Quantitative Biology 2019, 7(4): 293–301
https://doi.org/10.1007/s40484-019-0184-7

RESEARCH ARTICLE

WEDeepT3: predicting type III secreted effectors based on word embedding and deep learning

Xiaofeng Fu, Yang Yang*

Department of Computer Science and Engineering, Shanghai Jiao Tong University, and Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai 200240, China
* Correspondence: yangyang@cs.sjtu.edu.cn

Received June 17, 2019; Revised August 13, 2019; Accepted August 26, 2019

Background: **The type III secreted effectors (T3SEs) are one of the indispensable proteins in the growth and reproduction of Gram-negative bacteria. In particular, the pathogenesis of Gram-negative bacteria depends on the type III secreted effectors, and by injecting T3SEs into a host cell, the host cell's immunity can be destroyed. The high diversity of T3SE sequences and the lack of defined secretion signals make it difficult to identify and predict. Moreover, the related study of the pathological system associated with T3SE remains a hot topic in bioinformatics. Some computational tools have been developed to meet the growing demand for the recognition of T3SEs and the studies of type III secretion systems (T3SS). Although these tools can help biological experiments in certain procedures, there is still room for improvement, even for the current best model, as the existing methods adopt hand-designed feature and traditional machine learning methods.**

Methods: **In this study, we propose a powerful predictor based on deep learning methods, called WEDeepT3. Our work consists mainly of three key steps. First, we train word embedding vectors for protein sequences in a large-scale amino acid sequence database. Second, we combine the word vectors with traditional features extracted from protein sequences, like PSSM, to construct a more comprehensive feature representation. Finally, we construct a deep neural network model in the prediction of type III secreted effectors.**

Results: **The feature representation of WEDeepT3 consists of both word embedding and position-specific features. Working together with convolutional neural networks, the new model achieves superior performance to the state-of-the-art methods, demonstrating the effectiveness of the new feature representation and the powerful learning ability of deep models.**

Conclusion: **WEDeepT3 exploits both semantic information of *k*-mer fragments and evolutional information of protein sequences to accurately differentiate between T3SEs and non-T3SEs. WEDeepT3 is available at bcmi.sjtu.edu.cn/~yangyang/WEDeepT3.html.**

Keywords: type III secreted effectors; word2vector; PSSM; feature representation

Author summary: The computational identification of type III secreted effectors (T3SEs) is a very challenging task in bioinformatics. Due to the lack of conserved motifs and the great sequence diversity, it is difficult to extract informative features from T3SEs for the prediction. To represent features of T3SEs, we exploit word embedding method to capture semantic information of amino acid fragments and also combine commonly used position-specific features of sequence patterns. Driven by the latest deep learning technology, the proposed WEDeepT3 achieves the state-of-the-art prediction performance.

INTRODUCTION

The type III secreted effectors (T3SEs) play crucial roles in the interaction between bacteria and their hosts. They are produced by Gram-negative pathogenic bacteria and injected into the host cells through a needle-like apparatus called type III secretion systems (T3SSs) [1], which account for the vast majority of plant and animal pathogens, such as *Pseudomonas, Erwinia, Xanthomonas, Ralstonia, Salmonella, Yersinia, Shigella and Escherichia* [2,3]. Previous studies have shown that these toxic proteins can interfere with host immune signaling networks [1] and help pathogenic bacteria resist the attacks from host immune systems [4]. Moreover, T3SEs can evolve distinct functional domains similar to host cells to interfere with the normal metabolism of host cells [5], and new T3SEs can be evolved through adjustment of the existing T3SE sequences [6].

The T3SEs have important functions for the virulence of pathogens, which makes T3SEs powerful weapons for researchers to explore the immunity and functions of the host cells. Therefore, efficient recognition and large-scale analysis of T3SEs can contribute to the understanding of the mechanism of T3SS. Protein sequencing technology, like high-throughput sequencing technologies, has developed rapidly in the past few decades, and protein data has made great progress in quality and quantity. However, the identification and analysis of T3SEs are relatively slow due to the restriction of labor-intensive experimental methods, and a large proportion of T3SEs remain uncovered [7]. While computational methods have been demonstrated to be useful for revealing unknown T3SEs [6], a few machine learning-based predictors have been developed for the past decade [7–9]. Besides, as the known T3SE sequences accumulated rapidly, several large-scale T3SE databases have emerged, including T3SEdb [10], T3DB [11], *etc*.

Despite recent progress, the performance of these tools is limited by effective feature representation of protein sequences and learning capacity of the prediction model. Due to the lack of defined signals/motifs from known effectors, the recognition of T3SEs is subject to the feature representation of their amino acid sequences. The existing methods mainly adopted hand-designed features. For instance, Yang *et al.* [7,8] proposed the SSE-ACC method (amino acid composition in different secondary structures and solvent accessibility states) and topic models for T3SE recognition. Wang *et al.* [12] proposed a method to extract position-specific feature. Wang made use of the records of the position-specific occurrence time of each amino acid, and analysed the profile to compose features.

Most of these studies adopted shallow learning methods to perform a binary classification (effector and non-effector). For example, although Fu *et al.* utilized continous distributed features for representing amino acide sequences, they fed the features to support vector machines [9]. Some researchers enhanced the prediction performance by using a hierarchical classifier [13,14], *i.e.*, the combination of homology search and machine learning, while this strategy has little advantage for hard targets, which have no homolog in the database of verified effectors. As afore-mentioned, T3SEs evolve fast and have high sequence diversity, thus most of the unknown effectors could not be identified via homology search. Therefore, how to extract informative features from amino acid sequences is key to the prediction of T3SEs. Moreover, for the past decade, deep learning methods have been successfully applied to a lot of bioinformatics tasks related to sequence feature representation and classification. As far as we know, deep learning model has not yet been employed in the recognition of type III effectors.

In this study, we focus on both the feature representation and deep models to enhance the prediction accuracy. In order to employ deep learning models, amino acid sequences need to be first encoded into numeric values. The one-hot encoding is the most widely used method. For protein sequences, each residue is encoded as a 20-dimensional binary vector. The one-hot does not encode context or latent correlation of the residues, thus lose much important information. Instead of using discrete features, we generate continuous feature vectors to represent latent information in the amino acid sequences. Especially, by regarding protein sequences as a special biological language and *k*-mers as words, we develop a similar word embedding algorithm as used in natural language processing [15] to train the distributed representation for *k*-mers, based on an unsupervised learning using deep models. Besides the word embedding features, we also incorporate evolutionary information of amino acid sequences, *i.e.*, position-specific scoring matrix (PSSM), into the feature representation. The position specific features have been demonstrated as the most useful feature in previous studies [12]. Then we feed the combined feature vectors into a convolutional neural network, which further learns high-level abstract features for discriminating the effector from other proteins.

We name the new method WEDeepT3 (Word Embedding and Deep learning for predicting T3SEs). To assess the model performance, we conduct experiments on a cross-species dataset and compare WEDeepT3 with the existing methods on a new independent test dataset. The experimental results show that WEDeepT3 has a competitive performance against existing predictors, and both the word embedding features and deep learning classifier contribute to the performance enhancement.

RESULTS

Data sources

In order to compare the pros and cons of the model more objectively, we use the same data set as the current best model and collect a test set that has never been used before. We collect 713 T3SEs from BEAN 2.0 [13], which is the largest one among the existing T3SE databases (*e.g.,* T3SEdb [10], Effective [16] and Peffect [14]), and use CD-hit [17] to remove sequence redundancy with the sequence identity cutoff of 40%, remaining 239 T3SEs. Meanwhile, we set the same positive-to-negative ratio (1:2) as the training set of BEAN 2.0. Therefore, 478 negative samples are selected from the non-T3SE proteins released in BEAN 1.0 [18], where sequence redundancy has been removed with cutoff 40%.

The independent dataset is collected from the T3DB database [11], which is never used in the training procedure. CD-hit is used to remove sequence redundancy the same as above. Finally, we obtain 46 effectors and 92 non-effectors as the independent dataset. To better demonstrate the performance of our proposed method, we align the independent test set with the training set, where the independent test set has only three blast hits and the identity is below 40%, suggesting that there is no overlap between the independent test set and the training set.

Experimental settings and evaluation criteria

In WEDeepT3, we implement a deep neural network consisting of two one-dimensional convolutional layers and two fully connected layers. Small convolution kernels help reduce the number of parameters that need to be trained and alleviate overfitting. The two convolution kernel sizes are set to 5 and 3, respectively. The mini-batch size is 64, the dropout layer probability is 0.5. As for the focal loss, we use the recommended parameters, *i.e.,*α is 0.25 and γ is 2. In the comparative experiment, we conduct experiments using the traditional classifier SVM and different feature vectors (the result is discussed in Section of "Investigation on the feature representation and classifier"). Our implementation of SVMs adopts the RBF kernel function, where the parameters C and γ are obtained via a grid search using a nested cross-validation.

In order to assess the model performance, we use four metrics, including precision (Equation (1)), recall (Equation (2)), total accuracy (TA) (Equation (3)) and F_1-score (F_1)(Equation (4)).

$$Pre = \frac{TP}{TP + FP}, \tag{1}$$

$$Rec = \frac{TP}{TP + FN}, \tag{2}$$

$$TA = \frac{TP + TN}{TP + FP + TN + FN}. \tag{3}$$

$$F_1\text{-score} = \frac{2 \times TP}{2 \times TP + FP + FN}. \tag{4}$$

Investigation on the feature representation and classifier

As aforementioned, WEDeepT3 consists of three basic elements, *i.e.,* word embedding features, position-specific features and CNN classifier. In this section, we investigate the contributions of these three parts, respectively.

In order to assess the impact of the length of words and also the integration strategy for generating sequence representation from word vectors, we experiment five different lengths of words, *i.e.,* from 1 to 5. As can be seen in Figure 1, the best performance is obtained by the vectors when k equals 3. And, 2-mers, 3-mers and 4-mers have relatively close performance, while 1-mers and 5-mers have much worse performance, indicating that the appropriate length of k-mers is crucial to the accuracy. We did not examine the performance with a larger k as it would lead to much higher computation cost in training the word embedding from the corpus.

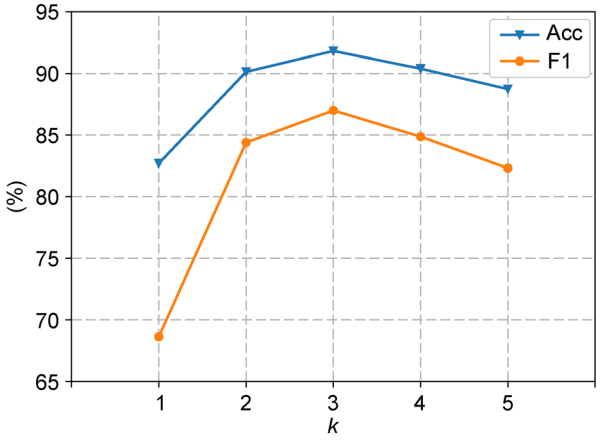

Figure 1. Performance under different length settings for the word embedding feature representation.

As many previous studies adopted PSSM as the major information for prediction T3SEs [13,14], and word embedding (WE) is a new kind of feature representation, we compare the performance of these two different types of features. Meanwhile, we have added two commonly used efficient feature extraction methods (PC-pseAAC and SC-pseAAC [19]) as the baselines. Specifically, we implement four predictors, corresponding to four different combinations of the two feature extraction methods and two classifiers, namely WE + SVM, PSSM + SVM, WE

+ PSSM + SVM and WEDeepT3 (note that WEDeepT3 can also be denoted by WE + PSSM + CNN), where WE + PSSM represents a combination of the word embedding and PSSM feature vectors. The results of four predictors are shown in Figure 2.

As can be seen in Figure 2, the embedding method has a very close performance compared with the PSSM feature extraction method, and far exceeds the other two traditional feature extraction methods, PC-pseAAC and SC-pseAAC working with SVMs. Note that the word embedding vectors and PSSM feature vectors have the dimensionalities of 120 and 400, respectively. With much lower dimensionality, the embedding method demonstrates its powerful representation ability in protein classification tasks.

Furthermore, we combine the embedding vector and the PSSM vector and obtain 520-D feature vectors for classification. The results show that the combination strategy results in a significant enhancement on the performance. Especially, compared with the 400-D PSSM features, the combined feature vectors increase the recall by 12.9%) and the F_1-score by 6.8%. This result suggests that the word embedding feature and PSSM features are complementary to each other, as the former one captures semantic correlation between words while the latter one focuses on position-specific features.

Figure 2 also shows an obvious performance gap between SVMs and CNNs. Considering the limited training data and the risk of overfitting, we only use two convolutional layers, but the CNNs still outperform SVMs by a large margin. The results show that the CNN component in the WEDeepT3 is competent for this classification task, with the precision of 100% and a total accuracy of nearly 100%. All the above experimental results are obtained from ten times ten-fold cross-validation on the same data, *i.e.*, accuracies are averaged over 100 tests.

Comparison with the state-of-the-art predictors

For a fair comparison, we collect an independent test set, and compare WEDeepT3 with 6 other methods on this set, including BPBAac [12], EffectiveT3 [20], T3 MM [21], DeepT3 [22], Bastion3 [23], and BEAN 2.0 [13]. All of these methods have publicly accessible tools and new updates released in past three years. The prediction results are obtained from their web servers or executable programs. The results are shown in Table 1.

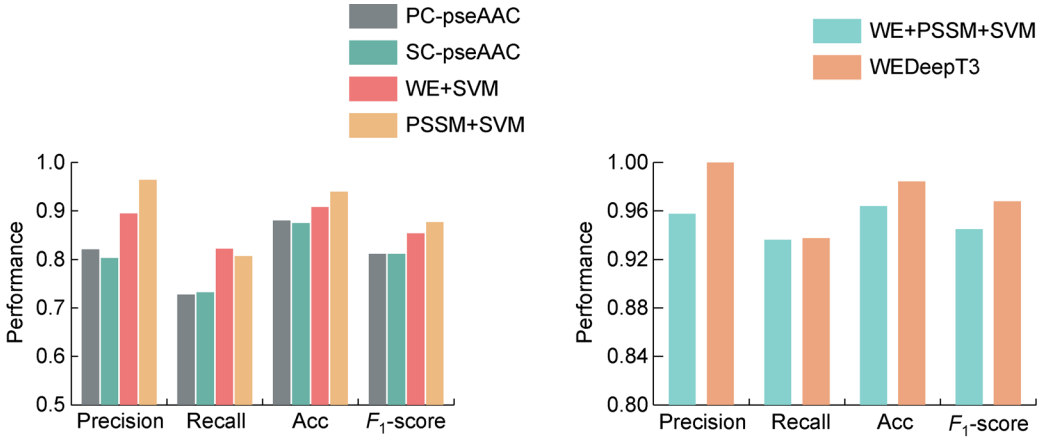

Figure 2. Performance of different feature representation methods and classifers.

Table 1 Result comparison of T3SE prediction methods

Method	Precision	Recall	TA	F_1-score
BPBAac[a]	**0.769**	0.217	0.609	0.339
EffectiveT3[b]	0.550	0.478	0.696	0.512
T3_MM[c]	0.574	0.587	0.718	0.581
DeepT3[d]	0.643	0.391	0.594	0.486
Bastion3[e]	0.578	**0.804**	0.739	0.673
BEAN 2.0[f]	0.607	**0.804**	0.7608	0.692
WEDeepT3	0.750	0.664	**0.812**	**0.705**

[a] BPBAac website(biocomputer.bio.cuhk.edu.hk/softwares/BPBAac), [b] EffectiveT3 website(www.chlamydiaedb.org), [c] T3 MM website(biocomputer.bio.cuhk.edu.hk/softwares/T3_MM), [d] DeepT3 website(github.com/lje00006/DeepT3), [e] Bastion3 website(bastion3.erc.monash.edu/), [f] BEAN 2.0 website(sysbio.cau.edu.cn/bean/)

As can be seen, WEDeepT3 achieves the best performance with the total accuracy of 81.2% and F_1-score of 70.5%, and the total accuracy is over 5% higher than the second-best method, BEAN 2.0. BEAN 2.0 has a close performance with WEDeepT3, which has a higher recall but a lower precision, indicating a higher false positive rate than WEDeepT3. Since the two methods use the same positive-to-negative ratio in the training set, a potential reason for the higher false positive rate is that some homologous proteins of the known effectors are not necessarily true effectors and BEAN 2.0 uses homology search as the initial classification step. By contrast, although BPBAac obtains the highest precision, its recall is much lower than other methods, resulting in poor total accuracy and F_1-score. DeepT3 and Bastion3 are the latest prediction methods for T3SEs. DeepT3 uses a simple one-hot encoding method and a deep learning framework. Bastion3 uses a vector-assembled method that combines multiple traditional vectors and finally uses GDBT as the classifier. Our method is superior to these two methods perhaps due to the feature representation of amino acid sequences and the framework of classifiers. Considering the low identity score of the independent test set and the training set we use, the generalization performance of WEDeepT3 can be guaranteed.

Visualization of the word embeddings

In order to get more insights on the features represented by word embeddings. We map the high-dimensional embedding vectors to a 2D space using t-SNE [24]. Especially, we focus on the first 50 amino acids in N-terminals of T3SEs in the training set, *i.e.*, all the 3-mers in the first 50 amino acids are mapped into the 2D space as shown in Figure 3. The bigger the word, the more frequently it occurs.

Interestingly, the words (*i.e.,* 3-mers) form distributed clusters. The biggest cluster is located at the bottom of the figure. Within this cluster, the most frequent word is SSS, and nearly all the words are centered by S, *e.g.,* SSA, SSK, YSS, ASP, and ISN. This is consistent with an observation in previous studies, *i.e.*, the first 50 amino acids of *P. syringae* effectors have a high proportion of Ser [6]. Besides, in most of the clusters, the words have a common letter in the center. The words within a cluster have close embedding vectors, indicating that they may be interchangeable in the context. By contrast, in previous studies, the hand-designed features, like *k*-mer frequency, are discrete, which can not capture the semantic correlation between words/*k*-mers or calculate the distance between words. It can be observed in the 2D space that the word vectors effectively represent the residue and the context information, thus leading to the performance improvement.

DISCUSSION

The word embedding technology has become the indispensable basis of NLP technology. Once trained, the bioprotein embeddings can be applied to all protein sequence representation tasks. Universality and ease of use make word embeddings stand out in all protein representation methods. Although the current experiments demonstrate the good performance of the word vectors, we can explore this method further, as the segmentation method is relatively simple, and may not be able to distinguish between useful words and useless words which brings noise to the training and prediction system. Thus, one of our future research direction is to develop automatic segmentation method and define more flexible words with varying length. Besides, the deep learning model used in WEDeepT3 is a simple CNN model. As the number of validated effectors increases, which enables the training of much deeper networks, we will explore more complex network architectures to further improve the accuracy.

CONCLUSIONS

In this paper, we propose a deep learning method to predict type III secreted effectors. First, we use an overlapping window to segment protein sequences into words (*k*-mers). Then we convert the words into numerical vectors using word embeddings well trained before via a large corpus of protein sequences. We integrate the word vectors in the sequence to obtain the sequence vector. Further, we incorporate PSSM information into the predictor, thus the predictor exploits both semantic information of the *k*-mers and evolutional information. By using a convolutional neural network, we construct a system to effectively distinguish T3SEs and non-T3SEs, which outperforms most existing prediction methods. In addition, this computational method has strong universality in biological sequence research and can be applied to several sequence analysis tasks.

METHODS

The key components of WEDeepT3 include word embedding features, PSSM features and a convolutional neural network (CNN), and there are five major steps to construct the predictor: (1) segmenting protein sequences into words; (2) learning the word embedding vectors; (3) obtaining the continuous word representation of protein sequences; (4) extracting PSSM vectors from sequence profile; (5) training the CNN classifier. Figure 4 shows the flowchart of WEDeepT3. Details of the 5 steps are described in the following subsections.

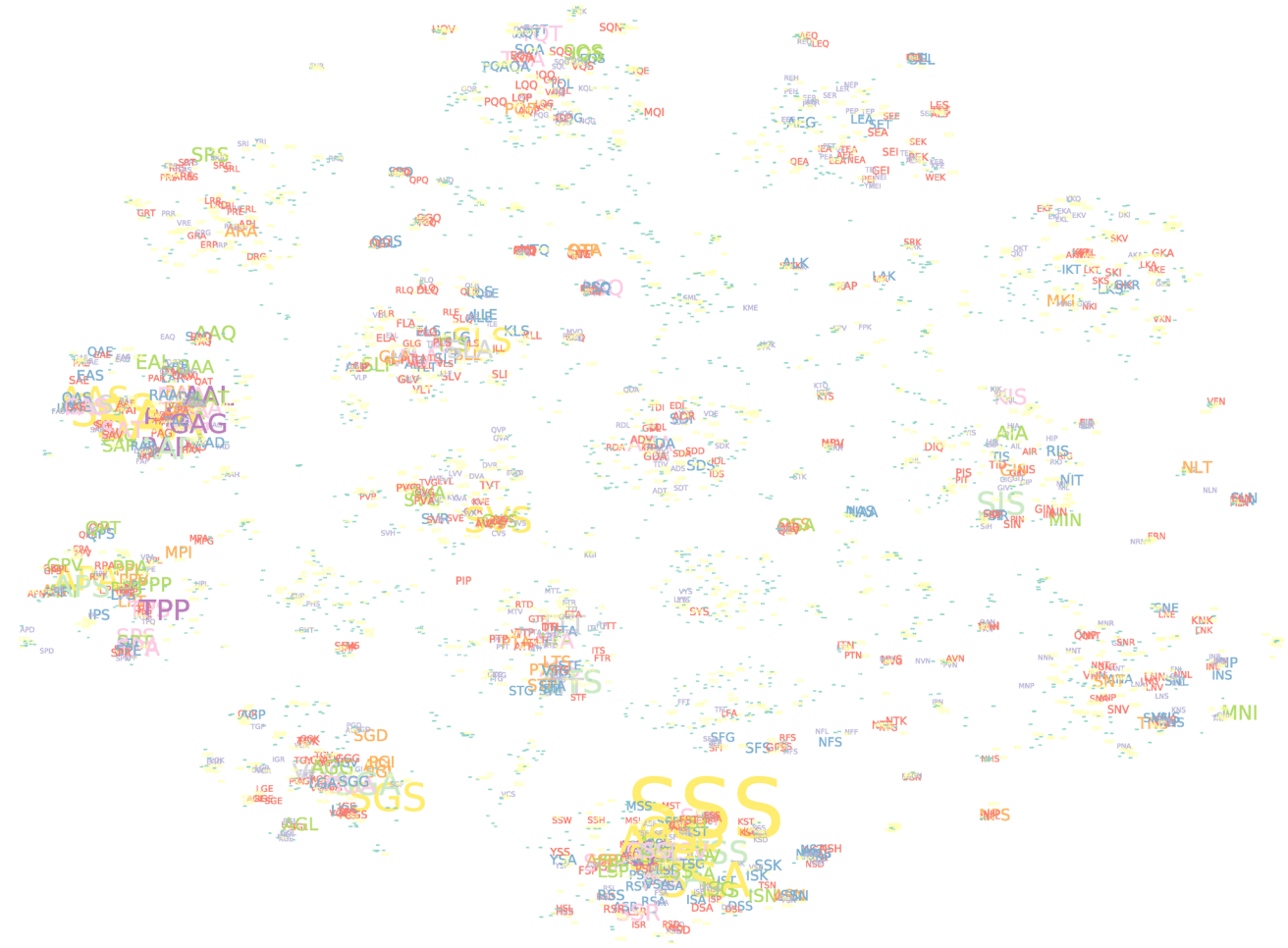

Figure 3. 2D distribution of word vectors for the N-terminal 50 AA of type III secreted effectors.

Word definition for protein sequences

The first type of features in WEDeepT3 is a kind of continuously distributed representation of protein sequences, which is based on the assumption that a biological sequence can be viewed as a sentence written in a special language [25]. However, there are no well defined words. We were inspired by Asgari *et al.* [26] to use the residue segments (*k*-mers) as biological words, and syntax and semantics may correspond to molecular structure and biological function. Analogous to natural language processing tasks, we convert *k*-mers into word embeddings and apply them to the inference of molecular structure, dynamics, and function. Before that, we have to convert a complete sequence of gap-free residues into a list of words according to certain rules.

The previous study [26] used the shifted non-overlapped method, which segments a sequence of length L into multiple lists containing L/k words according to the difference of the starting segmentation sites. However, such a method is equivalent to splitting information of a sequence into multiple copies. Each list of words can only contain a portion of the information, and the obtained word vectors may lose some important information, such as the relationship between the residues within each word. In this study, we have adopted n-gram modeling of protein informatics to segment a sequence into a list of fixed-length words with an overlapping window of length k (Figure 4 shows an example where k is equal to 3). This method could contain more sequence information compared to the method which segments the sequences non-overlappingly.

Generation of embedding vectors of words

With the rise of deep learning techniques in natural language processing (NLP), various word embedding methods have been developed to represent words, sentences and text by continous vectors, such as Word2Vec [15] and Glove [27]. All these algorithms require a large corpus to train the word embeddings.

In order to adapt the word embedding methods to

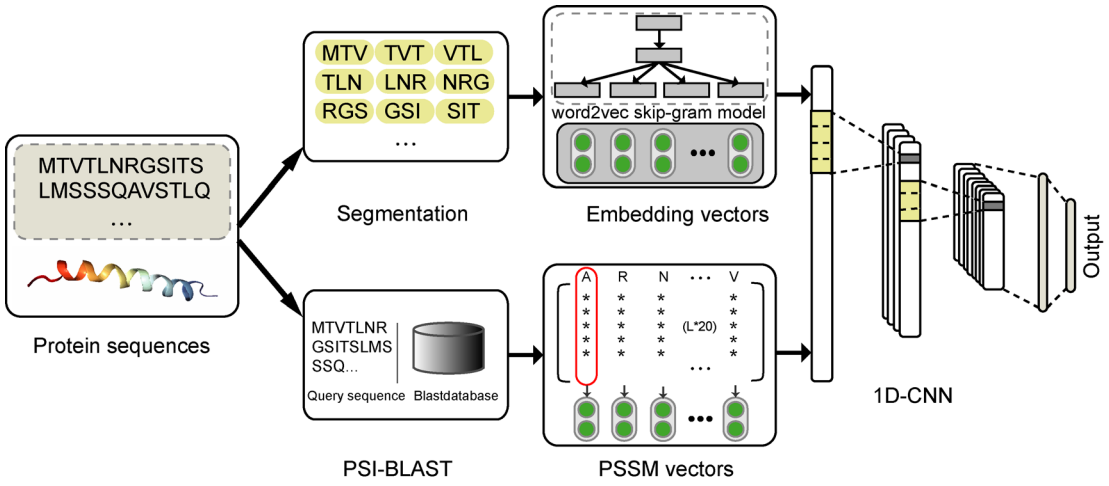

Figure 4. Flowchart of WEDeepT3. The raw sequences are processed by two feature extraction methods which produce embedding vectors and PSSM vectors, respectively. Then the two kinds of vectors are combined and further fed into a 1-D CNN model which yields the final output.

protein sequences, a very large protein database is necessary. At the beginning, Asgari *et al.* [26] used the Swiss-Prot database as the corpus of word embeddings, which contains less than 560,000 sequences. With similar vector spaces, the Swiss-Prot corpus is much smaller compared ones used in NLP tasks. More and more researches and our experimental results indicate that a larger corpus will have a better effect. In this study, we adopt UniRef50 as the corpus, which contains more than 25,000,000 sequences, which is more than the vast majority of NLP corpora. Based on the practice of Word2Vec tasks, this database is far superior to Swiss-Prot one in terms of the number of sequences. In addition, we do not adopt a larger data set, such as Uniref90. Because Uniref50 reduces the redundancy of the database compared to Uniref90, this can effectively alleviate the issue caused by high identity of the sequences.

Given the corpus, we adopt the commonly used Word2Vec algorithm to train the word embeddings for protein sequences. Word2Vec captures the contextual information and trains a fixed-length continuous feature vector for each word. There are two ways to implement Word2Vec algorithm. Here we adopt the Skip-gram model. The objective of the Skip-gram model is to maximize the sum of log-likelihood of each word and its context, as defined in Equation (5),

$$L = \frac{1}{N}\sum_{i=1}^{N}\sum_{i-c\leqslant j\leqslant i+c, i\neq j}\log p(\omega_j|\omega_i), \tag{5}$$

where N is the number of word vectors in the sequence, c is half of the window size, ω_i is the center word, ω_j is one of the context words. The equation of probability is defined as follows:

$$p(\omega_j|\omega_i) = \frac{\exp(v'^{T}_{\omega_j} v_{\omega_i})}{\sum_{\omega=1}^{W}\exp(v'^{T}_{\omega} v_{\omega_i})}, \tag{6}$$

where v_ω and v'_ω are the input and output vector of ω, and W is the total number of words in the corpus.

In natural languages, the number of common words is usually tens of thousands, while the number of k-mers in protein sequences is much larger when k is greater than 4. Therefore, in Equation (6), the softmax parameters may be very difficult to fit. Here, we use the negative sampling method to improve the computation efficiency. When we calculate the probability in Equation (6), we do not calculate all the W words. Instead, we select some negative words using a certain method, so the calculation of softmax function will not be very time-consuming or resource-intensive. The probability that a word is selected as a negative sample is not random, but related to the frequency of its occurrence. And, this probability follows the unigram distribution, as defined below in Equation (7),

$$P(\omega_i) = \frac{f(\omega_i)^{\frac{3}{4}}}{\sum_{j=1}^{W} f(\omega_j)^{\frac{3}{4}}}, \tag{7}$$

where $f(\omega)$ is the frequency of ω.

Continuous representation of protein sequences

The recognition of T3SE is a protein-level classification task. We need to integrate the k-mers' vectors belonging to a protein sequence into a sentence vector and then use it for the downstream classification. Many previous studies utilized N-terminal residues for prediction, while some

other studies showed that non-signal peptides are also helpful for the recognition of T3SEs [28]. And, although T3SEs may have secretory signals at their N-terminals, they have limited peptide sequence conservation. Therefore, here we extract features from full-length sequences for the prediction.

There are some simple ways to aggregate the word embeddings into a combined representation for a protein sequence. For instance, the word vectors can be concatenated as shown in Equation 8,

$$\chi = V_{\omega_1} \oplus V_{\omega_2} \oplus + \cdots V_{\omega_{L-k}} \oplus V_{\omega_{L-k+1}}, \quad (8)$$

where the $\oplus$ operator denotes vector concatenation. Since the overlapping window of length k segments the sequence of length L into $(L-k+1)$ words, each sequence can be represented by a vector of $(L-k+1) * d$ dimensions. The resulted sequence embeddings have varying lengths, and the dimensionality is high.

To avoid the above issue, each sequence can be represented by summing (Equation (9)) or averaging (Equation (10)) all word vectors in the sequence. In our last paper [9], we assess the performance of using both the sum vectors and mean vectors for representing protein sequences.

$$\chi = \sum_i V_{\omega_i}, i \in \{1,2,...,L-k+1\}, \quad (9)$$

$$\chi = \frac{\sum_i V_{\omega_i}}{L-1}, i \in \{1,2,...,L-k+1\}, \quad (10)$$

where ω_i denotes a word, V_{ω_i} denotes the vector of ω_i, and χ is the feature vector for the whole sequence. In these two cases, the dimentionality of all sequences is equal to d, which is the same as the dimension of word vectors.

Extracting evolutionary features from sequence profiles

The second type of features of WEDeepT3 is extracted from the sequence profile of each protein sequence, *i.e.,* position-specific scoring matrix (PSSM), one of the most important features in biological analysis. A PSSM for a query protein is a $L \times 20$ matrix, where L is the length of the protein sequence. It assigns a score $\{P_{ij} | i = 1,...,L \text{ and } j = 1,...,20\}$ for the jth amino acid in the ith position of the query sequence with a large value indicating a highly conserved position and a small value indicating a weakly conserved position. Position Specific Iterated BLAST (PSI- BLAST) [29] is the most commonly used program, which detects remotely related homologous proteins for generating PSSM profiles. In this paper, we use PSI-BLAST program with three iterations and the e-value threshold 0.0001 against the Uniref50 database to generate each PSSM profile. A lot of methods, such as PSSM-AAC and PSSM-DC, have been developed to extract PSSM features efficiently [30,31]. Here we adopt the method developed by Jeong *et al.* [32], which focuses more on domains with similar conservation rates. Specifically, for each particular column in the PSSM, we average the PSSM scores of all 20 amino acids with a PSSM value greater than 0 in the relevant column. We get a 20-dimensional vector from each probe and a 400-dimensional vector from all 20 probes.

The classifier

Here, we adopt a deep neural network as a classifier, which consists of 2 convolutional blocks and 2 fully connected layers. Each convolutional block consists of a convolutional layer, a max pooling layer, a dropout layer and an activation layer, where the activation layer employs the ReLU function. In addition, as effectors are much fewer than non-effectors in the real world, our data set also has an imbalanced class distribution. Here we use focal loss [33] to alleviate the data-imbalance issue.

ACKNOWLEDGEMENTS

This work has been supported by the National Natural Science Foundation of China (No. 61972251).

COMPLIANCE WITH ETHICS GUIDELINES

The authors Xiaofeng Fu and Yang Yang declare that they have no conflict of interests.

This article does not contain any studies with human or animal subjects performed by any of the authors.

REFERENCES

1. Galán, J. E. and Wolf-Watz, H. (2006) Protein delivery into eukaryotic cells by type III secretion machines. Nature, 444, 567–573
2. He, S. Y., Nomura, K. and Whittam, T. S. (2004) Type III protein secretion mechanism in mammalian and plant pathogens. Biochim. Biophys. Acta, 1694, 181–206
3. Cornelis, G. R. (2006) The type III secretion injectisome. Nat. Rev. Microbiol., 4, 811–825
4. Brodsky, I. E. and Medzhitov, R. (2009) Targeting of immune signalling networks by bacterial pathogens. Nat. Cell Biol., 11, 521–526
5. Dean, P. (2011) Functional domains and motifs of bacterial type III effector proteins and their roles in infection. FEMS Microbiol. Rev., 35, 1100–1125
6. Guttman, D. S., McHardy, A. C. and Schulze-Lefert, P. (2014) Microbial genome-enabled insights into plant-microorganism interactions. Nat. Rev. Genet., 15, 797–813
7. Yang, Y., Zhao, J., Morgan, R. L., Ma, W. and Jiang, T. (2010) Computational prediction of type III secreted proteins from gram-negative bacteria. BMC Bioinformatics, 11, S47

8. Yang, Y. and Qi, S. (2014) A new feature selection method for computational prediction of type III secreted effectors. Int. J. Data Min. Bioinform., 10, 440–454
9. Fu, X., Xiao, Y. and Yang, Y. (2018) Prediction of Type III Secreted Effectors Based on Word Embeddings for Protein Sequences. In: Bioinformatics Research and Applications, Zhang, F., Cai, Z., Skums, P., Zhang, S. (eds). Lecture Notes in Computer Science, vol 10847. Springer, Cham
10. Tay, D. M., Govindarajan, K. R., Khan, A. M., Ong, T. Y., Samad, H. M., Soh, W. W., Tong, M., Zhang, F. and Tan, T. W. (2010) T3SEdb: data warehousing of virulence effectors secreted by the bacterial Type III Secretion System. BMC Bioinformatics, 11, S4
11. Wang, Y., Huang, H., Sun, M., Zhang, Q. and Guo, D. (2012) T3DB: an integrated database for bacterial type III secretion system. BMC Bioinformatics, 13, 66
12. Wang, Y., Zhang, Q., Sun, M. A. and Guo, D. (2011) High-accuracy prediction of bacterial type III secreted effectors based on position-specific amino acid composition profiles. Bioinformatics, 27, 777–784
13. Dong, X., Lu, X. and Zhang, Z. (2015) Bean 2.0: an integrated web resource for the identification and functional analysis of type III secreted effectors. Database, 2015, bav064
14. Goldberg, T., Rost, B. and Bromberg, Y. (2016) Computational prediction shines light on type III secretion origins. Sci. Rep., 6, 34516
15. Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013) Efficient estimation of word representations in vector space. arXiv: 1301.3781
16. Jehl, M.-A., Arnold, R. and Rattei, T. (2011) Effective—a database of predicted secreted bacterial proteins. Nucleic Acids Res., 39, D591–D595
17. Li, W. and Godzik, A. (2006) Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics, 22, 1658–1659
18. Dong, X., Zhang, Y.-J. and Zhang, Z. (2013) Using weakly conserved motifs hidden in secretion signals to identify type-III effectors from bacterial pathogen genomes. PLoS One, 8, e56632
19. Chou, K. C. and Com, M. P. (2001) Prediction of protein cellular attributes using pseudo-amino acid composition. Proteins, 43, 246–255
20. Arnold, R., Brandmaier, S., Kleine, F., Tischler, P., Heinz, E., Behrens, S., Niinikoski, A., Mewes, H. W., Horn, M. and Rattei, T. (2009) Sequence-based prediction of type III secreted proteins. PLoS Pathog., 5, e1000376
21. Wang, Y., Sun, M., Bao, H. and White, A. P. (2013) T3_MM: a Markov model effectively classifies bacterial type III secretion signals. PLoS One, 8, e58173
22. Xue, L., Tang, B., Chen, W. and Luo, J. (2019) DeepT3: deep convolutional neural networks accurately identify Gram-negative bacterial type III secreted effectors using the N-terminal sequence. Bioinformatics, 35, 2051–2057
23. Wang, J., Li, J., Yang, B., Xie, R., Marquez-Lago, T. T., Leier, A., Hayashida, M., Akutsu, T., Zhang, Y., Chou, K.-C., *et al.* (2019) Bastion3: a two-layer ensemble predictor of type III secreted effectors. Bioinformatics, 35, 2017–2028
24. Maaten, L. d. and Hinton, G. (2008) Visualizing data using t-sne. J. Mach. Learn. Res., 9, 2579–2605
25. Klein-Seetharaman, J., Reddy, R. (2002) Biological language modeling: Convergence of computational linguistics and biological chemistry. In: Converging Technologies for Improving Human Performance, pp. 378, Springer
26. Asgari, E. and Mofrad, M. R. (2015) Continuous distributed representation of biological sequences for deep proteomics and genomics. PLoS One, 10, e0141287
27. Pennington, J., Socher, R. and Manning, C. (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543
28. Deng, W., Marshall, N. C., Rowland, J. L., McCoy, J. M., Worrall, L. J., Santos, A. S., Strynadka, N. C. J. and Finlay, B. B. (2017) Assembly, structure, function and regulation of type III secretion systems. Nat. Rev. Microbiol., 15, 323–337
29. Altschul, S. F. and Koonin, E. V. (1998) Iterated profile searches with PSI-BLAST—a tool for discovery in protein databases. Trends Biochem. Sci., 23, 444–447
30. Zuo, Y. C., Chen, W., Fan, G. L. and Li, Q. Z. (2013) A similarity distance of diversity measure for discriminating mesophilic and thermophilic proteins. Amino Acids, 44, 573–580
31. Zuo, Y. C. and Li, Q. Z. (2009) Using reduced amino acid composition to predict defensin family and subfamily: Integrating similarity measure and structural alphabet. Peptides, 30, 1788–1793
32. Jeong, J. C., Lin, X. and Chen, X. W. (2011) On position-specific scoring matrix for protein function prediction. IEEE/ACM Trans. Comput. Biol. Bioinformatics, 8, 308–315
33. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P. (2017) Focal loss for dense object detection. IEEE T. Pattern Anal. Mach. Intell., 99, 2999–3007

