MICROBIAL GENOMICS

RESEARCH ARTICLE

Wang *et al.*, *Microbial Genomics* 2021;7:000611

DOI 10.1099/mgen.0.000611

Prediction of prokaryotic transposases from protein features with machine learning approaches

Qian Wang[1], Jun Ye[2], Teng Xu[3], Ning Zhou[4], Zhongqiu Lu[4,*] and Jianchao Ying[4,5,*]

Abstract

Identification of prokaryotic transposases (Tnps) not only gives insight into the spread of antibiotic resistance and virulence but the process of DNA movement. This study aimed to develop a classifier for predicting Tnps in bacteria and archaea using machine learning (ML) approaches. We extracted a total of 2751 protein features from the training dataset including 14852 Tnps and 14852 controls, and selected 75 features as predictive signatures using the combined mutual information and least absolute shrinkage and selection operator algorithms. By aggregating these signatures, an ensemble classifier that integrated a collection of individual ML-based classifiers, was developed to identify Tnps. Further validation revealed that this classifier achieved good performance with an average AUC of 0.955, and met or exceeded other common methods. Based on this ensemble classifier, a stand-alone command-line tool designated TnpDiscovery was established to maximize the convenience for bioinformaticians and experimental researchers toward Tnp prediction. This study demonstrates the effectiveness of ML approaches in identifying Tnps, facilitating the discovery of novel Tnps in the future.

DATA SUMMARY

All the protein sequences used in this study were obtained from the National Centre for Biotechnology Information (NCBI) RefSeq, Swiss-Prot, and ISfinder databases. The TnpDiscovery program is publicly available at https://github.com/ying-jc/TnpDiscovery.

INTRODUCTION

Transposons (Tns) are DNA elements that can move from the DNA molecule to other places on the same DNA or other DNA molecules [1]. In bacteria, Tns are divided into four categories: insertion sequence (IS), composite Tns, non-composite Tns, and transposable phage Mu [2, 3]. Tns can transfer from a plasmid to other plasmids or from a DNA chromosome to a plasmid and vice versa that cause the transmission of antibiotic resistance genes in bacteria [4, 5]. Drug resistance genes carried by Tns and their transmission among bacteria is the most serious challenge in the treatment of infectious diseases [6]. In addition to antibiotic resistance, Tns can also cause an increase and decrease of bacterial virulence [7]. Transposase (Tnp) is an enzyme that binds to the end of a transposon and catalyses the movement of DNA segments and the associated genes, to new DNA sites by a cut and paste mechanism or a replicative transposition mechanism [1, 8]. Tnps have dramatic biological and evolutionary consequences that shape the genomes of organisms [8]. Identification of Tnps is a key process to understanding the role of Tns in the spread of antibiotic resistance and virulence. It is an important task to accurately and rapidly identify Tnps from large-scale proteins.

At present, several bioinformatics methods have been proposed to identify and annotate Tnps, which are mainly based on the sequence search of the Tnp library. ISfinder [9] is a dedicated database for bacterial ISs as well as Tnp

Received 24 November 2020; Accepted 18 May 2021; Published 26 July 2021

Author affiliations: [1]Department of Clinical Laboratory, Wenzhou People's Hospital, The Third Affiliated Hospital of Shanghai University, The Third Clinical Institute Affiliated to Wenzhou Medical University, Wenzhou, PR China; [2]Department of Clinical Laboratory, The Second Affiliated Hospital of Guizhou Medical University, Kaili, PR China; [3]Institute of Translational Medicine, Baotou Central Hospital, Baotou, PR China; [4]Wenzhou Key Laboratory of Emergency, Critical Care, and Disaster Medicine, Department of Emergency, The First Affiliated Hospital of Wenzhou Medical University, Wenzhou, PR China; [5]Central Laboratory, The First Affiliated Hospital of Wenzhou Medical University, Wenzhou, PR China.

***Correspondence:** Zhongqiu Lu, lzq640815@163.com; Jianchao Ying, yingjc@spaces.ac.cn; yingjc@wmu.edu.cn

Keywords: prokaryotic transposase; protein classifier; protein feature; feature selection; machine learning.

Abbreviations: AUC, the area under curve; CV, cross-validation; DL, deep learning; EC, ensemble classifier; GBM, gradient boosting machine; IS, insertion sequence; LASSO, least absolute shrinkage and selection operator; MCC, Matthews correlation coefficient; MI, mutual information; ML, machine learning; ROC, receiver operating characteristic; Tnp, transposase; t-SNE, t-distributed stochastic neighbour embedding; XGB, extreme gradient boost.

Data statement: All supporting data, code and protocols have been provided within the article or through supplementary data files. Five supplementary tables are available with the online version of this article.

protein sequences and is commonly used for Tnp identification by searching with the BLAST program. A web application termed ISSaga [10] was developed to automate IS annotation in complete genomes using BLAST seeded with the ISfinder sequences and classify them into families. IScan [11] is another application that makes use of BLAST to scan whole genomes for ISs and includes in its prediction pipeline searches for transposases and inverted and direct repeats. TnpPred [12] is a web service that provides HMM profiles of 19 prokaryotic transposase families and uses the HMMER search method for Tnp prediction. The library-based method has some limitations in identifying transposable elements. First, the efficiency of the library-based method is critically dependent on the quality and the exhaustiveness of the database used. Second, library-based methods are unable to identify new families that display no similarities with existing families [13]. For these reasons, the *de novo* method which relies on the structural properties of the transposable elements and does not require a set of reference sequences to work becomes an alternative method. Kamoun *et al.* [13] took advantage of the structural properties of the elements and combined with profile HMM searches to improve the discovery of ISs and miniature inverted-repeat transposable elements.

With the advent of the era of big data, machine learning (ML) techniques have been increasingly used as a powerful approach to identify important proteins in biology [14]. Although this method cannot replace biological experiments, it improves the accuracy of prediction and provides more clues for biological experiments [15]. There are many examples of protein identification using ML approaches, and most of them show good predictive performance. ACP-DL was developed based on deep learning to predict anticancer peptides and remarkably outperformed other comparison methods with high accuracy and satisfied specificity on benchmark datasets [16]. Han *et al.* applied support vector machine and random forest methods to predict ion channels and their types from protein sequences [17]. Hou *et al.* proposed a model combining 188D features with random forest to identify ABC transporters [15].

Considering that ML approaches have not been used for Tnp identification, we sought to explore their applicability in the development of classifiers for this behaviour. In this study, we extracted a wide variety of protein features, and further performed feature selection to select signatures for Tnp prediction based on ML algorithms. A set of ML-based classifiers was developed for these signatures and then integrated as an ensemble classifier to identify potential Tnps. To the best of our knowledge, this is the first study to construct an ML-based classifier for Tnp prediction. We envisage this classifier will be widely used to facilitate the discovery of novel Tnps.

Impact Statement

The identification and study of transposases are helpful to reveal the dissemination mechanism of antibiotic resistance and virulence in prokaryotes. However, the current identification methods are mainly based on library comparison, which limits the further discovery of novel transposases. In this study, we proposed a *de novo*-based approach for transposase prediction that utilizes machine learning to make predictions based on protein sequence features rather than relying on reference sequence comparisons. To facilitate the processing of large-scale protein sequences obtained from prokaryotic genomes or metagenomes, we developed a standalone command-line tool called TnpDiscovery. Compared with the library-based methods, TnpDiscovery shows comparable prediction accuracy, which may help to further improve the prediction performance of existing methods for transposases. This study demonstrates the effectiveness of machine learning approaches in identifying transposases, facilitating the discovery of novel transposases in the future.

METHODS

Data collection and preprocessing

Tnp protein sequences were obtained from the ISfinder [9] database and by retrieving Swiss-Prot (https://www.uniprot.org/) and NCBI RefSeq (https://www.ncbi.nlm.nih.gov/refseq/) databases using 'transposase' as the keyword. The resulted sequences were filtered and only sequences belonging to the archaea and bacteria were retained. We used all reviewed complete protein sequences of archaea and bacteria from Swiss-Prot as the control sequences by removing the Tnps. The following measures were taken to obtain reliable high-quality datasets. Firstly, the protein sequences containing blurred disabilities, such as those with amino acids 'X', 'Z', 'B', 'J', 'O', and 'U', as well as '*' were discarded. Secondly, the sequences of protein fragments were removed. Thirdly, to avoid any similarity bias in the following analysis, CD-HIT [18] program was used for protein clustering to maximally remove redundant data. All the collected proteins were clustered with a 40% sequence identity as the cut-off. The clusters containing both Tnps and controls were excluded because of their ambiguity, and the representative sequences in each of the remaining clusters were extracted to form the final dataset. This procedure produced a total of 18565 Tnps and 33457 controls (Table S1, available in the online version of this article). Given the imbalance between the number of Tnps and controls, we randomly selected 14852 (80%) of the Tnp group and the same number of controls as the training dataset (n=29704). Then, the remaining Tnps and 3713 randomly selected controls from the remaining 18605 controls formed a validation dataset (n=7426). This operation was conducted ten times and ended up with ten validation datasets. To make a relatively unbiased comparison of the various methods, we further generated a testing dataset containing 3530 Tnps that are not from the training dataset or ISfinder database, as well as 3530 controls. In this way, the number of Tnps and controls in each dataset is the same, and the sequences in the training

and validation/testing datasets do not overlap. In order to test the predictive performance of the proposed classifier for Tnp fragments, we randomly truncated the 3713 complete Tnp sequences in the validation dataset to set up a dataset. The sequences with a length less than 31 aa were removed according to the requirement of feature extraction. This whole process was repeated ten times, and a total of ten Tnp fragment datasets were obtained. The prokaryotic genomes used for speed and memory testing of the stand-alone tool were downloaded from the NCBI Genome database.

Data standardization is a common requirement for many ML algorithms, which might behave badly if the individual features do not conform to the standard normal distribution. In this study, all features of the training dataset were standardized by the Z-score method, that is, removing the mean and scaling to unit variance, then the same mean and standard deviation were stored to be used for validation and testing datasets using transform.

Feature extraction and visualization

A protein's amino acid sequence contains important intrinsic information that dictates its properties, such as composition, permutation, and combination modes of amino acids, orders of amino acids, and physicochemical properties, etc. [19]. Considering that each type of protein feature may contribute to the identification of Tnps, we incorporated a wide range of properties in this study to explore the optimal feature set for Tnp prediction. Through the implementation of iFeature [20], we extracted a comprehensive profile of 18 protein descriptors that encompass 2751 sequence features. These descriptors contain amino acid composition [21] (AAC), dipeptide deviation from expected mean [22] (DDE), and dipeptide composition [21] (DPC) in the category of amino acid composition; composition of k-spaced amino acid group pairs [23] (CKSAAGP), grouped amino acid composition [24] (GAAC), grouped dipeptide composition [20] (GDPC), and grouped tripeptide composition [20, 21] (GTPC) in the category of grouped amino acid composition; geary [25], moran [26], and normalized Moreau-Broto [27] (NMBroto) in the category of autocorrelation; composition [28] (CTDC), transition [28, 29] (CTDT), and distribution [28, 29] (CTDD) in the category of C/T/D; conjoint triad [30] (CTriad) in the category of conjoint triad; quasi-sequence-order descriptors [31] (QSOrder), and sequence-order-coupling number [32] (SOCNumber) in the category of quasi-sequence-order; amphiphilic pseudo-amino acid composition [33, 34] (APAAC), and pseudo-amino acid composition [33, 34] (PAAC) in the category of pseudo-amino acid composition.

The t-distributed stochastic neighbour embedding [35] (t-SNE), a dimensionality reduction tool based on non-linear manners, is particularly good at the visualization of high-dimensional datasets. Therefore, we used t-SNE to reduce the features of protein sequences to two-dimensional features and then visualized them in 2D. t-SNE was implemented in R-package Rtsne with dims=2 and perplexity=50. Unsupervised hierarchal clustering of sequences according to the pattern of selected features was performed using the R-package pheatmap, and the chi-square test was used to examine the significance of differences in sequence type between the clustered groups.

Feature selection

Since not all features contribute to the identification of Tnp proteins, aggregating all features may even decrease the predictive performance of the classifier [36]. For this, feature selection is a common method to obtain a panel of features with satisfying predictive performance. In this study, we applied a strategy that combines the results of two ML-based selection methods, mutual information (MI) and least absolute shrinkage and selection operator (LASSO), to select the features for classifier development. MI [37] is a univariate filtering method used to capture any relationship between each feature and label, including linear and non-linear relationship. It is equal to zero if and only if the feature is independent of the label, and higher values mean a stronger correlation. The features with an estimated MI value higher than 0 were retained for the following analysis. LASSO was employed to further select the feature set in the R-package glmnet, and the tuning parameters were determined according to the expected generalization error estimated from ten-fold cross-validation (CV). Since the results of LASSO were strongly dependent on the arbitrary choice of a random sample split for the data, we reduced this randomness by conducting this step ten times and averaging the error curves to achieve robust results.

Classifier construction

The optimal set of features were analysed using three popular classification algorithms, including deep learning (DL), gradient boosting machine (GBM), and extreme gradient boost (XGB). In this section, DL is based on a multi-layer feedforward artificial neural network that is trained with stochastic gradient descent using back-propagation. As one of the most common types of deep neural networks, it is suitable for tabular data. GBM is an ML technique for classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It can obtain good predictive results through increasingly refined approximations. XGB, a supervised learning algorithm based on the GBM algorithm, implements a process called boosting to yield accurate models. It can provide parallel tree boosting that solves many data science problems in a fast and accurate way. All these three algorithms were implemented in the H2O program (https://github.com/h2oai/h2o-3) which is an open-source, in-memory, distributed, fast, and scalable ML and predictive analytics platform. For each algorithm, a ten-fold CV is used to validate a classifier internally, and a random grid search was performed to optimize the hyperparameters. This process was performed ten times and ended up with a total of 30 classifiers for these three algorithms. It is noteworthy that the classifier's CV metrics were computed based on the combination of the ten holdout predictions from each CV step, rather than taking the average of the ten validation metrics. The ensemble ML

method uses multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms. Hence, we tried to find the optimal combination of a collection of prediction algorithms using a process called stacking.

Performance comparison and evaluation

Two commonly used methods for Tnp prediction - TnpPred [12] database with HMM search (termed TnpPred method) and ISfinder [9] database with BLAST search (termed ISfinder method) - were included in the method comparison using the testing dataset. In detail, hmmscan (HMMER 2.3.2, http://hmmer.org/) was used for searching the Hidden Markov Model (HMM) profiles of TnpPred for homologs of protein sequences, and for making sequence alignments. BLASTp (BLAST 2.6.0+, https://blast.ncbi.nlm.nih.gov/Blast.cgi) was deployed to compare protein sequences to the ISfinder database and calculate the statistical significance of matches.

To evaluate the performance of the features (or classifiers) in the prediction of Tnps, the Receiver Operating Characteristic (ROC) analysis was performed, and the area under curve (AUC) was calculated using the R-package pROC. Here, we assumed that the larger AUC of the ROC curve implies the better. In the section of performance comparison, the AUCs of different methods were compared with the bootstrap method. As the evaluation metrics of the classifier may rely on the selection of the cut-off point, we used the Youden method to determine the best cut-off value in the ROC curve and then applied it to sequence grouping. The confusion matrix was generated to describe the performance of the classifier in the Tnp prediction. The predictive performance was also evaluated by five metrics, including prediction accuracy (ACC), sensitivity (SN), specificity (SP), F-value, and Matthews correlation coefficient (MCC). These evaluation metrics are defined as follows:

$$ACC = \frac{TP+TN}{TP+TN+FP+FN}$$

$$SN = \frac{TP}{TP+FN}$$

$$SP = \frac{TN}{TN+FP}$$

$$F-value = \frac{2TP}{2TP+FN+FP}$$

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

where TP, TN, FP, and FN indicate the numbers of true positives, true negatives, false positives, and false negatives, respectively.

RESULTS

Identification of feature signatures for predicting Tnps

After the preprocessing procedure, the training dataset (n=29704) enrolled in this study contains 14852 Tnp and 14852 control protein sequences. A total of 18 structural and physicochemical descriptors, including 2751 protein features, were extracted from these sequences. To explore the predictive utility of these features in the classification of Tnps, each feature was assessed by ROC analysis (Fig. 1a, Table S2). There are 483 features with an AUC value greater than 0.60, among which the AUC value of three features even exceeded 0.80, indicating that these features have a certain potential for predicting Tnps.

Nevertheless, we further explored whether feature combinations have better predictive power than these single features. We reduced the dimensions of all features to two using the t-SNE algorithms, such that they could be projected and visualized in 2D. The differences of Tnp and control sequences in the higher-dimensional space can be represented by their mutual distances in the 2D space. As shown in Fig. 1B, the distribution of Tnp and control sequences is generally aggregated separately, suggesting that the combined features could indeed provide informative characteristics and patterns for Tnp classification.

Considering the limited contribution of some features to the characteristics of Tnps, it may even reduce the predictive performance of classification. Therefore, it is necessary to conduct a feature selection to determine the optimal feature set for Tnp prediction. We first used the MI method to examine the correlation between each feature and the sequence type, retaining 2695 features with positive MI values. Subsequently, the LASSO method was applied for further selection based on these 2695 features. This process was performed ten times to stabilize the results, which yields 75 features from 15 descriptors (Fig. 1c, Table 1). Of these descriptors, DDE had the highest frequency (19 features), followed by CKSAAGP (18 features) and CTriad (ten features). The three descriptors (APAAC, NMBroto, and SOCNumber) had no selected features in this study. As expected, the Tnps and controls were almost separated clearly through the visualization based on these 75 features (Fig. 1d). Obviously, the aggregation of different sequences based on the selected features is more concentrated than that based on all features. This suggests that these features contain enough information to distinguish Tnps from other proteins. Moreover, unsupervised hierarchal clustering of the training cohort sequences according to the pattern of these 75 features was performed, and the heatmap showed that most of the same types of sequences clustered together apart from a few exceptions (Chi-square test P<0.001, Fig. 1e). These results indicate that these 75 features may serve as signatures for Tnp prediction.

Construction of the classifiers for predicting Tnps

To develop a classifier with the best performance for discriminating Tnps from other proteins, three ML algorithms, including DL, GBM and XGB were applied to the training dataset in this section. For each algorithm, we trained ten prediction classifiers based on the 75 feature signatures with optimally-tuned parameters using the random grid search. All these 30 classifiers were subjected to ten-fold CV, which was then evaluated by ROC analysis and other classification

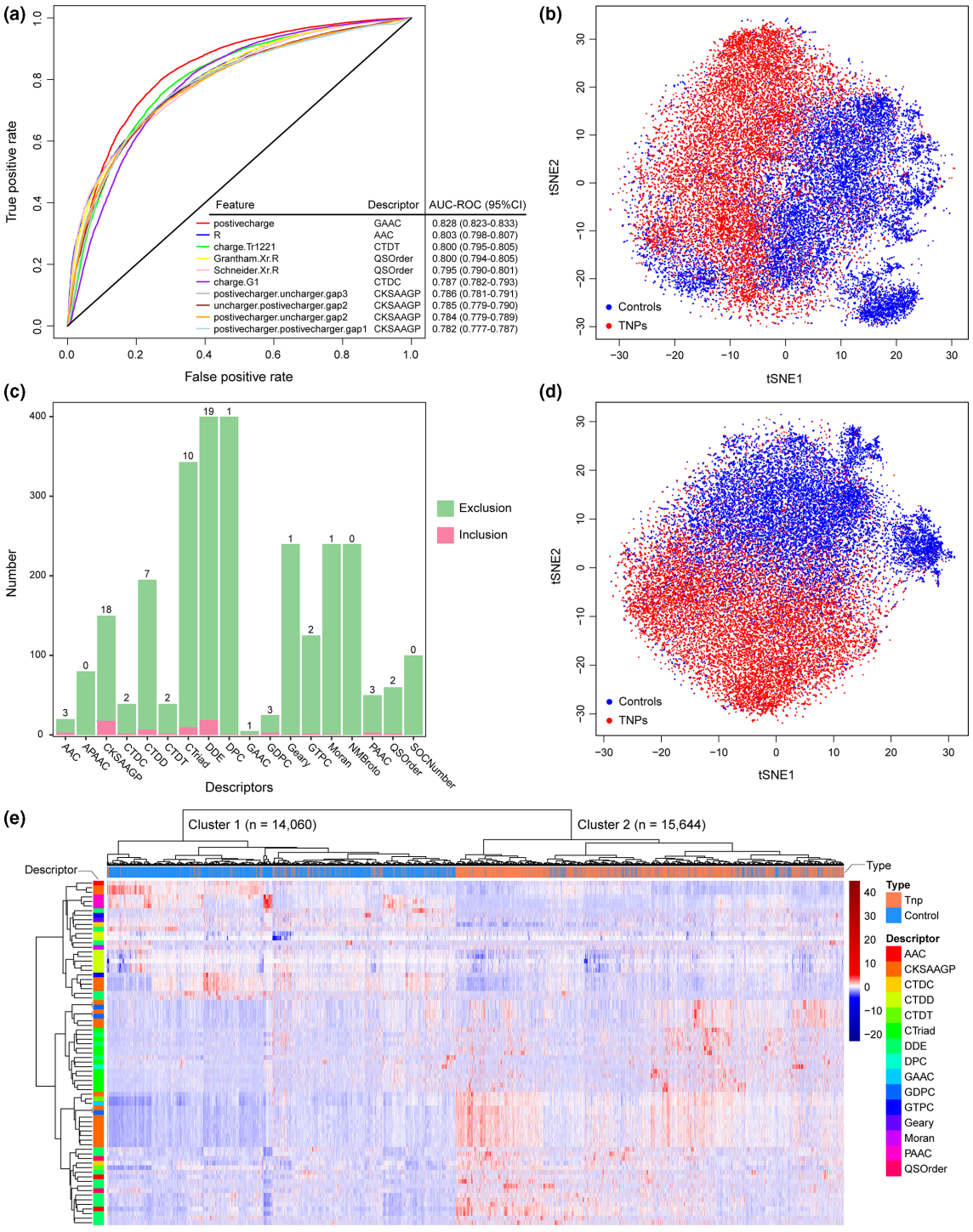

Fig. 1. Predictive potential of the 2751 protein features in the classification of Tnps. (a) ROC curves for the top ten features with the best predictive performance in the training dataset. (b) Embedding of 2751 features from 18 descriptors using t-SNE. The red and blue dots represent Tnps and controls, respectively. (c) Statistics of the features selected by both MI and LASSO methods. The number at the top of the bar represents the number of the selected features in each protein descriptor. (d) Embedding of 75 features from 15 descriptors using t-SNE. The red and blue dots represent Tnps and controls, respectively. (e) Unsupervised hierarchical clustering and heatmap of the training dataset based on the 75 features selected.

Table 1. The 75 feature signatures selected in this study

Descriptor	Dimension	Selection	Selected features
AAC	20	3	H, R, V
APAAC	80	0	–
CKSAAGP	150	18	uncharger.postivecharger.gap5, postivecharger.aromatic.gap0, uncharger.postivecharger.gap2, postivecharger.uncharger.gap0, postivecharger.uncharger.gap2, aromatic.postivecharger.gap4, alphaticr.negativecharger.gap3, alphaticr.alphaticr.gap3, postivecharger.uncharger.gap4, negativecharger.alphaticr.gap4, alphaticr.negativecharger.gap5, postivecharger.postivecharger.gap1, postivecharger.aromatic.gap1, uncharger.postivecharger.gap1, aromatic.postivecharger.gap0, alphaticr.alphaticr.gap2, uncharger.postivecharger.gap3, postivecharger.uncharger.gap3
CTDC	39	2	solventaccess.G3, polarizability.G2
CTDD	195	7	charge.2.residue100, charge.3.residue25, charge.2.residue75, hydrophobicity_FASG890101.1.residue75, polarity.3.residue100, charge.3.residue75, polarity.3.residue75
CTDT	39	2	charge.Tr1221, hydrophobicity_ENGD860101.Tr1221
CTriad	343	10	g3.g3.g4, g5.g4.g3, g5.g3.g5, g3.g5.g3, g5.g5.g3, g2.g5.g5, g3.g5.g4, g4.g5.g5, g2.g5.g3, g5.g5.g4
DDE	400	19	DR, RL, KR, RC, HL, HR, DI, VI, GE, RQ, RT, RK, RS, CL, RR, YS, PF, GD, RW
DPC	400	1	YN
GAAC	5	1	postivecharge
GDPC	25	3	postivecharger.aromatic, aromatic.postivecharger, postivecharger.uncharger
Geary	240	1	CHAM810101.lag3
GTPC	125	2	negativecharger.negativecharger.alphaticr, aromatic.negativecharger.alphaticr
Moran	240	1	CIDH920105.lag4
NMBroto	240	0	–
PAAC	50	3	Xc1.I, Xc1.V, Xc1.F
QSOrder	60	2	Grantham.Xr.C, Grantham.Xr.W
SOCNumber	100	0	–
Total	2751	75	

metrics. As can be seen, all these three algorithms performed well in the CV test, with their average AUC greater than 0.940, exceeding the performance of any single feature (Figs 1a and 2a). Notably, the GBM algorithm performed best in terms of AUC (0.950±0.002), ACC (0.883±0.002), F-value (0.885±0.002), and MCC (0.766±0.005) (Fig. 2a), which suggests that it may be the most suitable modelling method for Tnp prediction. Thus, the GBM classifier with the best performance termed GBM-best was chosen as the best candidate. Besides that, we also constructed two stacked ensemble classifiers (ECs) - one based on all trained individual classifiers (termed EC-all), another one on the best classifier of each algorithm (termed EC-best). Unexpectedly, the ECs performed better than GBM-best, and EC-all achieved the optimal overall performance with AUC of 0.956, ACC of 0.891, SN of 0.903, SP of 0.880, F-value of 0.893, and MCC of 0.783 (Fig. 2b). Based on the entire training dataset, the AUC values of all three classifiers reached or exceeded 0.994 (Fig. 2c). Subsequently, the best point for each classifier was determined according to the ROC curves for the sequence classification of the training dataset. From the plots of the confusion matrix, the overall correct classification rates of GBM-best, EC-all, and EC-best were 97.2, 96.1 and 96.0%, respectively (Fig. 2d).

Validation and evaluation of the Tnp classifiers

GBM-best, EC-all, and EC-best were further validated using the ten validation datasets. Similar to the CV test results of the training dataset, EC-all (AUC: 0.955±0.001) performed slightly better than GBM-best (AUC: 0.951±0.001) and EC-best (AUC: 0.954±0.001) (Fig. 3). According to the cut-off values identified from the training dataset, the average evaluation metrics of EC-all reached 0.891 (ACC), 0.876 (SN), 0.906 (SP), 0.889 (F-value), and 0.782 (MCC), respectively (Fig. 3, Table S3). Therefore, EC-all was chosen as the optimal classifier for predicting Tnps from all our constructed classifiers in this work.

Subsequently, ROC curve analysis was performed on the testing dataset to compare the predictive power of EC-all with two existing prediction methods - TnpPred and ISfinder (Fig. 4A). As shown in Fig. 4B, the AUC of EC-all

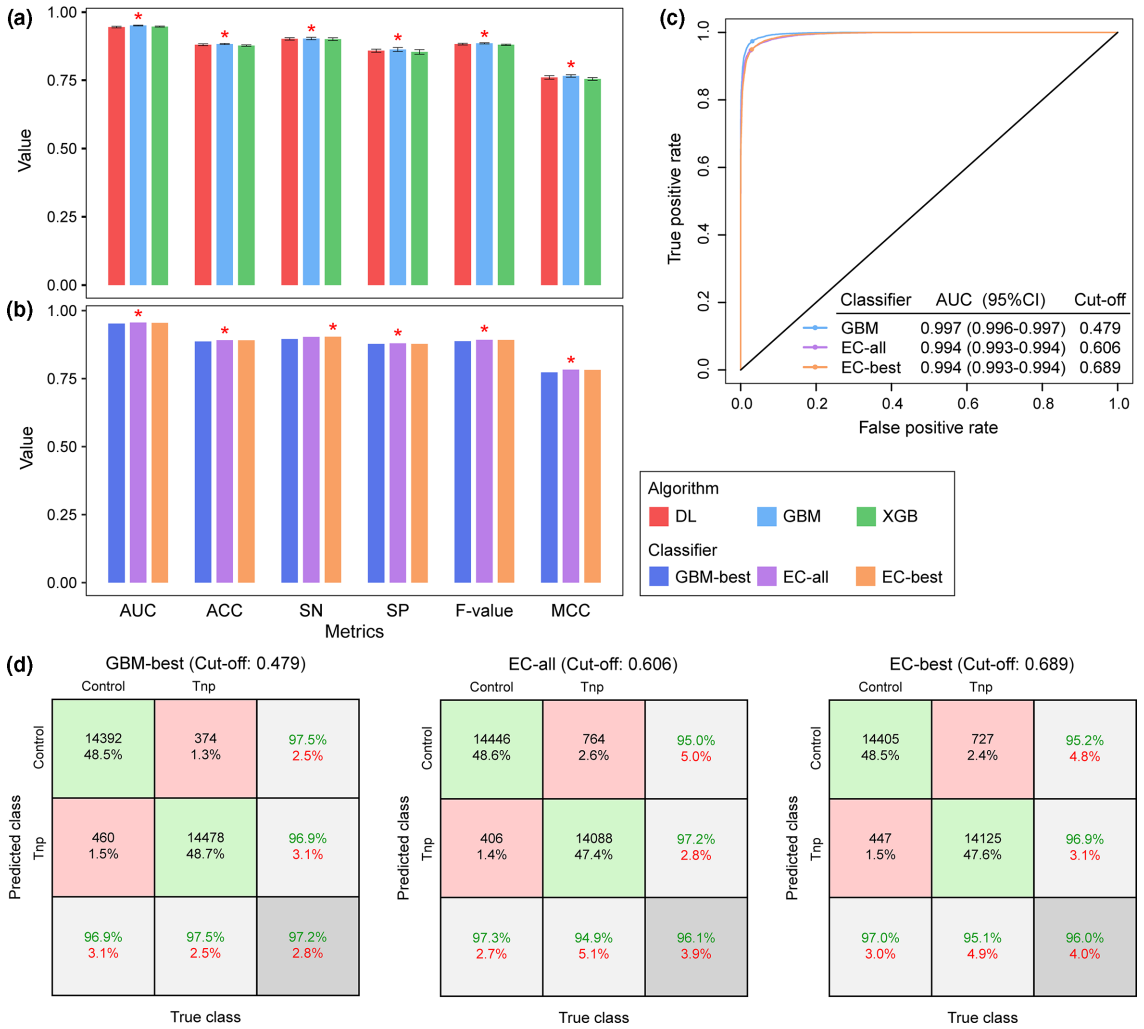

Fig. 2. Classifier construction for predicting Tnps using the training dataset. (a) Classification metrics for evaluating the performance of DL, GBM, and XGB algorithms based on ten-fold CV. (b) Classification metrics for evaluating the performance of the best performing classifier (GBM-best) and two ensemble classifiers based on ten-fold CV. The red star indicates the best performance amongst these algorithms or classifiers. ROC curves (c) and the confusion matrix plots (d) of GBM-best and two ensemble classifiers based on the entire training dataset. On the confusion matrix plot, the rows correspond to the predicted class and the columns correspond to the true class. The green cells correspond to observations that are correctly classified. The red cells correspond to incorrectly classified observations. Both the number of observations and the percentage of the total number of observations are shown in each cell. The column on the far right of the plot shows the percentages of all the samples predicted to belong to each class that are correctly (precision) and incorrectly (false discovery rate) classified. The row at the bottom of the plot shows the percentages of all the samples belonging to each class that are correctly (recall) and incorrectly (false negative rate) classified. The cell at the bottom right of the plot shows the overall accuracy.

was significantly higher than that of TnpPred and slightly higher than that of ISfinder. This result demonstrates that EC-all is a novel classifier with better predictive ability than these commonly used approaches. It is worth noting that the combination of multiple methods also performed well. The combination of EC-all and ISfinder achieved an AUC of 0.966, which was significantly higher than that of other methods, indicating that EC-all can help to improve the predictive accuracy for the existing methods (Fig. 4B).

Considering that there may be a large number of Tnp fragments in prokaryotic genomes, it is also important to be able to identify these incomplete Tnps. Ten datasets were established by randomly splitting the complete Tnp sequences to test the predictive performance of EC-all for Tnp fragments. The results showed that EC-all could identify about 84% of these Tnps, indicating that this classifier could be used for the prediction of Tnp fragments (Table S4).

Implementation of a stand-alone tool of Tnp classifiers

To facilitate the processing of large-scale protein sequences obtained from prokaryotic genomes or metagenomes, a

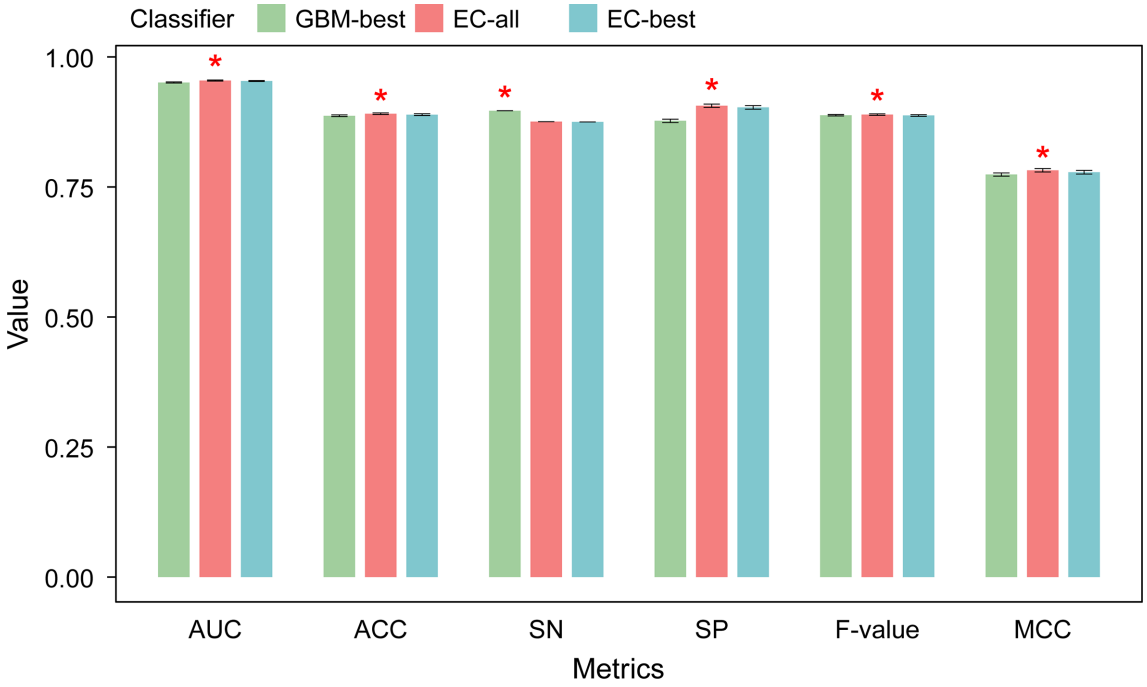

Fig. 3. Classifier evaluation using the validation datasets. Classification metrics were used to evaluate the performance of GBM-best, EC-all, and EC-best in ten randomly produced validation datasets. The red star indicates the best performance amongst these classifiers.

standalone command-line tool called TnpDiscovery has been developed with built-in EC-all as well as GBM-best. TnpDiscovery takes any number of amino acid sequences as input, automatically performs feature extraction and normalization, invokes a Tnp classifier, and finally returns the estimated probability and binary classification of Tnp for the given proteins. TnpDiscovery is freely available at https://github.com/ying-jc/TnpDiscovery.

The running speed and memory usage of TnpDiscovery were tested by using proteins from five bacterial genomes (Table S5). The results showed that it took about 10 min to predict a 4000-protein genome using 15 processes on a workstation computer. There is little difference in running time between the two built-in classifiers, but EC-all requires more memory than GBM-best. Given the accuracy of these two classifiers

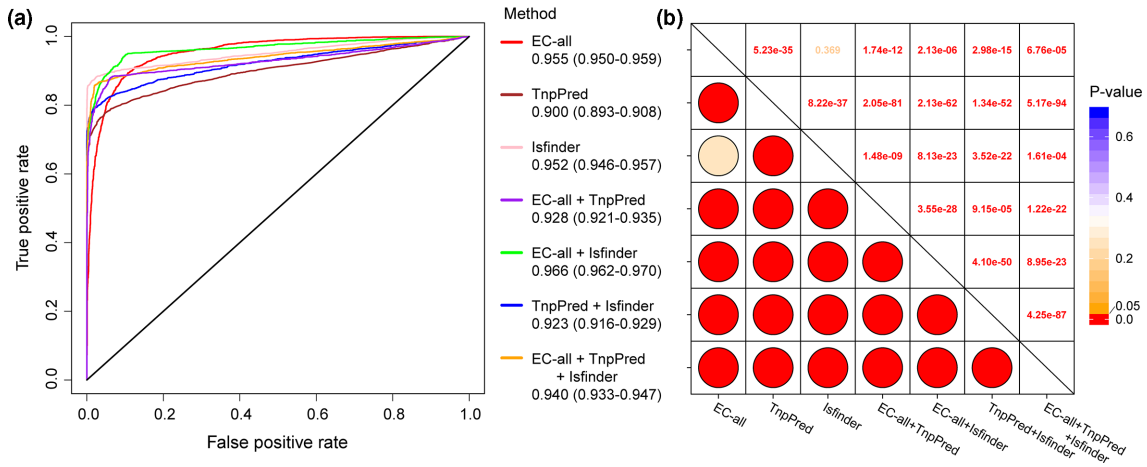

Fig. 4. Performance comparison between EC-all and two existing approaches for Tnp prediction using the testing dataset. Shown are ROC curves (a) and significance matrix plots (b) for these three methods and their corresponding combined methods. The entry values of the significance matrix represent the p-values for the comparison of the AUC of two ROC curves.

is comparable, we recommend using GBM-best as an option when computing resources are limited.

DISCUSSION

Prediction of Tnps in bacteria could help understand the spread of antibiotic resistance and virulence, as well as the process of DNA movement. The purpose of this study is to develop a classifier for Tnp prediction based on protein features with ML approaches. We incorporated a wide range of complementary and heterogeneous features of proteins in this study to accomplish this task. Our results showed that combining multiple features may provide more useful information for Tnp prediction than a single feature alone. Therefore, we further selected a set of features with satisfying predictive performances for Tnp prediction. ML approaches have been widely used in the selection of markers and construction of prediction models, and have been shown to improve the predictive performance of models in various human diseases [38, 39] and protein identification tasks [17, 40]. In this study, we applied a strategy of combining both MI and LASSO methods to reduce the dimension of protein features and finally selected 75 features as signatures for predicting Tnps. We leveraged three popular ML algorithms, DL, GBM, and XGB, to construct a classifier of Tnp prediction based on the selected feature signatures. And we found that each ML approach performed well on the datasets used in this study, but the GBM method appeared to be superior. Furthermore, we integrated these single ML-based classifiers to build stacked ensemble classifiers, which performed better than any individual ML-based classifier. Validation is an essential process to verify the predictive performance of the models, which also helps control the possibility of model overfitting [39]. In this study, the validation process was carried out using relatively large-scale sequences in both the training and validation datasets. In the step of classifier construction, we implemented a ten-fold cross-validation method in the training dataset to validate the classifiers and determine the best classifiers. For further validation, we set up ten sets as validation datasets to evaluate the robustness of these classifiers. Taking all these findings together, we finally developed a command-line prediction tool named TnpDiscovery with the best-performing classifier EC-all, with GBM-best as an alternative.

Before this study, Riadi *et al.* [12] proposed a web service termed TnpPred that supplements and extends currently available programs and HMM Profiles for the prediction of 19 prokaryotic transposase families. ISfinder database was set up by Siguier *et al.* [9] to collect bacterial insertion sequences. These two databases are commonly used for Tnp identification by homologous alignment using HMMER or BLAST programs. Since one of our dataset sources is the ISfinder database, for the fairness of the comparison, we built a testing dataset for the method comparison that does not contain sequences from ISfinder. Compared with these library-based methods, TnpDiscovery shows comparable prediction accuracy, which may help to further improve the prediction performance of existing methods for Tnps. However, unlike these library-based methods, TnpDiscovery is a *de novo* method for Tnp prediction that makes predictions based on protein sequence characteristics rather than relying on reference sequence comparisons. Therefore, TnpDiscovery does not need to prepare a sequence library in advance, and is easier to deploy and implement. Although TnpDiscovery cannot infer which IS family the Tnp belongs to as the library-based method can, it can be used to identify novel Tnps because it does not need to consider the similarity information of protein sequences. Besides, the running speed and memory consumption of TnpDiscovery are acceptable, and it is suitable for the primary screening of a large number of protein sequences obtained from prokaryotic genomes or metagenomes.

There are several limitations to this study. First, we still considered only a limited number of protein features, thus, the features identified here may not be the best signatures for Tnp prediction. Nevertheless, the focus of this study was to determine whether the ML approach could be applied to develop a classifier for Tnp prediction based on protein features. The proposed classifier exhibited potentially powerful abilities in the prediction of Tnps, which meets our requirements. Under such circumstances, we believe that the number of features included in this study is large enough to accomplish our goal, although we do agree that more features would be better. The second issue is that only a few ML algorithms have been applied to feature selection and classifier construction. Indeed, if more algorithms are applied, better predictive classifiers might be obtained, but this will greatly increase the time cost and computing power needed to analyse the data. Given this concern, we just examined several popular ML algorithms in the analyses. Furthermore, we implemented an ensemble model construction strategy by integrating multiple individual ML-based classifiers to achieve better predictive performance. Third, the comparison of TnpDiscovery with the existing methods may be biassed. The ISfinder database used by BLAST is not comprehensive enough, which may affect the sensitivity of BLAST. And the TnpPred database has not been updated for a long time. However, this has little impact on this study, and the purpose of this comparison is not to find the best method but to evaluate whether the classifier is effective.

In summary, we proposed a stacked ensemble classifier integrating a collection of individual ML-based classifiers that could accurately and effectively identify Tnp proteins in bacteria and archaea. On this basis, we implemented a free stand-alone tool called TnpDiscovery, to meet users' specific demands. We believe that this program can be a useful and alternative tool to predict Tnps in large-scale bacterial genome projects, which will expedite the discovery of novel Tnps.

Funding information

This study was supported by the Natural Science Foundation of Zhejiang Province, China (Grant Number: LQ20H150004); the Fundamental Research Funds for the Zhejiang Provincial Universities (Grant Number: KYYW201919); the Science and Technology Project of Inner

Mongolia Autonomous Region, China (grant number: 201802125); and the Start-up funds from the First Affiliated Hospital of Wenzhou Medical University (Grant Number: 2018QD014).

Author contributions

Q. W: data curation, formal analysis, writing - original draft preparation. J. Ye: investigation, validation, software. T. X: visualization, writing - reviewing and editing. N. Z: writing - reviewing and editing. Z. L: methodology, supervision. J. Ying: conceptualization, project administration, writing - reviewing and editing. Q. W. and J. Ye., contributed equally to this work.

Conflicts of interest

The authors declare that there are no conflicts of interest.

References

1. Babakhani S, Oloomi M. Transposons: the agents of antibiotic resistance in bacteria. *J Basic Microbiol* 2018;58:905–917.
2. Makalowski W, Pande A, Gotea V, Makalowska I. Transposable elements and their identification. *Methods Mol Biol* 2012;855:337–359.
3. Dziewit L, Baj J, Szuplewska M, Maj A, Tabin M, *et al*. Insights into the transposable mobilome of *Paracoccus* spp. (*Alphaproteobacteria*). *PloS one* 2012;7:e32277.
4. Iyer A, Barbour E, Azhar E, Salabi A, Hassan H, *et al*. Transposable elements in *Escherichia coli* antimicrobial resistance. *Adv Biosci Biotechnol* 2013;4:415–423.
5. van Hoek AH, Mevius D, Guerra B, Mullany P, Roberts AP, *et al*. Acquired antibiotic resistance genes: an overview. *Front Microbiol* 2011;2:203.
6. Wagner A. Cooperation is fleeting in the world of transposable elements. *PLoS Comput Biol* 2006;2:e162.
7. García-Contreras R. Unraveling resistance mechanisms against new antimicrobials using transposon mutagenesis. *Cloning and Transgenesis* 2013;02.
8. Rice PA, Baker TA. Comparative architecture of transposase and integrase complexes. *Nat Struct Biol* 2001;8:302–307.
9. Siguier P, Perochon J, Lestrade L, Mahillon J, Chandler M. ISfinder: the reference centre for bacterial insertion sequences. *Nucleic Acids Res* 2006;34:D32-6.
10. Varani AM, Siguier P, Gourbeyre E, Charneau V, Chandler M. ISsaga is an ensemble of web-based methods for high throughput identification and semi-automatic annotation of insertion sequences in prokaryotic genomes. *Genome Biol* 2011;12:R30.
11. Wagner A, Lewis C, Bichsel M. A survey of bacterial insertion sequences using IScan. *Nucleic Acids Res* 2007;35:5284–5293.
12. Riadi G, Medina-Moenne C, Holmes DS. TnpPred: A web service for the robust prediction of prokaryotic transposases. *Comp Funct Genomics* 2012;2012:678761.
13. Kamoun C, Payen T, Hua-Van A, Filee J. Improving prokaryotic transposable elements identification using a combination of *de novo* and profile HMM methods. *BMC genomics* 2013;14:700.
14. Tarca AL, Carey VJ, Chen XW, Romero R, Draghici S. Machine learning and its applications to biology. *PLoS Comput Biol* 2007;3:e116.
15. Hou R, Wang L, YJ W. Predicting atp-binding cassette transporters using the random forest method. *Front Genet* 2020;11:156.
16. HC Y, You ZH, Zhou X, Cheng L, Li X, *et al*. ACP-DL: A deep learning long short-term memory model to predict anticancer peptides using high-efficiency feature representation. *Molecular Therapy Nucleic acids* 2019;17:1–9.
17. Han K, Wang M, Zhang L, Wang Y, Guo M, *et al*. Predicting ion channels genes and their types with machine learning techniques. *Front Genet* 2019;10:399.
18. Li W, Godzik A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* 2006;22:1658–1659.
19. Wang J, Yang B, Leier A, Marquez-Lago TT, Hayashida M, *et al*. Bastion6: a bioinformatics approach for accurate prediction of type VI secreted effectors. *Bioinformatics* 2018;34:2546–2555.
20. Chen Z, Zhao P, Li F, Leier A, Marquez-Lago TT, *et al*. iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics* 2018;34:2499–2502.
21. Bhasin M, Raghava GP. Classification of nuclear receptors based on amino acid composition and dipeptide composition. *J Biol Chem* 2004;279:23262–23266.
22. Saravanan V, Gautham N. Harnessing computational biology for exact linear b-cell epitope prediction: A novel amino acid composition-based feature descriptor. *OMICS: A Journal of Integrative Biology* 2015;19:648–658.
23. Chen K, Kurgan LA, Ruan J. Prediction of flexible/rigid regions from protein sequences using k-spaced amino acid pairs. *BMC Struct Biol* 2007;7:25.
24. Lee TY, Lin ZQ, Hsieh SJ, Bretana NA, CT L. Exploiting maximal dependence decomposition to identify conserved motifs from a group of aligned signal sequences. *Bioinformatics* 2011;27:1780–1787.
25. Sokal RR, Thomson BA. Population structure inferred by local spatial autocorrelation: an example from an Amerindian tribal population. *Am J Phys Anthropol* 2006;129:121–131.
26. Lin Z, Pan XM. Accurate prediction of protein secondary structural content. *J Protein Chem* 2001;20:217–220.
27. Horne DS. Prediction of protein helix content from an autocorrelation analysis of sequence hydrophobicities. *Biopolymers* 1988;27:451–477.
28. Dubchak I, Muchnik I, Holbrook SR, Kim SH. Prediction of protein folding class using global description of amino acid sequence. *Proc Natl Acad Sci USA* 1995;92:8700–8704.
29. Dubchak I, Muchnik I, Mayor C, Dralyuk I, Kim SH. Recognition of a protein fold in the context of the Structural Classification of Proteins (SCOP) classification. *Proteins* 1999;35:401–407.
30. Shen J, Zhang J, Luo X, Zhu W, Yu K, *et al*. Predicting protein-protein interactions based only on sequences information. *Proc Natl Acad Sci USA* 2007;104:4337–4341.
31. Chou KC. Prediction of protein subcellular locations by incorporating quasi-sequence-order effect. *Biochem Biophys Res Commun* 2000;278:477–483.
32. Chou KC, Cai YD. Prediction and classification of protein subcellular location-sequence-order effect and pseudo amino acid composition. *J Cell Biochem* 2003;90:1250–1260.
33. Chou KC. Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes. *Bioinformatics* 2005;21:10–19.
34. Chou KC. Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins* 2001;43:246–255.
35. van der Maaten L, Hinton G. Viualizing data using t-SNE. *J Mach Learn Res* 2008;9:2579–2605.
36. Chen Z, Pang M, Zhao Z, Li S, Miao R, *et al*. Feature selection may improve deep neural networks for the bioinformatics problems. *Bioinformatics* 2020;36:1542–1552.
37. Ross BC. Mutual information between discrete and continuous data sets. *PloS one* 2014;9:e87357.
38. Xu Y, Cao L, Zhao X, Yao Y, Liu Q, *et al*. Prediction of smoking behavior from single nucleotide polymorphisms with machine learning approaches. *Front Psychiatry* 2020;11:416.
39. Wang Q, Xu T, Tong Y, Wu J, Zhu W, *et al*. Prognostic potential of alternative splicing markers in endometrial cancer. *Mol Ther Nucleic Acids* 2019;18:1039–1048.
40. Rao B, Zhou C, Zhang G, Su R, Wei L. ACPred-Fuse: fusing multi-view information improves the prediction of anticancer peptides. *Brief Bioinformatics* 2019.