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

Yongxian Fan[1,*], Wanru Wang[1], Qingqi Zhu[1]

[1] School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China

* Corresponding author

E-mail: yongxian.fan@gmail.com (YF)

Abstract

Terminator is a DNA sequence that give 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. It's prediction accuracy for two independent datasets reached 94.24% and 99.45% respectively. For the convenience of users, a software was developed with the same name on the basis of "iterb-PPse". 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.

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.

Fig 1. Transcriptional termination process. (A) The termination do not require Rho. The transcription stops when the RNA forms the stem loop structure. (B) The termination dependent on Rho.

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.

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.

2.1 Data Collection

In our study, the initial datasets were obtained from http://lin-group.cn/server/iTerm-PseKNC [2], which includes 280 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 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 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 of Supplementary data.

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

115 **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

116

117 2.2 Feature extraction

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

122

123 2.2.1 K-pwm

124 The new feature extraction method "K-pwm" mainly employed the Position
125 Weight Matrix[11-14], where K represents *k*-tuple nucleotides. Considering that the
126 length of negative samples is different from that of the positive samples in the
127 benchmark set. we made a little modification to the calculation of the final sequence
128 score to eliminate the negative impact of sequence length. A total of 6 feature sets were

129 obtained by using this method, namely the position weight features corresponding to k

130 =1, 2, 3, 4, 5, 6. The calculation steps are shown below.

131 $$p_0 = \frac{1}{4^k}, \tag{1}$$

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

133 $$p_{xi} = \frac{n_{xi}}{N_i}, \tag{2}$$

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

135 $$W_{xi} = \ln\left(\frac{p_{xi}}{p_0}\right), \tag{3}$$

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

137 $$F = \frac{1}{L}\sum_i W_{xi}, \tag{4}$$

138 where L is the length of the corresponding sequence.

139

140 ## 2.2.2 Base-content

141 Given that the rho-independent terminators are rich in GC base pairs, we extracted

142 a set of features and collectively referred to as Base-content[15, 16]. Specifically, we

143 mainly obtained the content features of the single nucleotide(A, C, G, T) in each DNA

144 sequence[17, 18]. In this paper, 5 kinds of base content features(atContent, gcContent,

145 gcSkew, atSkew, atgcRatio)[15, 16, 19-21] were took into account.

146 $$p_i^{\mathrm{A+T}} = \frac{m_i^{\mathrm{A+T}}}{m_i^{\mathrm{A+T+G+C}}}; \tag{5}$$

147 $$p_i^{\mathrm{G+C}} = \frac{m_i^{\mathrm{G+C}}}{m_i^{\mathrm{A+T+G+C}}}; \tag{6}$$

148 $$p_i^{\text{atgRatio}} = \frac{m_i^{\text{A+T}}}{m_i^{\text{G+C}}}; \quad (7)$$

149 $$p_i^{\text{gcSkew}} = \frac{m_i^{\text{G-C}}}{m_i^{\text{G+C}}}; \quad (8)$$

150 $$p_i^{\text{atSkew}} = \frac{m_i^{\text{A-T}}}{m_i^{\text{A+T}}}; \quad (9)$$

151 where mG i, mC $i$are the contents of G and C in the i-th sequence, respectively. mA+T
152 i, mG+C i, mA+T+G+C i are the contents of "A+T", "G+C" and "A+T+G+C",
153 respectively. mA-T i , mG-C i represent the content of "A-T" and "G-C",
154 respectively.
155

156 ## 2.2.3 Nucleotidepro

157 Nucleotide properties of DNA sequences play a key role in gene regulation[22].
158 Therefore, we proposed a new feature extraction method "Nucleotidepro" involving 47
159 properties[23] not covered previously, including 3 nucleotide chemical properties[24],
160 32 dinucleotide physicochemical properties and 12 trinucleotide physicochemical
161 properties.

162 To extract corresponding features, we employed a 47*L dimension matrix to
163 represent each sequence. L is the length of the corresponding sequence. As shown in
164 the Table 3, we used 0 and 1 to represent the chemical properties of different
165 nucleotides. Then we iterated through each sequence and assigned the values of

166 different properties for different nucleotide to the corresponding elements in the matrix.
167 The nucleotide properties and corresponding standard-converted values[23] for the 47
168 properties can be obtained from the Tables S8 and S9 from Supplementary data.
169

170 **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

171

172 ### 2.2.4 PseKNC-Ⅰ

173 PseKNC-I [6] is generally understood to mean the parallel correlation PseKNC. It
174 combines K-tuple nucleotides components [25] with 6 physicochemical properties [22]
175 (rise, slide, shift, twist, roll, tilt), not only considering the global or long-range sequence
176 information, but also calculating the biochemical information of DNA sequences. The
177 PseKNC-Ⅰ features can be obtained directly through the online tool Pse-in-one [26,
178 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.

2.4.1 Feature analysis

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].

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

2.4.2 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:

$$F_i = \frac{\left(\bar{x}_i^{(+)} - \bar{x}_i\right)^2 + \left(\bar{x}_i^{(-)} - \bar{x}_i\right)^2}{\frac{1}{n^+ - 1}\sum_{k=1}^{n^+}\left(\bar{x}_{k,i}^{(+)} - \bar{x}_i^{(+)}\right)^2 + \frac{1}{n^- - 1}\sum_{k=1}^{n^-}\left(\bar{x}_{k,i}^{(-)} - \bar{x}_i^{(-)}\right)^2}, \quad (10)$$

where $\bar{x}_i$, $\bar{x}(+)\ i$, $\bar{x}(-)$ 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:

$$q_i = m_i / M, \quad (11)$$

where q_i is the prior probability, m_i represents the number of i-th samples (i =1,2 indicates positive and negative respectively), and M is the number of all samples.

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

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.

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

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

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.

2.4.3 Incremental feature selection

The second step is Incremental Feature Selection(IFS)[33]. It uses a feature as the training set at first, then the sorted features are added to the training set one by one, finally find 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.

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

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

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

305

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.

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

312 Δ represents the step size.

313 2.7 Cross-validation test

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

320

321 2.8 Independent test

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

329

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].

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 Analysis 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.

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.

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

365 the end, a total of nine group features were obtained. Details of the combination method
366 are shown in Table 5. As shown in Fig 6, Group 8 stands out in terms of Sn, Sp, MCC
367 and Acc from other combined feature sets. Consequently, the three features Pse5NC-
368 Ⅰ, Pse5NC-Ⅱ, 47 nucleotide properties were applied to formulate all samples.
369
370 **Fig 5. Prediction results using different feature extraction methods.** All results are
371 obtained after 100 times 5-fold CV. The ones marked red represent the best of each
372 method.
373
374 **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	
	K-pwm	1-pwm, 6-pwm	
Group7	Nucleotidepro	3 nucleotide chemical properties 32 dinucleotide physicochemical properties 12 trinucleotide physicochemical properties	49
	PseKNC	Pse5NC-Ⅰ, Pse5NC-Ⅱ	
Group8	Nucleotidepro	3 nucleotide chemical properties 32 dinucleotide physicochemical properties 12 trinucleotide physicochemical properties	2600
	PseKNC	Pse5NC-Ⅰ, Pse5NC-Ⅱ	
	K-pwm	1-pwm, 6-pwm	
Group9	Nucleotidepro	3 nucleotide chemical properties 32 dinucleotide physicochemical properties 12 trinucleotide physicochemical properties	2132

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

376

377 **Fig 6. Classification results using different combined features.** These results are

378 obtained using XGBoost after 100 times 5-fold CV.

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.

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].

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.

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.

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.

As shown in Table 7 and Fig 7, the "iterb-PPse" is superior to the "iTerm-PseKNC" across the three datasets in Sn, Sp, MCC, Acc and AUC. 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% for the benchmark dataset and two independent datasets respectively.

3.5 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 Figure 8.

Figure 8. Main form of prediction tool. Just enter the sequence into the text box to get the prediction result.

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 not achieve the ideal classification effect. When 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

433 code and data used in our experiment are open source and are available at
434 https://github.com/Sarahyouzi/myexperiment, hopefully could provide some assistance
435 for related researches.

436

437 # References

438 1. M HT. Control of transcription termination in prokaryotes. Annual review of genetics.
439 1996;30:35-57.
440 2. De Hoon MJL, Makita Y, Nakai K, Miyano S. Prediction of Transcriptional
441 Terminators in Bacillus subtilis and Related Species. PLoS Computional Biology.
442 2005;1(3):e25.
443 3. Naville M, Ghuillot-Gaudeffroy A, Marchais A, Gautheret D. ARNold: A web tool for
444 the prediction of Rho-independent transcription terminators. RNA Biology.
445 2011;8(1):11-13.
446 4. Di Salvo M, Puccio S, Peano C, Lacour S, Alifano P. RhoTermPredict: an algorithm
447 for predicting Rho-dependent transcription terminators based on Escherichia coli,
448 Bacillus subtilis and Salmonella enterica databases. BMC Bioinformatics.
449 2019;20(1):117.
450 5. Feng CQ, Zhang ZY, Zhu XJ, Lin Y, Chen W, Tang H et al. iTerm-PseKNC: a
451 sequence-based tool for predicting bacterial transcriptional terminators.
452 Bioinformatics. 2019;35(9):1469-1477.

453 6. Lin H, Deng EZ, Ding H, Chen W, Chou KC. iPro54-PseKNC: a sequence-based
454 predictor for identifying sigma-54 promoters in prokaryote with pseudo k-tuple
455 nucleotide composition. Nucleic Acids Res. 2014;42(21):12961-12972.
456 7. Chou K-C. Some remarks on protein attribute prediction and pseudo amino acid
457 composition. Journal of Theoretical Biology. 2011;273(1):236-247.
458 8. Alberto S-Z, Heladia S, Socorro G-C, Mishael S-P, Laura G-R, Daniela L-T et al.
459 RegulonDB v 10.5: tackling challenges to unify classic and high throughput
460 knowledge of gene regulation in E. coli K-12. Nucleic Acids Research.
461 2019;47(D1):D212-D220.
462 9. T I, K Y, G T, Y F, K N. DBTBS: a database of Bacillus subtilis promoters and
463 transcription factors. Nucleic Acids Research. 2000;29(1):278-280.
464 10. Chou K-C. Impacts of Bioinformatics to Medicinal Chemistry. Medicinal Chemistry.
465 2015;11(3):218-234.
466 11. Xia X. Position weight matrix, gibbs sampler, and the associated significance tests
467 in motif characterization and prediction. Scientifica (Cairo). 2012;2012:917540.
468 12. Wu Q, Wang J, Yan H. An Improved Position Weight Matrix Method Based on an
469 Entropy Measure for the Recognition of Prokaryotic Promoters. International
470 Journal of Data Ming and Bioinformatics. 2009;5(1):22-37.
471 13. Sinha S (2006) On counting position weight matrix matches in a sequence, with
472 application to discriminative motif finding. Bioinformatics 22 (14):e454-e463.

473 14. Li Q-Z, Lin H. The recognition and prediction of σ70 promoters in Escherichia coli
474 K-12. Journal of Theoretical Biology. 2006;242(1):135-141.
475 15. Yuval B, P ST. Summarizing and correcting the GC content bias in high-throughput
476 sequencing. Nucleic Acids Research. 2012;40(10).
477 16. Sahyoun AH, Bernt M, Stadler PF, Tout K. GC skew and mitochondrial origins of
478 replication. Mitochondrion. 2014;17(2014):56-66.
479 17. Yang H, Qiu WR, Liu G, Guo FB, Chen W, Chou KC et al. iRSpot-Pse6NC:
480 Identifying recombination spots in Saccharomyces cerevisiae by incorporating
481 hexamer composition into general PseKNC. Int J Biol Sci. 2018;14(8):883-891.
482 18. Farnham PJ, Platt T. Rho-independent termination: dyad symmetry in DNA causes
483 RNA polymerase to pause during transcription in vitro. Nucleic Acids Research.
484 1981;9(3):563-577.
485 19. A G. Analyzing genomes with cumulative skew diagrams. Nucleic Acids Research.
486 1998;26(10):2286-2290.
487 20. Charneski CA, Honti F, Bryant JM, Hurst LD, Feil EJ. A typical AT Skew in Firmicute
488 Genomes Results from Selection and Not from Mutation. PLoS Genetics.
489 2011;7(9):e1002283.
490 21. Xiaoyong P, Kai X, Christian A, Poul H, K FK, Juhl JL et al. WebCircRNA:
491 Classifying the Circular RNA Potential of Coding and Noncoding RNA. Genes.
492 2018;9(11).

493 22. Fukue Y. A highly distinctive mechanical property found in the majority of human
494 promoters and its transcriptional relevance. Nucleic Acids Res. 2005;33(12):3821-
495 3827.
496 23. Chen W, Lei T-Y, Jin D-C, Lin H, Chou K-C. PseKNC: A flexible web server for
497 generating pseudo K-tuple nucleotide composition. Analytical Biochemistry.
498 2014;456:53-60.
499 24. Bari AT, Reaz MR, Choi H-J, Jeong B-S. DNA Encoding for Splice Site Prediction
500 in Large DNA Sequence. Database Systems for Advanced Applications.
501 Springer-Verlag New York, Inc, 2013, 46-58.
502 25. Ghandi M, Mohammad-Noori M, Beer MA. Robust k -mer frequency estimation
503 using gapped k -mers. Journal of Mathematical Biology. 2014;69(2):469-500.
504 26. Liu B, Liu F, Wang X, Chen J, Fang L, Chou K-C. Pse-in-One: a web server for
505 generating various modes of pseudo components of DNA, RNA, and protein
506 sequences. Nucleic Acids Research. 2015;43(W1):W65-W71.
507 27. Liu B, Wu H, Chou K-C. Pse-in-One 2.0: An Improved Package of Web Servers for
508 Generating Various Modes of Pseudo Components of DNA, RNA, and Protein
509 Sequences. Natural Science. 2017;09(04):67-91.
510 28. Chou K-C. A Key Driving Force in Determination of Protein Structural Classes.
511 Biochemical and Biophysical Research Communications. 1999;264(1):216-224.
512 29. Jiangning S, Fuyi L, André L, T M-LT, Tatsuya A, Gholamreza H et al.
513 PROSPERous: high-throughput prediction of substrate cleavage sites for 90

514 proteases with improved accuracy. Bioinformatics (Oxford, England).
515 2018;34(4):684-687.
516 30. Chen W, Lin H, Chou K-C. Pseudo nucleotide composition or PseKNC: an effective
517 formulation for analyzing genomic sequences. Molecular Biosystems.
518 2015;11(10):2620-2634.
519 31. Su ZD, Huang Y, Zhang ZY, Zhao YW, Wang D, Chen W et al. iLoc-lncRNA: predict
520 the subcellular location of lncRNAs by incorporating octamer composition into
521 general PseKNC. Bioinformatics. 2018;34(24):4196-4204.
522 32. Hong-Yan L, Xin-Xin C, Wei C, Hua T, Hao L. Sequence-based predictive modeling
523 to identify cancerlectins. Oncotarget. 2017;8(17):28169-28175.
524 33. Li F, Li C, Wang M, Webb GI, Zhang Y, Whisstock JC et al. GlycoMine: a machine
525 learning-based approach for predicting N-, C- and O-linked glycosylation in the
526 human proteome. Bioinformatics. 2015;31(9):1411-1419.
527 34. Feng P-M, Chen W, Lin H, Chou K-C. iHSP-PseRAAAC: Identifying the heat shock
528 protein families using pseudo reduced amino acid alphabet composition. Analytical
529 Biochemistry. 2013;442(1):118-125.
530 35. Basu S, Söderquist F, Wallner B. Proteus: a random forest classifier to predict
531 disorder-to-order transitioning binding regions in intrinsically disordered proteins.
532 Journal of Computer-Aided Molecular Design. 2017;31(5):453-466.

533 36. Xiaoyong P, Juhl JL, Jan G. Inferring disease-associated long non-coding RNAs
534 using genome-wide tissue expression profiles. Bioinformatics (Oxford, England).
535 2018;35(9):1494-1502.
536 37. A cross-validation scheme for machine learningalgorithms in shotgun proteomics.
537 BMC Bioinformatics. 2012;13 Suppl 16:S3.
538 38. Panwar B, Raghava GP. Prediction of uridine modifications in tRNA sequences.
539 BMC Bioinformatics. 2014;15(1):326.
540 39. Peng-Mian F, Hui D, Wei C, Hao L. Naïve bayes classifier with feature selection to
541 identify phage virion proteins. Computational and mathematical methods in
542 medicine. 2013;2013:530696.
543 40. Peng-Mian F, Hao L, Wei C. Identification of antioxidants from sequence
544 information using naïve Bayes. Computational and mathematical methods in
545 medicine. 2013;2013:567529.
546 41. Fuyi L, Chen L, T M-LT, André L, Tatsuya A, W PA et al. Quokka: a comprehensive
547 tool for rapid and accurate prediction of kinase family-specific phosphorylation sites
548 in the human proteome. Bioinformatics (Oxford, England). 2018;34(24):4223-4231.
549 42. Jiangning S, Yanan W, Fuyi L, Tatsuya A, D RN, I WG et al. iProt-Sub: a
550 comprehensive package for accurately mapping and predicting protease-specific
551 substrates and cleavage sites. Briefings in bioinformatics. 2019;20(2):638-658.

552 43. Liu B, Yang F, Huang D-S, Chou K-C. iPromoter-2L: a two-layer predictor for
553 identifying promoters and their types by multi-window-based PseKNC.
554 Bioinformatics. 2018;34(1):33-40.
555 44. W BG. Standard deviation, standard error. Which 'standard' should we use.
556 American journal of diseases of children. 1982;136(10).

557

558 Supporting information

559

560 **S1 Table. Dataset with 280 terminator sequences of *E. coli*.**

561 (CSV)

562 **S2 Table. Dataset with 560 non-terminator sequences of *E. coli*.**

563 (CSV)

564 **S3 Table. Dataset with 425 terminator sequences of *B. subtilis*.**

565 (CSV)

566 **S4 Table. Dataset with 147 terminator sequences of *E. coli*.**

567 (CSV)

568 **S5 Table. Dataset with 76 terminator sequences of *E. coli*.**

569 (CSV)

570 **S6 Table. Dataset with 159 non-terminator sequences of *E. coli*.**

571 (CSV)

572 **S7 Table. Dataset with 122 non-terminator sequences of *B. subtilis*.**

573 (CSV)

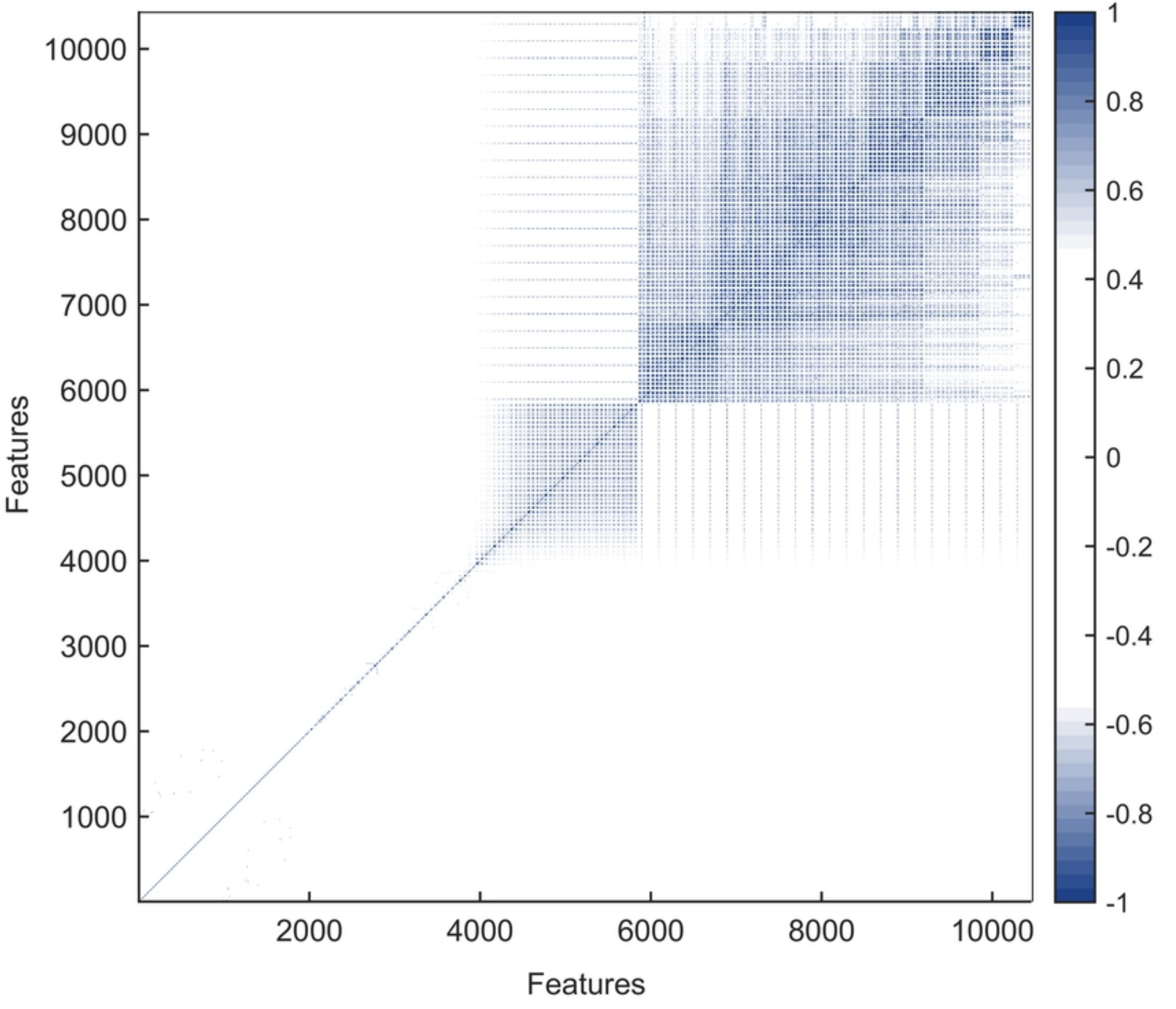
574 **S8 Table. Dinucleotide physicochemical properties.** This table contains 32
575 dinucleotide physicochemical properties we used and the corresponding standard
576 values.

577 (CSV)

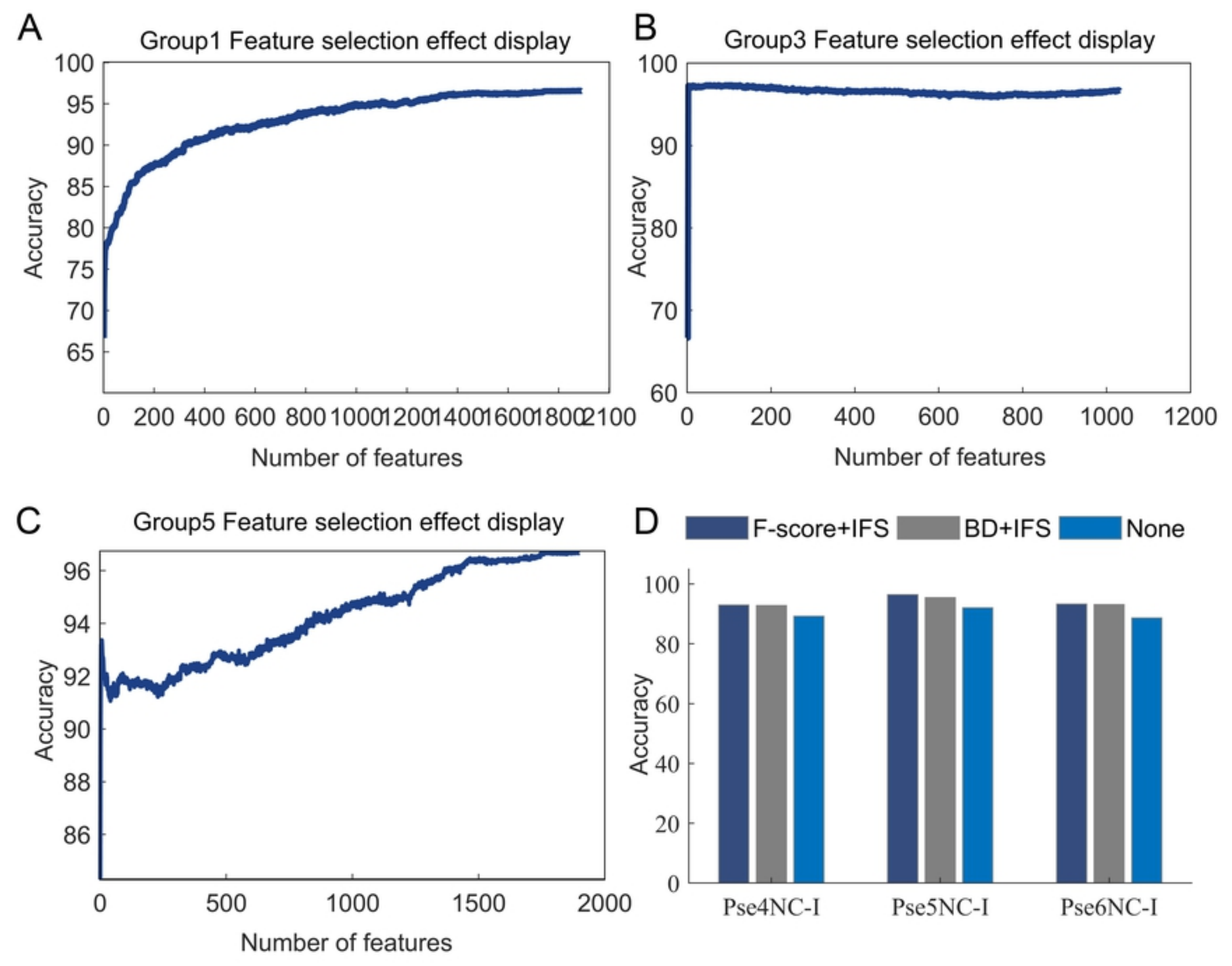
578 **S9 Table. Trinucleotide physicochemical properties.** This table contains 12
579 trinucleotide physicochemical properties we used and the corresponding standard
580 values.

581 (CSV)

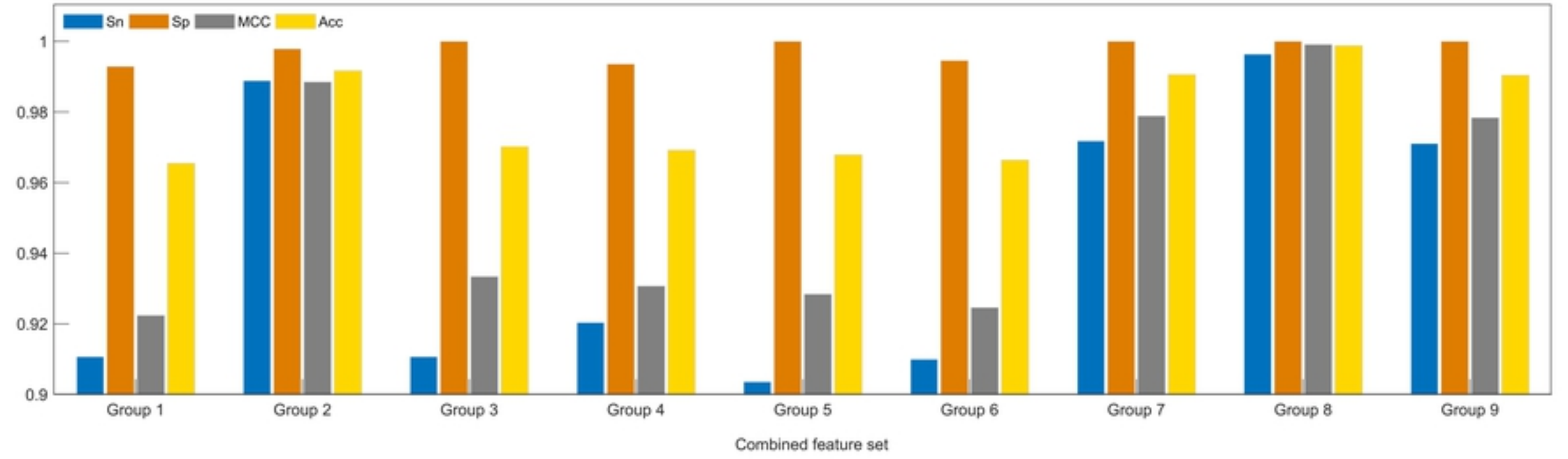
Figure


Features
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
2000
4000
6000
8000
10000
Features
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

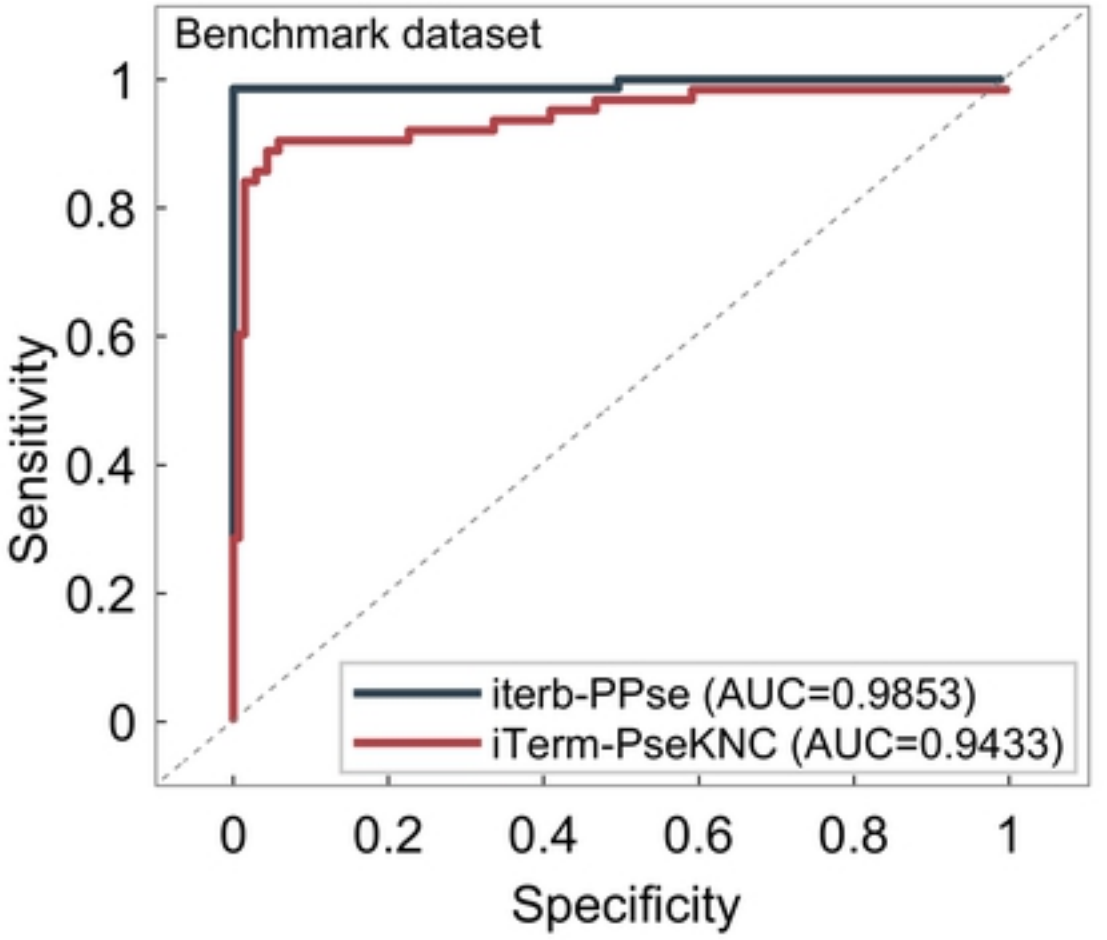
Figure

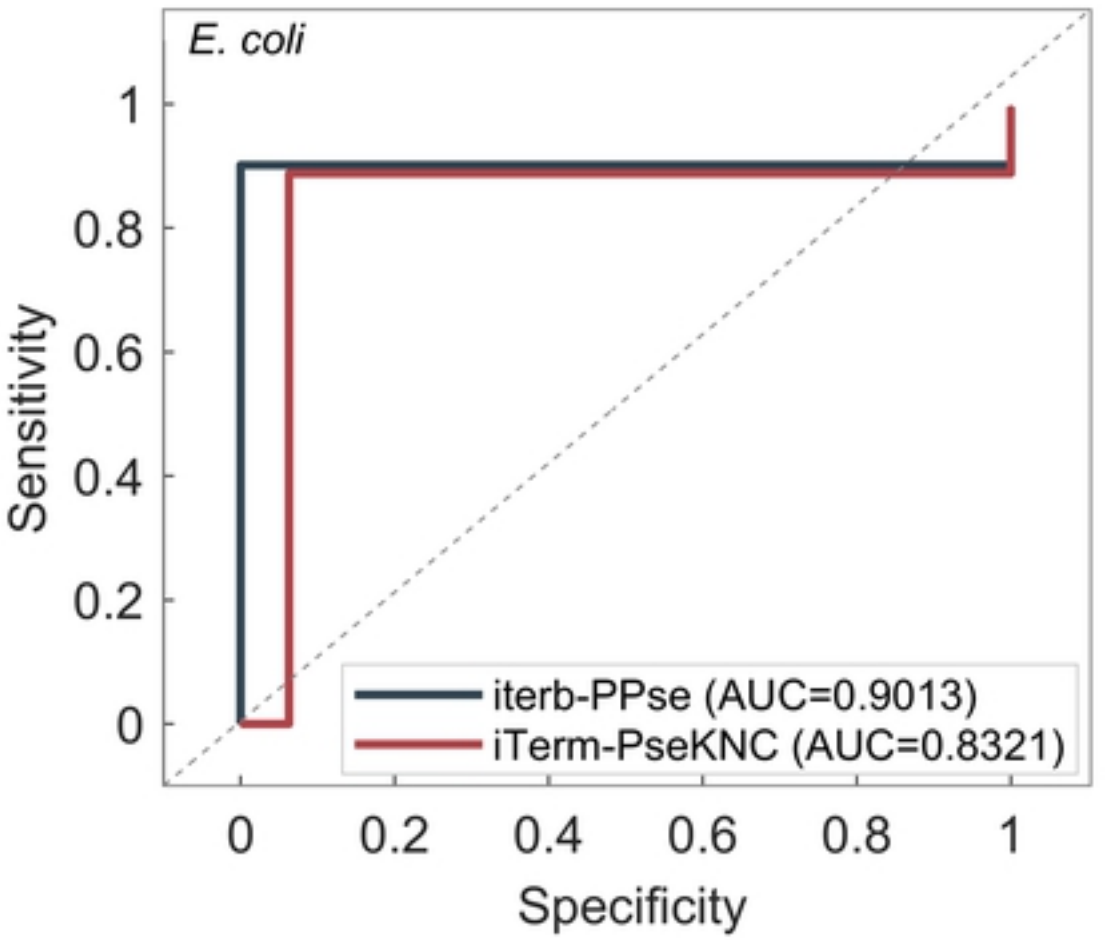

A
Group1 Feature selection effect display
Accuracy
Number of features
B
Group3 Feature selection effect display
Accuracy
Number of features
C
Group5 Feature selection effect display
Accuracy
Number of features
D
F-score+IFS
BD+IFS
None
Accuracy
Pse4NC-I
Pse5NC-I
Pse6NC-I

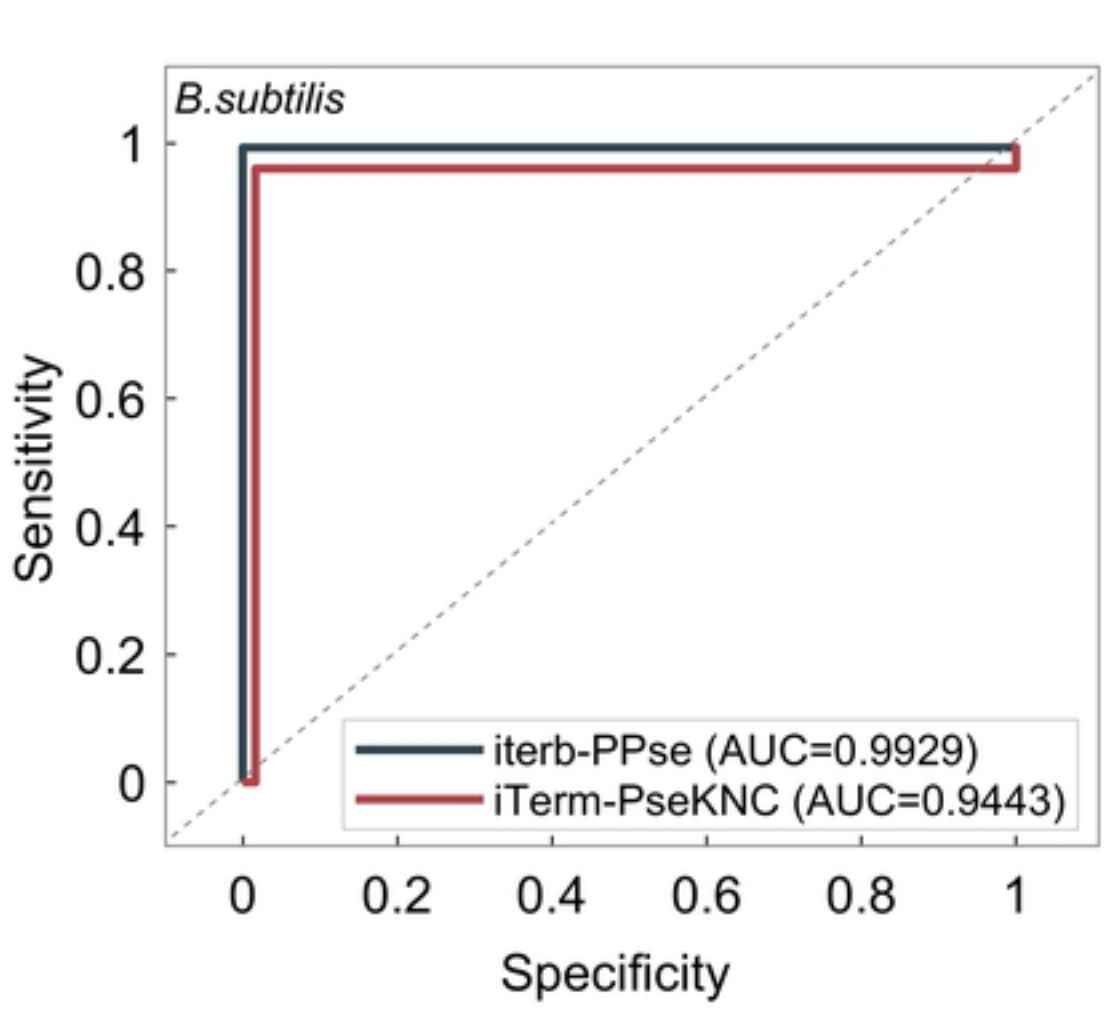
Figure

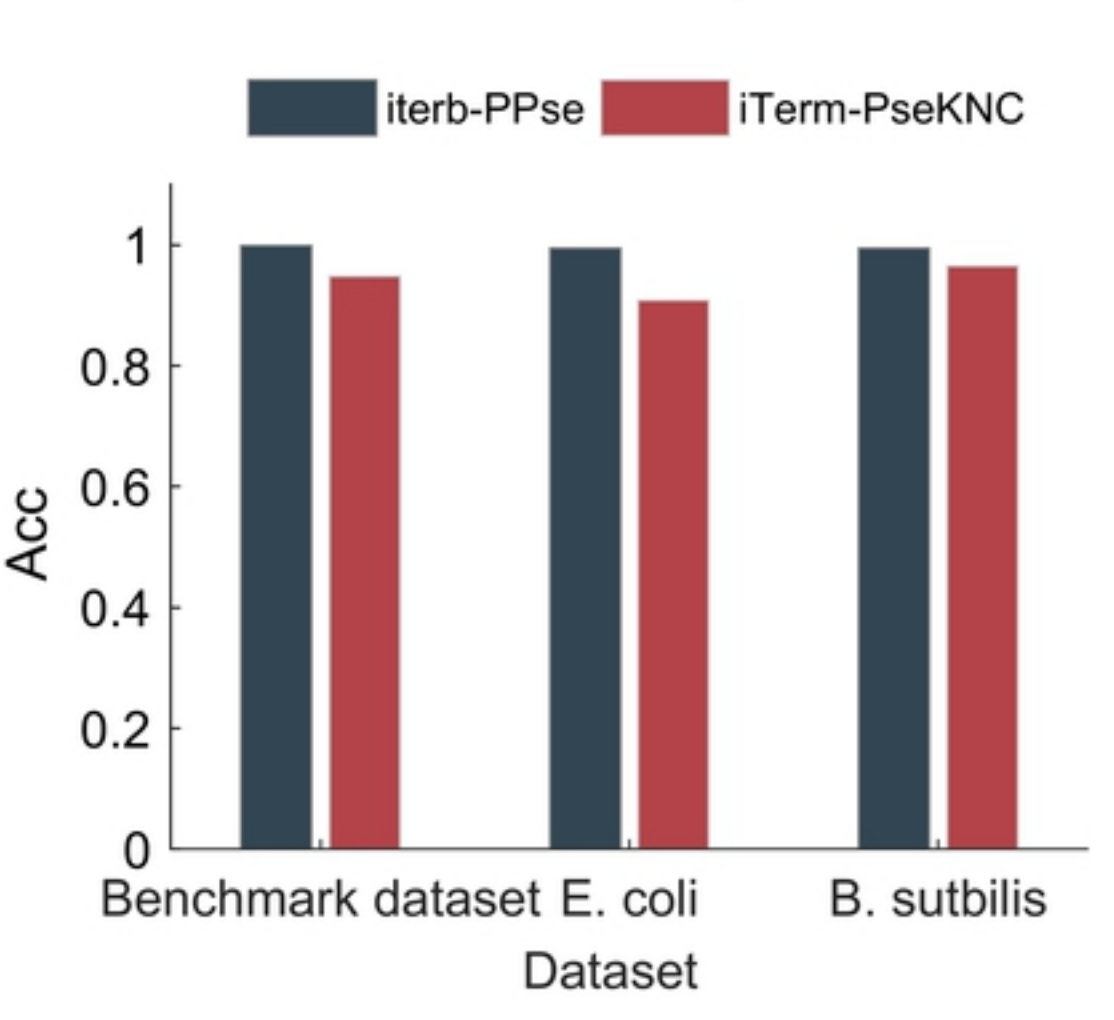

Sn
Sp
MCC
Acc
1
0.98
0.96
0.94
0.92
0.9
Group 1
Group 2
Group 3
Group 4
Group 5
Group 6
Group 7
Group 8
Group 9
Combined feature set

Figure

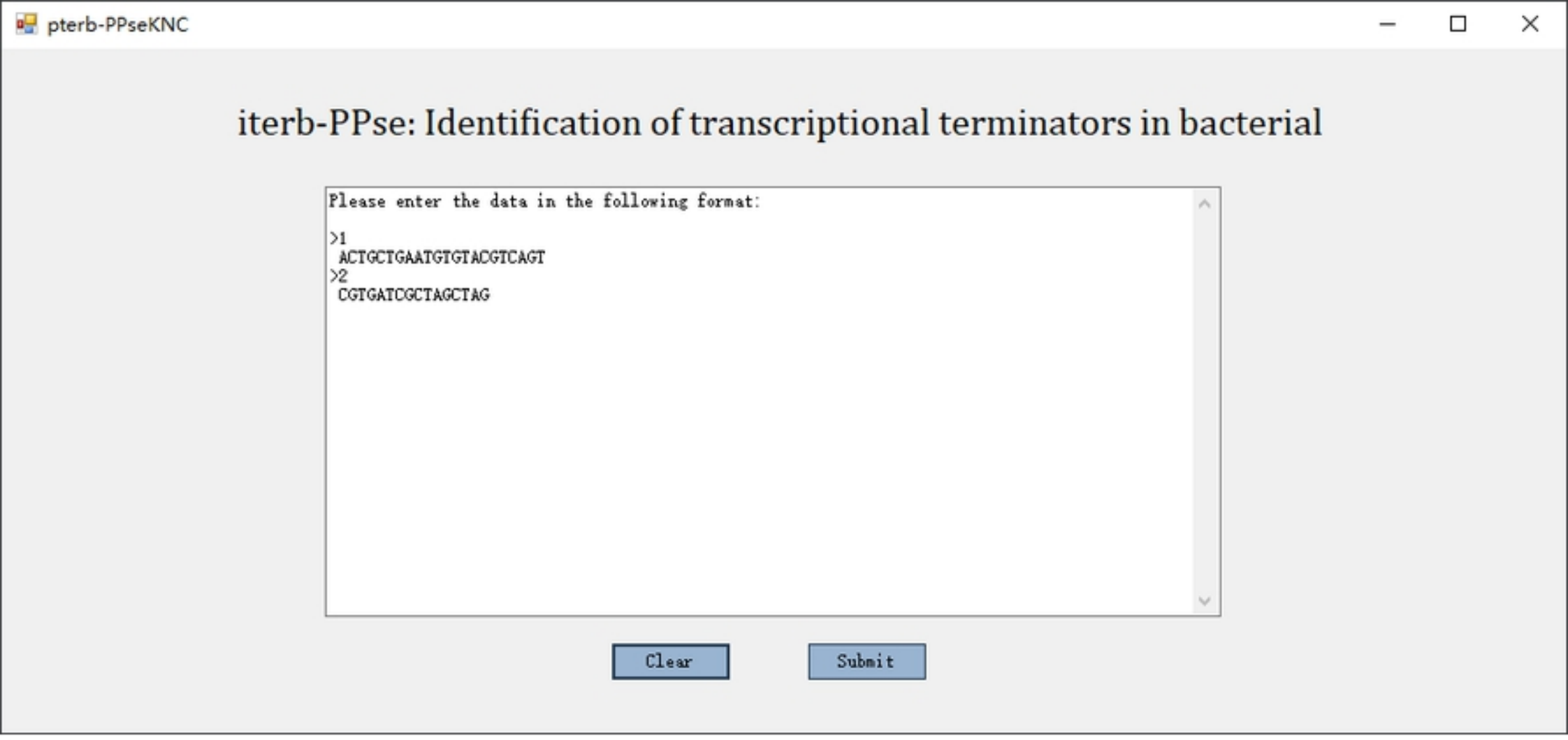

Benchmark dataset
Sensitivity
Specificity
iterb-PPse (AUC=0.9853)
iTerm-PseKNC (AUC=0.9433)


E. coli
Sensitivity
Specificity
iterb-PPse (AUC=0.9013)
iTerm-PseKNC (AUC=0.8321)


B.subtilis
Sensitivity
Specificity
iterb-PPse (AUC=0.9929)
iTerm-PseKNC (AUC=0.9443)

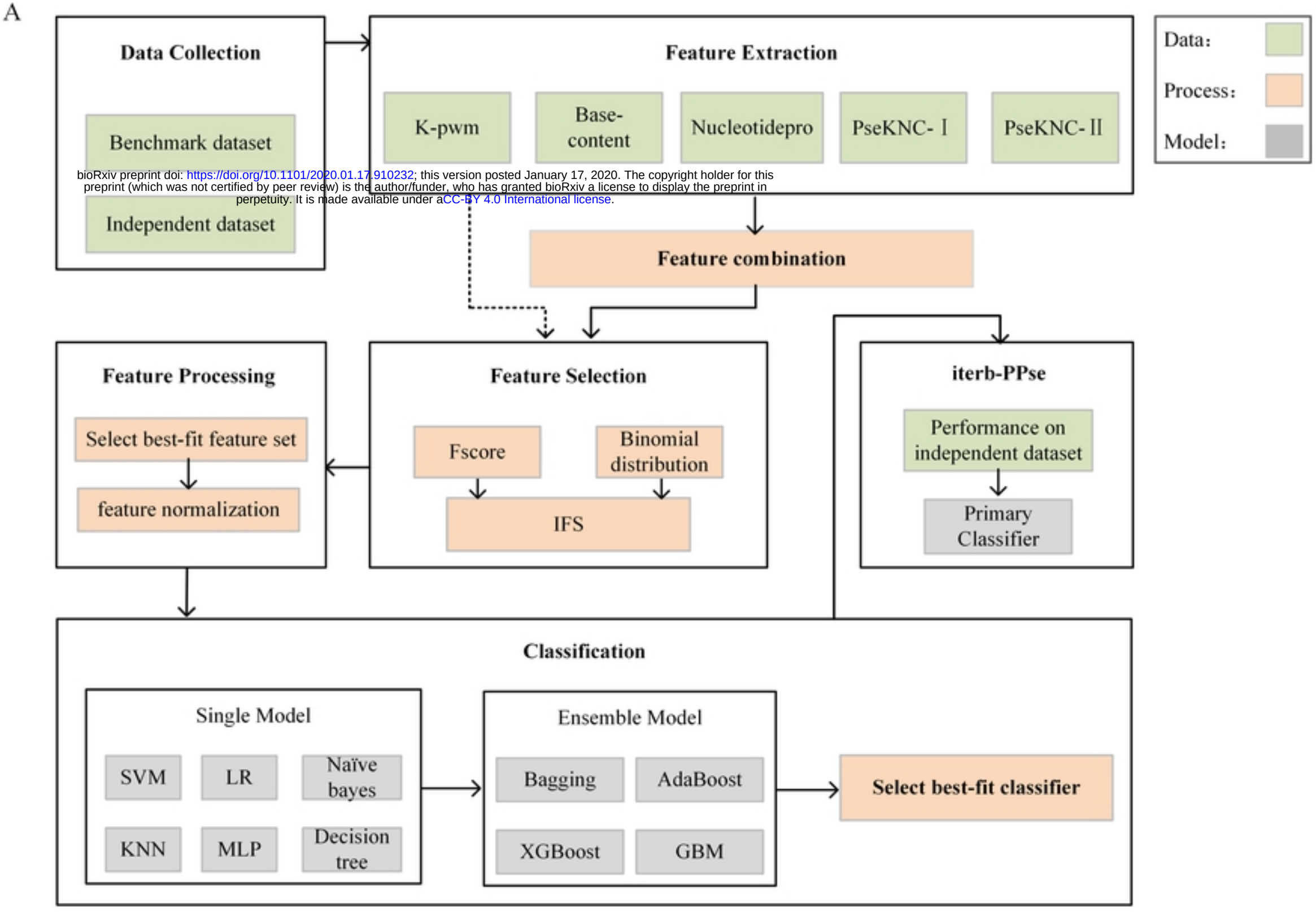

iterb-PPse
iTerm-PseKNC
Acc
Benchmark dataset
E. coli
B. sutbilis
Dataset

Figure

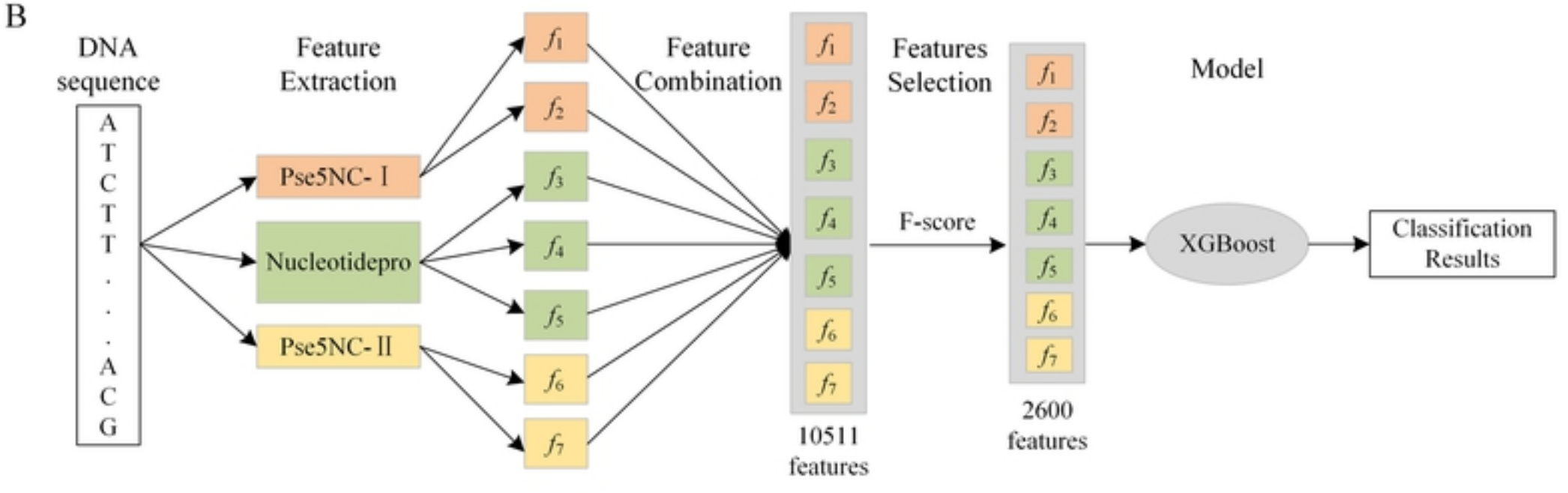

pterb-PPseKNC
iterb-PPse: Identification of transcriptional terminators in bacterial
Please enter the data in the following format:
>1
ACTGCTGAATGTGTACGTCAGT
>2
CGTGATCGCTAGCTAG
Clear
Submit

Figure

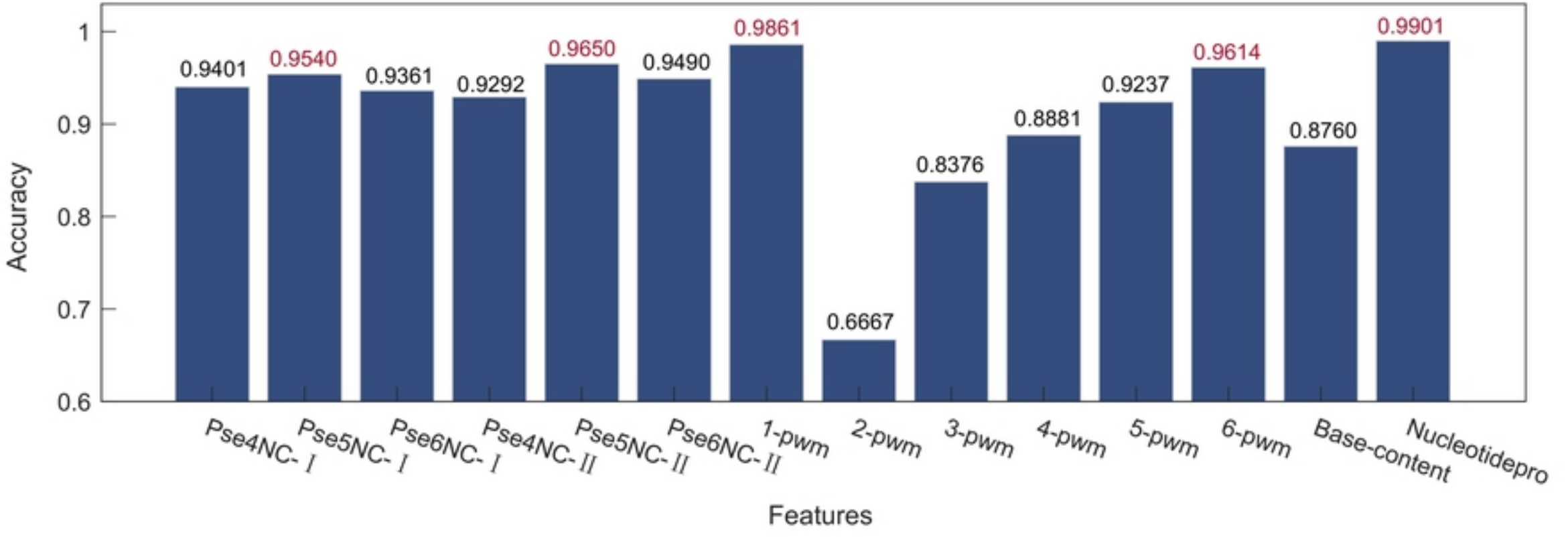
A



B



Figure


Accuracy
1
0.9
0.8
0.7
0.6
0.9401
0.9540
0.9361
0.9292
0.9650
0.9490
0.9861
0.6667
0.8376
0.8881
0.9237
0.9614
0.8760
0.9901
Pse4NC-I
Pse5NC-I
Pse6NC-I
Pse4NC-II
Pse5NC-II
Pse6NC-II
1-pwm
2-pwm
3-pwm
4-pwm
5-pwm
6-pwm
Base-content
Nucleotidepro
Features

Figure