Special Section on Feature Representation and Learning Methods with Applications in Large-Scale Biological Sequence Analysis

Received March 14, 2020, accepted March 29, 2020, date of publication April 2, 2020, date of current version May 5, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2985111

Identification of Type VI Effector Proteins Using a Novel Ensemble Classifier

CHUNYU WANG1, JIALIN LI1, YING ZHANG2, AND MAOZU GUO3,4
1School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China
2Department of Pharmacy, Heilongjiang Province Land Reclamation Headquarters General Hospital, Harbin 150088, China
3School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China
4Beijing Key Laboratory of Intelligent Processing for Building Big Data, Beijing 100044, China

Corresponding authors: Ying Zhang (zhangying_hmu@163.com) and Maozu Guo (guomaozu@bucea.edu.cn)

The work was supported in part by the Natural Science Foundation of China under Grant 61872114, Grant 61532014, and Grant 61871020, and in part by the National Key Research and Development Plan Task of China under Grant 2016YFC0901902.

ABSTRACT The type VI secretion system (T6SS) delivers effector proteins (Type VI secretion system effectors, termed T6SEs) into neighboring target cells. Many human pathogens express T6SEs, including Vibrio cholera, Burkholderia spp., and Pseudomonas aeruginosa. T6SEs play vital roles in the competitive survival and pathogenesis of bacterial populations. Several machine-learning methods are able to distinguish T6SEs from non-T6SEs. However, we believe there is room for further development. Therefore, herein we propose a more powerful ensemble predictor for identifying T6SEs. Initially, we construct a benchmark dataset from existing studies and databases. Then we use k-separated-bigrams-PSSM (a type of feature encoding) to convert the protein sequences to mathematical vectors. A synthetic minority oversampling technique (SMOTE) is next employed to solve the training data imbalance problem. Finally, we employ a soft voting strategy to construct an integrated model combining six fine-tuned base classifiers. The model we propose performs excellently in terms of accuracy (ACC, 99.0%), Matthew’s correlation coefficient (MCC, 97.8%), sensitivity (SN, 97.1%), and specificity (SP, 100%) in independent testing.

INDEX TERMS T6SE, SMOTE, k-separated-bigrams-PSSM, ensemble predictor, classification.

I. INTRODUCTION Type VI secretion is a recently described mechanism for protein transport across the cell envelope. The type VI secretion system (T6SS) is widely distributed in Gram-negative bacteria, and appears to inject toxins into competitor bacteria and/or eukaryotic cells [1]. T6SSs are phage tail spike-like injectisomes. The protein complex releases virulence factors into the extracellular medium and/or transports these factors directly into the target cells using a dynamic “firing” mechanism related to the action of contractile bacteriophage tails [4]. The effector proteins delivered by the T6SS are named type VI secretion effector proteins (T6SEs), and play important roles in the competitive survival of bacterial populations, as well as in the pathogenesis of bacteria [5]. Based on known effector transport mechanisms, effectors can be classified as “specialized” or “cargo” effectors. Specialized effector proteins are fused to the C-terminus of the T6SS structure, and include VgrG, and Hcp, and/or PAAR (Pro-Ala-Ala-Arg)-domain-containing proteins, known to sharpen the VgrG spike. Cargo effectors interact directly or require a specific chaperone or adaptor protein to load into the lumen of the Hcp tube or VgrG spike [6]. T6SEs can function inside target cells, as well as extracellularly. Many human pathogens, including Vibrio cholera, Burkholderia spp., and Pseudomonas aeruginosa, express the T6SS. The T6SS has also been discovered in plant pathogens, Agrobacterium tumefaciens, Pectobacterium atrosepticum, and Xanthomonas oryzae, among others. The T6SS loci in Yersinia pestis (NCBI Gene accession YPO0499–YPO0516) plays a crucial role in phagocytosis-promoting activity [7]. Although recent years have witnessed advances in the characterization of the role and function of these secretion systems, the number of identified T6SS effectors is quite limited, with almost all identified effectors being encoded with genetic linkage to the VgrG or Hcp locus in the T6SS main gene cluster or orphan VgrG/Hcp islands.

T6SEs can be experimentally identified by both discovery-driven and knowledge/hypothesis-based methodologies.
This includes bioinformatics analyses, the genetic analysis of T6SS-associated genes, proteomics-based methods, and mutant library screening [6]. Additionally, various online bioinformatics tools and databases are available. For example, the SecretEPDB [8] (Bacterial Secreted Effector Protein DataBase) is a knowledgebase for effector proteins of the type III, type IV, and type VI secretion systems. SecretEPDB provides detailed and experimental annotations of these effector proteins by manually extracting and integrating structural and functional information from currently available databases and literature in terms of protein characteristics, protein function, protein secondary structure, Pfam domains, metabolic pathways, and evolutionary details. SecReT6 is an integrated database providing comprehensive information on just the type VI secretion system in bacteria, including functional and genome-centric descriptions of T6SSs, T6SS components, cargo effectors, and immunity proteins according to sequence-derived data and findings reported in the literature [9].

Various attempts have been made toward in silico prediction of T3SEs, T4SEs, and T6SEs. Several predictive T3SEs computational models have previously been constructed [10]–[16], as well as for T4SEs [17], [18]. Bastion6 [5] was the first tool for T6 effector protein prediction, and is a two-layer support-vector machine (SVM)-based ensemble model, which obtains 94.3% accuracy. Apart from Bastion6, Zalguizuri et al. and An et al. attempted to predict T6SEs, but got unsatisfactory results [19], [20]. Recently, PyPredT6, a python-based tool, was proposed to predict T6SS effectors using 837 features [21]. These 837 features were selected from a position specific peptide sequence profile (PSPSP), a position specific nucleotide sequence profile (PSSNP), a position specific secondary structure profile (PSSSP), and a position specific solvent accessibility profile (PSSAP), and conjoint triad descriptors (CTD) by a randomized decision tree. PyPredT6 obtains 89.15% accuracy using the consensus of multi-layer perceptron (MLP), SVM, k-nearest neighbor (KNN), naïve Bayesian (NB), and random forest (RF) classifiers, in which the final class depends on the class predicted by the majority of the classifiers.

We think much space remains for improving T6SE prediction in view of the mediocre performance of Bastion6 and PyPredT6. Therefore, we developed and herein present a more advantageous machine learning method to identify T6SS effector proteins. The model is trained on a set of experimentally validated 138 T6SS effectors and 1,112 non-effectors, and tested on 107 T6SEs and 200 non-T6SEs. We demonstrate that our model can discriminate T6SS effectors from non-effectors effectively, with an accuracy (ACC) of 99.00%, a sensitivity (SN) of 97.10%, and a specificity (SP) of 100%, in independent testing.

Our method can be broken down into five main steps, as seen in Figure 1: (1) Collect protein sequences to construct a benchmark dataset for training and testing. (2) Convert the sequences to numeric vectors and find the most effective features. (3) Use SMOTE to balance the datasets. (4) Construct a powerful predictor that combines six base classifiers for final classification, and tune the parameters to improve the model’s performance. (5) Evaluate the accuracy of the predictor via cross-validation and discuss the results.

II. MATERIALS AND METHODS

A. DATA COLLECTION

The datasets consist of a training set for the creation of the model and an independent test set for testing the performance of the model. We used the same training set as Bastion6, which was originally collected from SecretEPDB. It contains 138 positive sample sequences and 1,112 negative, both of which have had homologous sequences removed at a threshold of 90% sequence identity by CD-HIT [22], [23]. To evaluate the performance of our model, we constructed an independent testing set containing 107 positive T6SEs (87 of which were collected from SecReT6 after removing homologous sequences at an identity threshold of 90%, and 20 from Bastion6) and 200 negative samples (all from Bastion6).

B. FEATURE EXTRACTION

Generally, the primary step in amino acid sequence-based bioinformatics problems is conversion of the protein sequence to a set of numerical features that can reveal intrinsic characteristics of the protein. Different feature descriptors reflect the diverse properties of proteins [24]–[28]. These properties roughly divide into the following categories: amino acid composition (AAC) [29]–[33], pseudo amino acid composition (PseAAC) [34]–[36], physicochemical properties, evolutionary and gene ontology information [37]–[39]. Feature extraction selection will directly influence the model’s efficiency and precision. Therefore, selecting the most useful features for extraction is key to protein sequence classification problems. Furthermore, using characteristics of these features, we may be able to explain differences between T6SS effector and T6SS non-effector proteins and inspire further research.

In our work, evolutionary information, as represented by the position specific scoring matrix (PSSM) is extremely important, especially as characterized by k-separated-bigrams-PSSM, which are introduced in detail below.

1) The POSITION SPECIFIC SCORING MATRIX (PSSM)

Before introducing k-separated-bigrams-PSSM, it is necessary to understand the PSSM. A PSSM describes protein sequence evolutionary information. Evolutionary conservation can reflect important biological functions. A PSSM can be generated using PSI-BLAST [40] by searching for homologous sequences to a query protein in protein sequence databases. PSSMs have been widely used in bioinformatics [41]–[44]. We used three iterations against Swiss-Prot, with 0.01 as the E-value cutoff for our analyses [45].
The PSSM of sequence $P$ is represented by equation (1).

$$P_{\text{PSSM}} = \begin{bmatrix} E_{1 \rightarrow 1} & E_{1 \rightarrow 2} & \cdots & E_{1 \rightarrow 20} \\ E_{2 \rightarrow 1} & E_{2 \rightarrow 2} & \cdots & E_{2 \rightarrow 20} \\ \vdots & \vdots & \ddots & \vdots \\ E_{L \rightarrow 1} & E_{L \rightarrow 2} & \cdots & E_{L \rightarrow 20} \end{bmatrix}.$$  \hspace{1cm} (1)

where in equation (1) $E_{i \rightarrow j}$ represents the score of the amino acid residue at the $i$-th position of the protein sequence being changed to amino acid residue type $j$ during the evolutionary process, where $L$ is the length of the query sequence, and numerical codes 1, 2, ..., 20 represent the 20 native amino acid residues in alphabetical order. Because $E_{i \rightarrow j}$ varies in a wide range, the following standardization is performed:

$$E_{i \rightarrow j} = \frac{E_{i \rightarrow j} - \frac{1}{20} \sum_{k=1}^{20} E_{i \rightarrow k}}{\sqrt{\frac{1}{20} \sum_{u=1}^{20} \left( E_{i \rightarrow u} - \frac{1}{20} \sum_{k=1}^{20} E_{i \rightarrow k} \right)^2}}.$$ \hspace{1cm} (2)

where $i = 1, 2, \ldots, L, j = 1, 2, \ldots, 20$, and $E_{i \rightarrow j}^0$ represents the original scores calculated by PSI-BLAST.

2) THE $k$-SEPARATED-BIGRAMS-PSSM

The $k$-separated-bigrams-PSSM was proposed to model the relationships between two separated amino acids in a protein sequence [46]. The amino acid bigram probabilities are extracted from the sequential evolutionary probabilities in a PSSM. The algorithm can be summarized mathematically as below:

$$T_{m,n} (k) = \sum_{i=1}^{L-K} N_{i,m} N_{i+k,n},$$ \hspace{1cm} (3)

$$T (k) = \begin{bmatrix} T_{1,1} (k) , T_{1,2} (k) , \ldots , T_{1,20} (k) , \\ T_{2,1} , \ldots , T_{20,20} (k) \end{bmatrix}.$$ \hspace{1cm} (4)

where $N$ is the PSSM representation for a protein sequence ($1 \leq m \leq 20$, $1 \leq n \leq 20$ and $k \leq K$), and $K$ usually is set at 11. The matrix $N$ has $L$ rows and 20 columns as equation (1). $L$ is the length of the protein sequence. The $k$ represents the distance between the amino acids, for example, when $k = 1$, the amino acids used to calculate the transition probabilities are separated by 0 amino acids, for $k = 2$, the amino acids used to calculate the transition probabilities are separated by 1 amino acid, and so forth. For each $k$, $T(k)$ generates 400 features. The $k$-separated-bigrams-PSSM is visualized in Figure 2 below. In this paper, we extract $k$-separated-bigrams-PSSM from POSSUM [47], which is a Web server offering users a comprehensive and flexible generator for various kinds of PSSM-base descriptors.

3) SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE (SMOTE)

In the area of computational biology, datasets often have more negative samples than positive samples [48]–[51]. Imbalanced datasets can lead to overfitting, and sample
FIGURE 2. Frmsample protein sequence with an illustration of k-separated-bigrams-PSSM.

prediction can be biased towards the category with more samples. Random oversampling adopts the strategy of simply copying samples to increase the size of a few samples. However, the main idea behind SMOTE is to create new minority class samples by interpolating between several minority class examples that lie close together [52]. Specifically, for every minority sample, initially, its k nearest neighbors from the minority class are selected. Then, according to the oversampling rate, N neighbors should be chosen randomly from the k nearest neighbors. Finally, synthetic examples \(x_{\text{new}}\) are created according to the following formula:

\[
x_{\text{new}} = x + \text{random}(0, 1) \times (x_k - x).
\] (5)

where \(x_k (j = 1, 2, \ldots, N)\) is one of the k randomly selected nearest neighbors of \(x\), and random \((0,1)\) is a function that generates a random number between 0 and 1. In this way, SMOTE creates a new synthetic sample along the line between the minority class sample and the selected nearest neighbors. We obtained 1,112 T6SE and 1,112 non-T6SE samples for the training set using this procedure.

4) MODEL CONSTRUCTION

The goal of ensemble methods is to combine the predictions of several estimators built with given learning algorithms to improve generalizability over a single estimate. Two families of ensemble methods exist, namely averaging methods and boosting methods. The voting classifier is one type of averaging method. It is a kind of wrapper containing different machine learning classifiers to classify data with combined voting. The idea behind the voting classifier is to conceptually combine different machine learning classifiers and use a majority vote or the average predicted probabilities to predict class labels. This type of ensemble classifier is useful for a set of equally well-performing models and balances out individual weaknesses [53]–[58].

“Hard” and “soft” voting methods exist to make decisions regarding the target class. In hard voting, the predicted class label for a particular sample is the class label that represents the majority of class labels predicted by each individual classifier. In contrast to hard voting, soft voting returns the class label as an argmax of the sum of predicted probabilities. The weighted average probabilities for a sample are calculated by multiplying the weight parameter assigned to each classifier and the predicted class probabilities for each classifier. The final class label is derived from the class with the highest average probability.

In our method, we use a soft voting strategy [59] that ensembles LogisticRegression, SVC [60], AdaBoostClassifier [61], LinearDiscriminantAnalysis [62], GradientBoostingClassifier, and LGBMClassifier [63]. The six base classifiers are available in the scikit-learn package [62], [64], which is an open source machine learning library providing various built-in machine learning algorithms and models. We trained every base classifier on the same training set, we used a grid search to find optimal parameters, and then we ensembled results using the VotingClassifier [62].

C. PERFORMANCE AND EVALUATION

Choosing appropriate performance metrics is a necessary step in measuring how well a model works, and in comparing new models with existing methods. We use ACC, SN, SP, Matthew’s correlation coefficient (MCC) [65]–[71], Area Under Receiver Operating Characteristic Curve (ROC AUC) [72]–[82], and F-score, all calculated by confusion matrices, gained according to true and predicted classes. The chosen metrics are defined as below:

\[
Sn = \frac{TP}{TP + FN}
\] (6)

\[
Sp = \frac{TN + FP}{TN + FP + FN + TN}
\] (7)

\[
Acc = \frac{TP + TN + FP + FN}{2 \times TP}
\] (8)

\[
F1 = \frac{2 \times TP}{2 \times TP + FP + FN}
\]

\[
MCC = \frac{TP \times TN - FN \times FP}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}
\] (9)

Both the training set and testing set were categorized into two classes: positive and negative. TP is defined as positive samples that classify positive, TN are negative samples that categorize negative, FP are negative samples that categorize positive, and FN are positive samples that categorize negative. SN measures the true positive rate, while SP measures the true negative rate; both are equally important for evaluating the model. ACC reflects the predictor’s overall accuracy. When the dataset is imbalanced, ACC may be misleading [83]. However, MCC remains informative for measuring a model’s overall quality. MCC ranges \(-1\) to \(1\), where \(-1\) represents that the predictor always predicts the wrong result, 0 indicates a random guess, and 1 denotes that the predictor predicts all samples accurately. Thus, MCC can be seen as a correlation coefficient between the true and predicted classes. The F1 score, also known as balanced F-score, can be interpreted as a weighted average of precision and recall. F1 ranges from 0 to 1, with precision and recall making equal contribution. ROC AUC is a plot of the true positive rate against the false positive rate. The greater the area under the curve is, the more accurate the prediction is. The maximum value of AUC is 1.0, which denotes a perfect prediction. A random guess gives an AUC value of 0.5. These metrics are usually adopted to evaluate prediction quality [84]–[86].

\[\text{FIGURE 2. Frmsample protein sequence with an illustration of k-separated-bigrams-PSSM.}\]
III. EVALUATION METRICS

Three primary cross-validation methods have been widely applied in evaluating the performance of bioinformatics predictors [87]–[102]. These are: jackknife cross-validation, 10-fold cross-validation, and independent testing. In this paper, we use two of them, namely 10-fold cross-validation and independent testing, to evaluate our model.

In k-fold cross-validation, the original samples are randomly partitioned into k equal size subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k − 1 subsamples are used as training data. The cross-validation process is then repeated k times, with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged to produce a single estimation. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once. For classification problems, stratified k-fold cross-validation is typically used, in which each fold contains roughly the same proportion of class labels. In practical use, k is usually set to k = 10. In 10-fold cross-validation, the dataset is divided into 10 parts (nine are used for training and the other for testing during each iteration). The average accuracy of the 10 results is seen as an estimate of the algorithm’s overall accuracy. Usually, 10-fold cross-validation is performed multiple times.

Independent testing, also known as holdout testing, has no overlap between training and test sets. In other words, the training set is completely different than the testing set. When the testing set is divided from the training set, the distribution of the testing set should be similar to that of the training set; otherwise, the results of this testing strategy may be misleading [103]. In this paper, we used two different datasets collected separately: one for training, and the other for testing.

IV. RESULTS AND DISCUSSIONS

A. COMPARISON BETWEEN SINGLE AND ENSEMBLE MODEL

Figures 3 and 4 show histograms obtained by different single models and the ensemble model. In this experiment, both the single models and the ensemble model were trained by the balanced training set after SMOTE, and the feature encoding is by k-separated-bigrams-PSSM. The most effective single predictor is SVC when considering both 10-fold cross-validation and independent testing. The ensemble predictor, which integrates all the single models, is better and more stable than its base classifiers in terms of two kinds of evaluation metrics.

B. EVALUATION ACROSS FEATURE ENCODING METHODS

For each feature encoding, the ensemble predictor was trained by the original training set and validated by independent testing. Figure 5 shows the ROC AUC obtained by different feature encoding. As shown, k-separated-bigrams-PSSM achieve the best AUC performance.

C. SMOTE EFFECTS

We used a comparison experiment to investigate SMOTE effectiveness. To ensure experimental objectivity and robustness, we conducted the same experiments on the different feature encodings. The predictor trained by the training set without and with SMOTE both achieved promising
results, especially the latter, as evaluated by an independent test set. SMOTE improved the performance of all majority feature encodings, except Directional Property-PSSM (DP_PSSM) [104]. The comparison is shown in Figure 6. Generally, we conclude that SMOTE is helpful for the classification of T6SEs.

D. EVALUATION WITH 10-FOLD-CROSS-VALIDATION AND INDEPENDENT TESTING

After the above experiments, we used \( k \)-separated-bigrams-PSSM for feature encoding and SMOTE to assist with the dataset imbalance problem. An ensemble classifier integrating LogisticRegression, SVC, AdaBoostClassifier, LinearDiscriminantAnalysis, GradientBoostingClassifier, and LGBMClassifier was employed using a soft voting strategy to predict T6SEs. The results shown in Figure 7 indicate that our model is both effective and robust.

E. COMPARISON WITH THE STATE-OF-ART METHODS

To further validate the performance of our proposed method, we compared our model’s results with the other existing methods listed in Table 1 below. Two other existing predictors can identify T6 effector proteins so far: Bastion6 and PyPredT6. As Table 1 shows, our method achieves the highest score in terms of accuracy and specificity. Compared with Basion6, our method is somewhat inferior in terms of sensitivity, but it behaves better overall, because its specificity is far ahead in independent testing. Furthermore, our model exceeds PyPredT6 in ACC by 9.9%, SN by 5.9%, and SP by 9.3% on an independent dataset.

V. CONCLUSION

We collected a new independent testing set and developed an ensemble predictor integrating six base classifiers for recognizing T6SEs. Several experiments were performed to validate the effectiveness and robustness of our method. At first, we filtered the most effective feature encodings (\( k \)-separated-bigrams-PSSM) from various feature encodings. The results of models trained by different feature encodings show that PSSM-based features are more helpful than other features. Then we demonstrated that SMOTE is significant for most feature encodings. We next compared the performances of different single classifiers, finding that SVC is the most effective single model. We ultimately proved our ensemble classifier to be the most effective and robust T6SE predictor available, both in 10-fold cross-validation and independent testing. Our method is far ahead in terms of ACC and SP, compared with other existing methods. Taking into account all of our results, we conclude that our method is the best available predictor for screening experimental targets of T6SEs. In the future, we will pay more attention on the deep learning prediction techniques [105]–[108].

VI. AUTHOR CONTRIBUTIONS

C.W and M.G initiated the idea, conceived the whole process and drafted the manuscript. J.L implemented the experiments and designed the figures. Y.Z helped with data analysis and revised the manuscript. M.G and finalized the paper. All authors have read and approved the final manuscript.

ACKNOWLEDGMENT

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. The authors would like to thank S. M. Thompson, from Liwen Bianji (www.liwenbianji.cn/ac), Edanz Editing China, for editing the English text of a draft of this manuscript.

REFERENCES

[1] V. Decoin, C. Barbey, D. Bergeau, X. Latour, M. G. J. Feuilloley, N. Orange, and A. Merieau, “A type VI secretion system is involved in pseudomonas fluorescens bacterial competition,” *PLoS ONE*, vol. 9, no. 2, 2014, Art. no. e89411.

[2] E. Cascales and C. Cambillau, “Structural biology of type VI secretion systems,” *Phil. Trans. Roy. Soc. B, Biol. Sci.*, vol. 367, no. 1592, pp. 1102–1111, Apr. 2012.

[3] Y. Xiao, J. Zhang, and L. Deng, “Prediction of lncRNA-protein interactions using HeteSim scores based on heterogeneous networks,” *Sci. Rep.*, vol. 7, no. 1, Dec. 2017, Art. no. 3664.
C. Wang et al.: Identification of Type VI Effector Proteins Using a Novel Ensemble Classifier

[4] F. R. Cianfanelli, L. Monlezun, and S. J. Coulthurst, “Aim, load, fire: The type VI secretion system, a bacterial nanowapen,” Trends Microbiol., vol. 24, no. 1, pp. 51–62, Jan. 2016.

[5] J. Wang, B. Yang, A. Leier, T. T. Marquez-Lago, M. Hayashida, A. Rocker, Y. Zhang, T. Akutsu, K.-C. Chou, R. A. Strugnell, J. Song, and T. Lithgow, “Bastion6: A bioinformatics approach for accurate prediction of type VI secreted effectors,” Bioinformatics, vol. 34, no. 15, pp. 2546–2555, Aug. 2018.

[6] Y.-W. Lien and E.-M. Lai, “Type VI secretion effectors: Methodologies and biology,” Frontiers Cellular Infection Microbiol., vol. 7, p. 254, Jun. 2017.

[7] J. B. Robinson, M. V. Telepnev, I. V. Zadina, D. Bouyer, J. A. Montenieri, S. W. Bearden, K. L. Gage, S. L. Agar, S. M. Foltz, S. Chauhan, A. K. Chopra, and V. L. Motin, “Evaluation of a yersinia pestis mutant infected in a thermoregulated type VI-like secretion system in flea, macrophage and murine models,” Microbial Pathogenesis, vol. 47, no. 5, pp. 243–251, Nov. 2009.

[8] Y. An, J. Wang, C. Li, J. Revote, Y. Zhang, T. Naderer, M. Hayashida, T. Akutsu, G. I. Webb, T. Lithgow, and J. Song, “SecretEPDB: A comprehensive Web-based resource for secreted effector proteins of the bacterial types III, IV and VI secretion systems,” Sci. Rep., vol. 7, no. 1, Feb. 2017, Art. no. 41031.

[9] J. Li, Y. Yao, H. H. Xu, L. Hao, Z. Deng, K. Rajakumar, and H.-Y. Ou, “SecReT6: A Web-based resource for type VI secretion systems found in bacteria,” Environ. Microbiol., vol. 17, no. 7, pp. 2196–2202, Jul. 2015.

[10] M. Löwer and G. Schneider, “Prediction of type III secretion signals in genomes of gram-negative bacteria,” PLoS ONE, vol. 4, no. 6, 2009, Art. no. e5917.

[11] L. Zhou, C. Nan, and F. Hu, “Accurate prediction of bacterial type IV secreted effectors using amino acid composition and PSSM profiles,” Bioinformatics, vol. 29, no. 24, pp. 3135–3142, Dec. 2013.

[12] J. Wang, B. Yang, Y. An, T. Marquez-Lago, A. Leier, J. Wilksch, Y. Zhang, M. Hayashida, T. Akutsu, G. I. Webb, R. A. Strugnell, J. Song, and T. Lithgow, “Systematic analysis and prediction of type IV secreted effector proteins by machine learning approaches,” Briefings Bioinf., vol. 20, no. 3, pp. 931–951, May 2019.

[13] R. Arnold, S. Brandmaier, F. Kleine, P. Tischler, E. Heinz, S. Behrens, A. Ninnokski, H.-W. Mewes, M. Horn, and T. Rattei, “Sequence-based prediction of type III secreted effectors,” PLoS Pathogens, vol. 5, no. 4, 2009, Art. no. e1000376.

[14] J. Wang, J. Li, B. Yang, R. Xie, T. T. Marquez-Lago, A. Leier, M. Hayashida, T. Akutsu, Y. Zhang, K.-C. Chou, J. Selkrig, T. Zhou, J. Song, and T. Lithgow, “Bastion3: A two-layer ensemble predictor of type III secreted effectors,” Bioinformatics, vol. 35, no. 12, pp. 2017–2028, Jun. 2019.

[15] L. Xue, B. Tang, W. Chen, and J. Luo, “DeepT3: Deep convolutional neural networks accurately identify gram-negative bacterial type III secreted effectors using the N-terminal sequence,” Bioinformatics, vol. 35, no. 12, pp. 2051–2057, Jun. 2019.

[16] J. Li, L. Wei, F. Guo, and Q. Zou, “EP3: An ensemble predictor that accurately identifies type III secreted effectors,” Briefings Bioinf., Feb. 2020, Art. no. bbaa008, doi: 10.1093/bib/bbaa008.

[17] J. E. McDermott, A. Corrigan, E. Peterson, C. Oehmen, G. Niemann, J. Li, L. Wei, F. Guo, and Q. Zou, “EP3: An ensemble predictor that accurately identifies type III secreted effectors,” Briefings Bioinf., vol. 19, no. 1, pp. 148–161, 2018.

[18] Y. An, “Comprehensive assessment and performance improvement of effector protein predictors for bacterial secretion systems III, IV and VI,” Briefings Bioinf., vol. 19, no. 1, pp. 148–161, 2018.
X. Ru, P. Cao, L. Li, and Q. Zou, “Selecting essential MicroRNAs using C. Cortes and V. Vapnik, “Support-vector networks,”
L. Song, D. Li, X. Zeng, Y. Wu, L. Guo, and Q. Zou, “NDNA-prot: Identification of DNA-binding proteins based on unbalanced classification,”
Y. Freund and R. E. Schapire, “A desicion-theoretic generalization of on-line learning and an application to boosting,” in Proc. Eur. Conf. Comput. Learn. Theory, 1995, pp. 23–37.
F. Pedregosa and G. Varoquaux, “Scikit-learn: Machine learning in Python,” J. Mach. Learn. Res., vol. 12, pp. 2825–2830, Oct. 2011.
G. Ke, “Lightgbm: A highly efficient gradient boosting decision tree,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 3146–3154.
[83] Y. Jiao and P. Du, “Performance measures in evaluating machine learning based bioinformatics predictors for classifications,” Quant. Biol., vol. 4, no. 4, pp. 320–330, Dec. 2016.

[84] C. Shen, Y. Ding, J. Tang, L. Jiang, and F. Guo, “LPI-KTASLP: Prediction of LncRNA-protein interaction by semi-supervised link learning with multivariate information,” IEEE/ACM Trans. Comput. Biol. Bioinf., vol. 17, no. 6, pp. 2346–2356, Dec. 2020.

[85] Y. Ding, J. Tang, and F. Guo, “Identification of drug-side effect association via semisupervised model and multiple kernel learning,” IEEE J. Biomed. Health Inform., vol. 23, no. 6, pp. 2619–2632, Nov. 2019, doi: 10.1109/JBHI.2018.288334.

[86] W. Li, J. Yu, B. Lian, H. Sun, J. Li, M. Zhang, L. Li, Y. Li, Q. Liu, and L. Xie, “Identifying prognostic features by bottom-up approach and correlating to drug repositioning,” PLoS ONE, vol. 10, no. 3, Mar. 2015, Art. no. e0111867, doi: 10.1371/journal.pone.0111867.

[87] M. S. Rahman, M. K. Rahman, M. Kaykobad, and M. S. Rahman, “IsGPT: An optimized model to identify sub-golgi protein types using SVM and random forest based feature selection,” Artif. Intell. Med., vol. 84, pp. 90–100, Jan. 2018.

[88] M. Hayat, M. Tahir, and S. A. Khan, “Prediction of protein structure classes using hybrid space of multi-profile bayes and bi-gram probability feature spaces,” J. Theor. Biol., vol. 346, pp. 8–15, Apr. 2014.

[89] Y. Sheng, J. Tang, and F. Guo, “Identification of protein subcellular localization via integrating evolutionary and physicochemical information into Chou’s general PseAAC,” J. Theor. Biol., vol. 462, pp. 230–239, Feb. 2019, doi: 10.1016/j.jtbi.2018.11.012.

[90] Y. Ding, J. Tang, and F. Guo, “Identification of drug-side effect association via multiple information integration with centered kernel alignment,” Neurocomputing, vol. 325, pp. 211–224, Jan. 2019, doi: 10.1016/j.neucom.2018.10.028.

[91] Y. C. Zuo, Y. Peng, L. Liu, W. Chen, L. Yang, and G.-L. Fan, “Predicting peroxidase subcellular localization by hybridizing different descriptors of Chou’s pseudo amino acid patterns,” Anal. Biochem., vol. 458, pp. 14–19, Aug. 2014, doi: 10.1016/j.ab.2014.04.032.

[92] L. Yu, S. Yao, L. Gao, and Y. Zha, “Conserved disease modules extracted from multilayer heterogeneous disease and gene networks for understanding disease mechanisms and predicting disease treatments,” Frontiers Genet., vol. 9, p. 18, Jan. 2019.

[93] L. Yu and L. Gao, “Human Pathway-Based Disease Network,” IEEE/ACM Trans. Comput. Biol. Bioinf., vol. 16, no. 4, pp. 1240–1249, Jul./Aug. 2019, doi: 10.1109/TCBB.2017.2774802.

[94] L. Xu, G. Liang, C. Liao, G.-D. Chen, and C.-C. Chang, “An efficient classifier for Alzheimer’s disease genes identification,” Molecules, vol. 23, no. 12, pp. 3140–3149, 2018, [Online]. Available: http://www.mdpi.com/1420-3049/23/12/3140.

[95] L. Xu, G. Liang, S. Shi, and C. Liao, “Seq2SVM: A sequence-based support vector machine method for identifying antioxidant proteins,” Int. J. Mol. Sci., vol. 19, no. 6, p. 1773, 2018.

[96] K. Yan, X. Fang, Y. Xu, and B. Liu, “Protein fold recognition based on multi-view modeling,” Bioinformatics, vol. 35, no. 17, pp. 2982–2990, Sep. 2019.

[97] B. Liu and K. Li, “iFpromoter-2L2.0: Identifying promoters and their types by combining smoothing cutting window algorithm and sequence-based features,” Mol. Therapy Nucleic Acids, vol. 18, pp. 80–87, Dec. 2019.

[98] Y. Gong, Y. Niu, W. Zhang, and X. Li, “A network embedding-based multiple information integration method for the MiRNA-disease association prediction,” BMC Bioinf., vol. 20, no. 1, p. 468, Dec. 2019, doi: 10.1186/s12859-019-3063-3.

[99] W. Zhang, X. Yue, G. Tang, W. Wu, F. Huang, and X. Zhang, “SFPEL-LPI: Sequence-based feature projection ensemble learning for predicting LncRNA-protein interactions,” PLOS Comput. Biol., vol. 14, no. 12, Dec. 2018, Art. no. e1006616, doi: 10.1371/journal.pcbi.1006616.

[100] S. Basith, B. Manavalan, T. H. Shin, and G. Lee, “SDM6A: A Web-based integrative machine-learning framework for predicting 6 mA sites in the rice genome,” Mol. Therapy Nucleic Acids, vol. 18, pp. 131–141, Dec. 2019, doi: 10.1016/j.omtn.2019.08.011.

[101] B. Manavalan, S. Basith, T. H. Shin, L. Wei, and G. Lee, “Meta-4mCpred: A sequence-based meta-predictor for accurate DNA 4mC site prediction using effective feature representation,” Mol. Therapy Nucleic Acids, vol. 16, pp. 733–744, Jun. 2019, doi: 10.1016/j.omtn.2019.04.019.

[102] B. Manavalan, S. Basith, T. H. Shin, L. Wei, and G. Lee, “MAHT-Pred: A sequence-based meta-predictor for improving the prediction of anti-hypertensive peptides using effective feature representation,” Bioinformatics, vol. 35, no. 16, pp. 2757–2765, Aug. 2019, doi: 10.1093/bioinformatics/bty1047.

[103] R. Kohavi, D. Sommerfield, and J. Dougherty, “Data mining using a machine learning library in C++,” Int. J. Artif. Intell. Tools, vol. 6, no. 4, pp. 537–566, 1997.

[104] E. Y. T. Juan, W. J. Li, J. H. Jiang, and C. H. Chiu, “Predicting protein subcellular localizations for gram-negative bacteria using DP-PSSM and support vector machines,” in Proc. Int. Conf. Complex, Intell. Softw. Intensive Syst., Mar. 2009, pp. 836–841.

[105] L. Yu, X. Sun, S. W. Tian, X. Y. Shi, and Y. L. Yan, “Drug and non-drug classification based on deep learning with various feature selection strategies,” Current Bioinf., vol. 13, no. 3, pp. 253–259, 2018, doi: 10.2174/1574893612666170125124538.

[106] Z. B. Li, C. Y. Ao, and Q. Zou, “Protein function prediction: From traditional classifier to deep learning,” Proteomics, vol. 19, no. 4, p. 2, Jul. 2019, Art no. 1900119, doi: 10.1002/pmic.201900119.

[107] L. Peng, M. Peng, B. Liao, G. Huang, W. Li, and D. Xie, “The advances and challenges of deep learning application in biological big data processing,” Current Bioinf., vol. 13, no. 4, pp. 352–359, Jul. 2018, doi: 10.2174/1574893612666170707095707.

[108] L. Nie, L. Deng, C. Fan, W. Zhan, and Y. Tang, “Prediction of protein S-Sulfenylation sites using a deep belief network,” Current Bioinf., vol. 13, no. 5, pp. 461–467, Sep. 2018, doi: 10.2174/1574893612666171122152208.

CHUNYU WANG received the B.S., M.S., and Ph.D. degrees in computer science and technology from the Harbin Institute of Technology. He was a Visiting Scholar with the Department of Electrical Engineering and Computer Science, University of Missouri, from 2016 to 2017. He is currently an Associate Professor with the Department of Computer Science and Technology. His interests include bioinformatics and machine learning.

JIALIN LI received the B.S. degree in computer science and technology from the Harbin Institute of Technology in Weihai. She is currently pursuing the M.S. degree with the School of Computer Science and Technology, Harbin Institute of Technology. Her main field of researches are machine learning and computational biology.

YING ZHANG received the B.S. and M.S. degrees in pharmacology from Harbin Medical University, in 2010. She is currently a medical doctor in Heilongjiang Province Land Reclamation Headquarters General Hospital. Her main field of researches are computational biology, AD and PD.

MAOZU GUO received the Ph.D. degree. He is currently a Professor and the Dean of the School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture. His research fields mainly include machine learning and data mining, computational biology and bioinformatics, and image understanding. He is also one of the guiding panel members in some Major Research Plan of National Natural Science Foundation Committee (NSFC), a member of Artificial Intelligence and Pattern Recognition Society of China Computer Federation (CCF), a Standing Committee Member of Machine Learning Society of Chinese Association for Artificial Intelligence (CAAI), and a member of Systems Biology Society of Operations Research Society of China (ORSC).