T4SE-XGB: interpretable sequence-based prediction of type IV secreted effectors using eXtreme gradient boosting algorithm

Tianhang Chen[1,#], Xiangeng Wang[1,#], Yanyi Chu[1,2], Dong-Qing Wei[1,2,*], and Yi Xiong[1, *]

[1] State Key Laboratory of Microbial Metabolism, and School of Life Sciences and Biotechnology, Shanghai Jiao Tong University, Shanghai 200240, China; [2] Peng Cheng Laboratory, Vanke Cloud City Phase I Building 8, Xili Street, Nanshan District, Shenzhen, Guangdong, 518055, China

Equal Contribution#

ABSTRACT

Type IV secreted effectors (T4SEs) can be translocated into the cytosol of host cells via type IV secretion system (T4SS) and cause diseases. However, experimental approaches to identify T4SEs are time- and resource-consuming, and the existing computational tools based on machine learning techniques have some obvious limitations such as the lack of interpretability in the prediction models. In this study, we proposed a new model, T4SE-XGB, which uses the eXtreme gradient boosting (XGBoost) algorithm for accurate identification of type IV effectors based on optimal protein sequence features. After trying 20 different features, the best result achieved when all features were fed into XGBoost by the 5-fold cross validation compared with different machine learning methods. Then, the ReliefF algorithm was adopted to optimize feature vectors and got final 1100 features for our dataset which obviously improved the model performance. T4SE-XGB exhibited highest predictive performance on the independent test set and clearly outperforms other recent prediction tools. What's more, the SHAP method was used to interpret the contribution of features to model predictions. The identification of key features can contribute to an improved understanding of multifactorial contributors to host-pathogen interactions and bacterial pathogenesis. In addition to type IV effector prediction, we believe that the proposed framework composed of model construction and model interpretation can provide more instructive guidance for further research of developing novel computational methods and mechanism exploration of biological problems. The data and code for this study can be found at https://github.com/CT001002/T4SE-XGB.

Key words: Type IV secretion system; T4SE; effector protein; XGBoost; SHAP

INTRODUCTION

Different secretion systems have been found in bacteria that secret proteins into the extracellular environment. Gram-negative bacterial secretion can be categorized into eight types (from type I to type VIII), and the secreted proteins (also called effectors) play a vital role in bacterial pathogenesis and bacterium-host interactions. Type IV secretion system (T4SS) are protein complexes found in various species that deliver proteins into the cytoplasm of host cell and thus cause infection, such as whooping cough [1], gastritis, peptic ulcer and crown-gall tumor [2]. Therefore, the identification of type IV secreted effector proteins (T4SEs) is a fundamental step toward understanding of the pathogenic mechanism of T4SS.

There are a variety of experimental methods for identifying new T4SEs such as immunoblot analysis and pull-down assay [3]. However, they are limited by both a *priori* knowledge about biological mechanisms and the sophisticated implementation of molecular experiments [4]. Furthermore, these experimental approaches are quite time-consuming and expensive. Instead, a large number of computational methods have been developed for the prediction of T4SEs in the last decade, which successfully speed up the process in terms of time and efficiency. These computational approaches can be categorized into two main groups: the first group of approaches infer new effectors based on sequence similarity with currently known effectors [5-10] or phylogenetic profiling analysis [11], and the second group of approaches involve learning patterns of known secreted effectors that distinguish them from non-secreted proteins based on machine learning and deep learning techniques [12-26]. In the latter group of methods, Burstein et al. [24] worked on *Legionella pneumophila* to identify T4SEs from non-effectors and validated 40 novel effectors which were predicted by machine learning algorithms. Several features including genomic organization, evolutionary based attributes, regulatory network attributes, and attributes specific to the *L. pneumophila* pathogenesis system were applied as input of different machine learning algorithms: naïve Bayes, Bayesian networks, support vector machine (SVM), neural network and a voting classifier based on these four algorithms. Then, Zou et al. [22] built the tool called T4EffPred based on the SVM algorithm with features such as amino acid composition (AAC), dipeptide composition (DPC), position specific scoring matrix composition (PSSM), auto covariance transformation of PSSM to identify T4SEs. Wang et al. [21] constructed an effective inter-species T4SS effector prediction software named T4SEpre, based on SVM by using C-terminal sequential and position-specific amino acid compositions, possible motifs and structural features. Later, Xiong et al. [17] used the same dataset as that of the previous study [19] and developed a stacked ensemble classifier PredT4SE-Stack including various machine learning algorithms, such as SVM, gradient boosting machine, and extremely randomized trees. Wang et al. [25] developed an ensemble classifier called Bastion4 which serves as an online T4SS effector predictor. They calculated 10 types of sequence-derived features as the input vectors. And then, Naïve Bayes (NB), K-nearest neighbor (KNN), logistic regression (LR), random forest (RF), SVM and multilayer perceptron (MLP) were trained and compared. Significantly improved predictive performance arose when they used the majority voting process based on the six classifiers where the PSSM-based features were used as input vectors. Ashari et al. developed the package called OPT4e [14], which assembled all the features used in prior studies for the purpose of predicting a set of candidate effectors for *A. phagocytophilum*. This tool yielded reasonable candidate effector predictions for most T4SS bacteria from the *Alphaproteobacteria* and *Gammaproteobacteria* classes.

Deep learning is a new technology based on neural network architecture and has been successfully applied in various problems in recent years. Some researchers have perceived the advantages of deep learning methods and applied them to achieve notable improvements in the field of identifying T4SEs. Xue et al. [16] proposed a deep learning method to identify T4SEs from primary sequences. The model called DeepT4 utilized a convolutional neural network (CNN) to extract T4SEs-related features from 50 N-terminal and 100 C-terminal residues of the proteins. This work provided the original idea about using the deep learning method. However, only few information of protein sequences can be extracted, which showed a slightly weaker performance compared with the Bastion4. Açıcı et al. [15] developed the CNN architecture based on the conversion from protein sequences to images using AAC, DPC and PSSM feature extraction methods. Recently, Hong et al. [12] developed the new tool CNN-T4SE based on CNN, which integrated three encoding strategies: PSSM, protein secondary structure & solvent accessibility (PSSSA) and one-hot encoding scheme (Onehot), respectively. Compared with other machine learning methods, CNN-T4SE outperform all other state-of-the-art sequence-based T4SEs predictors. However, the less-than-optimal features analysis causes the limited deep learning for protein data and it is not straightforward to understand which features extracted from a given protein sequence drive the final prediction.

In this research, we proposed T4SE-XGB, a XGBoost based model using sequence-based features from type IV effector and non-effector proteins. To overcome the limitations of existing methods, we selectively summarized the features covered in previous studies and added some new features. The main strength of our method hinges on two aspects. On the one hand, T4SE-XGB trained with features selected by the ReliefF algorithm significantly improved the overall performance on the benchmark dataset. On the other hand, T4SE-XGB uses a *post-hoc* interpretation technique: the SHAP method to demystify and explain specific features that led to deeper understanding of “black box” models.

MATERIALS AND METHODS

The overall workflow of T4SE-XGB is shown in Figure 1 and summarized to five stages: Dataset Collection, Feature Extraction, Feature Selection, Model Construction and Model Interpretation. The detailed steps of each stage are described in the following sections.

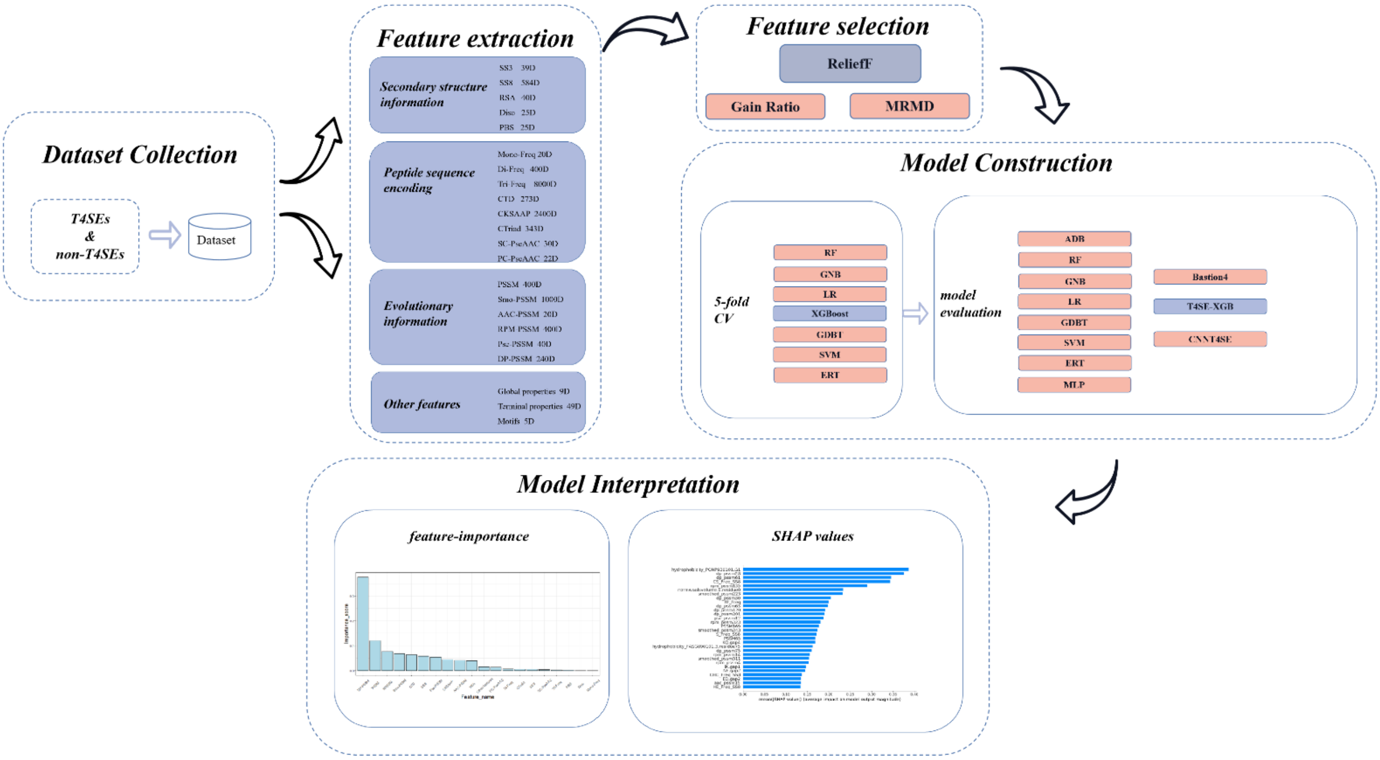

Figure 1. Overview workflow of T4SE-XGB. First, the benchmark dataset was collected. Next, 20 types of features were used to extract information from original protein sequences. Then, the ReliefF algorithm was employed to select optimal features.

Five-fold cross validation test and independent test were set to verify the validation of the model. Finally, we not only used the vanilla XGBoost method to get feature importance, but also got SHAP values to realize the model interpretation.

DATASET

In our study, type IV secreted effectors and non-effectors were selected as the benchmark dataset to construct the machine-learning model for proteins prediction. Our dataset was obtained from **Wang, J. et al.** [27] contained 420 T4SEs and 1262 non-T4SEs. These original proteins were already divided into training and independent test datasets. To reduce sequence redundancy, the tool named CD-HIT [28] was used to filter all proteins in the dataset having sequence similarity >30%. In the end, we got the final training dataset consisted of 365 T4SE and 1106 non-T4SE proteins and the independent test dataset including 20 T4SEs and 139 nonT4SEs.

FEATURE EXTRACTION

In this experiment, we took full advantage of features derived from protein sequence of T4SEs that former researchers have used and also added some novel features have been used in other large-scale protein-prediction problems. We utilized the following four aspects of features to characterize protein sequences: secondary structure information, peptide sequence encoding, evolutionary information and other underlying features. Details about feature extraction are listed in the following:

Secondary structure information.

(i) First, we used SCRATCH [29] to get predict **3- and 8-state secondary structure (SS) information** and then *mono*- (1 state i.e. turn, strand or coil), *di*- (two consecutive states) and *tri*-state (three consecutive states) frequencies from a given protein sequence were extracted. (ii) The fraction of exposed residues (FER) with 20 different **relative solvent accessibility (RSA)** cutoffs (0% to 95% cutoffs at 5% intervals) and the FER by the average hydrophobicity of these exposed residues at different RSA cutoffs were calculated. (ii) DISOPRED [30] can achieve predicting precise disordered region with annotated protein-binding activity. In the former study, Elbasir et al. [31] used DISOPRED to get 25 **disordered features** and 25 features of **protein binding sites (PBS)** in disordered region. We used the same strategy and details are provided in Supplementary Material.

Peptide sequence encoding.

(i) Frequencies of **20 amino acids, 400 di-peptides, 8000 tri-peptides** were extracted from the protein sequences. (ii) The **Composition, Transition and Distribution (CTD)** feature represents the amino acid distribution patterns of a specific structural or physicochemical property in a protein or peptide sequence. Different types of physicochemical properties including hydrophobicity, normalized Van der Waals Volume, polarity, polarizability, charge, secondary structures and solvent accessibility have been used for computing final feature vectors. (iii) The **Composition of k-spaced Amino Acid Pairs (CKSAAP)** feature encoding calculates frequencies of amino acid pairs separated by any k residues range from 0 to 5, We use the default maximum value of k which is 5, and got a 2400-dimensional feature vector for protein sequence. (iv) The **Conjoint Triad descriptor (CTriad)** considers the properties of one amino acid and its vicinal amino acids by regarding any three continuous amino acids as a single unit [32]. (v) Pseudo amino acid composition analyses protein sequences about the physicochemical properties of constituent amino acids. The final feature vectors include the global or long-range sequence order information. **Series correlation pseudo amino acid composition (SC-PseAAC)** [33] is a variant of PseAAC, generates protein feature vectors by combining the amino acid composition and global sequence-order effects via series correlation. **Parallel correlation pseudo amino acid composition (PC-PseAAC)** [34], derived from PseACC, incorporating the contiguous local sequence-order information and the global sequence-order information into feature vectors of protein sequences.

The iFeature [35] Sever is capable of calculating and extracting different sequences, structural and physiochemical features derived from protein sequences. The BioSeq-Analysis2.0 [36] Sever was employed to generate modes of pseudo amino acid compositions for protein sequences including SC-PseAAC and PC-PseAAC. There are several parameters to set: λ represents the counted rank (or tier) of the correlation along a biological sequence; w is the weight factor for the sequence-order effects and used to put weight to the additional pseudo components with respect to the conventional sequence components; two feature selection methods can be chosen. In brief, we kept the parameters: λ=5&w=0.1 for the SC-PseAAC mode while λ=2&w=0.1 for the PC-PseAAC mode, and we selected the feature selection method named mutual information for two modes (Supplementary Material).

Evolutionary information.

(i) **Position specific scoring matrix (PSSM)** of a protein sequence can be obtained in the form of L∗20 matrix (amino acid length is L). PSSM represents the evolutionary, residue and sequence information features of input proteins. In our study, we got 400 feature vectors from the original PSSM profile by summing rows corresponding to the same amino acid residue. (ii) **Smoothed-PSSM [37]** transformed from the standard PSSM encodes the correlation or dependency from surrounding residues significantly enhanced the performance of RNA-binding site prediction in proteins. The Smoothed-PSSM profile considered the first 50 amino acids starting from the protein's N-terminus to form a vector with the dimension 1000. (iii) **AAC-PSSM** [38] represents the correlation of evolutionary conservation of the 20 residues between two positions separated by a predefined distance along the sequence and successfully converts a protein into a fixed length feature vector with dimension 20. It reveals the possibility of the amino acid residues in the protein being mutated to different types during the evolution process. (iv) **RPM-PSSM** [39] filters all entities with values of less than 0 from the PSSM matrix by using the residue probing method, and the negative values were set to 0. For this method, original PSSM matrix finally transformed into the 20*20 matrix and can be constructed into a 400-dimensional vector. (v) **Pse-PSSM** [40] developed from PseAAC and encodes the PSSM of proteins with different lengths using a uniform length matrix. Pse-PSSM have been proved manually derive sequence-length-independent features from the sequence-length-dependent features and successfully avoid complete loss of the sequence-order information [41-43]. (iv) **DP-PSSM** [44] , a protein descriptor based on similarity , gets the hidden sequential order information by calculating from protein sequence and can avoid cancellation of positive or negative terms in the average process. As a result, we obtained a 400-dimensional vector from each sequence.

All PSSM-based algorithms above involved were achieved using the bioinformatics tool called POSSUM [45], including the original PSSM profiles, smoothed-PSSM , AAC-PSSM, RPM-PSSM, Pse-PSSM and DP-PSSM. All PSSM-based feature descriptors used default parameters the website provided: smoothing window=7 and sliding window=50 for smoothed-PSSM, ξ=1 for Pse-PSSM, and α=5 for DP-PSSM.

Other underlying features.

(i) **Global properties of the protein** were calculated including sequence length, molecular weight, total hydropathy et al. and the list is shown in Supplementary Material. (ii) **Terminal properties** like the frequencies of 20 amino acids length of 50 amino acids at the C-terminus or N-terminus adopted in former researches were also calculated [27, 46-48]. And the frequencies of di-peptides at the C-terminus, like SS, KE, EE, EK, AA, AG and LL involved in former studies have shown variances between effectors and the non-effectors were also calculated [49, 50]. (iii) We also searched for several types of protein **motifs** including nuclear localization signals (NLS), E-Block (EEXXE motif), conserved EPIYA motifs (EPIYA_CON), hypothetical EPIYA motifs (EPIYA_HYS) and Prenylation Domain (CaaX motif) that have been proposed and extracted before [18, 51-53].

Feature normalization

Normalization is a scaling technique in which values are shifted and rescaled so that they fall into the same numeric interval. Distance algorithms and distance-based feature selection methods are mostly affected by the range of features. Having features on a similar scale also help the gradient descent converge more quickly towards the minima. The following formula can be used to normalize all feature values and end up ranging between 0 and 1, which is known as Min-Max scaling:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{1}$$

Here, X_{max} and X_{min} are the maximum and the minimum values of the feature respectively. To realize this, we imported the MinMaxScalar from the python scikit learn library.

Extreme gradient boosting

Extreme gradient boosting also named XGBoost [54] is an optimized distributed gradient boosting algorithm designed to be highly efficient, flexible and portable [55]. XGBoost based on decision tree ensembles consists of a set of classification and regression trees. It uses the training data (with multiple features) x_i to predict a target variable y_i .

To begin with, Chen et al. defined the objective function as:

$$obj = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{i=1}^{t} \Omega(f_i) \tag{2}$$

where n is the number of trees, l is the training loss function, Ω is the regularization term.

Then, the XGBoost takes the Taylor expansion of the loss function up to the second order and removes all the constants, so the specific objective at step t becomes:

$$L^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \Omega(f_t) \tag{3}$$

where the g_i and h_i are defined as

$$\begin{cases} g_i = \partial_{\hat{y}_i^{(t-1)}} l\left(y_i, \hat{y}_i^{(t-1)}\right) \\ h_i = \partial^2_{\hat{y}_i^{(t-1)}} l\left(y_i, \hat{y}_i^{(t-1)}\right) \end{cases} \tag{4}$$

The value of the objective function only depends on g_i and h_i can optimize every loss function, including logistic regression and pairwise ranking.

The traditional treatment of tree learning only emphasized improving impurity, while the complexity control was left to heuristics. Chen et al. formally defined the complexity of the tree $\Omega(f)$ to obtain regularization, and the loss function in the *t*-th tree finally can be rewritten as:

$$L^{(t)} = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda} + \gamma T \tag{5}$$

where the G_j and H_j are defined as

$$\begin{cases} G_j = \sum_{i\in I_j} g_i \\ H_j = \sum_{i\in I_j} h_i \end{cases} \tag{6}$$

I_j is the sample set divided into the j-th leaf node according to the decision rules for a given tree. The formula (3) can be used as the score value to evaluate the quality of a tree. They also defined the score it gains when a leaf split into two leaves:

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma \tag{7}$$

This formula composed of the score on the new left leaf, the score on the new right leaf, the score on the original leaf and regularization on the additional leaf. We can find the best split efficiently by the maximum value of $Gain$ through a scan from left to right to get all possible split solutions.

XGBoost with many optimization techniques is able to solve problems using far fewer resources. It is simple to parallel and can greatly enhance the program efficiency with a fast model exploration. More details about XGBoost are given in [54].

Performance evaluation

In this work, confusion matrix obtained after prediction contains four cells: true positive (TP), false positive (FP), false negative (FN) and true negative (TN). In order to evaluate the overall predictive performance of different classification models, we used metrics including Sensitivity (SE), Specificity (SP), Precision (PRE), Accuracy (ACC), F-score and Matthew's correlation coefficient (MCC) to achieve the comparison [56, 57]. These metrics take values between 0 and 1 have been widely used in former studies, with a higher value indicating assessing better performances. The performance metrics can be defined as follows:

$$\begin{cases} Sensitivity = \dfrac{TP}{TP+FN} \\ Specificity = \dfrac{TN}{TN+FP} \\ Precision = \dfrac{TP}{TP+FP} \\ Accuracy = \dfrac{TP+TN}{TP+TN+FN+FP} \\ F-score = \dfrac{2TP}{2TP+FP+FN} \\ MCC = \dfrac{(TP\times TN)-(FP\times FN)}{\sqrt{(TP+FP)\times(TP+FN)\times(TN+FP)\times(TN+FN)}} \end{cases} \quad (8)$$

RESULTS AND DISCUSSION

Performance evaluation using 5-fold cross validation method

In this section, the extracted vectors of each feature were firstly classified by XGBoost classifier, the prediction result was evaluated by 5-fold cross validation method. For each of the 20 types of feature encodings, the training data set was randomly divided into five subsets. XGBoost were trained by four subsets and the remaining one was validated to estimate the performance of the model. All steps were repeated five times. The average performance like ACC, SE et.al of the training set were calculated and the results are shown in Table 1. It can be seen that some single feature classes based on PSSM have higher overall prediction power on the training data set. This observation indicates that encodings based on PSSM have a slightly upper hand in the prediction of T4SE when compared with other encodings.

The combination of different features could depict protein sequences in a more comprehensive manner [58]. As illustrated in Table 1, using combined features gave the ACC of 93.95% and the MCC of 0.8346, that are both higher than other PSSM-based methods. In summary, compared with single feature-based models, the combination of all features achieved consistently better performance.

Table 1. Prediction results of the training data set by 21 feature extraction methods

	5-validation result				
Feature name	ACC (%)	SE (%)	PRE (%)	F-score	MCC
ss3	79.94	43.84	64.52	0.5209	0.4125
ss8	81.44	48.77	68.57	0.5640	0.4650
RSA	84.23	58.90	72.80	0.6497	0.5556
Diso	79.81	35.62	68.16	0.4650	0.3856
PBS	79.47	36.16	65.74	0.4643	0.3760
Mono-Freq	85.18	60.00	75.54	0.6669	0.5809
Di-Freq	84.36	53.97	76.16	0.6302	0.5486
Tri-Freq	80.01	30.41	73.85	0.4301	0.3822
PSSM	92.11	78.08	88.99	0.8305	0.7833
smoothed-PSSM	88.51	70.14	81.03	0.7512	0.6806

AAC-PSSM	90.69	74.52	86.27	0.7976	0.7425
RPM-PSSM	91.98	77.26	**89.32**	0.8262	0.7797
Pse-PSSM	92.59	**81.37**	88.13	0.8446	0.7984
DP-PSSM	**92.79**	**81.37**	88.95	**0.8486**	**0.8039**
CKSAAP	84.02	51.78	76.31	0.6153	0.5359
CTD	87.70	67.12	80.24	0.7296	0.6562
CTraid	82.53	49.86	71.04	0.5816	0.4909
SC-PseAAC	85.66	61.92	75.96	0.6811	0.5958
PC-PseAAC	85.52	61.10	76.05	0.6769	0.5913
Other features	84.09	54.52	74.62	0.6288	0.5422
All	**93.95**	**81.92**	**93.04**	**0.8698**	**0.8346**

Performance evaluation of feature selection methods by 5-fold cross validation

The high-dimensional features would be time consuming for training model classification, and potentially biased toward model prediction performance among feature vectors. It is indispensable to reduce dimensionality so that we can reserve the important one.

In this study, three feature selection methods: gain ratio algorithm [59], maximum relevance–maximum distance (MRMD) [60] and ReliefF algorithm [61], were used to reduce the number of features. The ACC of different dimensions were obtained and compared under different algorithms to select most useful features. As shown in Table 2, when the MRMD algorithm was used for dimensionality reduction on the training data set, the highest ACC value was 92.93%. The gain ratio algorithm achieved ACC of 93.81% on the training data set. By comparing the prediction accuracy of three methods in different dimensions, we can find the ReliefF algorithm achieved the highest ACC value, 94.42% when the dimension was 1000, obviously better than models trained using all original features.

Thereby, the ReliefF algorithm can effectively eliminate redundant variables and improve prediction accuracy. In the following sections, the ReliefF algorithm was used for dimensionality reduction. To avoid overfitting, we selected 1100 as the final optimal dimension on the benchmark dataset.

Table 2. The training data set select the prediction results of ACC (%) obtained by three algorithms in different dimensions.

	500	600	700	800	900	1000	1100	1200	1300	1400	1500
GainRatio	92.79	92.73	93.34	93.34	93.41	93.47	93.34	93.54	93.60	93.41	**93.81**
MRMD	92.18	92.45	92.32	92.52	92.18	92.66	92.39	**92.93**	92.86	92.25	92.93
ReliefF	93.74	93.95	93.61	93.74	**94.36**	**94.42**	**94.22**	94.09	93.95	94.02	94.08

Performance evaluation of different classification algorithms by 5-fold cross validation

In order to objectively evaluate the predictions of the XGBoost method, we compared the results of this algorithm with other methods. Based on the same features selected, other classifiers, which like Random Forests (RF) [56], Gaussian Naive Bayes (NB), Logistic Regression (LR), Gradient Boost (GDBT), support vector machine (SVM), K-nearest neighbor (KNN), Extremely randomized trees (ERT) and Multi-layer Perceptron (MLP) were all applied. Tuning the hyperparameter combination help achieve better performance on both training data set and independent data set. The grid search method was employed in this work to optimize hyperparameters for each classifier [62], all search ranges are shown in Supplementary Table S2. For each ML classifier, we obtained the best hyper-parameter combination based on the highest accuracy (ACC) by the 5-fold cross validation. The optimal combination of parameters on two are shown in Supplementary Tables S3. Table 3 compares the performances of XGBoost with other prediction methods on the training data set assessed by 5-fold cross validation.

As shown, the ACCs under different classifiers were within the range from 90.89% to 94.42%, and their MCCs were from 0.76 to 0.84 on the training data set. The results of 10 annotation models showed XGBoost achieved the best performance, obtained ACC, F-score and MCC significantly higher than the other classifiers. All in all, the XGBoost algorithm found to be capable of performing better than the other machine learning methods when applied on the training data set.

Table 3. Prediction results of the training data set under nine classifiers.

	ACC (%)	SE (%)	PRE (%)	F-score	MCC
NB	90.89	84.11	80.76	0.8207	0.7631
ML	92.32	82.47	86.10	0.8409	0.7920
LR	93.00	83.01	88.28	0.8539	0.8101
KNN	93.20	80.82	91.20	0.8544	0.8148
RF	93.27	80.27	91.94	0.8554	0.8163
ERT	93.54	80.55	92.76	0.8604	0.8235
GDBT	93.81	84.11	90.55	0.8710	0.8323
SVM	94.36	83.56	93.28	0.8794	0.8466
XGB	**94.42**	**83.01**	**94.02**	**0.8803**	**0.8481**

Comparison with different classification algorithms and other existing state-of-the-art methods using the independent test.

To further validate the performance of the proposed model, we measured the performance of our T4SE-XGB model by comparing with other classification algorithms and other existing state-of-the-art methods on the independent data set. The performance results of these methods are provided in Table 4. To make a fair comparison, the same independent data set composes of 20

T4SEs and 139 non-T4SEs were used for all models.

Among these machine-learning methods, Performance comparisons showed that our T4SE-XGB model achieved the overall best performance with an ACC of 97.48%, F-value of 90.48% and MCC of 0.8916, followed by the state-of-the-art annotation machine-learning model called Bastion4 [27], which achieved 96.23% on ACC, 86.96% on F-value and 0.8579 on MCC. While achieving a better performance than Bastion4, the T4SE-XGB trained by fewer training samples also get a more stable prediction than the deep-learning method named CNN-T4SE (VOTE 2/3) considers two votes of the three identified best-performing convolutional neural network-based models (CNN-PSSM, CNNPSSSA and CNN-Onehot). In all models, the CNN-PSSM, a deep-learning model based on PSSM features, achieved the best results, compared with our model gets two less false positive.

In summary, there is a consistent observation (with results obtained from the 5-fold cross validation test) that our T4SE-XGB model achieved great performance in terms of sensitivity, specificity, accuracy and MCC on the training data set and independent data set successively.

Table 4. Comparison among different classification algorithms and other existing state-of-the-art methods based on the independent data set.

Model	Independent test results									
	TP	FN	TN	FP	ACC (%)	SE (%)	SP (%)	PRE (%)	F-score	MCC
SVM	19	1	134	5	96.23	95.00	96.40	79.17	0.8636	0.8467
LR	19	1	131	8	94.34	95.00	94.24	70.37	0.8085	0.7882
NB	19	1	126	13	91.19	95.00	90.65	59.38	0.7308	0.7084
GDBT	19	1	131	8	94.34	95.00	94.24	70.37	0.8085	0.7882
RF	19	1	132	7	94.97	94.96	95.68	73.08	0.8261	0.8066
ERT	19	1	134	5	96.23	95.00	96.40	79.17	0.8636	0.8467
KNN	**20**	**0**	128	11	93.08	**100.0**	92.09	64.52	0.7843	0.7708
ML	18	2	129	10	92.45	90.0	92.81	64.29	0.7500	0.7209
Bastion4	**20**	**0**	133	6	96.23	**100.0**	95.68	76.92	0.8696	0.8579
CNNT4SE(PSSSA)	14	6	138	1	95.60	70.00	99.28	93.33	0.8000	0.7860
CNNT4SE(Onehot)	14	6	**139**	**0**	96.23	70.00	**100.0**	**100.0**	0.8235	0.8192
CNNT4SE(PSSM)	19	1	138	1	**98.74**	95.00	99.28	95.00	**0.9500**	**0.9428**
CNNT4SE(VOTE 2/3)	16	4	**139**	**0**	**97.48**	80.00	**100.0**	**100.0**	0.8889	0.8818
T4SE-XGB	19	1	136	3	**97.48**	95.00	97.84	86.36	**0.9048**	**0.8916**

Model interpretation

Computing feature-importance estimates

As a tree-based non-linear machine learning technique, XGBoost can exploit the interactions between the engineered features. In contrast to black-box modeling techniques like SVM, ANN, CNN, we can easily obtain feature importance scores for all input features. XGboost can also obtain the importance quickly and efficiently based on the frequency of a feature is used to split data or according to the average gain a feature brings when used during node splitting across all trees established. For the 1100 features constructed on the benchmark dataset, the importance of each feature during training is available in Supplementary Material, which is the sum of information gained when used for splits (tree branching).

The total feature importance contribution of all features according to their feature types are shown in Table 5 and Figure 2. We can see that the DP-PSSM feature gets the maximum value of importance scores which is 0.3758. This may mean that the DP-PSSM feature is more important and has a wide range of effectiveness. Besides, the PSSM feature incorporated evolutionary information contributing the importance of 0.1199, followed by other features based on the transformation of the standard PSSM profile, which like RPM-PSSM and Smoothed-PSSM. And there are also other features showing high importance. Next, CTD accounts for 6.46% of all feature importance score. SS8 makes up 5.84% of the total variable importance.

Table 9. Importance percentages grouped by feature classes for the T4SE-XGB model.

Feature name	Importance score
DP-PSSM	0.3758
PSSM	0.1199
RPM-PSSM	0.0764
Smoothed-PSSM	0.0690
CTD	0.0646
SS8	0.0584
Pse-PSSM	0.0532
CKSAAP	0.0449
AAC-PSSM	0.0411
RSA	0.0394
Other features	0.0158
PC-PseAAC	0.0143
Di-Freq	0.0075
CTraid	0.0057
SS3	0.0053
SC-PseAAC	0.0038
Tri-Freq	0.0032
PBS	0.0015
Diso	0
Mono-Freq	0

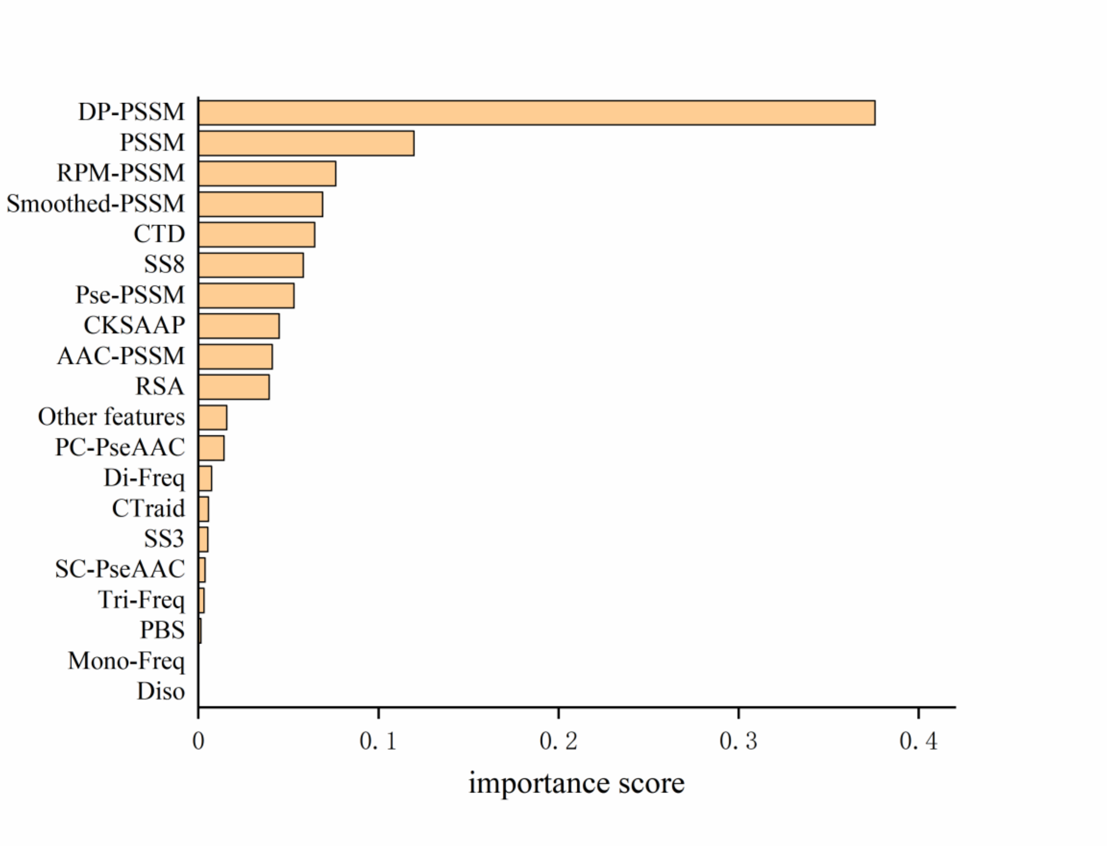

Figure 2. Comparison of importance percentages grouped by feature classes for the T4SE-XGB model.

Computing SHAP values and get summaries of entire model and individual features

SHAP (SHapley Additive exPlanations), a unified framework for interpreting predictions, assigns each feature an importance value for a particular prediction[63] and have improved the interpretability of tree-based models such as random forests, decision trees, and gradient boosted trees[64, 65]. The SHAP method has the ability to provide interpretable predictions and also overcomes limitation that the feature importance scores obtained from XGBoost model lack of directivity, unable to correspond with specific eigenvalues.

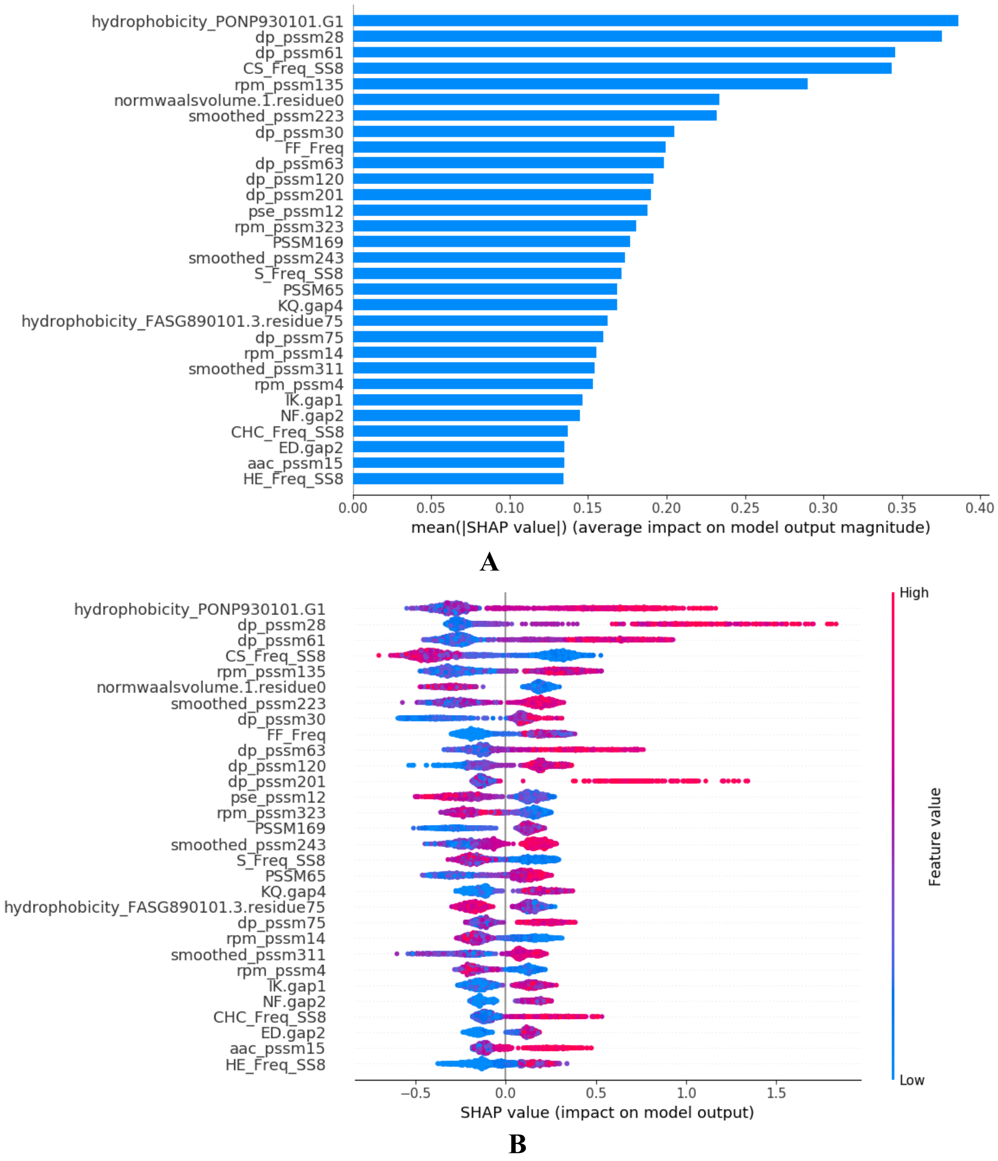

Figure 3. SHAP analysis results for T4SE-XGB.

Figure 3B is the standard bar-chart based on the average magnitude of the SHAP values over all training instances. Higher values indicate higher feature importance. It can be seen that DP-PSSM has the largest number of features, accounting for 7, among the 30 most important features. Meanwhile, other features based on PSSM also form the majority. Among them, the hydrophobicity_PONP930101.G1 came from the feature unit of CTD can be obviously identified as the most important. Hydrophobicity_PONP930101 is one physicochemical attribute based on the main clusters of the amino acid indices of Tomii and Kanehisa [66]**.** The hydrophobicity_PONP930101.G1={ N(r)/N, r ∈ {KPDESNQT}} represents the global compositions (percentage) of polar residues of the protein under the hydrophobicity_PONP930101 attribute [35]. Several studies have suggested that type IV effector proteins exhibited some specificities in regard to amino acid frequency before. Zou et al.[49] computed the ACC and the variance in their dataset called T4_1472. They found that Asn (N), Glu (E) and Lys (K) have higher compositions in type IVB effectors than non-effectors, and Ala (A), Glu (E) and Ser (S) have higher compositions in type IVA effectors than non-effectors. Some polar amino acids, such as Asp (D), Cys (C) and His (H), have small differences between secreted proteins and non-secreted proteins. Similarly, The Mann–Whitney U-test and the permutation test on amino acid frequencies were conducted by Yi et al. [67] , it showed that Ala (A), Gly (G), Met (M), Arg (R), Val (V), occurred less frequently in type IV effectors than in cytoplasmic proteins, meanwhile, Phe (F), Ile (I), Lys (K), Asn (N), Ser (S), Tyr (Y) ,Thr (T)occurred more frequently in type IV effectors than in cytoplasmic proteins. Nevertheless, because of different benchmark datasets were selected, the final results are debatable and incomplete. However, this is the first time to pay attention on the feature named hydrophobicity_PONP930101.G1, which not only according to the amino acid frequency, but also represents the corresponding hydrophilicity. SHAP summary plots from TreeExplainer [65] succinctly display the magnitude, prevalence, and direction of a feature's effect. Each dot in Figure 3B corresponds to a protein sample in the study. The position of the dot on the x-axis is the impact that feature has on the model's prediction for that protein. For example, higher values on hydrophobicity_PONP930101.G1 having a higher contribution on predicting a protein being an effector. In contrast, when the values of top features such as CS_Freq_SS8 and normwaalsvolume.1.residue0 are high, the corresponding Shapley values are negative driving the model prediction towards non-effector class. Besides, there are many long tails mean features with a low global importance can yet be extremely important for specific samples. All in all, from the analysis above, it is necessary and effective to consider many characteristics at the same time.

Conclusion

In this study, we have presented T4SE-XGB, a predictor developed for accurate identification of T4SE proteins based on the XGBoost algorithm. Especially, we have achieved the state-of-the-art performance compared with previous predictors on the benchmark dataset. There are three major conclusions can be drawn. First, compared with different algorithm, the XGBoost algorithm gives more stable and accurate prediction performance for T4SEs. Second, the feature selection method called ReliefF was presented to optimize feature vectors, which extracted significant features from large scale data and improved the model performance distinctly. Furthermore, unlike other sequence-based T4SEs predictors, T4SE-XGB can provide meaningful explanation based on samples provided using the feature importance and the SHAP method. It gives us the details about how some features, such as DP-PSSM features and hydrophobicity_PONP930101.G1 from CTD contributed to the final direction of prediction. Meanwhile, it explains the reason why it is essential to pay attention to some certain identities, and also consider a variety of features at the same time.

The final result showed that T4SE-XGB achieved a satisfying and promising performance which is stable and credible. However, the model is still constrained by the quantity of T4SE proteins which needs to be further improved and the characteristics of T4SEs need to be discovered. Besides, some potential relationships between features need to be explored. In the future, we plan to find and extract as many features as possible from a large amount of collected data to discriminate type IV secreted effectors from non-effectors.

References

1. Dorji D, Mooi F, Yantorno O, Deora R, Graham RM, Mukkur TK: **Bordetella Pertussis virulence factors in the continuing evolution of whooping cough vaccines for improved performance**. *Med Microbiol Immunol* 2018, **207**(1):3-26.

2. Kuzmanovic N, Pulawska J, Hao L, Burr TJ: **The Ecology of Agrobacterium vitis and Management of Crown Gall Disease in Vineyards**. *Curr Top Microbiol Immunol* 2018, **418**:15-53.
3. Cunha LD, Ribeiro JM, Fernandes TD, Massis LM, Khoo CA, Moffatt JH, Newton HJ, Roy CR, Zamboni DS: **Inhibition of inflammasome activation by Coxiella burnetii type IV secretion system effector IcaA**. *Nat Commun* 2015, **6**:10205.
4. Zeng C, Zou L: **An account of in silico identification tools of secreted effector proteins in bacteria and future challenges**. *Brief Bioinform* 2019, **20**(1):110-129.
5. Chen C, Banga S, Mertens K, Weber MM, Gorbaslieva I, Tan Y, Luo ZQ, Samuel JE: **Large-scale identification and translocation of type IV secretion substrates by Coxiella burnetii**. *Proc Natl Acad Sci U S A* 2010, **107**(50):21755-21760.
6. Lockwood S, Voth DE, Brayton KA, Beare PA, Brown WC, Heinzen RA, Broschat SL: **Identification of Anaplasma marginale type IV secretion system effector proteins**. *PLoS ONE* 2011, **6**(11):e27724.
7. Marchesini MI, Herrmann CK, Salcedo SP, Gorvel JP, Comerci DJ: **In search of Brucella abortus type IV secretion substrates: screening and identification of four proteins translocated into host cells through VirB system**. *Cell Microbiol* 2011, **13**(8):1261-1274.
8. Meyer DF, Noroy C, Moumene A, Raffaele S, Albina E, Vachiery N: **Searching algorithm for type IV secretion system effectors 1.0: a tool for predicting type IV effectors and exploring their genomic context**. *Nucleic Acids Res* 2013, **41**(20):9218-9229.
9. Sankarasubramanian J, Vishnu US, Dinakaran V, Sridhar J, Gunasekaran P, Rajendhran J: **Computational prediction of secretion systems and secretomes of Brucella: identification of novel type IV effectors and their interaction with the host**. *Mol Biosyst* 2016, **12**(1):178-190.
10. Noroy C, Lefrancois T, Meyer DF: **Searching algorithm for Type IV effector proteins (S4TE) 2.0: Improved tools for Type IV effector prediction, analysis and comparison in proteobacteria**. *PLoS Comput Biol* 2019, **15**(3):e1006847.
11. Zalguizuri A, Caetano-Anolles G, Lepek VC: **Phylogenetic profiling, an untapped resource for the prediction of secreted proteins and its complementation with sequence-based classifiers in bacterial type III, IV and VI secretion systems**. *Brief Bioinform* 2019, **20**(4):1395-1402.
12. Hong J, Luo Y, Mou M, Fu J, Zhang Y, Xue W, Xie T, Tao L, Lou Y, Zhu F: **Convolutional neural network-based annotation of bacterial type IV secretion system effectors with enhanced accuracy and reduced false discovery**. *Brief Bioinform* 2019.
13. Esna Ashari Z, Brayton KA, Broschat SL: **Using an optimal set of features with a machine learning-based approach to predict effector proteins for Legionella pneumophila**. *PLoS ONE* 2019, **14**(1):e0202312.
14. Esna Ashari Z, Brayton KA, Broschat SL: **Prediction of T4SS Effector Proteins for Anaplasma phagocytophilum Using OPT4e, A New Software Tool**. *Front Microbiol* 2019, **10**:1391.
15. Acici K, Asuroglu T, Erdas CB, Ogul H: **T4SS Effector Protein Prediction with Deep Learning**. *Data* 2019, **4**(1).
16. Xue L, Tang B, Chen W, Luo JS: **A deep learning framework for sequence-based bacteria type IV secreted effectors prediction**. *Chemom Intell Lab Syst* 2018, **183**:134-139.
17. Xiong Y, Wang Q, Yang J, Zhu X, Wei DQ: **PredT4SE-Stack: Prediction of Bacterial Type IV Secreted Effectors From Protein Sequences Using a Stacked Ensemble Method**. *Front Microbiol* 2018, **9**:2571.

18. Esna Ashari Z, Dasgupta N, Brayton KA, Broschat SL: **An optimal set of features for predicting type IV secretion system effector proteins for a subset of species based on a multi-level feature selection approach**. *PLOS ONE* 2018, **13**(5):e0197041.

19. Wang Y, Guo Y, Pu X, Li M: **Effective prediction of bacterial type IV secreted effectors by combined features of both C-termini and N-termini**. *J Comput Aided Mol Des* 2017, **31**(11):1029-1038.

20. Ashari ZE, Brayton KA, Broschat SL: **Determining Optimal Features for Predicting Type IV Secretion System Effector Proteins for Coxiella burnetii**. *ACM-Bcb' 2017: Proceedings of the 8th Acm International Conference on Bioinformatics, Computational Biology,And Health Informatics* 2017:346-351.

21. Wang Y, Wei X, Bao H, Liu SL: **Prediction of bacterial type IV secreted effectors by C-terminal features**. *BMC Genomics* 2014, **15**:50.

22. Zou L, Nan C, Hu F: **Accurate prediction of bacterial type IV secreted effectors using amino acid composition and PSSM profiles**. *Bioinformatics* 2013, **29**(24):3135-3142.

23. Lifshitz Z, Burstein D, Peeri M, Zusman T, Schwartz K, Shuman HA, Pupko T, Segal G: **Computational modeling and experimental validation of the Legionella and Coxiella virulence-related type-IVB secretion signal**. *Proc Natl Acad Sci U S A* 2013, **110**(8):E707-715.

24. Burstein D, Zusman T, Degtyar E, Viner R, Segal G, Pupko T: **Genome-scale identification of Legionella pneumophila effectors using a machine learning approach**. *PLoS Pathog* 2009, **5**(7):e1000508.

25. Wang J, Yang B, An Y, Marquez-Lago T, Leier A, Wilksch J, Hong Q, Zhang Y, Hayashida M, Akutsu T *et al*: **Systematic analysis and prediction of type IV secreted effector proteins by machine learning approaches**. *Brief Bioinform* 2019, **20**(3):931-951.

26. Yan ZH, Chen D, Teng ZX, Wang DH, Li YJ: **SMOPredT4SE: An Effective Prediction of Bacterial Type IV Secreted Effectors Using SVM Training With SMO**. *Ieee Access* 2020, **8**:25570-25578.

27. Wang J, Yang B, An Y, Marquez-Lago T, Leier A, Wilksch J, Hong Q, Zhang Y, Hayashida M, Akutsu T: **Systematic analysis and prediction of type IV secreted effector proteins by machine learning approaches**. *Briefings in bioinformatics* 2019, **20**(3):931-951.

28. Huang Y, Niu B, Gao Y, Fu L, Li W: **CD-HIT Suite: a web server for clustering and comparing biological sequences**. *Bioinformatics* 2010, **26**(5):680-682.

29. Cheng J, Randall AZ, Sweredoski MJ, Baldi P: **SCRATCH: a protein structure and structural feature prediction server**. *Nucleic acids research* 2005, **33**(Web Server issue):W72-W76.

30. Ward JJ, McGuffin LJ, Bryson K, Buxton BF, Jones DT: **The DISOPRED server for the prediction of protein disorder**. *Bioinformatics* 2004, **20**(13):2138-2139.

31. Elbasir A, Mall R, Kunji K, Rawi R, Islam Z, Chuang G-Y, Kolatkar PR, Bensmail H: **BCrystal: an interpretable sequence-based protein crystallization predictor**. *Bioinformatics* 2019, **36**(5):1429-1438.

32. Shen J, Zhang J, Luo X, Zhu W, Yu K, Chen K, Li Y, Jiang H: **Predicting protein-protein interactions based only on sequences information**. *Proceedings of the National Academy of Sciences of the United States of America* 2007, **104**(11):4337-4341.

33. Chou K-C: **Using amphiphilic pseudo amino acid composition to predict enzyme subfamily classes**. *Bioinformatics* 2004, **21**(1):10-19.

34. Chou K-C: **Prediction of protein cellular attributes using pseudo-amino acid composition**. *Proteins: Structure, Function, and Bioinformatics* 2001, **43**(3):246-255.

35. Chen Z, Zhao P, Li F, Leier A, Marquez-Lago TT, Wang Y, Webb GI, Smith AI, Daly RJ, Chou K-C *et al*: **iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences**. *Bioinformatics* 2018, **34**(14):2499-2502.

36. Liu B, Gao X, Zhang H: **BioSeq-Analysis2.0: an updated platform for analyzing DNA, RNA and protein sequences at sequence level and residue level based on machine learning approaches**. *Nucleic Acids Research* 2019, **47**(20):e127-e127.

37. Cheng C-W, Su EC-Y, Hwang J-K, Sung T-Y, Hsu W-L: **Predicting RNA-binding sites of proteins using support vector machines and evolutionary information**. *BMC bioinformatics* 2008, **9 Suppl 12**(Suppl 12):S6-S6.

38. Liu T, Zheng X, Wang J: **Prediction of protein structural class for low-similarity sequences using support vector machine and PSI-BLAST profile**. *Biochimie* 2010, **92**(10):1330-1334.

39. Jeong Jc, Lin X, Chen X: **On Position-Specific Scoring Matrix for Protein Function Prediction**. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 2011, **8**(2):308-315.

40. Chou K-C, Shen H-B: **MemType-2L: A Web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM**. *Biochemical and Biophysical Research Communications* 2007, **360**(2):339-345.

41. Kuo-Chen C: **Pseudo Amino Acid Composition and its Applications in Bioinformatics, Proteomics and System Biology**. *Current Proteomics* 2009, **6**(4):262-274.

42. Chou K-C: **Some remarks on protein attribute prediction and pseudo amino acid composition**. *Journal of Theoretical Biology* 2011, **273**(1):236-247.

43. Li Y, Wang S, Umarov R, Xie B, Fan M, Li L, Gao X: **DEEPre: sequence-based enzyme EC number prediction by deep learning**. *Bioinformatics* 2017, **34**(5):760-769.

44. Juan EYT, Li WJ, Jhang JH, Chiu CH: **Predicting Protein Subcellular Localizations for Gram-Negative Bacteria Using DP-PSSM and Support Vector Machines**. In: *2009 International Conference on Complex, Intelligent and Software Intensive Systems: 16-19 March 2009 2009*. 836-841.

45. Wang J, Yang B, Revote J, Leier A, Marquez-Lago TT, Webb G, Song J, Chou K-C, Lithgow T: **POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles**. *Bioinformatics* 2017, **33**(17):2756-2758.

46. An Y, Wang J, Li C, Leier A, Marquez-Lago T, Wilksch J, Zhang Y, Webb GI, Song J, Lithgow T: **Comprehensive assessment and performance improvement of effector protein predictors for bacterial secretion systems III, IV and VI**. *Briefings in Bioinformatics* 2016, **19**(1):148-161.

47. Zeng C, Zou L: **An account of in silico identification tools of secreted effector proteins in bacteria and future challenges**. *Briefings in Bioinformatics* 2017, **20**(1):110-129.

48. Wang Y, Guo Y, Pu X, Li M: **Effective prediction of bacterial type IV secreted effectors by combined features of both C-termini and N-termini**. *Journal of Computer-Aided Molecular Design* 2017, **31**(11):1029-1038.

49. Zou L, Nan C, Hu F: **Accurate prediction of bacterial type IV secreted effectors using amino acid composition and PSSM profiles**. *Bioinformatics* 2013, **29**(24):3135-3142.

50. Zou L, Chen K: **Computational prediction of bacterial type IV-B effectors using C-terminal signals and machine learning algorithms**. In: *2016 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB): 5-7 Oct. 2016 2016*. 1-5.

51. Meyer DF, Noroy C, Moumène A, Raffaele S, Albina E, Vachiéry N: **Searching algorithm for type IV secretion system effectors 1.0: a tool for predicting type IV effectors and exploring their

genomic context. *Nucleic Acids Research* 2013, **41**(20):9218-9229.

52. Noroy C, Lefrançois T, Meyer DF: **Searching algorithm for Type IV effector proteins (S4TE) 2.0: Improved tools for Type IV effector prediction, analysis and comparison in proteobacteria**. *PLOS Computational Biology* 2019, **15**(3):e1006847.
53. Esna Ashari Z, Brayton KA, Broschat SL: **Prediction of T4SS Effector Proteins for Anaplasma phagocytophilum Using OPT4e, A New Software Tool**. *Frontiers in Microbiology* 2019, **10**(1391).
54. Chen T, Guestrin C: **XGBoost: A scalable tree boosting system**. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining: 2016*. 785-794.
55. Wang X, Wang Y, Xu Z, Xiong Y, Wei D: **ATC-NLSP: prediction of the classes of anatomical therapeutic chemicals using a network-based label space partition method**. *Frontiers in pharmacology* 2019, **10**:971.
56. Zhang Y-F, Wang X, Kaushik AC, Chu Y, Shan X, Zhao M-Z, Xu Q, Wei D-Q: **SPVec: A Word2vec-Inspired Feature Representation Method for Drug-Target Interaction Prediction**. *Front Chem* 2020, **7**:895.
57. Chu Y, Kaushik AC, Wang X, Wang W, Zhang Y, Shan X, Salahub DR, Xiong Y, Wei D-Q: **DTI-CDF: a cascade deep forest model towards the prediction of drug-target interactions based on hybrid features**. *Briefings in Bioinformatics*.
58. Wang X, Zhu X, Ye M, Wang Y, Li C-D, Xiong Y, Wei D: **STS-NLSP: a network-based label space partition method for predicting the specificity of membrane transporter substrates using a hybrid feature of structural and semantic similarity**. *Frontiers in bioengineering and biotechnology* 2019, **7**:306.
59. Shannon CE: **A Mathematical Theory of Communication**. *Bell System Technical Journal* 1948, **27**(3):379-423.
60. Zou Q, Zeng J, Cao L, Ji R: **A novel features ranking metric with application to scalable visual and bioinformatics data classification**. *Neurocomputing* 2016, **173**:346-354.
61. Kira K, Rendell LA: **The feature selection problem: traditional methods and a new algorithm**. *AAAI-92 Proceedings Tenth National Conference on Artificial Intelligence* 1992:129-134.
62. Shan X, Wang X, Li C-d, Chu Y, Zhang Y, Xiong Y, Wei D-Q: **Prediction of CYP450 Enzyme–Substrate Selectivity Based on the Network-Based Label Space Division Method**. *Journal of chemical information and modeling* 2019, **59**(11):4577-4586.
63. Lundberg S, Lee S-I: **A Unified Approach to Interpreting Model Predictions**. In.; 2017.
64. Lundberg SM, Nair B, Vavilala MS, Horibe M, Eisses MJ, Adams T, Liston DE, Low DK-W, Newman S-F, Kim J *et al*: **Explainable machine-learning predictions for the prevention of hypoxaemia during surgery**. *Nature Biomedical Engineering* 2018, **2**(10):749-760.
65. Lundberg SM, Erion G, Chen H, DeGrave A, Prutkin JM, Nair B, Katz R, Himmelfarb J, Bansal N, Lee S-I: **From local explanations to global understanding with explainable AI for trees**. *Nature Machine Intelligence* 2020, **2**(1):56-67.
66. Tomii K, Kanehisa M: **Analysis of amino acid indices and mutation matrices for sequence comparison and structure prediction of proteins**. *Protein Engineering, Design and Selection* 1996, **9**(1):27-36.
67. An Y, Wang J, Li C, Leier A, Marquez-Lago T, Wilksch J, Zhang Y, Webb GI, Song J, Lithgow T: **Comprehensive assessment and performance improvement of effector protein predictors for bacterial secretion systems III, IV and VI**. *Briefings in Bioinformatics* 2018, **19**(1):148-161.