Briefings in Bioinformatics, 21(5), 2020, 1825–1836

doi: 10.1093/bib/bbz120
Advance Access Publication Date: 6 November 2019
Problem solving protocol

Downloaded from https://academic.oup.com/bib/article/21/5/1825/5612159 by National Science and Technology Library -Root user on 05 October 2020

Convolutional neural network-based annotation of bacterial type IV secretion system effectors with enhanced accuracy and reduced false discovery

Jiajun Hong, Yongchao Luo, Minjie Mou, Jianbo Fu, Yang Zhang, Weiwei Xue, Tian Xie, Lin Tao, Yan Lou and Feng Zhu

Corresponding author: Feng Zhu, College of Pharmaceutical Sciences, Zhejiang University, Hangzhou 310058, China. E-mail: zhufeng@zju.edu.cn; Yan Lou, The First Affiliated Hospital, Zhejiang University, Hangzhou 310000, China. E-mail: yanlou@zju.edu.cn; Lin Tao, School of Medicine, Hangzhou Normal University, Hangzhou 310036, China. E-mail: taolin@hznu.edu.cn.

Abstract

The type IV bacterial secretion system (SS) is reported to be one of the most ubiquitous SSs in nature and can induce serious conditions by secreting type IV SS effectors (T4SEs) into the host cells. Recent studies mainly focus on annotating new T4SE from the huge amount of sequencing data, and various computational tools are therefore developed to accelerate T4SE annotation. However, these tools are reported as heavily dependent on the selected methods and their annotation performance need to be further enhanced. Herein, a convolution neural network (CNN) technique was used to annotate T4SEs by integrating multiple protein encoding strategies. First, the annotation accuracies of nine encoding strategies integrated with CNN were assessed and compared with that of the popular T4SE annotation tools based on independent benchmark. Second, false discovery rates of various models were systematically evaluated by (1) scanning the genome of *Legionella pneumophila subsp. ATCC 33152* and (2) predicting the real-world non-T4SEs validated using published experiments. Based on the above analyses, the encoding strategies, (*a*) position-specific scoring matrix (PSSM), (*b*) protein secondary structure & solvent accessibility (PSSSA) and (*c*) one-hot encoding scheme (Onehot), were identified as well-performing when integrated with CNN. Finally, a novel strategy that collectively considers the three well-performing models (CNN-PSSM, CNN-PSSSA and CNN-Onehot) was proposed, and a new tool (CNN-T4SE, https://idrblab.org/cnnt4se/) was constructed to facilitate T4SE annotation. All in all, this study conducted a comprehensive analysis on the performance of a collection of encoding strategies when integrated with CNN, which could facilitate the suppression of T4SS in infection and limit the spread of antimicrobial resistance.

Key words: bacterial secretion system; T4SE; effector protein; function annotation; convolution neural network

Jiajun Hong, Yongchao Luo, Jianbo Fu and Yang Zhang are doctoral, master and undergraduate students of the College of Pharmaceutical Sciences in Zhejiang University and jointly cultivated by School of Pharmaceutical Sciences in Chongqing University. They are interested in Artificial Intelligence.
Minjie Mou is the undergraduate student of the College of Pharmaceutical Sciences in Zhejiang University.
Weiwei Xue is an associate professor of the School of Pharmaceutical Sciences in Chongqing University, China. He is interested in the area of computer-based drug design and molecular dynamics simulation.
Tian Xie and Lin Tao are professors of the School of Medicine in Hangzhou Normal University, China. They are interested in the area of Traditional Chinese Medicine, bioinformatics and machine learning.
Yan Lou is an associate professor of the First Affiliated Hospital in Zhejiang University, China. She is interested in the area of clinical pharmacology, precision medicine, metabolomics and bioinformatics.
Feng Zhu is a professor of College of Pharmaceutical Sciences in Zhejiang University, China. He got his PhD degree from the National University of Singapore. His research group (https://idrblab.org/) has been working in the fields of bioinformatics, OMIC-based drug discovery, system biology and medicinal chemistry since 2014. Welcome to visit his personal website at https://idrblab.org/Peoples.php.
Submitted: 23 June 2019; **Received (in revised form):** 12 August 2019

© The Author(s) 2019. Published by Oxford University Press. All rights reserved. For Permissions, please email: journals.permissions@oup.com

Introduction

Bacterial secretion system (SS) plays pivotal roles in the invasion of pathogenic bacteria into the host cell by transporting virulence factors [1]. Among those diverse types of SS characterized [2], type IV SS (T4SS) is reported to be one of the most ubiquitous ones in nature [3], which can induce whooping cough [4], gastritis [5] and crown-gall tumor [6] by secreting type IV SS effectors (T4SEs) into host cells [7]. Recent studies mainly focus on annotating novel T4SEs from the huge amount of sequencing data [8] and revealing their mechanisms underlying the bacteria invasion [9–12]. Particularly, a variety of experimental methods, such as host hypersensitive response suppression [13], immunoblot analysis [14] and pull-down assay [15], are applied for characterizing novel T4SEs and result in the discovery of >450 T4SEs, which can be used to suppress T4SS and drug resistance during infection [16].

However, the available experimental approaches are reported to be very inefficient in identifying new T4SEs [17], and they are incapable of accomplishing the large-scale screenings of the entire bacterial genome [9, 18]. Therefore, various computational methods are designed to ensure the comprehensive and timely annotation of T4SEs [19], which can be classified into the methods based on the *similarity* [20] and *machine learning* [21]. On the one hand, *similarity*-based methods identify T4SEs by sequence homology or the similar secondary/tertiary structure to known effectors [20, 22]. Since T4SEs show great sequence variations among different bacterial species or even strains [7], these *similarity*-based methods are reported to be not always applicable [11]. On the other hand, the *machine learning*-based methods are used to identify key characteristics of sequence, conservation profile, regulatory element, cognate chaperone, secretion signal or physicochemical property [10, 21, 23]. Because of their ability in detecting new T4SEs regardless of the sequence similarity to the existing effector [24], various *machine learning*-based methods (such as *T4SEpre* [8] & *T4EffPred* [11]) are developed, but they are reported to be dependent on the selected method and their performance requires further enhancement [21].

To cope with the problems of *similarity*-based and *machine learning*-based methods, a combinatorial strategy of the majority votes by different *machine learning*-based methods is proposed, and several novel tools, such as *Bastion4* [9], are developed. *Bastion4* integrates six *machine learning* methods and is distinguished by not only its independency on the sequence similarities but also its higher prediction performance than any of these integrated machine learning approaches [9]. However, the combination of multiple methods can heavily enhance the complexity of the annotation model [9, 25], which makes it infeasible for researchers, especially the non-programmers, to combine multiple approaches for T4SE annotation. As an independent technique, *deep learning* has been frequently and successfully applied in sequence/omics analyses [26–29] and medical imaging/signal processing [30–33], which shows the remarkable performance [34–37]. Thus, it is essential to develop new tools to simultaneously enhance T4SE annotation performance and improve the practical application of the annotation tool.

Herein, a novel *convolution neural network* (CNN) technique was identified and applied to annotate T4SEs by integrating multiple protein encoding strategies. First, the annotation accuracies of multiple encoding strategies integrated with CNN were assessed and compared with that of a variety of popular T4SE annotation tools (*Bastion4*, T4SEpre_bpbAac, T4SEpre_Joint & T4SEpre_psAac) based on an independent benchmark. Second, false discovery rates (FDRs) of various models were systematically evaluated by (1) scanning the whole genome of *Legionella pneumophila subsp. pneumophila (strain Philadelphia 1/ATCC 33152/DSM 7513)* and (2) predicting the real-world true non-T4SEs validated using published experiment. Based on above analyses, three encoding strategies, (*a*) position-specific scoring matrix (PSSM), (*b*) protein secondary structure & solvent accessibility (PSSSA) and (*c*) one-hot encoding scheme (Onehot), were identified in this study to be powerful in T4SE annotation when integrated with CNN. Finally, to ensure both the high enrichment and low false positive rate (FPR), a novel strategy that collectively considers the three newly identified best-performing models (CNN-PSSM, CNN-PSSSA & CNN-Onehot) was proposed, and a new software tool (CNN-T4SE) was constructed to facilitate the annotation of T4SEs. All in all, this study conducted a comprehensive analysis on the performance of a collection of encoding strategies when integrated with CNN, which could facilitate the suppression of T4SS during infection and limit the spread of antimicrobial resistance.

Materials and methods

Benchmark datasets collected for the analyses in this study

In total, 420 T4SEs and 1262 non-T4SEs were directly collected from a pioneer study that annotated the T4SE protein [9]. Using the same strategy in that study, these proteins were divided into training and independent test datasets. Particularly, there were 390 T4SE and 1112 non-T4SE proteins in the training dataset after considering sequence redundancy [9]. Meanwhile, the independent test dataset was made up of 30 T4SEs and 150 non-T4SEs from the same study [9].

Moreover, in order to assess the FDR of studied methods, two additional datasets were collected. (1) 2950 proteins encoded in the genome of *Legionella pneumophila subsp. pneumophila (strain Philadelphia 1/ATCC 33152/DSM 7513)* were directly downloaded from the *UniProt* database [38]. This strain was known as containing a rather exceptional number of SSs [39] as well as including the largest number of validated T4SEs among other T4SE-related strains [9, 40]. In other words, since it was estimated to encode a repertoire of over 300 T4SEs [24, 40], this strain was selected and collected here to evaluate the FDR of the studied method. (2) 1385 experimentally validated real-world true non-T4SEs were collected from an independent study [21] that explored the annotation of T4SE. Both datasets were used to assess the level of false discovery in this study.

A variety of techniques adopted in this study for protein sequence encoding

Currently, a variety of protein-encoding techniques were available for protein function prediction [9]. Particularly, ≥8 popular techniques were frequently applied, which included the (1) PSS [41], (2) protein solvent accessibility (PSA) [42], (3) Onehot [43], (4) native disorder (Diso) [44], (5) PSSM [45], (6) smoothed PSSM (SmoPSSM) [46], (7) amino acid composition (AAC) [47] and (8) composition, transition & distribution (CTD) features [18]. All these techniques were applied in this study to facilitate T4SE annotation.

PSSSA

Three encoding strategies under this technique were adopted, and their annotation performances were assessed, which included PSS, PSA and PSSSA. Taking PSSSA as example, it

Downloaded from https://academic.oup.com/bib/article/21/5/1825/5612159 by National Science and Technology Library -Root user on 05 October 2020

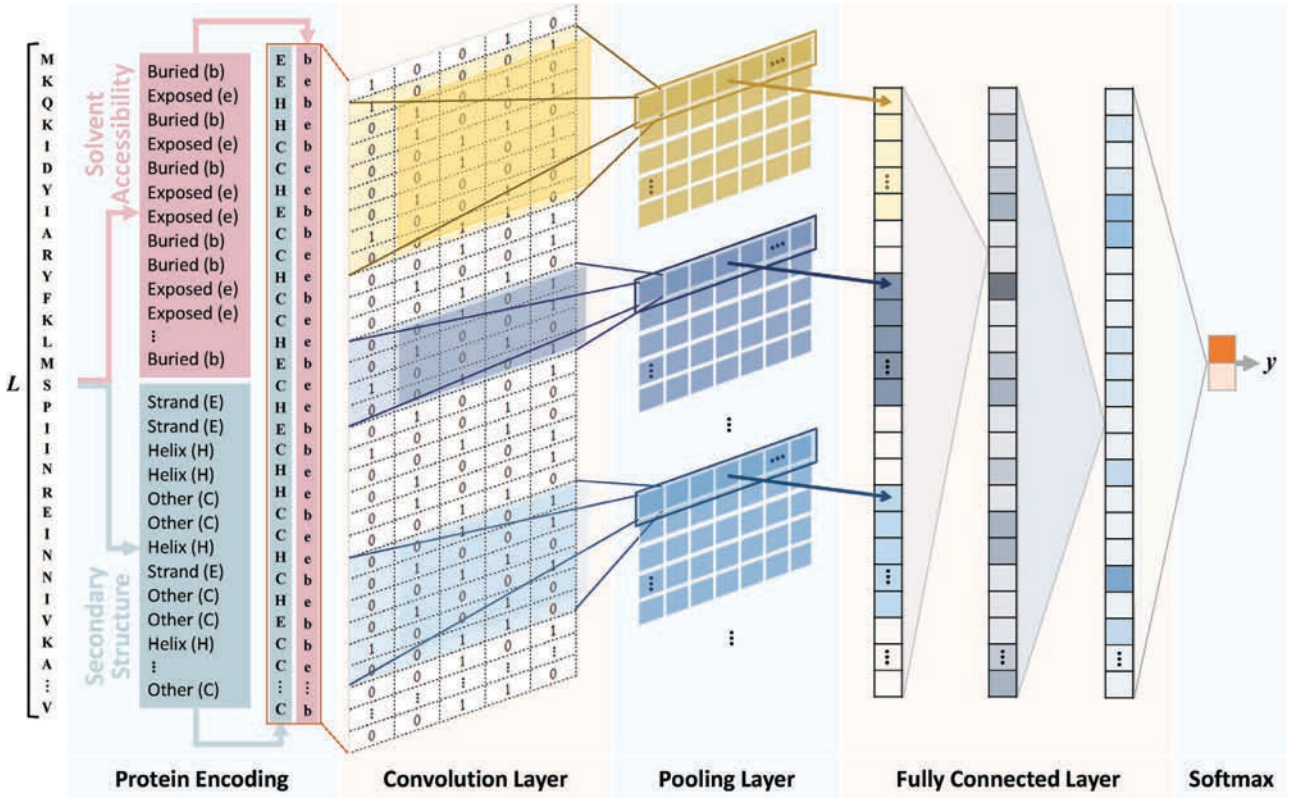

Figure 1. Workflow of the CNN strategy applied in this study together with a graphic illustration of the encoding technique, combining *solvent accessibility* with *secondary structure*, adopted in this study. The SCRATCH Protein Predictor [41] tool (both online version and stand-alone version) was applied to achieve the protein encoding.

generated a 1000×5 binary array for any protein sequence. As illustrated in the *Protein Encoding* part of Figure 1, an amino acid (aa) sequence was first represented by its (*a*) secondary structure ('H', 'E' and 'C' were applied to indicate helix, strand and other, respectively, PSS in Supplementary Figure S1) and (*b*) solvent accessibility ('b' and 'e' were used to denote buried and exposed, respectively, PSA in Supplementary Figure S1). In other words, PSS and PSA helped to transform the given aa sequence to two new sequences. Second, each 'aa' represented by PSS was encoded by a three-dimensional vector ([0,0,1], [0,1,0] and [1,0,0] denoted 'C', 'H' and 'E', respectively), and each 'aa' represented by PSA was encoded by a two-dimensional vector ([0,1] and [1,0] indicated 'e' and 'b', respectively). Third, as shown in the *Convolution Layer* part of Figure 1, each position of the sequence could be encoded by a five-dimensional vector through combining the vectors of PSS and PSA. Moreover, only the proteins whose sequence length was no more than 1000 aa were analyzed in this study. For sequence >1000, its C-terminal 1000-aa fragment was chosen. For the sequence ≤1000, their empty aa positions were complemented using [0,0,0,0,0]. To achieve the protein encoding based on the above three strategies (PSS, PSA and PSSSA) in this study, the SCRATCH Protein Predictor [41] was applied.

Onehot

Onehot was applied to represent a studied protein based on its aa sequence, which had been widely applied to predict acetylation site [48] and annotate RNA-binding protein [49]. It did not require to convert sequence to other forms of feature. Each aa was represented by a corresponding 20-dimensional vector, in which the legal combinations of values were only those with a single '1' bit and all the others '0'. In other words, as illustrated in Supplementary Figure S2, a protein sequence of length *L* was encoded as an $L \times 20$ matrix, where the number 20 was corresponding to the twenty common aas. Each row in the matrix consists of nineteen '0' and a single '1', with the position of the '1' indicating the aa at that position in the proteins. The aas other than these 20 were represented by a 20-dimensional vector with every dimension setting to '0'. For the sequences >1000, their C-terminal 1000-aa fragment was chosen, and for the sequences ≤1000, their empty aa positions were complemented using twenty '0'. To achieve the one-hot protein encoding, the corresponding program was realized by *Python* programming.

Native disorder of proteins

The intrinsically disordered regions with the annotated protein-binding site were applied in this study as another protein encoding technique [44]. To get Diso encoding, DISOPRED3 [44] was first adopted, which converted the protein sequences (*fasta* format) to two types of file (*.pbdat* and *.diso*). In *.pbdat* file, protein-binding disordered residues were marked with carets (^), disordered residues not binding proteins were marked with dashes (–) and ordered residues were marked with dots (.). In *.diso* file, an asterisk (*) referred to disordered residue and a dot (.) referred to ordered residue. Then, as shown in Supplementary Figure S3, each residue represented by intrinsically disordered regions was encoded by a two-dimensional vector ([0,1] and [1,0] referred to '*', and '.', respectively), and each aa represented by annotated protein-binding sites was encoded by a three-dimensional vec-

Downloaded from https://academic.oup.com/bib/article/21/5/1825/5612159 by National Science and Technology Library -Root user on 05 October 2020

tor ([0,0,1], [0,1,0] and [1,0,0] indicated '^', '–' and '.', respectively). For the sequences >1000, their C-terminal 1000-aa fragment was used, and for the sequences ≤1000, their empty aa positions were complemented using five '0'. Finally, these two were combined to construct a five-dimensional vector for each residue, and a 1000×5 feature matrix was therefore generated to establish annotation models in this study.

PSSM and SmoPSSM

PSSM was a global sequence encoding strategy that provided the evolutionary information of protein. Supplementary Figure S4 illustrates a standard PSSM profile. As shown, the (*i*, *j*)-th entry of the matrix indicated the log of the probability that residue in the *i*-th position mutated to aa type *j* [50], which was realized by PSI-BLAST [11] with default parameter *j*=3 & *h*=0.001. For sequences >1000, their 1000-aa fragment was used, and for sequences ≤1000, their empty aa sites were complemented using 20 '0'. As a result, it generated a 1000×20 binary array for any protein sequence. Moreover, as a derivation of the PSSM coding strategy, SmoPSSM was a transformation of the standard PSSM encoding by replacing the vector of residue αi with the sum of surrounding row vectors [50]. The SmoPSSM profile considered the first 50 aas at the protein's N-terminus to generate a 50×20 matrix (shown in Supplementary Figure S4). Both standard PSSM and SmoPSSM encodings could be achieved using the POSSUM server [45].

CTD and AAC

CTD features were a popular encoding technique that converted a protein sequence into a digital feature vector based on the characteristics of each residue within that protein [18]. The studied characteristics included (1) AAC, (2) hydrophobicity, (3) PSS, (4) surface tension, (5) polarizability, (6) solvent accessibility, (7) polarity, (8) charge and (9) van der Waals volume [51]. Then, three features (*composition*, *transition* & *distribution*) were used to describe each property [36]: (*a*) *composition* (number of residues of particular property over the total number of residues; (*b*) *transition* (the percentage of residues with a certain property was followed by residues with a different property); (*c*) distribution (the sequence lengths within which the 1st, one fourth, half, three-quarters and all of the residues of specific property were localized). Detailed information on how to construct the CTD characteristics was provided in previous studies [52, 53]. Moreover, AAC was also a popular technique for protein encoding, which converts a protein sequence to a 20-dimensional vector, where each number referred to the global composition of a given aa. In other words, the AAC was a simplified CTD technique that encoded the protein by a 188-dimensional feature vector according to the characteristics of each residue within the protein [53].

Based on the above analysis, nine protein encoding techniques were applied in this study for annotating T4SE proteins, which included the (1) PSSSA, (2) PSS, (3) PSA, (4) One-hot, (5) Diso, (6) PSSM, (7) SmoPSSM, (8) AAC and (9) CTD. All encoded features were analyzed using the newly constructed CNN strategy in this study. Different from the first seven encoding strategies, the AAC and CTD converted a sequence to a vector other than a $n \times m$ binary array, which made them suitable to be analyzed by the traditional machine learning methods (such as support vector machine (SVM)) [52, 53]. Therefore, these two techniques were further assessed using SVM in this study.

The *Deep Learning* strategy adopted for T4SE annotation in this work

To apply *deep learning* method to T4SE annotation, the CNN technique was applied in this study to construct the function annotation models. As illustrated in Figure 1, CNN consisted of five distinct layers: one convolutional layer, one pooling layer, two fully connected layers and one softmax layer. First, the encoding array connected directly with the convolution layer which scanned the encoding array through a $m_k \times 5$ convolution kernel and resulted in a feature vector. Second, a max pooling layer was adopted, and the maximum neuron output value of the feature vector was selected to be the output of the pooling layer. To fully extract the protein feature, eight different lengths of convolution kernel (there are 120 kernels for each length) were used for scanning the protein encoding array. Thus, after the pooling layer, a vector containing 960 outputs for each sequence was obtained. Third, based on this vector, the fully connected layers generated the output for each layer. Finally, the output vector of the fully connected layer was further used as the input of the softmax layer, which gave corresponding classification probability (the y in Figure 1) to a query protein.

This newly constructed annotation model was implemented with *Python* programming language and *TensorFlow* library. The *binary cross-entropy loss* function was adopted to train the models, and the adaptive moment estimation (Adam) [54–56] was used to optimize the parameters through computing the adaptive learning rates. Particularly, the Adam not only stored an exponentially decaying average of the past squared gradients v_t, but also kept an exponentially decaying average of the past gradients m_t [57]. The m_t and v_t were estimates of the first moment (mean) together with the second moment (uncentered variance) of the gradients, respectively, and biased towards zero (especially in initial time steps). Moreover, a variety of parameters were set as default (which included (1) learning rate (η) = 0.001, (2) exponential decay rate $\beta 1$ for the running average of gradient = 0.9, (3) exponential decay rate $\beta 2$ for the running average of the square of gradient = 0.999 and (4) smoothing term (ε) = 10^{-8}) during back-propagation optimization. The weights of each neuron in the neural network constructed in this study were initialized by the *He* initialization method [58], and the *biases* were initialized to zero. The *batch* normalization was applied in the fully connected layers before *ELU activation* function for accelerating the speed of convergence. To prevent the potential overfitting problem of the constructed model, the effective strategy (*dropout* [59] and *regularization* [60]) helping to avoid the model overfitting dilemma was applied in this study. Particularly, a dropout strategy (randomly removing a certain number of neurons at each training step) was used to a fully connected layer by setting the drop rate to 0.6 [59], and a weight decay parameter *lambda* of L2 regularization strategy (forcing weights to decay towards zero, but not exactly zero) was set to 0.001 [59]. Detailed information of this CNN strategy was fully provided in the Supplementary Method S1.

The *Machine Learning* strategy applied for annotating T4SEs in this study

The machine learning strategy assessed in this study was SVM, which had been widely applied in protein function annotation [51–53]. As one of the supervised learning methods, SVM was used here to classify proteins into two groups (T4SEs versus non-T4SEs). The details of the SVM algorithm and computational procedure could be found in previous publications [51–53]. In this

Downloaded from https://academic.oup.com/bib/article/21/5/1825/5612159 by National Science and Technology Library -Root user on 05 October 2020

study, a nonlinear SVM was applied based on a kernel function as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{e}^{-|\mathbf{x}_j - \mathbf{x}_i|^2/2\sigma^2}$$

SVM *first* projects feature vectors (generated using CTD and AAC techniques) into a high-dimensional feature space based on the above function. *Then*, a linear hyper-plane was drawn to divide all studied proteins into T4SEs and non-T4SEs. The proteins in the training dataset included 390 T4SE and 1112 non-T4SE proteins, and the independent test dataset was made up of 30 T4SEs and 150 non-T4SEs.

The T4SE annotation models studied and assessed in this work

Based on the nine techniques for protein encoding, CNN and SVM were applied and resulted in 11 T4SE annotation models, which included CNN-PSSSA, CNN-PSS, CNN-PSA, CNN-Onehot, CNN-Diso, CNN-PSSM, CNN-SmoPSSM, CNN-AAC, CNN-CTD, SVM-AAC and SVM-CTD. All the models were comprehensively analyzed and assessed in the following studies. Moreover, two popular models in T3SE annotation were further assessed and compared with these 11 models. First, the popular tool *T4SEpre* [8] was assessed, which contained three powerful models: *T4SEpre_Joint* (trained on the joint features of position-specific AAC, secondary structure and solvent accessibility), *T4SEpre_bpbAac* (trained on the bi-profile *Bayesian* aa compositions) & *T4SEpre_psAac* (trained on the position-specific single-profile *Bayesian* & sequence-based aa compositions). These models were found to considerably outperform previous models in terms of sensitivity, specificity, accuracy, AUC and MCC and expected to be combined in practice for T4SE annotations [8]. Second, another state-of-the-art annotation model assessed in this study was *Bastion4*, which integrated six different *machine learning* techniques and was distinguished by its independency on sequence similarity and its higher prediction performance than any of those integrated techniques [9, 61, 62].

Assessing the accuracy and FDR of the studied annotation models

Five popular metrics, *Accuracy* (AC), *Precision* (PR), *Sensitivity* (SE), *Specificity* (SP) and *Matthews correlation coefficient* (MCC) [9, 18], were adopted in this study to evaluate the performance of each T4SE annotation model. Particularly, AC denoted the percentage of the correctly predicted T4SE and non-T4SE in all studied sequences, and MCC reflected the stability of the particular annotation model and described the correlation between the prediction results and actual protein function [63]. AC was one of the most popular metrics applied for assessing the performances of protein function annotation, and MCC was considered as one of the most comprehensive parameters in any category of predictors due to its full consideration of multiple metrics. To further estimate the performance of the annotation models, 5-fold cross validation (CV) was applied by dividing training dataset into five subsets. In each CV step, one subset constituted the validation set and the remaining four subsets were combined to form a training set. This procedure was repeated five times until all subsets were used as both training and validation. The average performance across all five trials was then calculated.

To assess the FDR of each studied model, a real-world application of genome scanning was performed. Particularly, 2950 sequences encoded within the genome of *Legionella pneumophila subsp. pneumophila (strain Philadelphia 1/ATCC 33152/DSM 7513)*. This strain contained the rather exceptional number of SSs [39] and the largest number of validated T4SEs among other T4SE-related strains [9, 40]. In other words, since it was estimated to encode a repertoire of over 300 T4SEs [24, 40], this strain was collected to evaluate FDR of each studied model. The FDR was evaluated using enrichment factor (EF) for discovering T4SEs. The value of EFs would be no less than zero, and only when the EF value was larger than 1 was there an enrichment. The larger the EF, the lower the false annotation rate of T4SE. The detail information of the EF calculation could be found in the Supplementary Method S2.

Results and discussion

Models' performances assessed by 5-fold CV and independent test data

Performances of the annotation models (CNN-PSSSA, CNN-PSS, CNN-PSA, CNN-Onehot, CNN-Diso, CNN-PSSM, CNN-SmoPSSM, CNN-AAC, CNN-CTD, SVM-AAC, SVM-CTD) constructed in this study were calculated and assessed via the 5-fold CV based on the training dataset of 390 T4SEs and 1112 non-T4SEs. As illustrated in Table 1, SEs, SPs, PRs, ACs and MCCs of each fold were provided. Taking CNN-PSSSA as an example, its average SE, SP, PR, AC and MCC for the 5-fold CV equaled to 71.5%, 93.5%, 81.4%, 87.8%, and 0.68, respectively. The ACs of all models were within the range from 75.6% to 95.3%, and their MCCs were from 0.44 to 0.88. Except for CNN–Diso (MCC = 0.44), all the other models could reach an MCC higher than 0.5.

Moreover, the same independent test data as that used for constructing *Bastion4* [9] were adopted to assess and compare the annotation performances among those studied models. As illustrated in Table 2, the SEs of all 11 newly constructed models (from 63.3% to 96.7%) were found significantly higher than that of those three *T4SEpre* models (43.3%, 50.0% and 36.7% for T4SEpre_bpbAac, T4SEpre_Joint and T4SEpre_psAac, respectively); the SPs of some newly constructed models (such as CNN-PSSSA, CNN-PSA, CNN-Onehot and CNN-PSSM) were comparable to/slightly higher than that of the *T4SEpre* model (98.0%, 98.7% and 99.3% for the T4SE_bpbAac, T4SE_Joint and T4SE_psAac, respectively); the PRs of some new models (such as CNN-PSSSA, CNN-PSA, CNN-Onehot and CNN-PSSM) were higher than those of the *T4SEpre* tools (81.3%, 88.2% and 91.7% for T4SEpre_bpbAac, T4SEpre_Joint and T4SEpre_psAac, respectively); the ACs of the majority of the new models (except for CNN–Diso) were substantially higher than those of the *T4SEpre* model (88.9%, 90.6% and 88.9% for T4SEpre_bpbAac, T4SEpre_Joint and T4SEpre_psAac, respectively); the MCCs of the majority of the 11 new models (except CNN–Diso) were significantly higher than those of the three *T4SEpre* models (0.54, 0.62 and 0.54 for T4SE_bpbAac, T4SE_Joint and T4SE_psAac, respectively). Moreover, ROC curves of 14 studied models (11 newly constructed and 3 *T4SEpre* models) on independent test datasets are illustrated in Figure 2. As shown, the majority of those 11 models (CNN-PSSM, CNN-SmoPSSM, CNN-Onehot, CNN-PSSSA, CNN-PSA, SVM-AAC, SVM-CTD & CNN-AAC) outperformed those three *T4SEpre* models by providing higher AUCs (from 0.928 to 0.996). All in all, based on the ROC curve and overall performance assessment metrics (AC and MCC) on independent data, these new models constructed using the CNN technique were found to be capable of performing better than those three models of *T4SEpre*.

Downloaded from https://academic.oup.com/bib/article/21/5/1825/5612159 by National Science and Technology Library -Root user on 05 October 2020

Table 1. The performances of 11 annotation models constructed in this study assessed by 5-fold CV. TP: true positive; FN: false negative; TN: true negative; FP: false positive; SE: sensitivity; SP: specificity; PR: precision; AC: accuracy; MCC: Matthews correlation coefficient

	Fold	TP	FN	TN	FP	SE	SP	PR	AC	MCC
CNN-PSSSA	1	66	12	195	28	84.6%	87.4%	70.2%	86.7%	0.68
	2	52	26	215	8	66.7%	96.4%	86.7%	88.7%	0.69
	3	48	30	219	3	61.5%	98.6%	94.1%	89.0%	0.70
	4	58	20	213	9	74.4%	95.9%	86.6%	90.3%	0.74
	5	55	23	198	24	70.5%	89.2%	69.6%	84.3%	0.60
	AVE	–	–	–	–	**71.5%**	**93.5%**	**81.4%**	**87.8%**	**0.68**
CNN-PSS	1	54	36	225	15	60.0%	93.8%	78.3%	84.5%	0.59
	2	54	36	207	33	60.0%	86.2%	62.1%	79.1%	0.47
	3	88	2	208	2	97.8%	99.0%	97.8%	98.7%	0.97
	4	34	26	194	16	56.7%	92.4%	68.0%	84.4%	0.52
	5	37	23	200	12	61.7%	94.3%	75.5%	87.1%	0.60
	AVE	–	–	–	–	**67.2%**	**93.2%**	**76.3%**	**86.8%**	**0.63**
CNN-PSA	1	41	49	231	9	45.6%	96.2%	82.0%	82.4%	0.52
	2	53	37	214	26	58.9%	89.2%	67.1%	81.0%	0.50
	3	57	33	182	28	63.3%	86.7%	67.1%	79.7%	0.51
	4	33	27	203	7	55.5%	96.7%	82.5%	87.4%	0.60
	5	40	20	201	11	66.7%	94.8%	78.4%	88.6%	0.65
	AVE	–	–	–	–	**57.9%**	**92.7%**	**75.4%**	**83.8%**	**0.56**
CNN-Onehot	1	62	28	233	7	68.9%	97.1%	89.9%	89.4%	0.72
	2	80	10	191	49	88.9%	79.6%	62.0%	82.1%	0.62
	3	72	18	172	38	80.0%	81.9%	83.9%	85.6%	0.53
	4	26	34	205	5	43.3%	97.6%	83.9%	85.6%	0.53
	5	48	12	183	29	80.0%	96.3%	62.3%	84.9%	0.61
	AVE	–	–	–	–	**72.2%**	**88.5%**	**72.7%**	**84.7%**	**0.62**
CNN-PSSM	1	74	16	215	25	82.2%	89.6%	74.7%	87.6%	0.70
	2	90	0	239	1	100.0%	99.6%	98.9%	99.7%	0.99
	3	72	18	239	1	80.0%	99.6%	98.6%	94.2%	0.85
	4	51	9	210	0	85.0%	100.0%	100.0%	96.7%	0.90
	5	58	2	180	2	96.7%	98.9%	96.7%	98.3%	0.96
	AVE	–	–	–	–	**88.8%**	**97.5%**	**93.8%**	**95.3%**	**0.88**
CNN-SmoPSSM	1	67	23	235	5	74.4%	97.9%	93.1%	91.5%	0.78
	2	72	18	216	24	80.0%	90.0%	75.0%	87.3%	0.69
	3	80	10	189	21	88.9%	90.0%	79.2%	89.7%	0.76
	4	44	16	208	2	73.3%	99.0%	95.7%	93.3%	0.80
	5	49	11	205	7	81.7%	96.7%	87.5%	93.4%	0.80
	AVE	–	–	–	–	**79.7%**	**94.7%**	**86.1%**	**91.0%**	**0.77**
CNN-Diso	1	62	28	205	35	68.9%	85.4%	63.9%	80.9%	0.53
	2	65	25	202	38	72.2%	84.2%	63.1%	80.9%	0.54
	3	58	32	185	25	64.4%	88.1%	69.9%	81.0%	0.54
	4	36	24	189	21	60.0%	90.0%	63.2%	83.3%	0.51
	5	34	26	107	105	56.7%	50.5%	24.5%	51.8%	0.06
	AVE	–	–	–	–	**64.4%**	**79.6%**	**56.9%**	**75.6%**	**0.44**
CNN-AAC	1	53	37	235	5	58.9%	97.9%	91.4%	87.3%	0.66
	2	76	14	209	31	84.4%	87.1%	71.0%	86.4%	0.68
	3	73	17	181	29	81.1%	86.2%	71.6%	84.7%	0.65
	4	57	3	191	19	95.0%	91.0%	75.0%	91.9%	0.79
	5	43	17	192	20	71.7%	90.6%	68.3%	86.4%	0.61
	AVE	–	–	–	–	**78.2%**	**90.5%**	**75.4%**	**87.3%**	**0.68**
CNN-CTD	1	72	18	206	34	80.0%	85.8%	67.9%	84.2%	0.63
	2	42	48	230	10	46.7%	95.8%	80.8%	82.4%	0.52
	3	81	9	157	53	90.0%	74.8%	60.4%	79.3%	0.60
	4	20	40	207	3	33.3%	98.6%	87.0%	84.1%	0.48
	5	50	10	154	58	83.3%	72.6%	46.3%	75.0%	0.47
	AVE	–	–	–	–	**66.7%**	**85.5%**	**68.5%**	**81.0%**	**0.54**
SVM-AAC	1	36	42	212	11	46.2%	95.1%	76.6%	82.4%	0.50
	2	34	44	218	5	43.6%	97.8%	87.2%	83.7%	0.54
	3	37	41	214	8	47.4%	96.4%	82.2%	83.7%	0.54
	4	39	39	213	9	50.0%	95.9%	81.2%	84.0%	0.55
	5	44	34	210	12	56.4%	94.6%	78.6%	84.7%	0.57
	AVE	–	–	–	–	**48.7%**	**96.0%**	**81.2%**	**83.7%**	**0.54**

Continued

Downloaded from https://academic.oup.com/bib/article/21/5/1825/5612159 by National Science and Technology Library -Root user on 05 October 2020

Table 1. (continued)

	Fold	TP	FN	TN	FP	SE	SP	PR	AC	MCC
SVM-CTD	1	50	28	208	15	64.1%	93.3%	76.9%	85.7%	0.61
	2	55	23	204	19	70.5%	91.5%	74.3%	86.0%	0.61
	3	52	26	210	12	66.7%	94.6%	81.2%	87.3%	0.66
	4	52	26	206	16	66.7%	92.8%	76.5%	86.0%	0.65
	5	61	17	197	25	78.2%	88.7%	70.9%	86.0%	0.65
	AVE	–	–	–	–	**69.2%**	**92.2%**	**76.0%**	**86.2%**	**0.63**

AVE: the average performance of 5-fold CV results of each model.

Table 2. Comparison among the annotation performances of different models based on the benchmark independent test data provided in a previous study [9]. CNN: convolution neural network; SVM: support vector machine; PSSSA: protein secondary structure solvent accessibility; PSS: protein secondary structure; PSA: protein solvent accessibility; Onehot: one-hot scheme; Diso: protein native disorder; PSSM: position-specific scoring matrix; SmoPSSM: smoothed PSSM; AAC: amino acid composition; CTD: composition, transition & distribution features; TP: true positive; FN: false negative; TN: true negative; FP: false positive; SE: sensitivity; SP: specificity; PR: precision; AC: accuracy; MCC: Matthews correlation coefficient

Studied model		TP	FN	TN	FP	SE	SP	PR	AC	MCC
CNN	PSSSA	23	7	149	1	76.7%	99.3%	95.8%	95.6%	0.83
	PSS	22	8	146	4	73.3%	97.3%	84.6%	93.3%	0.75
	PSA	21	9	148	2	70.0%	98.7%	91.3%	93.9%	0.77
	Onehot	24	6	150	0	80.0%	100.0%	100.0%	96.7%	0.88
	Diso	19	11	132	18	63.3%	88.0%	51.4%	83.9%	0.47
	PSSM	29	1	149	1	96.7%	99.3%	96.7%	98.9%	0.96
	SmoPSSM	25	5	142	8	83.3%	94.7%	75.8%	92.8%	0.75
	AAC	21	9	141	9	70.0%	94.0%	70.0%	90.0%	0.64
	CTD	22	8	143	7	73.3%	95.3%	75.9%	91.7%	0.70
SVM	AAC	20	10	145	5	66.7%	96.7%	80.0%	91.7%	0.68
	CTD	25	5	141	9	83.3%	94.0%	73.5%	92.2%	0.74
Bastion4		29	1	142	8	96.7%	94.7%	78.4%	95.0%	0.84
T4SEpre_bpbAac		13	17	147	3	43.3%	98.0%	81.3%	88.9%	0.54
T4SEpre_Joint		15	15	148	2	50.0%	98.7%	88.2%	90.6%	0.62
T4SEpre_psAac		11	19	149	1	36.7%	99.3%	91.7%	88.9%	0.54

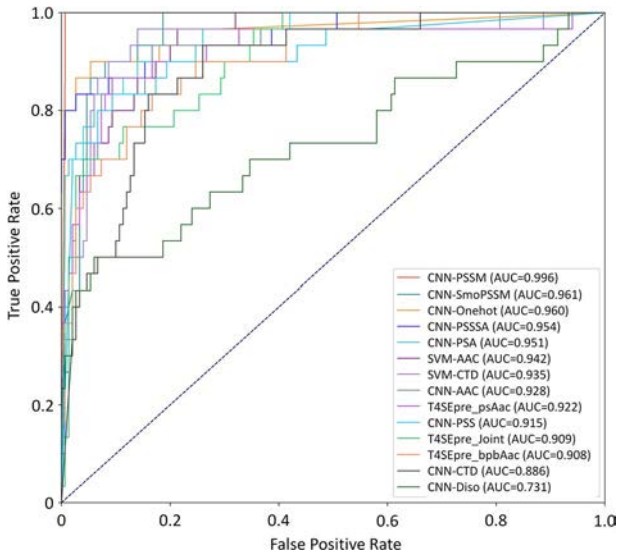

Figure 2. ROC curves of the 14 annotation models (9 CNN models of various encoding strategies, 2 SVM models of distinct encoding strategies and 3 T4SEpre models popular in the previous study) based on independent test data. The AUC value for each model was provided in the lower-right corner, and all models were arranged in the descending order of their corresponding AUC values.

When it came to another popular T4SE annotation tool *Bastion4*, several new models could reach the comparable AC and MCC to that of *Bastion4*. These new models included CNN-PSSM, CNN-Onehot and CNN-PSSSA with the MCC equaling to 0.96, 0.88 and 0.83, respectively. These results indicated the similar annotation accuracy on independent test datasets among those four models. However, the SPs of the three new models (CNN-PSSM, CNN-Onehot and CNN-PSSSA) were substantially higher than that of *Bastion4* (Table 2). As known, the bacteria genomes were usually composed of thousands of proteins, and the slight loss in annotation SP would result in greatly enhanced FDRs [18, 51]. Therefore, it is essential to further assess the false annotation rate of each model.

Considering that the benchmark data used in this study were relatively small and CNN-based algorithm typically require larger datasets for training a robust model, the overfitting problem should be especially considered [59] and, if necessary, an effective strategy helping to avoid the model overfitting dilemma should thus be applied. The commonly applied strategies include dropout [60] and regularization [64]. In this study, both strategies were used. Particularly, the dropout strategy (randomly removing a certain number of neurons at each training step) was applied to the fully connected layers by setting the drop rate to 0.6 [60], and the weight decay parameter *lambda* of L2 regularization strategy (forcing the weights to decay towards zero but not exactly zero) was set to 0.001 [60]. As shown in Supplementary Table S1, the ACs and MCCs with and without the application of dropout and regularization in training and independent test datasets were provided. As demonstrated, both ACs and MCCs of the training dataset with and without

Downloaded from https://academic.oup.com/bib/article/21/5/1825/5612159 by National Science and Technology Library -Root user on 05 October 2020

Table 3. 2950 proteins in the genome of *Legionella pneumophila subsp. pneumophila (Philadelphia 1/ATCC 33152/DSM 7513)* were scanned by 18 different models studied in this work (nine CNN models of various encoding strategies, two SVM models of distinct encoding strategies, four T4SE models popular in the previous study and three voting strategies collectively considering three best-performing models) for assessing their false annotation rates in real-world. The '**Total no. of true T4SEs**' & '**Total no. of proteins**' did not include the proteins used for model construction

Studied model		No. of T4SEs predicted	No. of true T4SEs identified	Total no. of true T4SEs	Total no. of proteins	EF
CNN-T4SE	VOTE 3/3	309	4	6	2950	6.36
	VOTE 2/3	356	6	6	2950	**8.29**
	VOTE 1/3	514	6	6	2950	5.74
	PSSSA	366	5	6	2950	**6.72**
	PSS	452	4	6	2950	4.35
	PSA	385	3	6	2950	3.83
CNN	**Onehot**	382	5	6	2950	**6.44**
	Diso	400	1	6	2950	1.23
	PSSM	431	6	6	2950	**6.84**
	SmoPSSM	497	5	6	2950	4.95
	AAC	682	3	6	2950	2.16
	CTD	573	3	6	2950	2.57
SVM	AAC	617	5	6	2950	3.98
	CTD	435	2	6	2950	2.26
Bastion4		670	5	6	2950	3.67
T4SEpre_bpbAac		349	1	6	2950	1.41
T4SEpre_Joint		496	4	6	2950	3.96
T4SEpre_psAac		303	1	6	2950	1.62

dropout & regularization were roughly consistent with each other, while the AC and MCC of independent test data with dropout & regularization were substantially enhanced compared to that without dropout & regularization. Taking CNN–PSSM as an example, its AC was increased from 94.4% (without) to 98.9% (with), and its MCC was enhanced from 0.82 (without) to 0.96 (with), which indicated an effective overcome of the model overfitting dilemma in this study.

Evaluating the false annotation rate based on genome scanning and non-T4SE data

Besides SP, the EF was known as one of the most popular and effective measures for assessing FDR of any functional annotation method [18]. As known, the SP assessed FDR by only considering the annotation performance on the independent negative test data, while the EF evaluated the false annotation by fully considering the real-world true T4SEs. Therefore, the EF was used in this study to complement SP and make in-depth evaluations on the FDR of studied models. In other words, in order to assess the FDR of each model in the real world, all 15 models (11 new models and 4 popular previous models) were adopted for scanning 2950 proteins in the genome of *Legionella pneumophila subsp. pneumophila (strain Philadelphia 1/ATCC 33152/DSM 7513)*. As shown in Table 3, the total numbers of T4SEs predicted and true T4SEs correctly identified from this particular genome by all the models together with their corresponding EFs were provided. Particularly, five models (CNN-PSSSA, CNN-Onehot, CNN-SmoPSSM, SVM-AAC & *Bastion4*) were identified to correctly annotate five true T4SEs, and the CNN-PSSM was the only model that could discover all six true T4SEs. This result was consistent with above assessments that the annotation accuracies on independent test datasets among the six models were comparable with each other. However, the numbers of T4SEs identified by the *Bastion4* (670) and SVM-AAC (617) were extensively higher than those of CNN-PSSM (431), CNN-PSSSA (366), CNN-Onehot (382) and CNN-SmoPSSM (497), which made the EFs of the later four models significantly higher than those of *Bastion4* and SVM-AAC. Especially, the EFs of CNN-PSSM (6.84), CNN-PSSSA (6.72) & CNN-Onehot (6.44) were 1.75∼1.86 times of *Bastion4*'s EF (3.67). Similar to *Bastion4*, T4SEpre_Joint correctly identified high number of true T4SEs (4), but the number of T4SEs predicted by T4SEpre_Joint (496) was much larger than that of the identified models (CNN-PSSM, CNN-PSSSA & CNN-Onehot), which made their EFs 1.63∼1.73 times of T4SEpre_Joint's EF (3.96). These results indicated the great improvement in controlling the FDRs by three CNN-based models comparing with both *Bastion4* and T4SEpre_Joint.

Moreover, the numbers of T4SEs predicted by the T4SEpre_bpbAac and T4SEpre_psAac equaled to 349 and 303, which were smaller than all those 11 newly constructed models. However, the numbers of true T4SEs correctly identified by these two models (only one for both models) were much lower than those of the three identified powerful models (CNN-PSSM, CNN-PSSSA & CNN-Onehot), which made the EFs of CNN-PSSM (6.84), CNN-PSSSA (6.72) & CNN-Onehot (6.44) 3.98∼4.85 times of EFs of T4SEpre_bpbAac and T4SEpre_psAac. This result indicated great improvement in controlling the FDR by these three newly identified powerful models compared with both T4SEpre_bpbAac and T4SEpre_psAac. Additionally, FDRs could be further assessed by 1385 real-world true non-T4SEs reported in the previous study [21]. As shown in Table 4, the FPRs of CNN-PSSSA, CNN-Onehot and CNN-PSSM were <3%, which indicated relatively low FDR by these models.

Constructing the software tool CNN-T4SE to facilitate T4SE annotation

To construct the software tool of great functionality, the genome scan results by those three identified powerful models were further assessed. As illustrated in Figure 3(A), the Venn diagram of the scanning results among these three models of the best performance was provided. Although the majority of the scanning results of these three models were overlapped with

Downloaded from https://academic.oup.com/bib/article/21/5/1825/5612159 by National Science and Technology Library -Root user on 05 October 2020

Table 4. 1385 real-world non-T4SEs that were reported in previous study [21] were predicted by 18 models (9 CNN models of various encoding strategies, 2 SVM models of distinct encoding strategies, 4 T4SE models popular in previous study and 3 voting strategies collectively considering three best-performing models) studied in this work to assess their FPRs

Studied model		No. of T4SEs predicted	Total no. of true non-T4SEs reported	FPR
CNN-T4SE	VOTE 3/3	4	1385	0.3%
	VOTE 2/3	12	1385	0.9%
	VOTE 1/3	46	1385	3.3%
CNN	PSSSA	18	1385	1.3%
	PSS	39	1385	2.8%
	PSA	21	1385	1.5%
	Onehot	15	1385	1.1%
	Diso	65	1385	4.7%
	PSSM	29	1385	2.1%
	SmoPSSM	41	1385	3.0%
	AAC	109	1385	7.9%
	CTD	100	1385	7.2%
SVM	AAC	123	1385	8.9%
	CTD	101	1385	7.3%
Bastion4		88	1385	6.4%
T4SEpre_bpbAac		23	1385	1.7%
T4SEpre_Joint		135	1385	9.7%
T4SEpre_psAac		4	1385	0.3%

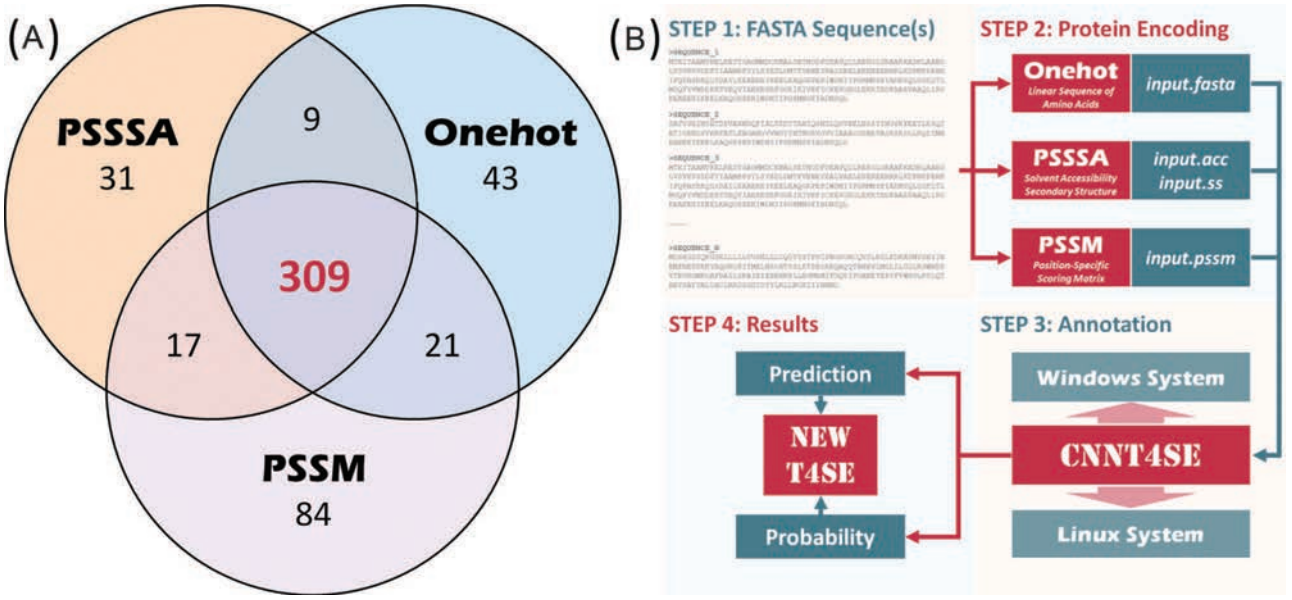

Figure 3. (**A**) Venn diagram of three identified best-performing models (CNN-PSSSA, CNN-PSSM and CNN-Onehot) in scanning the genome (the numbers indicated the T4SEs predicted by each model). (**B**) General workflow of the newly constructed T4SE annotation tool (CNN-T4SE), which was based on a voting strategy that collectively considered the above three best-performing models.

each other (309 proteins were annotated as T4SEs by all three models), there were substantial variations (15.6%∼28.3%) among their scanning results. This finding indicated a great difference among the underlying theories of the three encoding techniques (PSSSA, Onehot & PSSM), which inspired us to further enhance annotation performances by collectively considering three best-performing models (CNN-PSSM, CNN-PSSSA and CNN-Onehot). Thus, three additional novel strategies were considered: (VOTE 1/3) if a protein is annotated as T4SE by any of these three models, this protein is considered to be predicted as a T4SE; (VOTE 2/3) only if a protein is annotated as T4SE by no less than two of these three models can it be considered to be a T4SE; and (VOTE 3/3) only when a protein is annotated by all three models can it be considered to be a T4SE. Based on these three additional strategies, their performances were further assessed, as shown in Tables 3 and 4. As demonstrated in Table 3, VOTE 2/3 achieved the highest EF of 8.29, which is substantially higher than any of the three best-performing models. Although VOTE 3/3 achieved the best FPR, its EF value was significantly hampered by its failure in predicting true T4SEs (EF = 6.36). To ensure both the

Downloaded from https://academic.oup.com/bib/article/21/5/1825/5612159 by National Science and Technology Library -Root user on 05 October 2020

high enrichment and the low FPR, VOTE 2/3 was finally selected in this study to construct the T4SE annotation tool of great functionality.

Therefore, a novel T4SE annotation tool CNN-T4SE (accessible at: https://idrblab.org/cnnt4se/) was constructed to provide the simultaneously enhanced accuracy and reduced FDR, which could thus be adopted as effective complement to other available T4SE annotation tools. The CNN-T4SE was an executable program that could run within both *Windows* and *Linux* operating systems. The users could fully download the program package and the exemplar testing datasets directly from the website. CNN-T4SE was written in *Python*, and a number of python libraries were therefore utilized to ensure the full operation of this annotation tool (which included *Pandas*, *Numpy*, *tensorflow*, *re*, *sys*, *os*, etc.). In the package downloaded from the CNN-T4SE website, CNNT4SE.zip and CNNT4SE.tar.gz were for *Windows* and *Linux*, respectively. The 'predict' file (predict.exe for *Windows*, and predict for *Linux*) was the executable file for T4SE annotations. The folders named as 'PSSSA', 'Onehot' and 'PSSM' contained the parameters of three constructed CNN models, and the 'lib' folder was composed of all python libraries essential for CNN-T4SE. The files in the 'mpl-data' folder were the system files, which could not be removed during prediction. Most importantly, there were two types of input documents: (T1) three files under the 'CNNT4SE' main folder with the file name extension of .acc, .ss and .fasta, which provided the data of *solvent accessibility sequences* (.acc), *secondary structure sequences* (.ss) and *protein sequence in FASTA format*. The files of .acc and .ss were generated by SCRATCH [41]. Since the calculation speed of the online version of SCRATCH was relatively slow, its local version was highlighted in the CNN-T4SE website. Please directly download the local version of SCRATCH from its official website (http://scratch.proteomics.ics.uci.edu/). (T2) the files under the 'pssm_files' folder with the file name extension of .pssm, which gave the evolutionary information in the form of a PSSM for each analyzed protein.

The general workflow of the new software tool CNN-T4SE is illustrated in Figure 3(B). In STEP 1, those studied proteins were provided in FASTA format. Since multiple sequences could be annotated simultaneously by CNN-T4SE, this tool could also be applied to scan the genome of a given bacteria. In STEP 2, the studied proteins were converted using three protein encoding strategies. The generated solvent accessibility sequence was stored in .acc file; the generated secondary structure sequence was put into the .ss file; the generated PSSM evolutionary information was in the .pssm file; and sequence information was provided in .fasta to conduct protein conversion using the Onehots integrated in the CNN-T4SE source code. In STEP 3, CNN-T4SE was applied to annotate all the studied protein sequences. The program was available for both *Windows* and *Linux* operating systems. Finally, a sequence was predicted as T4SE or non-T4SE by three different models based on its corresponding prediction probabilities. Only when a protein was annotated as T4SE by no less than two of the three models could it be predicted as T4SE. The CNN-T4SE manual was provided in Supplementary Method S3 and could be downloaded from the CNN-T4SE website (https://idrblab.org/cnnt4se/).

Conclusion

A CNN technique was applied in this study to annotate T4SEs through integrating multiple protein encoding strategies, and three encoding strategies (PSSM, PSSSA & Onehot) were identified to be powerful in T4SE annotation when integrated with CNN. As a result, a novel strategy that collectively considers the three identified best-performing models (CNN-PSSM, CNN-PSSSA and CNN-Onehot) was proposed, and a new software tool (CNN-T4SE, https://idrblab.org/cnnt4se/) was constructed to facilitate the annotation. All in all, CNN-T4SE was expected to annotate bacterial T4SEs with improved accuracy and reduced false discovery.

Key Points

- Recent studies mainly focus on annotating new T4SEs from the huge amount of sequencing data.
- Three encoding strategies were identified as powerful in T4SE annotation when integrated with CNN.
- A software tool for T4SE was finally constructed and could be downloaded for T4SE annotation.

Supplementary Data

Supplementary data are available online at https://academic.oup.com/bib.

Funding

National Key Research and Development Program of China (2018YFC0910500); National Natural Science Foundation of China (81872798, 81872935 and 81573502); the Innovation Projects on Industrial Generic Key Technologies of Chongqing (cstc2015zdcy-ztzx120003); Fundamental Research Funds for Central University (2018QNA7023, 10611CDJXZ238826, 2018CDQYSG0007 and CDJZR14468801); Key Projects of National Natural Science Foundation of China (81730108); Zhejiang Province Ministry of Science and Technology (2015C03055).

References

1. Bhuwan M, Arora N, Sharma A, *et al*. Interaction of *Mycobacterium tuberculosis* virulence factor RipA with chaperone MoxR1 is required for transport through the TAT secretion system. *MBio* 2016;**7**:e02259.
2. An Y, Wang J, Li C, *et al*. Comprehensive assessment and performance improvement of effector protein predictors for bacterial secretion systems III, IV and VI. *Brief Bioinform* 2018;**19**:148–61.
3. Costa TR, Felisberto-Rodrigues C, Meir A, *et al*. Secretion systems in Gram-negative bacteria: structural and mechanistic insights. *Nat Rev Microbiol* 2015;**13**:343–59.
4. Dorji D, Mooi F, Yantorno O, *et al*. *Bordetella pertussis* virulence factors in the continuing evolution of whooping cough vaccines for improved performance. *Med Microbiol Immunol* 2018;**207**:3–26.
5. Vacca I. Bacterial pathogenesis: activating *Helicobacter* effector delivery. *Nat Rev Microbiol* 2017;**15**:708–9.
6. Kuzmanovic N, Pulawska J, Hao L, *et al*. The ecology of *Agrobacterium vitis* and management of crown gall disease in vineyards. *Curr Top Microbiol Immunol* 2018;**418**:15–53.
7. Wang Y, Guo Y, Pu X, *et al*. 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**:1029–38.

Downloaded from https://academic.oup.com/bib/article/21/5/1825/5612159 by National Science and Technology Library -Root user on 05 October 2020

8. Wang Y, Wei X, Bao H, *et al.* Prediction of bacterial type IV secreted effectors by C-terminal features. *BMC Genomics* 2014;**15**:50.
9. Wang J, Yang B, An Y, *et al.* Systematic analysis and prediction of type IV secreted effector proteins by machine learning approaches. *Brief Bioinform* 2019;**20**:931–51.
10. Zeng C, Zou L. An account of in silico identification tools of secreted effector proteins in bacteria and future challenges. *Brief Bioinform* 2019;**20**:110–29.
11. Zou L, Nan C, Hu F. Accurate prediction of bacterial type IV secreted effectors using amino acid composition and PSSM profiles. *Bioinformatics* 2013;**29**:3135–42.
12. Chen X, Yin J, Qu J, *et al.* MDHGI: matrix decomposition and heterogeneous graph inference for miRNA-disease association prediction. *PLoS Comput Biol* 2018;**14**:e1006418.
13. Ramachandran SR, Yin C, Kud J, *et al.* Effectors from wheat rust fungi suppress multiple plant defense responses. *Phytopathology* 2017;**107**:75–83.
14. Wang C, Fu J, Wang M, *et al.* Bartonella quintana type IV secretion effector BepE-induced selective autophagy by conjugation with K63 polyubiquitin chain. *Cell Microbiol* 2019; **21**:e12984.
15. Cunha LD, Ribeiro JM, Fernandes TD, *et al.* Inhibition of inflammasome activation by *Coxiella burnetii* type IV secretion system effector IcaA. *Nat Commun* 2015;**6**:10205.
16. Grohmann E, Christie PJ, Waksman G, *et al.* Type IV secretion in Gram-negative and Gram-positive bacteria. *Mol Microbiol* 2018;**107**:455–71.
17. Lorrain C, Hecker A, Duplessis S. Effector-mining in the poplar rust fungus *Melampsora larici-populina* secretome. *Front Plant Sci* 2015;**6**:1051.
18. Yu CY, Li XX, Yang H, *et al.* Assessing the performances of protein function prediction algorithms from the perspectives of identification accuracy and false discovery rate. *Int J Mol Sci* 2018;**19**:E183.
19. Sankarasubramanian J, Vishnu US, Dinakaran V, *et al.* 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**:178–90.
20. de Guillen K, Ortiz-Vallejo D, Gracy J, *et al.* Structure analysis uncovers a highly diverse but structurally conserved effector family in phytopathogenic fungi. *PLoS Pathog* 2015;**11**:e1005228.
21. Xiong Y, Wang Q, Yang J, *et al.* PredT4SE-Stack: prediction of bacterial type IV secreted effectors from protein sequences using a stacked ensemble method. *Front Microbiol* 2018; **9**:2571.
22. Sonah H, Deshmukh RK, Belanger RR. Computational prediction of effector proteins in fungi: opportunities and challenges. *Front Plant Sci* 2016;**7**:126.
23. Teper D, Burstein D, Salomon D, *et al.* Identification of novel *Xanthomonas euvesicatoria* type III effector proteins by a machine-learning approach. *Mol Plant Pathol* 2016;**17**: 398–411.
24. Burstein D, Zusman T, Degtyar E, *et al.* Genome-scale identification of *Legionella pneumophila* effectors using a machine learning approach. *PLoS Pathog* 2009;**5**:e1000508.
25. Sperschneider J, Dodds PN, Gardiner DM, *et al.* Improved prediction of fungal effector proteins from secretomes with EffectorP 2.0. *Mol Plant Pathol* 2018;**19**:2094–110.
26. Zhang ZQ, Zhao Y, Liao XK, *et al.* Deep learning in omics: a survey and guideline. *Brief Funct Genomics* 2019;**18**:41–57.
27. Zou Q, Xing PW, Wei LY, *et al.* Gene2vec: gene subsequence embedding for prediction of mammalian N-6-methyladenosine sites from mRNA. *RNA* 2019;**25**: 205–18.
28. Li B, Tang J, Yang Q, *et al.* NOREVA: normalization and evaluation of MS-based metabolomics data. *Nucleic Acids Res* 2017;**45**:W162–70.
29. Chen X, Huang L. LRSSLMDA: laplacian regularized sparse subspace learning for miRNA-disease association prediction. *PLoS Comput Biol* 2017;**13**:e1005912.
30. Fa R, Cozzetto D, Wan C, *et al.* Predicting human protein function with multi-task deep neural networks. *PLoS One* 2018;**13**:e0198216.
31. Zeng NY, Zhang H, Song BY, *et al.* Facial expression recognition via learning deep sparse autoencoders. *Neurocomputing* 2018;**273**:643–9.
32. Chen X, Xie D, Zhao Q, *et al.* MicroRNAs and complex diseases: from experimental results to computational models. *Brief Bioinform* 2019;**20**:515–39.
33. Peng L, Peng MM, Liao B, *et al.* The advances and challenges of deep learning application in biological big data processing. *Curr Bioinform* 2018;**13**:352–9.
34. Min S, Lee B, Yoon S. Deep learning in bioinformatics. *Brief Bioinform* 2017;**18**:851–69.
35. Wei LY, Su R, Wang B, *et al.* Integration of deep feature representations and handcrafted features to improve the prediction of N-6-methyladenosine sites. *Neurocomputing* 2019; **324**:3–9.
36. Long H, Liao B, Xu X, *et al.* A hybrid deep learning model for predicting protein hydroxylation sites. *Int J Mol Sci* 2018; **19**:E2817.
37. Yu L, Sun X, Tian SW, *et al.* Drug and nondrug classification based on deep learning with various feature selection strategies. *Curr Bioinform* 2018;**13**:253–9.
38. UniProt Consortium T. UniProt: the universal protein knowledgebase. *Nucleic Acids Res* 2018;**46**:2699.
39. Costa J, d'Avo AF, da Costa MS, *et al.* Molecular evolution of key genes for type II secretion in *Legionella pneumophila*. *Environ Microbiol* 2012;**14**:2017–33.
40. Shames SR, Liu L, Havey JC, *et al.* Multiple *Legionella pneumophila* effector virulence phenotypes revealed through high-throughput analysis of targeted mutant libraries. *Proc Natl Acad Sci U S A* 2017;**114**:E10446–54.
41. Magnan CN, Baldi P. SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning and structural similarity. *Bioinformatics* 2014;**30**:2592–7.
42. Wang S, Li W, Liu S, *et al.* RaptorX-Property: a web server for protein structure property prediction. *Nucleic Acids Res* 2016;**44**:W430–5.
43. Seo S, Oh M, Park Y, *et al.* DeepFam: deep learning based alignment-free method for protein family modeling and prediction. *Bioinformatics* 2018;**34**:i254–62.
44. Jones DT, Cozzetto D. DISOPRED3: precise disordered region predictions with annotated protein-binding activity. *Bioinformatics* 2015;**31**:857–63.
45. Wang J, Yang B, Revote J, *et al.* POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles. *Bioinformatics* 2017;**33**: 2756–8.
46. Pai PP, Dattatreya RK, Mondal S. Ensemble architecture for prediction of enzyme-ligand binding residues using evolutionary information. *Mol Inform* 2017;**36**: 1700021.
47. Li K, Xu C, Huang J, *et al.* Prediction and identification of the effectors of heterotrimeric G proteins in rice (*Oryza sativa L.*). *Brief Bioinform* 2017;**18**:270–8.

Downloaded from https://academic.oup.com/bib/article/21/5/1825/5612159 by National Science and Technology Library -Root user on 05 October 2020

48. Wu M, Yang Y, Wang H, *et al*. A deep learning method to more accurately recall known lysine acetylation sites. *BMC Bioinformatics* 2019;**20**:49.
49. Pan X, Rijnbeek P, Yan J, *et al*. Prediction of RNA-protein sequence and structure binding preferences using deep convolutional and recurrent neural networks. *BMC Genomics* 2018;**19**:511.
50. Cheng CW, Su EC, Hwang JK, *et al*. Predicting RNA-binding sites of proteins using support vector machines and evolutionary information. *BMC Bioinformatics* 2008;**9**:S6.
51. Li YH, Xu JY, Tao L, *et al*. SVM-Prot 2016: a web-server for machine learning prediction of protein functional families from sequence irrespective of similarity. *PLoS One* 2016;**11**:e0155290.
52. Zhu F, Han L, Zheng C, *et al*. What are next generation innovative therapeutic targets? Clues from genetic, structural, physicochemical, and systems profiles of successful targets. *J Pharmacol Exp Ther* 2009;**330**:304–15.
53. Han LY, Cai CZ, Ji ZL, *et al*. Predicting functional family of novel enzymes irrespective of sequence similarity: a statistical learning approach. *Nucleic Acids Res* 2004;**32**: 6437–44.
54. Hamm CA, Wang CJ, Savic LJ, *et al*. Deep learning for liver tumor diagnosis part I: development of a convolutional neural network classifier for multi-phasic MRI. *Eur Radiol* 2019;**29**:3337–47.
55. Hsieh MH, Sun LM, Lin CL, *et al*. Development of a prediction model for colorectal cancer among patients with type 2 diabetes mellitus using a deep neural network. *J Clin Med* 2018; **7**:E277.
56. Vidotto M, De Momi E, Gazzara M, *et al*. FCNN-based axon segmentation for convection-enhanced delivery optimization. *Int J Comput Assist Radiol Surg* 2019;**14**:493–9.
57. Arcos-García Á, Álvarez-García JA, Soria-Morillo LM, *et al*. Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods. *Neural Netw* 2018;**99**:158–65.
58. Kim J, Calhoun VD, Shim E, *et al*. Deep neural network with weight sparsity control and pre-training extracts hierarchical features and enhances classification performance: evidence from whole-brain resting-state functional connectivity patterns of schizophrenia. *Neuroimage* 2016;**124**:127–46.
59. Mummadi SR, Al-Zubaidi A, Hahn PY. Overfitting and use of mismatched cohorts in deep learning models: preventable design limitations. *Am J Respir Crit Care Med* 2018;**198**:544–5.
60. Sato M, Horie K, Hara A, *et al*. Application of deep learning to the classification of images from colposcopy. *Oncol Lett* 2018;**15**:3518–23.
61. Chen X, Xie D, Wang L, *et al*. BNPMDA: bipartite network projection for miRNA-disease association prediction. *Bioinformatics* 2018;**34**:3178–86.
62. Chen X, Wang L, Qu J, *et al*. Predicting miRNA-disease association based on inductive matrix completion. *Bioinformatics* 2018;**34**:4256–65.
63. Chen W, Feng PM, Lin H, *et al*. iRSpot-PseDNC: identify recombination spots with pseudo dinucleotide composition. *Nucleic Acids Res* 2013;**41**:e68.
64. Lemm S, Blankertz B, Dickhaus T, *et al*. Introduction to machine learning for brain imaging. *Neuroimage* 2011;**56**: 387–99.

Downloaded from https://academic.oup.com/bib/article/21/5/1825/5612159 by National Science and Technology Library -Root user on 05 October 2020