the content of "A-T" and "G-C", respectively.

2.2.3 Nucleotidepro. Nucleotide properties of DNA sequences play a key role in gene regulation [22]. Therefore, we proposed a new feature extraction method "Nucleotidepro" involving 47 properties [23] not covered previously, including 3 nucleotide chemical properties [24], 32 dinucleotide physicochemical properties and 12 trinucleotide physicochemical properties.

To extract corresponding features, we employed a 47^*L dimension matrix to represent each sequence. L is the length of the corresponding sequence. As shown in the Table 3, we used 0 and 1 to represent the chemical properties of different nucleotides. Then we iterated through each sequence and assigned the values of different properties for different nucleotide to the corresponding elements in the matrix. The nucleotide properties and corresponding standard-converted values [23] for the 47 properties can be obtained from the S9 and S10 Tables.

2.2.4 PseKNC-I. PseKNC-I [6] is generally understood to mean the parallel correlation PseKNC. It combines K-tuple nucleotides components [25] with 6 physicochemical properties [22] (rise, slide, shift, twist, roll, tilt), not only considering the global or long-range sequence information, but also calculating the biochemical information of DNA sequences. The

Table 3. Corresponding values for different chemical properties.

Chemical	Category	Nucleotides	Value
Ring structure	Purine	AG	0
	Pyrimidine	CT	1
Hydrogen bond	Strong	CG	0
	Weak	AT	1
Functional group	Amino	AC	0
	Keto	GT	1

https://doi.org/10.1371/journal.pone.0228479.t003

PseKNC-I features can be obtained directly through the online tool Pse-in-one [26, 27], or run our code to process multiple sequences at the same time.

By changing the value of *K*, more features could be obtained. However, as the dimension of the feature matrix increases, it may lead to over-fitting and generate a large amount of redundant data [28]. Therefore, only three feature sets were extracted when $K = 4$, 5 and 6, respectively.

2.2.5 PseKNC-II. PseKNC-II, also known as the series correlation PseKNC [5]. PseKNC-II also calculated the K-tuple pseudo nucleotide properties, but unlike PseKNC-I, it considered the difference between properties. By changing the value of *K*, We extracted three feature sets when $K = 4$, 5, 6 respectively.

2.3 Feature combination

Each feature extraction method can extract distinctive features of the DNA sequence with different emphasis. To further optimize the prediction results, we analyzed the performance of five feature extraction methods by training XGBoost to predict terminators and selected the more effective features from each method to combine. The specific combination method will be introduced in the section **Results**.

2.4 Feature selection

Feature selection is an important data process, which could not only reduce the computation time, but also remove redundant data, and select more effective features, finally greatly improve the prediction accuracy [28].Hence, the two-step method was adopted to select features.

To present the correlation between features, the Pearson correlation coefficients were calculated to construct correlation matrix. If the two properties change in the opposite direction, it is a opposite effect. As shown in Fig 3, the features contain some redundant data, so it is necessary to utilize the two-step feature selection method [5, 17, 29].

2.4.1 Feature *sorting*. The first step is utilizing feature sorting methods. The main task of feature sort is to analyze the importance of each feature for prediction of terminators. The top features are more helpful in predicting terminators.

F-score. F-score [6] is a method for measuring the ability of a feature to distinguish between two classes. Given the training set *x*, if n^+ and n^- stand for the number of positive and negative samples, respectively. The F-score of the *i*-th feature is inferred to be:

$$\mathrm{F}_i = \frac{(\bar{x}_i^{(+)} - \bar{x}_i)^2 + (\bar{x}_i^{(-)} - \bar{x}_i)^2}{\frac{1}{n^+ - 1}\sum_{k=1}^{n^+}(\bar{x}_{k,i}^{(+)} - \bar{x}_i^{(+)})^2 + \frac{1}{n^- - 1}\sum_{k=1}^{n^-}(\bar{x}_{k,i}^{(-)} - \bar{x}_i^{(-)})^2}, \tag{10}$$

where $\bar{x}_i$, $\bar{x}_i^{(+)}$, *i*, $\bar{x}_{\mathrm{i}}^{(-)}$ represent the average of the *i*-th feature in all samples, positive samples, and negative samples, respectively. $\bar{x}_{k,i}^{(+)}$ is the *i*-th feature of the *k*-th positive sample, $\bar{x}_{k,i}^{(-)}$ is the *i*-th, feature of the *k*-th negative sample. The larger the F-score, the more distinctive this feature. The existing feature sorting toolkit fselect.py can be obtained from http://www.csie.ntu.edu.tw/~cjlin/.

Binomial distribution. As well as, binomial distribution [27, 30] were used to sort the features [31, 32]. The specific process is as follows:

$$q_i = m_i / M, \tag{11}$$

where q_{i} is the prior probability, m_i represents the number of *i*-th samples ($i = 1,2$ indicates

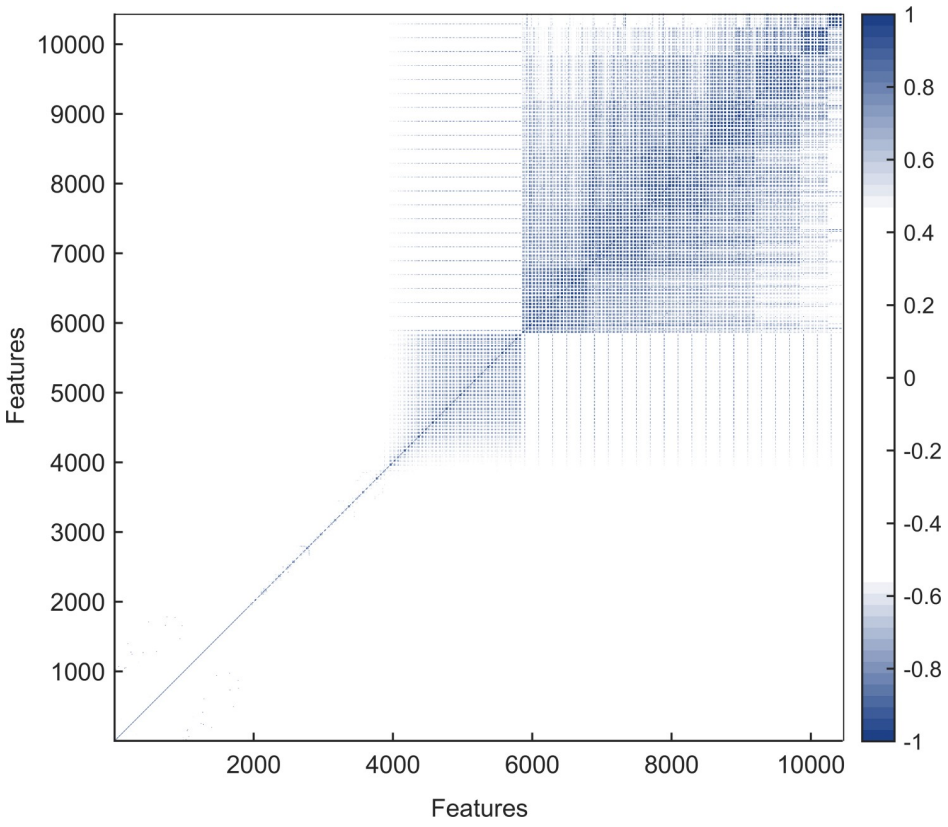

Fig 3. Correlation of all features. The correlation between all features obtained by calculating the Pearson correlation coefficient.

https://doi.org/10.1371/journal.pone.0228479.g003

positive and negative respectively), and M is the number of all samples.

$$P(n_{ij}) = \sum_{m=n_{ij}}^{N_j} \frac{N_j!}{m!(N_j-m)} q_i^m (1-q_i)^{N_j-m}, \tag{12}$$

where n_{ij} represents the times of the j-th feature appears in the i-th samples, and N_j is the times of the j-th feature appears in all samples.

$$CL_{ij} = 1 - P(n_{ij}). \tag{13}$$

$$CL_j = \max(CL_{i1}, CL_{i2}), \tag{14}$$

where CL_j is the confidence level, the higher the confidence level, the higher the credibility. Therefore, the confidence level of each feature was ranked in descending order according to the corresponding CL_j.

2.4.2 Incremental feature selection. The second step is Incremental Feature Selection (IFS) [33]. It uses a feature as the training set at first, then adds the sorted features to the training set one by one, finally finds the number of features corresponding to highest classification accuracy.

2.5 Data normalization

It is necessary to process the data into the required format before conducting experiments, such as normalized. Our study first employed function "mapminmax" for data normalization, its purpose is to make data limited in a certain range, such as [0, 1] or [-1, 1], thereby eliminating singular sample data leading to negative impact.

In addition, it should be noted that data normalization is not applicable to all classification algorithms, and sometimes it may lead to a decrease in accuracy. Data normalization applies to optimization problems like AdaBoost, Support Vector Machine, Logistic regression, K-Nearest Neighbor but not probability models such as decision tree.

2.6 Model

2.6.1 Single model. *SVM*. The principle of SVM [34] is using a series of kernel functions to map the initial feature sets to high-dimensional space, and then finding a hyperplane in high-dimensional space to classify samples. The SVM pattern classification and regression package LIBSVM is available at https://www.csie.ntu.edu.tw/~cjlin/libsvm/oldfiles/.

Naïve Bayes. Naïve Bayes uses the prior probability of an object to calculate posterior probability belongs to one of the categories by using the Bayes formula. The object belongs to the class whose corresponding posterior probability is the greatest.

LR. LR usually utilizes known independent variables to fit the model $y = w^{T}x+b$. Then, predict the value of a discrete dependent variable (whether true or false). Besides its output value should be 0~1, so it is very suitable for dealing with the two-class problem.

KNN. The main principle of the K-Nearest Neighbor is to find k samples closest to the sample to be classified. Then count which category has the largest number of samples, and the current sample belongs to this category.

Decision Tree. Decision Tree is based on the tree structure which usually formed by a root node, several leaf nodes and some branches. A node represents an attribute, each branch indicates an option, and each leaf represents a classification result. The principle is to construct a tree with the maximum information gain as a criterion, combine various situations through a tree structure, and then employ it to predict new samples.

MLP. MLP with multiple neuron layers, also be known as Deep Neural Networks. Similar to a common neural network, it has an input layer, implicit layers, an output layer, and optimizes the model by information transfer between layers.

2.6.2 Ensemble model. *Bagging*. Bagging's main principle is to integrate multiple base models of the same kind in order to obtain better learning and generalization performance. Single model SVM, Naïve Bayes, Decision Tree [35] and LR were employed as the base classifier respectively. First, the training set is separated into multiple training subsets to train different models. Then make final decision through the voting method.

AdaBoost. AdaBoost is a typical iterative algorithm whose core idea is to train different classifiers (weak classifiers) using the same training set. It adjusts the weight based on whether the sample in each training set is correct and the accuracy of the last round. Then, the modified weights are sent to next layer for training, the classifier obtained by each training are integrated as the ultimate classifier. In our study, Decision Tree, SVM, LR and Naïve Bayes were mainly adopted as the weak classifier for iterative algorithm.

GBM. It finds the maximum value of a function by exploring it along the gradient direction. The gradient operator always points to the fastest growing direction. Because of the high computational complexity, the improved algorithm only uses one sample point to update the regression coefficient at a time, which greatly improves the computational complexity of the algorithm.

XGBoost. XGBoost which utilizes the cart tree that can get the predicted score as the base classifier, optimizes different trees in turn during training, adds them to the integrated classifier, and finally get the predicted scores of all trees. The scores are added together to get the classification results.

2.6.3 Parameter optimization. Before applying various models, we studied the parameters of each model and selected some more important to optimize by grid search using 100 times 5-fold cv scheme [36], as shown in Table 4.

2.7 Cross-validation test

The 5-fold cross-validation (5-fold CV) can effectively avoid over-fitting and under-learning [37], and the results obtained are more convincing. First randomly divide the dataset into 5 pieces. One of them was employed as the test set and the other four were used as training sets. The above process is repeated until each of the five datasets serves as the test set [38]. Since the datasets are randomly divided, the results are accidental. The stability of the results can be improved by performing repeatedly.

2.8 Independent test

To test the prediction performance, we utilized the independent set to test prediction performance of terminators. The initial independent sets were obtained from http://lin-group.cn/server/iTerm-PseKNC [2], containing sequences of *E. coli* and *B. subtilis*, respectively. However, both of them do not include negative samples, which result in the test results are not convincing. Therefore, we collected another 159 non-terminator sequences of *E. coli* and 122 non-terminator sequences of *B. subtilis* from database RegulonDB and DBTBS to construct two reliable independent sets.

2.9 Performance measures

For the sake of better presentation and comparison of the experiments results, we mainly calculated the following four evaluation parameters [39–41].

$$\begin{cases} \mathrm{Sn} = 1 - \dfrac{N_-^+}{N^+} & 0 \le \mathrm{Sn} \le 1 \\ \mathrm{Sp} = 1 - \dfrac{N_+^-}{N^-} & 0 \le \mathrm{Sp} \le 1 \\ \mathrm{Acc} = 1 - \dfrac{N_-^+ + N_+^-}{N^+ + N^-} & 0 \le \mathrm{Acc} \le 1 \\ \mathrm{MCC} = \dfrac{1 - \left(\dfrac{N_-^+}{N^+} + \dfrac{N^-{}_+}{N^-}\right)}{\sqrt{\left(1 + \dfrac{N_+^- - N_-^+}{N^+}\right)\left(1 + \dfrac{N_-^+ - N_+^-}{N^-}\right)}} & -1 \le \mathrm{MCC} \le 1 \end{cases}, \quad (15)$$

where N^+ represents the number of terminator sequences, and N^- is the number of non-terminator sequences, $N+$ -indicates the number of positive samples mistaken as negative samples, and $N-+$ indicates the number of negative samples mistaken as positive samples. Sn and Sp delegate the ability of the model to accurately predict samples. Acc reflects the prediction accuracy of models. MCC measures the performance of model [5] on the unbalanced benchmark dataset [42, 43].

Table 4. Parameters and the value range of parameter adjustment.

Model	Parameter	Value
SVM	*c*, *g*	$[2^{-5}, 2^{15}]$ Δ = 2, $[2^{-15}, 2^{-5}]$ Δ = 2^{-1}
LR	*c*, *solver*	[0.1, 1] Δ = 0.1 newton-cg, lbfgs, liblinear, sag
MLP	*alpha*	0.001, 0.01, 0.1, 0.5, 1, 1.5
Decision Tree	*min_sample_split*, *max_depth*	[2, 30] Δ = 2, [1, 10] Δ = 1
Bagging	*n_estimators*	[10, 1000] Δ = 50
AdaBoost	*n_estimators*, *learning_rate*	[10, 1000] Δ = 50, [0.1, 1] Δ = 0.1
GBM	*learning_rate*, *n_estimators max_depth*, *max_features*, *random_state*	[0.1, 1] Δ = 0.1, [10, 1000] Δ = 50 [1, 10] Δ = 1
XGBoost	*n_estimators*, *learning_rate*	[10, 1000] Δ = 50, [0.1, 1] Δ = 0.1

Δ represents the step size.

https://doi.org/10.1371/journal.pone.0228479.t004

In addition to the above four evaluation parameters, the ROC curve was adopted to evaluate the comprehensive performance of different method. It is a comprehensive indicator of continuous variables of sensitivity and specificity. AUC is the area below the ROC curve. Generally, the higher the value of AUC, the higher the classification accuracy [17].

3 Results and discussion

3.1 Result of feature selection

As shown in Fig 4, we compared the experimental results with and without feature selection, and drew the accuracy corresponding to different number of features after IFS. It is clear that the number of features has a great influence on the classification accuracy, and too many characteristics are bad, so it is necessary to select features. Furthermore, F-score is better than binomial distribution. Therefore, "F-score+IFS" was chose to conduct feature selection.

3.2 Comparison of different feature extraction methods

We compared the performance of different feature extraction methods by training XGBoost to predict terminators. As shown in Fig 5, PseKNC-I, PseKNC-II, k-pwm, and nucleotidepro are all effective, but the performance of base content is not ideal. Hence, the more effective features were selected to construct combined feature sets. In the end, a total of nine group features were obtained. Details of the combination method are shown in Table 5. As shown in Fig 6, Group 8 stands out in terms of Sn, Sp, MCC and Acc from other combined feature sets. Consequently, the three features Pse5NC-I, Pse5NC-II, 47 nucleotide properties were applied to formulate all samples.

3.3 Comparison of different models

To compare different methods, the above experimental process was repeated using 16 different models. What can be clearly seen in Table 6 is that the classification performance of some ensemble models is better than that of a single model. For example, the accuracy of AdaBoost (SVM) and Bagging (SVM) are significantly higher than SVM. Decision tree, AdaBoost (Decision Tree) and XGBoost perform well, but XGBoost achieved the highest prediction accuracy in all models. Hence, it is reasonable and wise to choose XGBoost as the classifier.

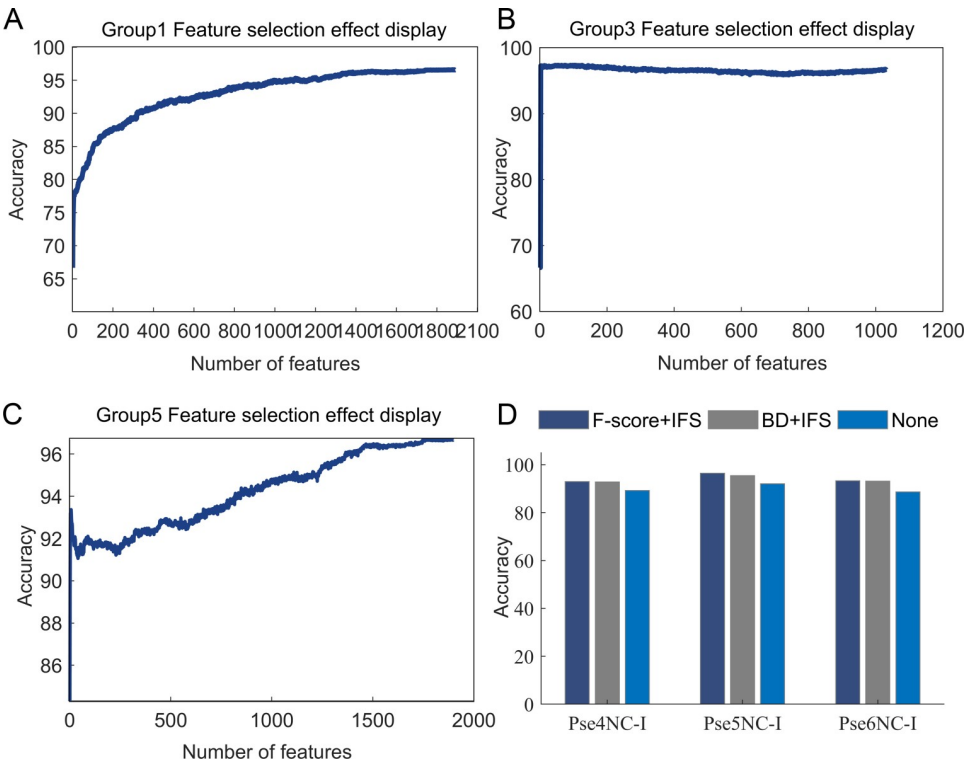

Fig 4. Performance of feature selection. (A)-(C) Relationship between the number of features and classification accuracy of three combined feature sets respectively. (D) Comparison of prediction results using three PseKNC-I features and different feature sorting methods. The combined feature set is described in detail in the next section.

https://doi.org/10.1371/journal.pone.0228479.g004

3.4 Comparison with existing state-of-the-art methods

To verify the advantage of our method "iterb-PPse", we made a comprehensive comparison with "iTerm-PseKNC" [5], the current best tool for classifying two kinds of terminators, on the benchmark dataset and two independent sets we constructed using four evaluation parameters and ROC curves, as shown in Table 7 and Fig 7. The benchmark set we utilized is exactly the same with "iTerm-PseKNC", so the comparison between the two methods is fair and objective.

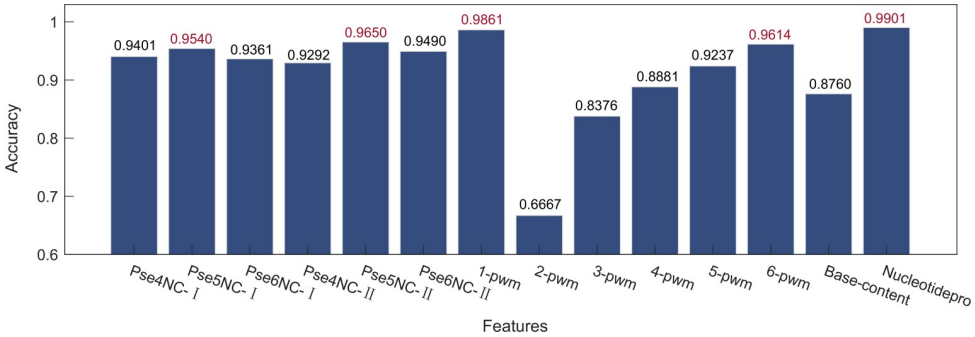

Fig 5. Prediction results using different feature extraction methods. All results are obtained after 100 times 5-fold CV. The ones marked red represent the best of each method.

https://doi.org/10.1371/journal.pone.0228479.g005

Table 5. Combination of feature extraction methods.

Combination	Method	Feature	Number
Group1	PseKNC	Pse5NC-I, Pse5NC-II	2083
Group2	K-pwm	1-pwm, 6-pwm	2
Group3	PseKNC-I	Pse5NC-I	1031
	K-pwm	1-pwm, 6-pwm	
Group4	PseKNC	Pse5NC-II	1056
	K-pwm	1-pwm, 6-pwm	
Group5	PseKNC	Pse5NC-I, Pse5NC-II	2085
	K-pwm	1-pwm, 6-pwm	
Group6	PseKNC	Pse5NC-I, Pse5NC-II	2088
	K-pwm	1-pwm, 6-pwm	
	Base-content	3 base content features	
Group7	K-pwm	1-pwm, 6-pwm	49
	Nucleotidepro	3 nucleotide chemical properties	
		32 dinucleotide physicochemical properties	
		12 trinucleotide physicochemical properties	
Group8	PseKNC	Pse5NC-I, Pse5NC-II	2600
	Nucleotidepro	3 nucleotide chemical properties	
		32 dinucleotide physicochemical properties	
		12 trinucleotide physicochemical properties	
Group9	PseKNC	Pse5NC-I, Pse5NC-II	2132
	K-pwm	1-pwm, 6-pwm	
	Nucleotidepro	3 nucleotide chemical properties	
		32 dinucleotide physicochemical properties	
		12 trinucleotide physicochemical properties	

The "Number" refers to the number of features after feature selection.

https://doi.org/10.1371/journal.pone.0228479.t005

As shown in Table 7 and Fig 7, the "iterb-PPse" performs better than the "iTerm-PseKNC" across the three datasets in Sn, Sp, MCC, Acc and AUC after 100 times 5-fold CV. Besides, the ROC curves in also show that the overall performance of our method is better. To be more precise, we improved the prediction accuracy (Acc) by 5.08%, 3.4%, 2.92% after 100 times 5-fold CV for the benchmark dataset and two independent datasets respectively.

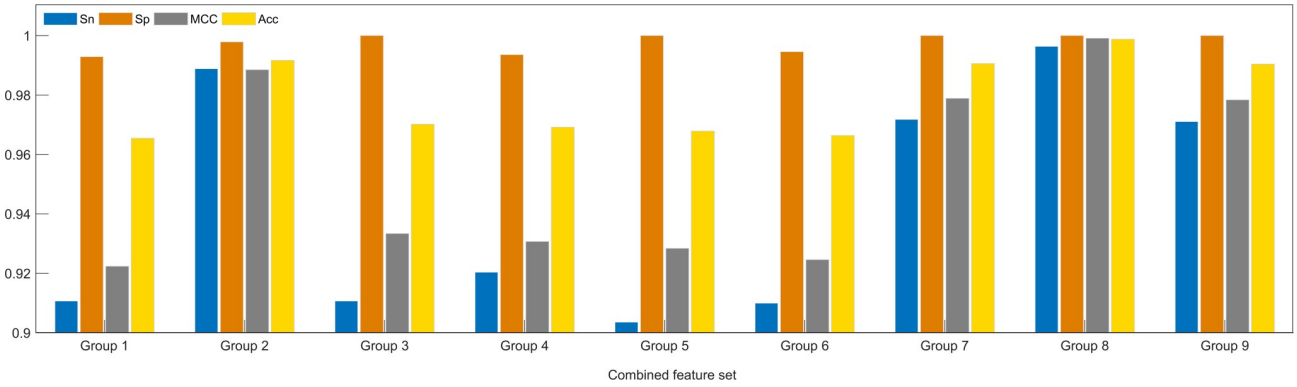

Fig 6. Classification results using different combined features. These results are obtained using XGBoost after 100 times 5-fold CV.

https://doi.org/10.1371/journal.pone.0228479.g006

Table 6. Display of all model classification results.

Model	Sn	Sp	MCC	Acc
SVM	0.9754±0.0003	1	0.9816±0.0002	0.9918±0.0001
Decision tree	0.9939±0.0012	0.9979±0.0002	0.9984±0.0002	0.9979±0.0398
LR	0.9904±0.0018	1	0.9975±0.0004	0.9967±0.0006
Naïve bayes	0.9933±0.0017	0.9935±0.0052	0.9984±0.0003	0.9978±0.0005
MLP	0.9911±0.0013	1	0.9977±0.0003	0.9970±0.0004
KNN	0.9921±0.0016	0.9994+0.0003	0.9966±0.0009	0.9970±0.0005
AdaBoost (LR)	0.9561±0.0028	1	0.9893±0.0008	0.9854±0.0010
AdaBoost (Naïve Bayes)	0.9917±0.0012	1	0.9979±0.0002	0.9972±0.0003
AdaBoost (Decision Tree)	0.9956±0.0013	0.9987±0.0005	0.9989±0.0003	0.9985±0.0004
AdaBoost (SVM)	0.9933±0.0015	0.9980±0.0004	0.9984±0.0003	0.9978±0.0004
Bagging (Decision Tree)	0.9910±0.0010	1	0.9976±0.0002	0.9969±0.0003
Bagging (SVM)	0.9840±0.0019	1	0.9959±0.0004	0.9946±0.0006
Bagging (LR)	0.9885±0.0010	1	0.9971±0.0002	0.9961±0.0003
Bagging (Naïve Bayes)	0.9931±0.0019	0.9903±0.0001	0.9983±0.0005	0.9977±0.0006
GBM	0.9921±0.0015	1	0.9980±0.0003	0.9973±0.0005
XGBoost	**0.9964±0.0023**	**1**	**0.9991±0.0005**	**0.9988±0.0007**

These results are obtained after 100 times 5-fold CV with standard error [44].

https://doi.org/10.1371/journal.pone.0228479.t006

3.5 Latest sequence prediction

In order to further evaluate iterb-PPse, we compiled two up-to-date terminator data sets [45,46] in *E. coli*. These data are from recent new sequencing methods. The details of new sequences and the corresponding recognition accuracy are shown in Table 8. The 1615 terminator sequences can be found in S8 Table. The final prediction results show that our method can identify 99.87% of terminators, proving that our method is effective and accurate.

3.6 Feature analysis

In order to analysis the feature of terminators, we further analyzed the selected 2600 features. First, we draw a heat map based on the F-score of each feature. It can be seen from the Fig 8, that there are extremely effective features among 2,600 features, for example, 18 features out of 1024 pentamer nucleotides have higher scores. Therefore, we analyzed the preferences of terminators and non-terminators for these 18 pentamers using statistical methods. As shown in the Fig 9, we plotted the distribution ratio of the pentamers in the terminator sequences and non-terminator sequences. The figure clearly shows the 16 pentamer nucleotides that often appear in the terminators, of which the preference for "TTTTT" is the most obvious.

Table 7. Comparison of "iTerm-PseKNC" and "iterb-PPse".

Dataset	Method	Sn	Sp	MCC	Acc
Benchmark dataset	iterb-PPse	**0.9964**	**1**	**0.9991**	**0.9988**
	iTerm-PseKNC	0.8545	0.9993	0.8846	0.9480
E. coli	iterb-PPse	**0.9013**	**1**	**0.8898**	**0.9424**
	iTerm-PseKNC	0.8879	0.9371	0.8166	0.9084
B. subtilis	iterb-PPse	**0.9929**	**1**	**0.9844**	**0.9945**
	iTerm-PseKNC	0.96	0.9836	0.9066	0.9653

The prediction results were obtained after 100 times 5-fold CV.

https://doi.org/10.1371/journal.pone.0228479.t007

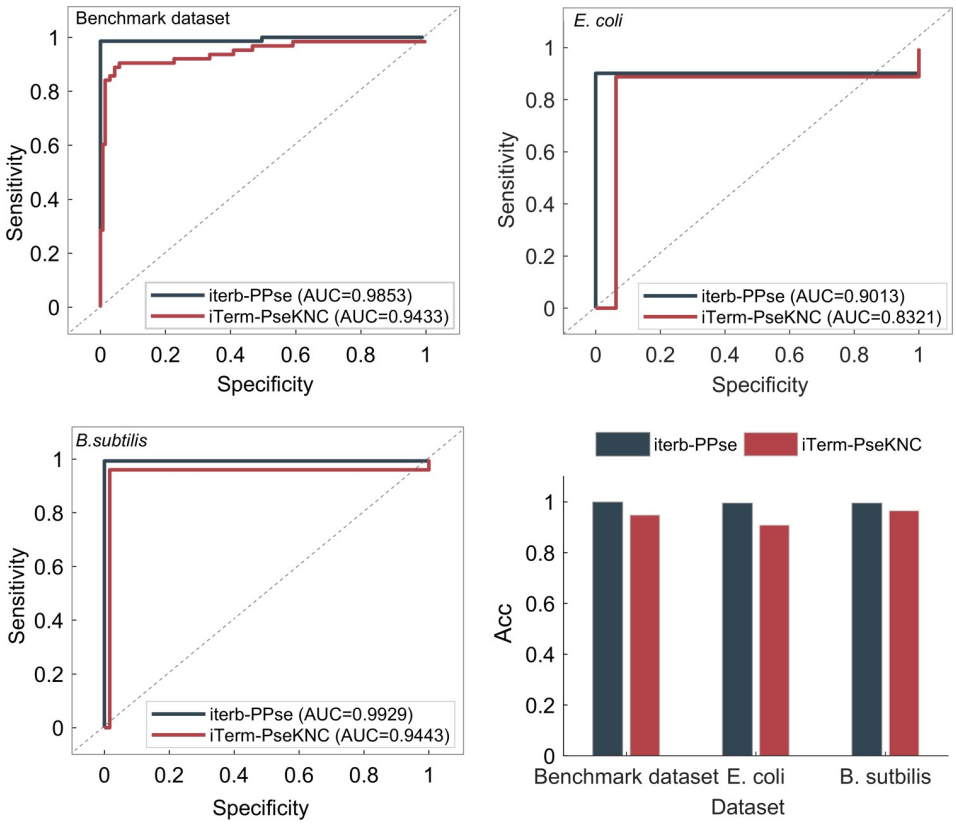

Fig 7. Comparison of "iTerm-PseKNC" and "iterb-PPse". (A)-(C) ROC curves of two methods' performance on the benchmark dataset and independent sets. (D) Prediction accuracy of two methods on different datasets.

https://doi.org/10.1371/journal.pone.0228479.g007

After that, in order to further analyze the characteristics of the terminator, we used the tool MEME [47] to analyze the motif information of 928 terminator sequences. Then we got the common motif of the terminator sequence `[AC]A[TAC]AAAAAA[AG][CG]C[CG][CG][CG] [GAC]G[GC][GC]G[CG]TTTTT A[GT][GA][CA]CTGATAAG[CG]G[CA]AG[CG]GC`. As shown in Fig 10, we drew a motif diagram of the terminator sequence. This motif corresponds to the terminator-preferred pentamer nucleotide we obtained, indicating that our experiment is effective.

3.7 Availability of software "iterb-PPse"

In addition to providing all codes of the prediction method, we developed a prediction software which could directly predict whether a DNA sequence is a terminator by simply installing it according to our software manual. The interface of the software is shown in the Fig 11.

Table 8. Sequence details and recognition accuracy.

Species	Category	Number	Identification accuracy
E. coli	Rho-dependent terminator	790	99.87%
	Rho-independent terminator	411	100%
	Terminator of undetermined classification	414	99.75%

https://doi.org/10.1371/journal.pone.0228479.t008

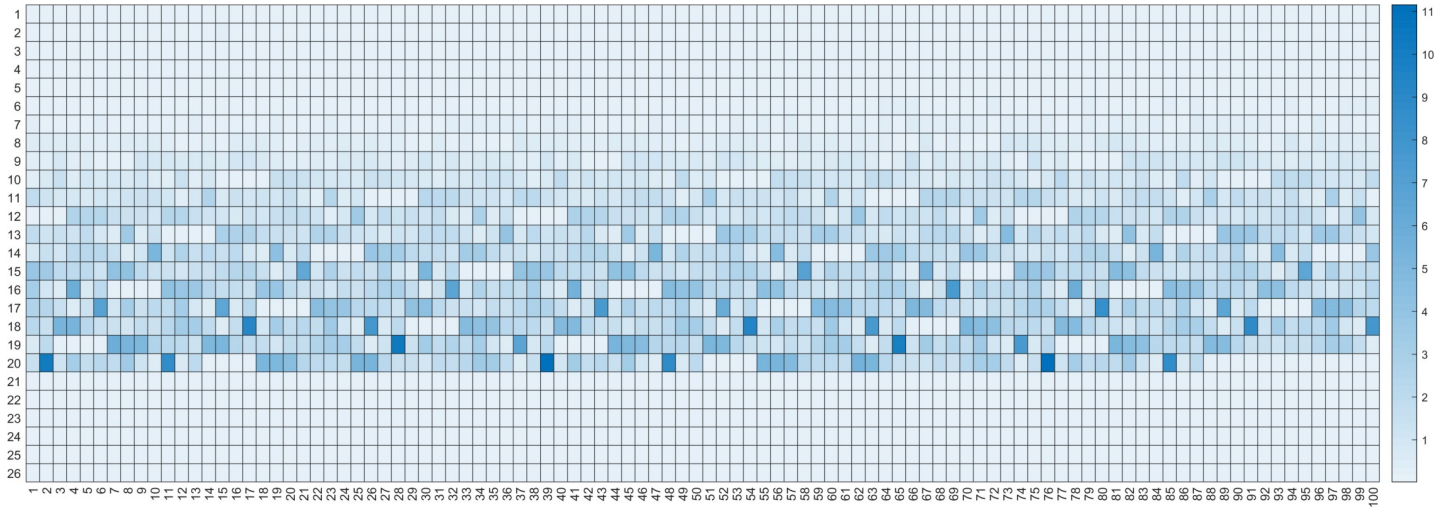

Fig 8. Heat map of each feature score. This figure shows the score of 2600 features we got using F-score.

https://doi.org/10.1371/journal.pone.0228479.g008

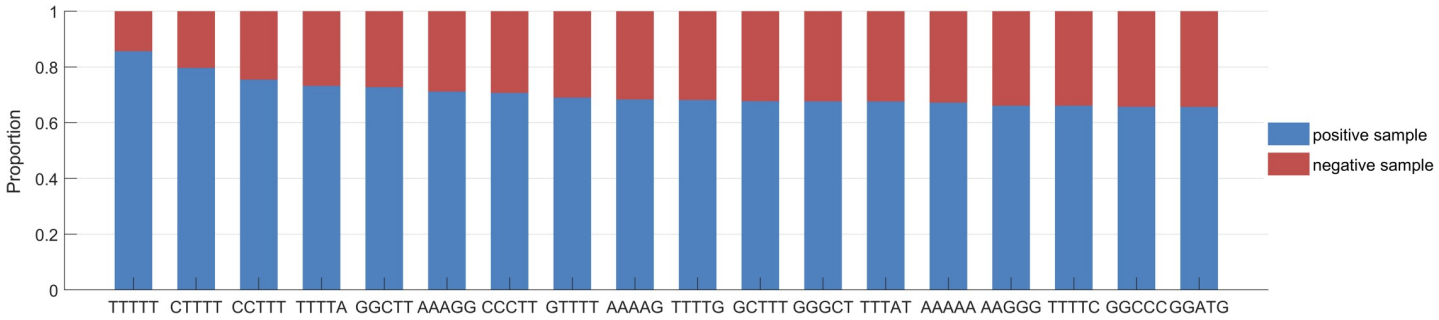

Fig 9. Pentamer nucleotide distribution. The figure shows the distribution ratio of each pentamer nucleotide in terminator sequences and non-terminator sequences.

https://doi.org/10.1371/journal.pone.0228479.g009

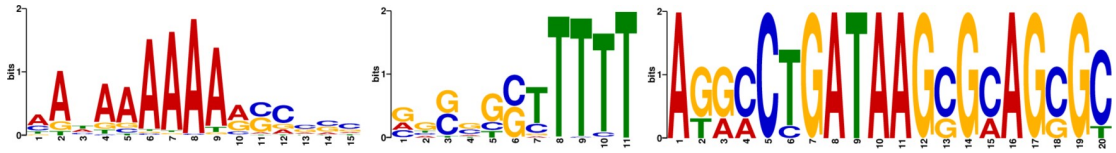

Fig 10. Motif diagram of the terminator sequence. The figure shows three motif diagrams we discovered of terminator sequences.

https://doi.org/10.1371/journal.pone.0228479.g010

4 Conclusions

In this work, we made miscellaneous comparisons of different feature extraction methods and models in many aspects. Eventually we proposed an accurate classification method "iterb-PPse" with 99.64%, 100%, 99.91% 99.88% in Sn, Sp, MCC, Acc respectively which is superior to the state-of-art prediction method and came to the following conclusions: (1) PseNC-I, PseNC-II, nucleotidepro are appropriate for formulating all samples. It proofs that nucleotide properties and the nucleotide components play a significant role in terminator classification and using the single GC content feature can't achieve the ideal classification effect. When

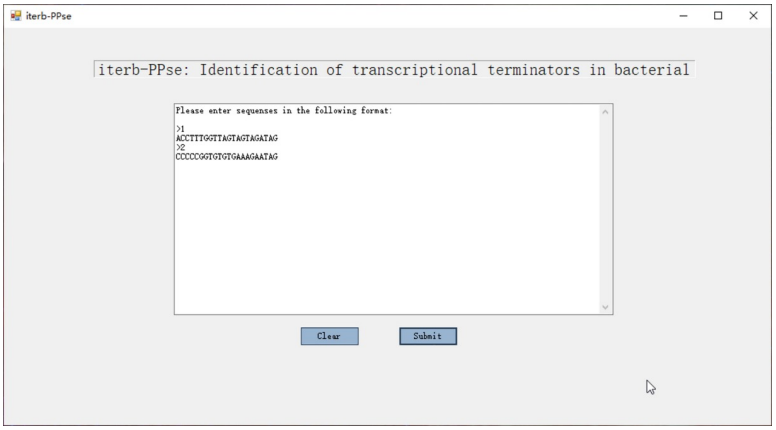

Fig 11. Main form of prediction tool. Just enter the sequence into the text box to get the prediction result.

https://doi.org/10.1371/journal.pone.0228479.g011

using K-pwm feature extraction methods, we found that position-weight features of oligonucleotides and hexanucleotides are effective for predicting terminators (2) XGBoost works best on predicting terminators among all models based on the features we extracted. All the code and data used in our experiment are open source and the full laboratory protocol are available online at https://www.protocols.io/view/prediction-of-terminational-terminators-in-bacteri-beccjasw, hopefully could provide some assistance for related researches.

Supporting information

S1 Table. Dataset with 280 terminator sequences of *E. coli*.
(CSV)

S2 Table. Dataset with 560 non-terminator sequences of *E. coli*.
(CSV)

S3 Table. Dataset with 425 terminator sequences of *B. subtilis*.
(CSV)

S4 Table. Dataset with 147 terminator sequences of *E. coli*.
(CSV)

S5 Table. Dataset with 76 terminator sequences of *E. coli*.
(CSV)

S6 Table. Dataset with 159 non-terminator sequences of *E. coli*.
(CSV)

S7 Table. Dataset with 122 non-terminator sequences of *B. subtilis*.
(CSV)

S8 Table. Dataset with 1615 terminator sequences of *E. coli*.
(CSV)

S9 Table. Dinucleotide physicochemical properties. This table contains 32 dinucleotide physicochemical properties we used and the corresponding standard values.
(CSV)

S10 Table. Trinucleotide physicochemical properties. This table contains 12 trinucleotide physicochemical properties we used and the corresponding standard values.
(CSV)

Author Contributions

Conceptualization: Yongxian Fan.

Data curation: Wanru Wang.

Formal analysis: Wanru Wang, Qingqi Zhu.

Funding acquisition: Yongxian Fan.

Investigation: Wanru Wang.

Methodology: Wanru Wang.

Project administration: Yongxian Fan.

Resources: Yongxian Fan.

Software: Wanru Wang.

Supervision: Yongxian Fan.

Validation: Wanru Wang.

Visualization: Wanru Wang.

Writing – original draft: Wanru Wang.

Writing – review & editing: Qingqi Zhu.

References

1. Henkin TM. Control of transcription termination in prokaryotes. Annual review of genetics. 1996; 30 (1):35–57.
2. De Hoon MJL, Makita Y, Nakai K, Miyano S. Prediction of Transcriptional Terminators in Bacillus subtilis and Related Species. PLoS Computional Biology. 2005; 1(3):e25.
3. Naville M, Ghuillot-Gaudeffroy A, Marchais A, Gautheret D. ARNold: A web tool for the prediction of Rho-independent transcription terminators. RNA Biology. 2011; 8(1):11–13. https://doi.org/10.4161/rna.8.1.13346 PMID: 21282983
4. Di Salvo M, Puccio S, Peano C, Lacour S, Alifano P. RhoTermPredict: an algorithm for predicting Rho-dependent transcription terminators based on Escherichia coli, Bacillus subtilis and Salmonella enterica databases. BMC Bioinformatics. 2019; 20(1):117. https://doi.org/10.1186/s12859-019-2704-x PMID: 30845912
5. Feng CQ, Zhang ZY, Zhu XJ, Lin Y, Chen W, Tang H, et al. iTerm-PseKNC: a sequence-based tool for predicting bacterial transcriptional terminators. Bioinformatics. 2019; 35(9):1469–1477. https://doi.org/10.1093/bioinformatics/bty827 PMID: 30247625
6. Lin H, Deng EZ, Ding H, Chen W, Chou KC. iPro54-PseKNC: a sequence-based predictor for identifying sigma-54 promoters in prokaryote with pseudo k-tuple nucleotide composition. Nucleic Acids Res. 2014; 42(21):12961–12972. https://doi.org/10.1093/nar/gku1019 PMID: 25361964
7. Chou KC. Some remarks on protein attribute prediction and pseudo amino acid composition. Journal of Theoretical Biology. 2011; 273(1):236–247. https://doi.org/10.1016/j.jtbi.2010.12.024 PMID: 21168420
8. Santos-Zavaleta A, Salgado H, Gama-Castro S, Sánchez-Pérez M, Gómez-Romero L, Ledezma-Tejeida D, et al. RegulonDB v 10.5: tackling challenges to unify classic and high throughput knowledge of gene regulation in E. coli K-12. Nucleic Acids Research. 2019; 47(D1):D212–D220. https://doi.org/10.1093/nar/gky1077 PMID: 30395280
9. Ishii T, Yoshida K, Terai G, Fujita Y, Nakai K. DBTBS: a database of Bacillus subtilis promoters and transcription factors. Nucleic Acids Research. 2000; 29(1):278–280.

10. Chou KC. Impacts of Bioinformatics to Medicinal Chemistry. Medicinal Chemistry. 2015; 11(3):218–234. https://doi.org/10.2174/1573406411666141229162834 PMID: 25548930

11. Xia X. Position weight matrix, gibbs sampler, and the associated significance tests in motif characterization and prediction. Scientifica (Cairo). 2012; 2012:917540. https://doi.org/10.6064/2012/917540 PMID: 24278755

12. Wu Q, Wang J, Yan H. An Improved Position Weight Matrix Method Based on an Entropy Measure for the Recognition of Prokaryotic Promoters. International Journal of Data Ming and Bioinformatics. 2009; 5(1):22–37.

13. Sinha S. On counting position weight matrix matches in a sequence, with application to discriminative motif finding. Bioinformatics. 2006; 22(14):e454–e463. https://doi.org/10.1093/bioinformatics/btl227 PMID: 16873507

14. Li QZ, Lin H. The recognition and prediction of σ70 promoters in Escherichia coli K-12. Journal of Theoretical Biology. 2006; 242(1):135–141. https://doi.org/10.1016/j.jtbi.2006.02.007 PMID: 16603195

15. Benjamini Y, Speed TP. Summarizing and correcting the GC content bias in high-throughput sequencing. Nucleic Acids Research. 2012; 40(10).

16. Sahyoun AH, Bernt M, Stadler PF, Tout K. GC skew and mitochondrial origins of replication. Mitochondrion. 2014; 17(2014):56–66.

17. Yang H, Qiu WR, Liu G, Guo FB, Chen W, Chou KC, et al. iRSpot-Pse6NC: Identifying recombination spots in Saccharomyces cerevisiae by incorporating hexamer composition into general PseKNC. Int J Biol Sci. 2018; 14(8):883–891. https://doi.org/10.7150/ijbs.24616 PMID: 29989083

18. Farnham PJ, Platt T. Rho-independent termination: dyad symmetry in DNA causes RNA polymerase to pause during transcription in vitro. Nucleic Acids Research. 1981; 9(3):563–577. https://doi.org/10.1093/nar/9.3.563 PMID: 7012794

19. Grigoriev A. Analyzing genomes with cumulative skew diagrams. Nucleic Acids Research. 1998; 26 (10):2286–2290. https://doi.org/10.1093/nar/26.10.2286 PMID: 9580676

20. Charneski CA, Honti F, Bryant JM, Hurst LD, Feil EJ. A typical AT Skew in Firmicute Genomes Results from Selection and Not from Mutation. PLoS Genetics. 2011; 7(9):e1002283. https://doi.org/10.1371/journal.pgen.1002283 PMID: 21935355

21. Pan X, Xiong K, Anthon C, Hyttel P, Freude KK, Jensen LJ, et al. WebCircRNA: Classifying the Circular RNA Potential of Coding and Noncoding RNA. Genes. 2018; 9(11).

22. Fukue Y, Sumida N, Tanase J, Ohyama T. A highly distinctive mechanical property found in the majority of human promoters and its transcriptional relevance. Nucleic Acids Res. 2005; 33(12):3821–3827. https://doi.org/10.1093/nar/gki700 PMID: 16027106

23. Chen W, Lei TY, Jin DC, Lin H, Chou KC. PseKNC: A flexible web server for generating pseudo K-tuple nucleotide composition. Analytical Biochemistry. 2014; 456:53–60. https://doi.org/10.1016/j.ab.2014.04.001 PMID: 24732113

24. Bari ATMG Reaz MR, Choi HJ Jeong BS. DNA Encoding for Splice Site Prediction in Large DNA Sequence. International Conference on Database Systems for Advanced Applications. New York: Springer; 2013.p. 46–58.

25. Ghandi M, Mohammad-Noori M, Beer MA. Robust k-mer frequency estimation using gapped k -mers. Journal of Mathematical Biology. 2014; 69(2):469–500. https://doi.org/10.1007/s00285-013-0705-3 PMID: 23861010

26. Liu B, Liu F, Wang X, Chen J, Fang L, Chou KC. Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. Nucleic Acids Research. 2015; 43 (W1):W65–W71. https://doi.org/10.1093/nar/gkv458 PMID: 25958395

27. Liu B, Wu H, Chou KC. Pse-in-One 2.0: An Improved Package of Web Servers for Generating Various Modes of Pseudo Components of DNA, RNA, and Protein Sequences. Natural Science. 2017; 09 (04):67–91.

28. Chou KC. A Key Driving Force in Determination of Protein Structural Classes. Biochemical and Biophysical Research Communications. 1999; 264(1):216–224. https://doi.org/10.1006/bbrc.1999.1325 PMID: 10527868

29. Song J, Li F, Leier A, Marquez-Lago TT, Akutsu T, Haffari G, et al. PROSPERous: high-throughput prediction of substrate cleavage sites for 90 proteases with improved accuracy. Bioinformatics (Oxford, England). 2018; 34(4):684–687.

30. Chen W, Lin H, Chou KC. Pseudo nucleotide composition or PseKNC: an effective formulation for analyzing genomic sequences. Molecular Biosystems. 2015; 11(10):2620–2634. https://doi.org/10.1039/c5mb00155b PMID: 26099739

31. Su ZD, Huang Y, Zhang ZY, Zhao YW, Wang D, Chen W, et al. iLoc-lncRNA: predict the subcellular location of lncRNAs by incorporating octamer composition into general PseKNC. Bioinformatics. 2018; 34(24):4196–4204. https://doi.org/10.1093/bioinformatics/bty508 PMID: 29931187

32. Lai HY, Chen XX, Chen W, Tang H, Lin H. Sequence-based predictive modeling to identify cancerlectins. Oncotarget. 2017; 8(17):28169–28175. https://doi.org/10.18632/oncotarget.15963 PMID: 28423655

33. Li F, Li C, Wang M, Webb GI, Zhang Y, Whisstock JC, et al. GlycoMine: a machine learning-based approach for predicting N-, C- and O-linked glycosylation in the human proteome. Bioinformatics. 2015; 31(9):1411–1419. https://doi.org/10.1093/bioinformatics/btu852 PMID: 25568279

34. Feng PM, Chen W, Lin H, Chou KC. iHSP-PseRAAAC: Identifying the heat shock protein families using pseudo reduced amino acid alphabet composition. Analytical Biochemistry. 2013; 442(1):118–125. https://doi.org/10.1016/j.ab.2013.05.024 PMID: 23756733

35. Basu S, Söderquist F, Wallner B. Proteus: a random forest classifier to predict disorder-to-order transitioning binding regions in intrinsically disordered proteins. Journal of Computer-Aided Molecular Design. 2017; 31(5):453–466. https://doi.org/10.1007/s10822-017-0020-y PMID: 28365882

36. Pan X, Jensen LJ, Gorodkin J. Inferring disease-associated long non-coding RNAs using genome-wide tissue expression profiles. Bioinformatics (Oxford, England). 2018; 35(9):1494–1502.

37. Granholm V, Noble WS, Käll L. A cross-validation scheme for machine learning algorithms in shotgun proteomics. BMC Bioinformatics. 2012; 13 Suppl 16:S3.

38. Panwar B, Raghava GP. Prediction of uridine modifications in tRNA sequences. BMC Bioinformatics. 2014; 15(1):326.

39. Feng PM, Ding H, Chen W, Lin H. Naïve bayes classifier with feature selection to identify phage virion proteins. Computational and mathematical methods in medicine. 2013; 2013:530696. https://doi.org/10.1155/2013/530696 PMID: 23762187

40. Feng PM, Ding H, Chen W, Lin H. Identification of antioxidants from sequence information using naïve Bayes. Computational and mathematical methods in medicine. 2013; 2013:567529. https://doi.org/10.1155/2013/567529 PMID: 24062796

41. Li F, Li C, Marquez-Lago TT, Leier A, Akutsu T, Purcell AW, et al. Quokka: a comprehensive tool for rapid and accurate prediction of kinase family-specific phosphorylation sites in the human proteome. Bioinformatics (Oxford, England). 2018; 34(24):4223–4231.

42. Song J, Wang Y, Li F, Akutsu T, Rawlings ND, Webb GI, et al. iProt-Sub: a comprehensive package for accurately mapping and predicting protease-specific substrates and cleavage sites. Briefings in bioinformatics. 2019; 20(2):638–658. https://doi.org/10.1093/bib/bby028 PMID: 29897410

43. Liu B, Yang F, Huang DS, Chou KC. iPromoter-2L: a two-layer predictor for identifying promoters and their types by multi-window-based PseKNC. Bioinformatics. 2018; 34(1):33–40. https://doi.org/10.1093/bioinformatics/btx579 PMID: 28968797

44. Brown GW. Standard deviation, standard error. Which 'standard' should we use. American journal of diseases of children. 1982; 136(10).

45. Dar D, Sorek R. High-resolution RNA 3′-ends mapping of bacterial Rho-dependent transcripts. Nucleic Acids Research. 2018; 46(13):6967–6805.

46. Ju X, Li D, Liu S. Full-length RNA profiling reveals pervasive bidirectional transcription terminators in bacteria. Nat Microbiol. 2019; 4(11):1907–1918. https://doi.org/10.1038/s41564-019-0500-z PMID: 31308523

47. Bailey TL, Elkan C. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. Proceedings. International Conference on Intelligent Systems for Molecular Biology. 1994; 2:28–36. PMID: 7584402