6-2020

Learning transferable deep convolutional neural networks for the classification of bacterial virulence factors

Dandan ZHENG

Guansong PANG
Singapore Management University, gspang@smu.edu.sg

Bo LIU

Lihong CHEN

Jian YANG

Citation

ZHENG, Dandan; PANG, Guansong; LIU, Bo; CHEN, Lihong; and YANG, Jian. Learning transferable deep convolutional neural networks for the classification of bacterial virulence factors. (2020). *Bioinformatics*. 36, (12), 3693-3702.
Available at: https://ink.library.smu.edu.sg/sis_research/7038

Bioinformatics, 36(12), 2020, 3693–3702
doi: 10.1093/bioinformatics/btaa230
Advance Access Publication Date: 6 April 2020
Original Paper

Sequence analysis

Learning transferable deep convolutional neural networks for the classification of bacterial virulence factors

Dandan Zheng[1,†], Guansong Pang[2,†], Bo Liu[1], Lihong Chen[1] and Jian Yang [1,*]

[1]NHC Key Laboratory of Systems Biology of Pathogens, Institute of Pathogen Biology, Chinese Academy of Medical Sciences and Peking Union Medical College, Beijing 100176, China and [2]Australian Institute for Machine Learning, The University of Adelaide, Adelaide, SA 5005, Australia

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.
Associate Editor: Jinbo Xu

Received on December 2, 2019; revised on March 25, 2020; editorial decision on March 27, 2020; accepted on April 1, 2020

Abstract

Motivation: Identification of virulence factors (VFs) is critical to the elucidation of bacterial pathogenesis and prevention of related infectious diseases. Current computational methods for VF prediction focus on binary classification or involve only several class(es) of VFs with sufficient samples. However, thousands of VF classes are present in real-world scenarios, and many of them only have a very limited number of samples available.

Results: We first construct a large VF dataset, covering 3446 VF classes with 160 495 sequences, and then propose deep convolutional neural network models for VF classification. We show that (i) for common VF classes with sufficient samples, our models can achieve state-of-the-art performance with an overall accuracy of 0.9831 and an F1-score of 0.9803; (ii) for uncommon VF classes with limited samples, our models can learn transferable features from auxiliary data and achieve good performance with accuracy ranging from 0.9277 to 0.9512 and F1-score ranging from 0.9168 to 0.9446 when combined with different predefined features, outperforming traditional classifiers by 1–13% in accuracy and by 1–16% in F1-score.

Availability and implementation: All of our datasets are made publicly available at http://www.mgc.ac.cn/VFNet/, and the source code of our models is publicly available at https://github.com/zhengdd0422/VFNet.

Contact: yangj@ipbcams.ac.cn

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Despite advances in diagnosis, treatment and prevention, bacterial infectious diseases still significantly threaten public health worldwide (van Oosten *et al.*, 2015). Virulence factors (VFs) are elements (i.e. gene products) that enable microorganisms to establish colonies, achieve immune escape, survive in host environments and cause tissue damage or systemic inflammation (Cui *et al.*, 2013). The molecular mechanisms of bacterial VFs, which are involved in various functional classes/categories (classes and categories are used interchangeably hereafter), form the cellular and molecular basis of pathogenesis. Therefore, the characterization of VFs is not only essential to the comprehensive understanding of bacterial pathogenesis but also valuable to the effective prevention and therapy of bacterial infectious diseases.

The rapid development and wide application of DNA sequencing technologies in recent years have led to the availability of a large amount of bacterial genomic data, which in turn enables sequence analysis as the primary approach for potential VF identification. On the one hand, an increasing number of VFs have been identified by sequence similarity search methods (e.g. Basic Local Alignment Search Tool, BLAST) against known VFs (Vinatzer *et al.*, 2005). In addition, hidden Markov models (HMMs) are used to assist sequence similarity searches to identify more distinct homologs of VFs in bacterial genomes (Manuel Martinez-Garcia *et al.*, 2015). More recently, via the combination of BLAST- and HMM-based methods, VFanalyzer (Liu *et al.*, 2019) conducts iterative and thorough sequence similarity searches in the virulence factor database (VFDB) to identify known/potential VFs from bacterial genomes. Nevertheless, homology-based methods are generally computationally expensive, especially when handling large-scale sequences with long lengths.

On the other hand, traditional machine learning approaches, e.g. the support vector machine (SVM) and random forest (RF)

algorithms, have recently been introduced to predict new VFs based on predefined sequence features, such as frequency components, physicochemical properties, protein functional domains and position-specific scoring matrices (PSSMs). For instance, VirulentPred (Garg and Gupta, 2008) extracts features including the amino acid composition (AAC), dipeptide composition (DPC) and PSSMs and then employs an SVM and BLAST to predict the bacterial VFs. MP3 (Gupta *et al.*, 2014) further uses an integrated SVM–HMM approach to predict bacterial VFs in both genomic and metagenomic data. Although the training of classification models can be computationally costly, they are generally highly efficient at the testing stage since they use trained models to directly make predictions.

However, the aforementioned machine learning-based methods are focused on binary predictions of bacterial VFs (e.g. whether virulence related or not). Since there are usually tens to hundreds of VFs involved in the pathogenesis of any individual well-established bacterial pathogen, further classification of new VFs into fine-grained functional categories is critical for in-depth follow-up biological validations of the VFs. Indeed, some studies have attempted to address this issue by performing prediction on only specific class(es) of VFs, i.e. a subset of known bacterial VFs. For example, SPAAN (Sachdeva *et al.*, 2005) applies an SVM to predict adhesins and adhesin-like proteins. Su *et al.* (2014) extract functional domain features and apply an SVM to predict endotoxins and exotoxins. Many other machine learning-based approaches focus on the prediction of secreted proteins or effectors of known bacterial secretion systems, including SecretP (Yu *et al.*, 2010), SSPred (Pundhir and Kumar, 2011), T4SEpre (Wang *et al.*, 2014), T4EffPred (Zou *et al.*, 2013) and Bastion6 (Wang *et al.*, 2018). Nevertheless, existing methods typically involve only one or several VF classes rather than all known bacterial VFs, which consist of thousands of functional categories. Furthermore, traditional machine learning methods are heavily dependent on predefined sequence features, which are derived from training samples. The effectiveness of these approaches is normally built upon the assumption that there are sufficient high-quality training samples for each class. Unfortunately, this is not always true in real-world bacterial VF scenarios. Pathogenic bacteria have acquired various VFs, many of which are commonly shared by different strains, indicating universal requirements for causing infection. However, there are some narrowly distributed VFs that are encoded by only a small subset of known genomes, which determine species- and/or strain-specific characteristics. Therefore, the limited sample sizes of these uncommon VFs obstruct the effective application of current approaches to all categories of bacterial VFs. In addition, the predefined features are extracted independently from downstream classification methods. By doing so, the discriminative information embedded in the extracted features is generic rather than specifically defined or learned for specific classification methods. As a result, the underlying discriminative information in the data cannot be fully exploited.

In recent years, due to the capability to automatically learn semantic-rich abstract representations from raw data, deep learning has transformed many different machine learning tasks and achieved state-of-the-art performance in a broad range of real-world applications, including many biological sequence-based applications (Min *et al.*, 2017). For instance, convolutional neural network (CNN) models are widely used for various biological feature predictions, including enhancer–promoter interactions (Zhuang *et al.*, 2019), promoters in the human genome (Umarov *et al.*, 2019), functional categories of proteins (Seo *et al.*, 2018), genomic sequence polymorphisms (Poplin *et al.*, 2018) and taxa from metagenomic sequences (Fiannaca *et al.*, 2018). Furthermore, some studies use recurrent neural network (RNN) or the combination of CNN and RNN for the prediction of antimicrobial peptides (Hamid and Friedberg, 2019; Veltri *et al.*, 2018). Motivated by these successful applications, especially DeepFam (Seo *et al.*, 2018) and ProtCNN (Bileschi *et al.*, 2019), which use CNN models to classify over 2000 protein families well, we propose to explore the potential of deep learning models for the classification of large-scale datasets of bacterial VFs.

In this work, we aim to leverage deep learning techniques to address the aforementioned issues of traditional machine learning-based methods that hinder comprehensive prediction and classification. First, we compile a comprehensive VF dataset of 32 genera of human bacterial pathogens, which contains 3446 VF classes with 160 495 sequence samples. This dataset has the largest number of VF classes and samples ever built so far, providing an important benchmark for developing and testing novel methods for the prediction of bacterial VFs. Then, we present an end-to-end deep CNN model, termed VFNet, which can automatically learn expressive feature representations for the classification of bacterial VFs. In addition, we empirically justify three critical characteristics of our VFNet, including its capability of harnessing expanded data from easily accessible databases to significantly enhance the accuracy, effectively combining automatically learned and predefined features to achieve state-of-the-art performance, and learning transferable features for accurate prediction of uncommon VFs that have very limited samples. Impressively, for common VFs with sufficient training samples available (i.e. >10), VFNet achieves the desired performance with an overall accuracy of 0.9831 and an F1-score of 0.9803; in the case of uncommon VFs in divergent datasets, our model also achieves remarkable performance with an accuracy of 0.9512, a precision of 0.9542, a recall of 0.9499 and an F1-score of 0.9446, which outperforms traditional classifiers by about 1–13% in accuracy and 1–16% in terms of the F1-score when combined with different predefined features. In addition, we show that the VFNet model can well capture conserved regions or motifs that are highly similar to the known patterns of protein domains, which provides important insights into the underlying driving factors of the effectiveness of our model.

2 Materials and methods

2.1 Dataset construction

VFDB is a comprehensive repository of bacterial VFs. It has been dedicated to providing up-to-date knowledge of VFs from various medically significant bacterial pathogens for over a decade (Chen *et al.*, 2016). In particular, the intragenus VF comparison dataset of the VFDB covers known and potential VFs of 474 genomes from 32 genera of human bacterial pathogens. The dataset is well classified and contains many more samples than any VF datasets previously used for modeling (Garg and Gupta, 2008; Gupta *et al.*, 2014), so the VFDB data are collected as the original dataset in this study (accessed on April 2019), which consists of 24 739 protein sequences from 4100 VF classes (Supplementary Table S1). However, the sequence samples are unevenly distributed across different VF classes with 3548 classes containing no more than 10 sequences. It is known that the VFDB usually includes only representative genomes for brevity, whereas there are thousands of fully sequenced bacterial genomes available in the public domain. We, therefore, download a total of 7183 complete bacterial genomes from the NCBI server (accessed on May 2019), which are all from the 32 genera included in the original dataset (Supplementary Table S2). VFanalyzer is then employed to conduct an exhaustive similarity search for known and potential VFs within each genome. The homologous VF sequences identified in the new genomes are assigned the same class as their counterparts in the VFDB. Thus, the original dataset is substantially expanded to 164 119 sequences after excluding identical sequences. Supplementary Figure S1 shows the sample size of each genus before and after data expansion.

Based on the expanded dataset, we perform the following data screening procedures before further analysis. We first remove sequences related to more than one VF class, which yields 163 417 unique sequences from 4058 unambiguous VF classes. Then, we manually merge the corresponding categories related to components of type III/IV secretion systems of various pathogens into a set of pan-genera VF superclasses (Supplementary Table S1) since these bacterial secretion systems are generally conserved in terms of both biological function and genetic organization. Therefore, the resulting data contain 3778 VF (super) classes. As shown in Supplementary Figure S2, the majority of the VF sequences (98.57%) have a length no longer than 2500 amino acids. To limit

the required computing memory resources in the subsequent modeling analysis, we further remove 2330 sequences longer than 2500 amino acids. Besides, 332 VF categories with no more than three sequences are excluded from further analysis since the extremely small sample size is insufficient to produce statistically significant results. Thus, we produce an expanded nonredundant dataset of 3446 VF classes with 160 495 sequences.

Due to the intrinsic genetic diversity of pathogenic bacteria, there are some species-/strain-specific VFs for many pathogens, which are encoded by only a small subset of known genomes. Therefore, although we already excluded extremely undersampled classes from the expanded dataset, severe imbalance in the sample size of VF classes is inevitable. We divide the expanded dataset into two subsets, namely, VFG-2706 and VFG-740, which consist of 2706 and 740 classes, respectively. VFG-2706 covers all the common VFs, which are empirically defined as classes with more than 10 labeled samples, whereas VFG-740 includes uncommon VF classes (i.e. ≤ 10 samples). More details on the VFG-2706 and VFG-740 datasets are provided in Supplementary Table S3.

To examine the potential effect of high sequence homology on the overall performance of our methods, we apply CD-HIT (Fu *et al.*, 2012) to each class of the original dataset and the VFG-2706 dataset to remove highly homologous sequences (defined as having >90% sequence identity), which respectively result in two divergent datasets: VFG-564 and VFG-2706-1066. Lower CD-HIT thresholds might help further reduce the potential bias introduced by sequence homology and produce more reliable models (Wang *et al.*, 2018), but 86.54% of the classes in the original dataset contain no more than 10 sequences, which limits us from using those thresholds. The resulting VFG-564 is derived from the original dataset and covers only 3382 sequences from 564 uncommon VF classes (i.e. those that contain 4–10 labeled samples), whereas VFG-2706-1066 is produced from the VFG-2706 dataset and covers 34 078 sequences from 1066 common VF classes (i.e. those that contain >10 labeled samples). We do not use CD-HIT on the VFG-740 dataset because of the limited sample size in each class. More details on the different VF datasets used in this study are given in Supplementary Figure S3.

Transfer learning (TL) is widely used to improve the performance of a learner on one domain of insufficient data by transferring information from a related domain that has abundant data (Weiss *et al.*, 2016). To improve the classification performance on the uncommon VFs in the VFG-564 dataset, we apply popular TL techniques to a large dataset from the database of Clusters of Orthologous Groups of proteins (COGs) (Tatusov, 2000). Specifically, we exclude sequences longer than 1000 amino acids in COGs and keep the categories with 500–1000 valid samples, which results in a COG dataset, namely, COG-755, with 755 classes and 514 310 labeled sequences.

2.2 VFNet: our proposed deep learning model

2.2.1 The proposed CNN model

We propose to use an end-to-end sequence-based CNN model to classify bacterial VFs. A CNN model is used rather than an RNN-based model, such as a gated recurrent unit (GRU) or long short-term memory (LSTM) model, mainly because CNN models are able to capture sequential spatial correlations while being significantly more efficient than RNN models (LeCun *et al.*, 2015; Zhang *et al.*, 2015).

The model we introduce is called VFNet, and it is a CNN model with seven layers. As shown in Figure 1, VFNet consists of one one-hot encoding input layer, two 1D convolution (Conv1D) layers, two 1D global max pooling (Maxpooling1D) layers and two fully connected (fc) layers. The model is summarized as follows:

$$y_i = g(f(x_i; \Theta_f); \Theta_g), \tag{1}$$

where x_i is a raw input sequence; $f : \{0,1\}^{2500\times 20} \mapsto \mathbb{R}^{512}$ is a feature representation learner that maps the one-hot encoding matrix of a sequence onto a 512D space, which is composed of all the convolution layers, max pooling layers and the first fc layer; Θ_f is the set of parameters to be learned; $g : \mathbb{R}^{512} \mapsto \mathbb{R}^{2706}$ represents the softmax layer, which is a classifier that takes the learned representations as inputs and yields the prediction probability of each sequence belonging to each class, in which Θ_g is the associated parameter set.

In particular, each input protein sequence is converted into a one-hot encoding matrix as follows. Each standard amino acid is represented by a vector of length 20, where an entry is equal to one if a specific amino acid is present and is equal to zero otherwise. For instance, A is encoded as [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0] because A occurs in the first entry only, and similarly, C is encoded as [0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0], etc. In addition, sequences are preprocessed to have the same length before being fed to VFNet. Specifically, based on the length distribution of all the sequences (Supplementary Fig. S2), we found that 2500 is a desired fixed length to retain most sequence information while not being biased by a small number of exceptionally long sequences. Thus,

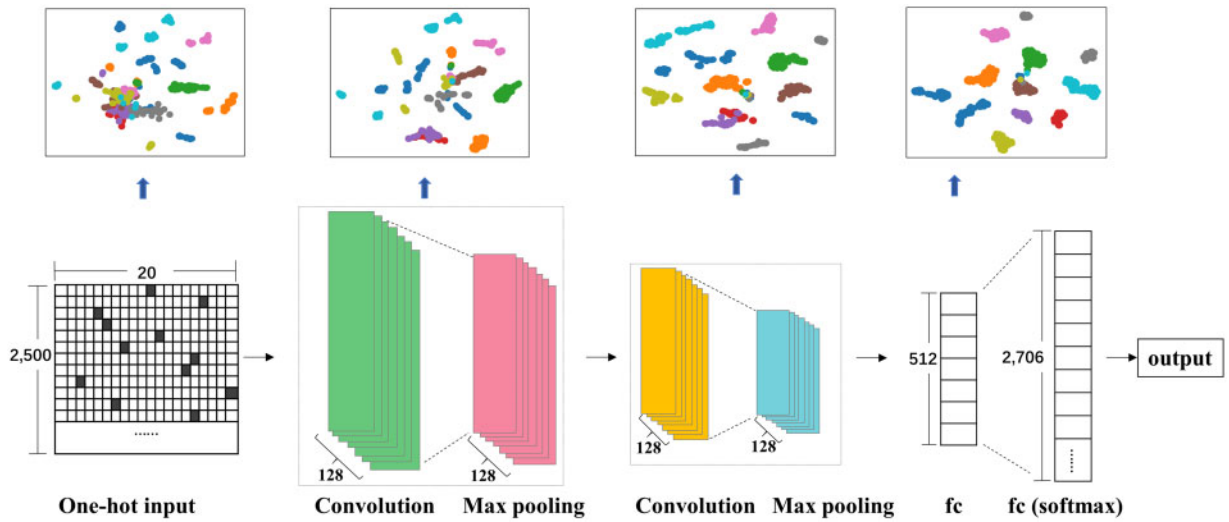

Fig. 1. Our proposed VFNet model and a visualization of its four key layers. VFNet is a CNN model with seven layers, including one input encoding layer, two convolution layers, two max pooling layers, one fc layer and one softmax layer. Specifically, the encoding layer converts each peptide sequence into a matrix of size 2500×20. The convolution layers are Conv1D using 128 filters with 7 kernel sizes. The pooling layer is a max pooling layer of size 5. The fc layer consists of 512 units. The softmax layer contains 2706 units to classify the categories. To clearly demonstrate the feature representations that VFNet learns from the raw data, we visualize 10 VF classes randomly chosen from the 2706 categories in a 2D projected space using t-SNE. The visualization is performed in the four key layers, including the one-hot encoding input layer, two convolution layers and the fc layer

sequences longer than 2500 are removed, while the widely used strategy, zero padding, is applied to sequences that are shorter than the fixed length.

Accordingly, each sequence becomes a matrix of 2500×20. Let $\mathbf{X}_i$ be the one-hot encoding matrix of the sequence x_i; then, the feature representation learner can be represented by a compound function of two replicate convolution functions f^{con} and one fc mapping function f^{fc}:

$$f = f^{\text{fc}} \circ f^{\text{con}} \circ f^{\text{con}}(\mathbf{X}_i; \Theta_f), \tag{2}$$

where '$\circ$' is a compound operation and f^{con} consists of nonlinear convolution and pooling operations as follows:

$$f^{\text{con}} = \text{Maxpooling1D}(\text{ReLU}(\text{Conv1D}(\mathbf{X}_i))). \tag{3}$$

The Conv1D layer is composed of 128 filters with 7 kernel sizes. It computes the dot product between its weights and a small region in the input to extract shallow features and yields an output feature map of size 2494×128. The rectified linear unit (ReLU) (Agarap, 2018) function a is used as a nonlinear activation function in the Conv1D layer to transform the data from one volume to another. ReLU can be represented by $a(u) = \max(0, u)$, in which we retain the input scalar u if it is not <0 and omit the input otherwise. A Maxpooling1D layer of size 5 is used after Conv1D to reduce the output dimension to 498×128. The second f^{con}, employed in the second convolution and max pooling layers, is exactly the same as the first f^{con} to extract more features and produces an output of size 98×128. The 98×128 feature map is then flattened and fed to the fc layer f^{fc} with 512 units to learn more expressive high-level abstract features, yielding a 512D vector for each input raw sequence. These vectors are further fed into the final softmax layer, which is equivalent to a classifier g using 2706 units with a softmax function to learn the probability of the input sequence belonging to each VF category. Finally, each input sequence is classified into the corresponding class with the highest probability value. Note that dropout with a rate of 0.5 is used after each pooling layer as well as between the two fc layers to avoid overfitting by randomly masking positions of the output (Srivastava *et al.*, 2014). The specific structural details of VFNet are shown in Supplementary Figure S4.

The cross-entropy loss and 'Adam' optimizer (Kingma and Ba, 2014) are used to learn the parameters Θ_f and Θ_g of VFNet. Since our datasets contain hundreds/thousands of categories, it is difficult to determine the length dependence between sequences. Therefore, a number of convolution options were probed on VFG-2706, including filter size of 64, 128, 256 and 512, kernel sizes of 5, 7, 11 and 13, max pooling sizes of 3, 5 and 7 and fc layer units of 64, 128, 256 and 512. By using the validation set, we found that setting the filter size to 128, the kernel size to 7, the max pooling size to 5 and the fc layer units to 512 achieved the best performance. These hyperparameters are used by default in our experiments. The Keras (http://www.keras.io) library with a TensorFlow (http://tensorflow.org/) backend is used to implement all deep learning models, which are executed with a GeForce RTX 2080 Ti graphics card.

2.2.2 Visualization of the learned features

To explore what our model VFNet has learned, we randomly choose 10 categories (Supplementary Table S1) from VFG-2706, each with more than 100 samples, and visualize this data subset in a 2D projected space using t-distributed stochastic neighbor embedding (t-SNE) (van der Maaten and Hinton, 2008). To gain insights into the feature learning process, a visualization is performed in four key layers: the one-hot encoding input layer, the two convolutional layers and the fc layer. Particularly, for each specific layer, we first obtain the representations of the sequences of these 10 classes, and we then use t-SNE to further reduce the data dimensions to two. To obtain a more meaningful visualization, principal component analysis (Wold *et al.*, 1987) is first applied to project the data onto 20D space before using t-SNE. The visualization results are illustrated in the upper part of Figure 1. It is clear that the sequence samples of these 10 classes are highly entangled in the one-hot encoding input layer, but they become more separable with deeper representations. This finding demonstrates that VFNet can effectively learn disentangled representations from original highly entangled input data.

Note that VFNet learns disentangled representations for all the classes. Here, we focus on 10 randomly selected categories mainly because it is difficult to clearly visualize all 2706 classes in a single figure.

2.3 Competing methods

2.3.1 Two relevant deep learning methods

To examine the effectiveness of the design of our deep learning model, we compare VFNet to the methods 'Embed + CNN + LSTM' (Veltri *et al.*, 2018) and 'Word2vec + biGRU' (Hamid and Friedberg, 2019) proposed previously. They are two representative models using different sequence encoding methods and deep architectures from our VFNet, which uses one-hot encoding and CNN.

Specifically, instead of one-hot encoding, 'Embed + CNN + LSTM' utilizes the distributed representation idea in the word embedding (Levy and Goldberg, 2014) to encode sequence data. In addition, it uses a convolutional–recurrent neural network architecture to make predictions, while we use a convolutional network only. 'Embed + CNN + LSTM' first uses a 128D vector in the embedding layer to learn sequence embeddings, followed by a Conv1D layer with 64 filters, a kernel size of 16 and a Maxpooling1D layer of size 5. It is then connected to a LSTM layer with 100 units before feeding to the output softmax layer to make predictions.

'Word2vec + biGRU' uses the UniProt/TrEMBL dataset (Boeckmann, 2003) as the document corpus and pretrains a Word2vec model based on the skip-gram method (Goldberg and Levy, 2014) to represent the sequences. It builds a deep learning model consisting of one embedding layer with 200 units, two-layer bidirectional GRUs (Chung *et al.*, 2014) and a softmax layer to make predictions.

2.3.2 Four traditional classifiers using predefined features

To verify the advantages of deep learning methods over traditional machine learning methods using predefined features, we apply four well-established classification algorithms widely used in predicting bacterial VFs (Zeng and Zou, 2019) as the baselines. These algorithms include logistic regression (LR), decision tree (DT), RF and SVM models. For the RF model, we searched the number of base models from 10 up to 100 and found that 100 base models worked best (data not shown), which is used by default in our experiments. For the SVM model, we apply the linear kernel function with the penalty hyperparameter C set to 1. We also tried an SVM model with nonlinear kernel functions, but it works less effectively than the linear kernel on our datasets. We extracted five types of widely used predefined features, including AAC, DPC, composition, transition and distribution (CTD) and pseudo amino acid composition (PseAAC1 and PseAAC2). These five types of features cover both sequence-based features and physicochemical features and can be further concatenated to produce a full feature set (designated as 'ALL' hereafter) to cover information from all the aforementioned statistics and properties of amino acids (see Supplementary Methods).

2.4 A variant of VFNet: combining automatically learned and predefined features

VFNet is designed to achieve the end-to-end classification of bacterial VFs, but it is also flexible enough to incorporate any readily available predefined features into the end-to-end classification process. To achieve this flexibility, we introduce a variant of VFNet, called VFNet-H, which employs a hybrid of automatically learned features and predefined features to make predictions. The key motivation is to incorporate some statistical information or property of the amino acids into VFNet, which may otherwise not be captured by VFNet. This method may help further enhance the capability of VFNet. The procedure of VFNet-H is shown in Supplementary

Figure S5. Specifically, as shown at the top of Supplementary Figure S5, VFNet-H uses exactly the same architecture and hyperparameters as VFNet for learning features from the raw data, while adding another parallel branch to take predefined features and fuse the automatically learned and predefined features through an fc layer. Unlike the deep architecture in the feature learning component, the new branch of the network consists of an input layer only since the inputs are extracted features rather than raw data. The number of units of this input layer depends on the type of predefined features used: AAC, 20; DPC, 400; CTD, 147; PseAAC1, 30; PseAAC2, 40; ALL, 637. Finally, we concatenate the features from the two branches and feed them to the fc layer before using the softmax layer to perform classifications. VFNet-H is trained using the same optimizer and hyperparameters as VFNet.

2.5 Performance evaluation

Our datasets are divided into three exclusive datasets: training, validation and testing using the ratio of 6:2:2 with class-wise stratified sampling by default. The training dataset is used to train the proposed models, while the validation dataset is used to monitor the generalization capabilities and turn the hyperparameters of the models. The test dataset is used as an independent dataset to evaluate the final performance. All the methods are performed by 5 folds with different dataset splitting random seeds. The reported performance is averaged over the results of the five implementations. The accuracy, precision, recall and F1-score (see Supplementary Methods) are used to evaluate the performance. We report both the macro and micro average values of the above metrics. Note that since our task is multiclass classification, the micro averages of precision and recall are always the same. As a result, the micro average of F1-score also equals to that of precision and recall, as well as the overall accuracy. Thus, we report only the overall accuracy for the micro average metrics while report precision, recall and F1-score for the macro average metrics.

3 Results

3.1 Comparison to other deep learning models

3.1.1 Classification performance

Table 1 and Supplementary Figure S6 show the classification performance of our VFNet model and two previously proposed deep learning models on the VFG-2706 dataset. VFNet achieves the best performance in terms of all the classification performance metrics, including the training accuracy, test accuracy, precision, recall and F1-score, as well as the training and testing runtime. Particularly, VFNet achieves a nearly perfect model fit to the training data by having a training accuracy of more than 0.99. More importantly, it generalizes well to the independent test data by achieving the desired performance of a test accuracy of 0.9663, a precision of 0.9693, a recall of 0.9551 and an F1-score of 0.9577. Compared to 'Embed + CNN + LSTM' and 'Word2vec + biGRU' that use more complex sequence encoding methods and network architectures, VFNet achieves respective 3% and 1% improvement in both F1-score and accuracy. This performance is remarkable given that the VFG-2706 dataset contains as many as 2706 classes. Since VFNet performs better in both the macro average F1-score and the overall accuracy than the two competing deep methods, it indicates VFNet is more effective in classifying both large and small classes in VFG-2706. Impressively, VFNet is the most efficient deep model here, which runs more than 210 times and 16 times faster than the 'Word2vec + biGRU' and 'Embed + CNN + LSTM' methods, respectively, at the training stage; VFNet is also highly efficient at the testing stage, taking only 21 s in total to classify 31 086 sequence samples.

3.1.2 Visualization of learned feature representations

We further explore the underlying reasons to explain the performance difference between the three deep learning models in Table 1 by looking into the feature representations learned by these models. Specifically, we apply the 3 models to the sequences of 10 randomly selected VF classes (Supplementary Table S1) and visualize the final feature representations extracted from the penultimate layer of each model. The visualization using t-SNE is presented in Supplementary Figure S7. It is clear that VFNet disentangles the sequences of different classes much better than those of 'Embed + CNN + LSTM' and 'Word2vec + biGRU'. Particularly, VFNet learned separable feature representations for all the classes except a few outlier sequences, whereas 'Embed + CNN + LSTM' and 'Word2vec + biGRU' effectively disentangle most of the classes but mix up the two classes 'C04' and 'C05'. Indeed, classes C04 and C05 are integral membrane proteins FagA and iron enterobactin transporter FagB, respectively (Supplementary Table S1). They are two essential components of the same iron uptake system of *Corynebacterium*. In particular, FagA and FagB share ~30% overall sequence similarity since both of them are extremely hydrophobic proteins with at least seven predicted transmembrane helices (Billington *et al.*, 2002). This demonstrates that VFNet learns better abstraction representations of both heterogeneous and homologous VFs than the other two deep learning methods, enabling VFNet to achieve the best VF classification performance.

3.2 Comparison of VFNet performance before and after data expansion

To construct VFNet, we deliberately expand the original VF dataset of VFDB to include additional homologous VFs within more bacterial genomes available from NCBI. In an attempt to investigate whether the data expansion procedure effectively improves the performance of VFNet as expected, we select a subset of VFG-2706, namely, VFG-552, which contains only 552 of 2706 classes that have more than 10 sequence samples before data expansion. Then, the sample size for each class of the VFG-552 dataset is sufficient to avoid potential impacts due to data insufficiency. The VFG-552 dataset originally contains 10 530 sequences (VFG-552a), whereas the total number of sequences increases to 68 673 after data expansion (VFG-552b), obtaining an approximately 7-fold expansion. Figure 2 and Supplementary Table S4 present the performance of VFNet on the VFG-552 test dataset before and after data expansion. Undoubtedly, benefiting from the data expansion, the performance

Table 1. Performance (means ± SDs) of VFNet and two other deep learning methods on the VFG-2706 dataset

Method	Training stage		Testing stage				
	Accuracy	Runtime	Accuracy (micro)	Precision (macro)	Recall (macro)	F1-score (macro)	Runtime
VFNet	0.9915 (±0.0016)	**46 m 44 s**	**0.9663** (±0.0011)	**0.9693** (±0.0018)	**0.9551** (±0.0019)	**0.9577** (±0.0018)	**21 s**
Embed + CNN + LSTM	0.9893 (±0.0022)	12 h 39 m 45 s	0.9405 (±0.0046)	0.9491 (±0.0031)	0.9276 (±0.0050)	0.9314 (±0.0043)	1 m 3 s
Word2vec + biGRU	0.9672	6 d 19 h 23 m 31 s	0.9570	0.9614	0.9479	0.9492	10 m 3s

Note: The runtime in the training stage denotes the training time of 100 epochs. Word2vec + biGRU is too computationally expensive, so we report its performance in a single run only. The best performance on a metric is highlighted in bold.

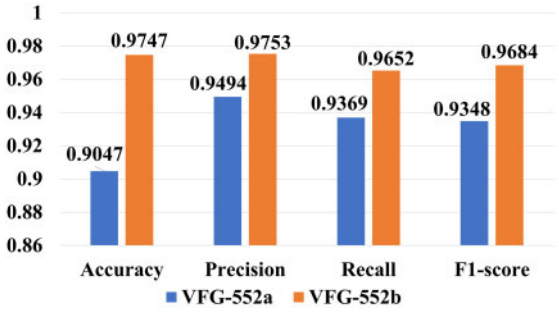

Fig. 2. Performance of VFNet on the VFG-552 datasets before (VFG-552a) and after (VFG-552b) data expansion from NCBI. VFG-552a and VFG-552b contain 10 530 and 68 673 sequences, respectively

of VFNet is significantly improved by 8%, 3%, 3% and 4% for the test accuracy, precision, recall and F1-score, respectively.

3.3 Comparison to traditional classifiers

The performance of VFNet, VFNet-H and the best-competing traditional machine learning method per predefined feature on VFG-2706 is shown in Table 2 (see Supplementary Table S5 and Fig. S8 for the full results). It is clear that as an end-to-end sequence classification method, VFNet automatically learns highly expressive feature representations from the raw VF data, enabling the model to achieve a performance that is very comparable to the traditional classifiers that heavily rely on predefined features. This finding verifies the applicability of deep learning models to the classification of bacterial VFs. Furthermore, VFNet-H is able to leverage both its automatically learned features and predefined features to achieve the best performance in terms of the test accuracy (0.9831), recall (0.9783) and F1-score (0.9803). It is interesting that some traditional classifiers (e.g. RF) using predefined features can perform very well in most cases and obtain better performance than VFNet, while other classifiers (e.g. LR and DT) often perform poorly.

Since VFG-2706 is a highly imbalanced dataset, we try to investigate whether balanced class weighting methods can be used to further enhance the prediction performance. A popular class weighting method that assigns class weights inversely proportional to the class size (King and Zeng, 2001) is incorporated into VFNet, VFNet-H and the best-competing classifier per predefined feature on VFG-2706. The results presented in Supplementary Table S6 show that the introduction of balanced class weights to the training model does not consistently improve the results of the methods. To better understand the results, we calculate the correlation between the F1-score and the class weights. As shown in Supplementary Figure S9, there is no clear correlation. The added class weights only contribute to some extra improvements in a small subset of classes while deteriorate the performance in the others. Thus, we do not use class weights in this work.

As the VFG-2706 dataset covers expanded data from NCBI based on sequence similarity, it will contain highly similar sequences within some VF classes, which may lead to an overestimated view of the prediction performance. To address this issue, we examine the correlation between the sequence identity and the F1-score across all the classes. Our results show that there is a low linear Pearson correlation ($\rho = 0.2280$) between the F1-score of VFNet and the sequence identity in VFG-2706 (see Supplementary Fig. S10). In other words, the superior performance of VFNet is not related to the potential sequence identity/homology issue, i.e. VFNet obtains similarly good/bad performance on classes of either high or low sequence similarities. In addition, we further apply VFNet and four traditional classifiers to the divergent dataset VFG-2706-1066, which has excluded highly homologous sequences by CD-HIT. The results of VFG-2706-1066, as shown in Supplementary Table S7, present the same performance trend as that in the VFG-2706 dataset (Supplementary Table S5), which confirms that the VFNet model is consistently the best classifier. Note that as expected, the removal of highly homologous sequences leads to some loss in the performance accuracy for all classifiers, but it is very impressive that the performance of VFNet and VFNet-H has significantly smaller drops than the competing methods, e.g. 4–6% drop in VFNet/VFNet-H versus 10–14% in RF when comparing the performance on the VFG-2706 and VFG-2706-1066 datasets. This demonstrates that VFNet learn well generalized representations of different VF classes rather than simply memorizing the training data.

3.4 Experiments on genome-independent data

VF samples expanded from the same data source may lead to some levels of sample dependency in our data, which can violate the i.i.d. (independent and identically distributed) assumption (Cover and Thomas, 2006) commonly made by machine learning-based prediction, rendering our cross-validation less effective. To address this potential issue, we train and test the models on genome-independent data. Particularly, the VF classes in our data are originally created based on bacterial genera, with a number of genomes within each VF class. To create more reliable datasets, instead of randomly splitting the data within each VF class, we use the genome source information of each sequence to divide the VFG-2706 dataset into genome-independent training and testing datasets. This guarantees that the sequence samples from each genome appear in either training or testing data; the training and test datasets contain no overlapping genomes. Since there are two classes in VFG-2706 that have only one genome sequences, we exclude them to obtain a dataset called VFG-2706-i.i.d., covering 2704 classes and 155 368 sequences. We take all sequences from ~20% of genomes (at least one) within each class of VFG-2706-i.i.d. as the independent test dataset. The remaining sequences from the other genomes are used to perform 5-fold cross-validation. After the cross-validation, all models are also evaluated on independent testing dataset. As shown in Supplementary Table S8, VFNet-H (DPC) achieves the best test accuracy, precision and F1-score, while VFNet-H (ALL) achieves the best recall performance at the 5-fold cross-validation stage. Similarly, on the independent testing dataset, VFNet-H (DPC) outperforms other methods with the best F1-score of 0.9788 and VFNet-H (ALL) achieves the best recall of 0.9765. It is worth noting that the results in Supplementary Table S8 are generally consistent with those based on the random sampling scheme in Supplementary Table S5. This implies that our primary sampling scheme does not clearly overestimate the generalization ability of these classification methods and is applicable to follow-up analysis.

3.5 Classification of uncommon VFs

Traditional classifiers rely on predefined features that are extracted from training datasets, so it is difficult to transfer the predefined features through the classification models. To examine whether VFNet and VFNet-H can transfer the feature representations learned from relevant abundant datasets to uncommon VFs (i.e. the VF classes with no more than 10 labeled samples), we apply a popular deep TL method to the nonredundant dataset VFG-740 and the divergent dataset VFG-564, respectively.

For VFG-740, we first pretrain VFNet and VFNet-H using its correlated large-scale dataset VFG-2706, and then initialize the weight parameters of VFNet and VFNet-H on VFG-740 using the corresponding pretrained models, and finally fine-turn the models using the VFG-740 dataset. This process transfers the knowledge of VFNet/VFNet-H learned from VFG-2706 to the classification on VFG-740. Four aforementioned traditional baseline methods are applied for comparison. As shown in Supplementary Table S9, by pretraining on the VFG-2706 dataset, both VFNet (TL) and VFNet-H (TL) significantly outperform the models without TL, whereas VFNet-H (TL) and SVM combining with ALL predefined features achieve the best accuracy and F1-score performance. However, a Pearson correlation coefficient analysis indicates a linear correlation of $\rho = 0.5286$ between the mean macro F1-score of VFNet and the sequence identity of each class in VFG-740 (Supplementary Fig. S11), implying that, to some extent, the performance of all methods on the VFG-740 dataset might have been overestimated due to the

Table 2. Performance (means ± SDs) of VFNet, VFNet (TL), VFNet-H, VFNet-H (TL) and four traditional classifiers on VFG-2706, VFG-2706-1066 and VFG-564

Dataset	Feature	Method	Train accuracy	Test accuracy (micro)	Precision (macro)	Recall (macro)	F1-score (macro)
VFG-2706	Raw data	VFNet	0.9915 (±0.0016)	0.9663 (±0.0011)	0.9693 (±0.0018)	0.9551 (±0.0019)	0.9577 (±0.0018)
	AAC	RF	1.0000 (±0.0000)	0.9631 (±0.0008)	0.9772 (±0.0007)	0.9477 (±0.0014)	0.9584 (±0.0012)
		VFNet-H	0.9849 (±0.0009)	0.9661 (±0.0021)	0.9683 (±0.0019)	0.9553 (±0.0027)	0.9574 (±0.0025)
	DPC	RF	1.0000 (±0.0000)	0.9767 (±0.0036)	**0.9896 (±0.0003)**	0.9645 (±0.0006)	0.9743 (±0.0005)
		VFNet-H	0.9984 (±0.0003)	**0.9831 (±0.0004)**	0.9868 (±0.0012)	**0.9783 (±0.0129)**	**0.9803 (±0.0015)**
	CTD	RF	1.0000 (±0.0000)	0.9630 (±0.0010)	0.9755 (±0.0008)	0.9479 (±0.0012)	0.9581 (±0.0011)
		VFNet-H	0.9887 (±0.0057)	0.9684 (±0.0081)	0.9704 (±0.0089)	0.9582 (±0.0103)	0.9602 (±0.0104)
	PseAAC1	RF	1.0000 (±0.0000)	0.9640 (±0.0011)	0.9793 (±0.0010)	0.9490 (±0.0018)	0.9604 (±0.0013)
		VFNet-H	0.9861 (±0.0031)	0.9664 (±0.0023)	0.9690 (±0.0033)	0.9559 (±0.0039)	0.9587 (±0.0039)
	PseAAC2	RF	1.0000 (±0.0000)	0.9706 (±0.0005)	0.9841 (±0.0003)	0.9565 (±0.0013)	0.9671 (±0.0009)
		VFNet-H	0.9891 (±0.0023)	0.9685 (±0.0010)	0.9713 (±0.0013)	0.9594 (±0.0016)	0.9614 (±0.0016)
	ALL	RF	1.0000 (±0.0000)	0.9758 (±0.0014)	0.9853 (±0.0016)	0.9649 (±0.0018)	0.9718 (±0.0021)
		VFNet-H	0.9976 (±0.0002)	0.9826 (±0.0006)	0.9857 (±0.0005)	0.9782 (±0.0010)	0.9798 (±0.0007)
VFG-2706-1066	Raw data	VFNet	0.9961 (±0.0011)	0.9244 (±0.0024)	0.9354 (±0.0036)	0.9153 (±0.0028)	0.9164 (±0.0032)
	AAC	RF	1.0000 (±0.0000)	0.8447 (±0.0064)	0.8660 (±0.0033)	0.8073 (±0.0035)	0.8155 (±0.0025)
		VFNet-H	0.9873 (±0.0013)	0.9161 (±0.0024)	0.9242 (±0.0039)	0.9040 (±0.0025)	0.9039 (±0.0032)
	DPC	SVM	0.9873 (±0.0008)	0.9275 (±0.0042)	0.9432 (±0.0030)	0.9380 (±0.0025)	0.9334 (±0.0028)
		VFNet-H	0.9983 (±0.0043)	**0.9414 (±0.0032)**	**0.9599 (±0.0019)**	**0.9439 (±0.0018)**	**0.9458 (±0.0019)**
	CTD	RF	1.0000 (±0.0000)	0.8838 (±0.0021)	0.8930 (±0.0053)	0.8549 (±0.0042)	0.8585 (±0.0038)
		VFNet-H	0.9886 (±0.0028)	0.9132 (±0.0035)	0.9203 (±0.0043)	0.9031 (±0.0037)	0.9017 (±0.0040)
	PseAAC1	RF	1.0000 (±0.0000)	0.8565 (±0.0039)	0.8841 (±0.0061)	0.8216 (±0.0053)	0.8320 (±0.0038)
		VFNet-H	0.9885 (±0.0019)	0.9155 (±0.0027)	0.9240 (±0.0049)	0.9057 (±0.0039)	0.9051 (±0.0046)
	PseAAC2	RF	1.0000 (±0.0000)	0.8672 (±0.0059)	0.8954 (±0.0075)	0.8334 (±0.0073)	0.8448 (±0.0072)
		VFNet-H	0.9896 (±0.0018)	0.9171 (±0.0025)	0.9266 (±0.0050)	0.9074 (±0.0038)	0.9070 (±0.0043)
	ALL	SVM	0.9939 (±0.0007)	0.9387 (±0.0039)	0.9510 (±0.0048)	0.9477 (±0.0026)	0.9434 (±0.0039)
		VFNet-H	0.9959 (±0.0021)	0.9376 (±0.0020)	0.9541 (±0.0026)	0.9397 (±0.0038)	0.9401 (±0.0027)
VFG-564	Raw data	VFNet	0.9969 (±0.0005)	0.8516 (±0.0046)	0.8440 (±0.0116)	0.8427 (±0.0063)	0.8236 (±0.0087)
		VFNet (TL)	0.9911 (±0.0007)	0.9262 (±0.0058)	0.9268 (±0.0088)	0.9235 (±0.0063)	0.9146 (±0.0078)
	AAC	RF	1.0000 (±0.0000)	0.8287 (±0.0113)	0.8153 (±0.0137)	0.8185 (±0.0098)	0.7971 (±0.0113)
		VFNet-H	0.9944 (±0.0021)	0.8565 (±0.0040)	0.8530 (±0.0081)	0.8481 (±0.0036)	0.8313 (±0.0054)
		VFNet-H (TL)	0.9939 (±0.0014)	0.9327 (±0.0084)	0.9363 (±0.0058)	0.9315 (±0.0070)	0.9245 (±0.0075
	DPC	SVM	0.9997 (±0.0004)	0.9307 (±0.0066)	0.9288 (±0.0075)	0.9267 (±0.0079)	0.9174 (±0.0077)
		VFNet-H	0.9999 (±0.0002)	0.8940 (±0.0079)	0.8901 (±0.0093)	0.8871 (±0.0079)	0.8739 (±0.0093)
		VFNet-H (TL)	1.0000 (±0.0000)	0.9503 (±0.0058)	0.9520 (±0.0048)	0.9478 (±0.0050)	0.9427 (±0.0052)
	CTD	SVM	0.9924 (±0.0009)	0.8902 (±0.0051)	0.8771 (±0.0091)	0.8808 (±0.0072)	0.8648 (±0.0087)
		VFNet-H	0.9929 (±0.0010)	0.8619 (±0.0086)	0.8552 (±0.0131)	0.8512 (±0.0093)	0.8349 (±0.0120)
		VFNet-H (TL)	0.9950 (±0.0023)	0.9277 (±0.0067)	0.9287 (±0.0079)	0.9242 (±0.0068)	0.9168 (±0.0078)
	PseAAC1	RF	1.0000 (±0.0000)	0.8359 (±0.0088)	0.8246 (±0.0142)	0.8268 (±0.0101)	0.8063 (±0.0126)
		VFNet-H	0.9947 (±0.0013)	0.8575 (±0.0075)	0.8531 (±0.0108)	0.8489 (±0.0074)	0.8323 (±0.0090)
		VFNet-H (TL)	0.9933 (±0.0009)	0.9302 (±0.0054)	0.9304 (±0.0048)	0.9274 (±0.0045)	0.9191 (±0.0047)
	PseAAC2	SVM	0.9770 (±0.0025)	0.8808 (±0.0041)	0.8611 (±0.0077)	0.8704 (±0.0055)	0.8509 (±0.0068)
		VFNet-H	0.9938 (±0.0012)	0.8597 (±0.0085)	0.8581 (±0.0094)	0.8529 (±0.0073)	0.8378 (±0.0087)
		VFNet-H (TL)	0.9945 (±0.0089)	0.9336 (±0.0071)	0.9323 (±0.0071)	0.9301 (±0.0062)	0.9215 (±0.0069)
	ALL	SVM	0.9994 (±0.0007)	0.9434 (±0.0060)	0.9441 (±0.0067)	0.9418 (±0.0067)	0.9343 (±0.0068)
		VFNet-H	0.9994 (±0.0005)	0.8952 (±0.0078)	0.8929 (±0.0064)	0.8897 (±0.0063)	0.8771 (±0.0067)
		VFNet-H (TL)	1.0000 (±0.0000)	**0.9512 (±0.0053)**	**0.9542 (±0.0069)**	**0.9499 (±0.0045)**	**0.9446 (±0.0058)**

Note: The best performance of each dataset on a metric is highlighted in bold.

relatively high sequence homology. Unfortunately, VFG-740 is not suitable to further exclude highly homologous sequences since it contains no more than 10 sequences in each class.

Then, we employ the same TL method on VFG-564, which is the uncommon VFs in divergent datasets. There is a considerable overlap between VFG-564 and VFG-2706 since both of which are derived from the original dataset. Thus, we pretrain VFNet and VFNet-H on the dataset of COG-755 rather than VFG-2706 and then fine-tune the models using VFG-564. Given the limited overall sample size of VFG-564, we divide the dataset into training and testing subsets using the ratio of 6:4 to trade-off the samples for testing and training. For brevity, Table 2 shows the best-competing traditional machine learning method per predefined feature, with the full performance results given in Supplementary Table S10.

Impressively, by pretraining on the large COG-755 dataset, VFNet (TL) significantly outperforms the counterpart without pretraining, VFNet, by 9%, 10%, 10% and 11% improvement in test accuracy, precision, recall and F1-score, respectively. This is very encouraging in the sense that the features transferred from the COG-755 dataset significantly boost the performance of our deep model on VFG-564. VFNet-H consistently outperforms VFNet on raw data in F1-score. This indicates that the predefined features can

also provide complementary information to the automatically learned features. It is remarkable that VFNet-H (TL) combined with different types of predefined features performs rather stably, achieving the test accuracy ranging from 0.9277 to 0.9512 and the F1-score ranging from 0.9168 to 0.9446. In contrast, the best traditional classifier varies for different predefined features and the best performance they achieve fluctuates largely when using different predefined features. For example, the best traditional classifier is RF with an accuracy of 0.8287 and an F1-score of 0.7971 only when the AAC features are used, while the best-competing classifier changes to SVM with an accuracy of 0.9307 and an F1-score of 0.9174 when the DPC features are employed. Overall, VFNet-H (TL) consistently outperforms the best-competing classifier in using all types of predefined features, achieving 1–13% improvement in accuracy and 1–16% in F1-score. VFNet and VFNet-H without using the transferred features as well as some traditional classifiers (e.g. DT and RF) can obtain nearly perfect training accuracy, but perform poorly on the test data. This is because these classifiers lack sufficient samples to train and generalize their classification models, even though these models (especially RF) indeed have the capability to perform excellent classification on the VF dataset of VFG-2706, as evidenced in Table 2. These results demonstrate the importance of our deep learning model's capability in leveraging auxiliary large datasets to learn transferable features for classifying uncommon VFs.

3.6 Interpretation of the VFNet model with known domains

To gain a biological explanation of why VFNet performs well in classifying VF datasets, following a previous strategy (Wang and Wang, 2019), we use the 128 filters in the first convolutional layer of VFNet to detect potential motifs and compare them to known protein domains. However, it is infeasible to perform this comparison for all VF classes in this study since our datasets cover thousands of VF classes. Instead, we focus on two well-studied pan-genera classes, T3SS-C05 and T3SS-C11 of VFG-2706. T3SS-C05 is the major needle complex component of T3SS, which is arranged in a helical fashion to offer an extended cylindrical structure with a central channel for the translocation of effectors into host cell (Cordes *et al.*, 2003). T3SS-C11 is one of the inner membrane components of the export apparatus of T3SS, which form a unique assembly on the periplasmic side of the inner membrane (Filloux and Whitfield, 2016). These two VF classes are selected because both of them have conserved protein domains that have already been established in the Pfam database of protein families (El-Gebali *et al.*, 2019).

Specifically, we first train VFNet on the training dataset of VFG-2706, and then feed all sequences of each class to the trained model to collect the amino acid subsequences that successfully activate each filter of the model with the largest positive activation value. The length of the subsequence is seven as it is determined by the kernel size of the VFNet filters. Finally, all subsequences reported by each filter are used to generate sequence logos using the WebLogo software (Crooks, 2004). The comparison baseline is the WebLogo visualization of the known protein domains of the two classes based on multiple alignments of the seed sequences available from the Pfam database (accessions PF09392 and PF01313 for T3SS-C05 and T3SS-C11, respectively). Among the alignments, the sites with the gaps presented in the majority of seed sequences are manually excluded since the VFNet filters produce ungapped motifs only. The results are illustrated in Supplementary Figure S12, in which for each Pfam domain we present a set of nonoverlapped motifs that are generated by the VFNet filters and matched the baseline motifs. Given the fact that the Pfam domains consist of sequences from various bacteria whereas our VF datasets are only composed by 32 genera of important human pathogens, it is impressive that the discovered amino acid motifs revealed by VFNet exhibit high similarity to the known patterns of protein domains. This result indicates that our VFNet model has very good capability to capture conserved regions or motifs related to protein families, explaining its accurate classification of sequences from hundreds/thousands of VF classes.

4 Discussion

In this study, we first construct a comprehensive bacterial VF dataset containing 3446 VF classes with 160 495 sequences from 32 genera of medically significant pathogens. To the best of our knowledge, this is the largest bacterial VF dataset used for machine learning studies reported in the literature so far. Previous studies use only selected representative samples available from public resources such as the VFDB database, whereas we further expand the original VFDB dataset to include more additional homologous VFs available from NCBI. Our results highlight the importance of the large dataset for enhancing the performance of machine learning models. Therefore, the VF dataset produced in this work provides a valuable benchmark for future development and evaluation of novel methods for the classification of bacterial VFs. Furthermore, our dataset is well categorized into thousands of bacterial VF functional classes, which are informative for further biological studies for real-world application of bacterial VF prediction.

Second, we propose an end-to-end VFNet model to automatically learn expressive feature representations for the classification of bacterial VFs. Compared with the other two deep learning methods, VFNet is substantially more efficient and accurate for VF data classification. We therefore recommend that CNNs with one-hot encoding should be considered first rather than bidirectional GRU or 'CNN + LSTM' models to classify bacterial VF data. In addition, we employ four popular traditional machine learning methods as baseline methods and show that they can also classify VF data successfully. The feature DPC outperforms the other predefined features for all methods, even enabling equivalent performance to the use of all predefined features (ALL), which implies that the amino acid pair frequency of each sequence makes the main contribution to VF classification. It is worth noting that traditional classifiers heavily rely on predefined features. For instance, from the results of the four baseline methods on the VFG-564, the SVM model with ALL achieves the best performance, while the SVM model with the other features (i.e. AAC, CTD, PseAAC1 and PseAAC2) performs poorly (see Supplementary Table S10). Although evolutionary information-based features such as PSSMs have been used to improve the accuracy of secreted effector prediction (Wang *et al.*, 2018), we do not include these features in this study, as extracting such features from our dataset with more than 160 000 samples is extremely time-consuming. Besides, we employ a popular balanced class weighting method to improve the performance on minority classes, but it does not consistently improve the performance of VFNet on the VFG-2706 dataset. We believe that having an appropriate class weight per class is an important way to further improve the model's performance, but more advanced methods are required to automatically learn adaptive class weights for different classes in our dataset that contains thousands of classes. In this work, we focus on the design of effective deep learning models to work at scale in classifying common/uncommon VF classes. We plan to explore this adaptive class weighting idea in future.

Third, we empirically justify three critical characteristics of our VFNet model. We demonstrate that VFNet has the capability of harnessing expanded data from easily accessible databases. With the expanded data, the generalization ability of our VFNet has been greatly improved, which provides important insight that public databases (e.g. VFDB and NCBI) should be well leveraged to improve prediction on sequences when using deep learning models. Then, we show that we can combine our automatically learned features with predefined features by a variant of VFNet, VFNet-H, to perform more effective classification of the bacterial VF dataset. The generalization ability of VFNet-H improves on nearly all features, especially DPC, with which VFNet-H obtains the best performance. This shows that our VFNet is very flexible and can learn expressive features from raw data to perform the desired classification and combine the automatically learned and predefined features to further enhance its performance. Finally, we verify that VFNet and its variant VFNet-H can transfer the feature representations learned from an auxiliary large dataset, COG-755, to empower the classification of uncommon VF classes (e.g. VFG-564). The resulting models significantly improve the classification performance of VFNet on

uncommon VFs and substantially outperform the best traditional classifier. This suggests that the TL approach is helpful and effective in improving the classification accuracy of CNN models on VF classes with limited samples. In addition, our empirical results also show that our model can well capture conserved motifs that exhibit high similarity to a set of known patterns of protein domains, enabling us to explain the effectiveness of our model from a biological perspective.

5 Conclusion

In this study, we first construct a large bacterial VF dataset of 32 genera of human pathogens from the public domain. We believe that these well-organized VF data are important for further identification and prediction of bacterial VFs, providing an important benchmark for future development and testing of novel methods for computational prediction of VFs. We further introduce the end-to-end CNN-based classification model VFNet and its variant VFNet-H to effectively and efficiently classify the bacterial VFs. To the best of our knowledge, this is the first successful application of deep learning algorithms on the classification of bacterial VF data with thousands of categories. Our results emphasize the power of deep learning models to automatically combine learning features and traditional predefined features to achieve striking performance. The predefined features (e.g. DPC and PSSMs) are good representatives of the current knowledge in the biological sciences. However, we are far from a full understanding of the law of life. The capability of hybrid automatically learned and predefined features highlighted in this study is particularly valuable to future applications of deep learning models on biological data. Moreover, we reveal that our deep learning models can leverage large auxiliary data to learn transferable features to enhance the classification of uncommon VFs. The specific characteristics of deep learning models on TL illustrated here provide new avenues for computational studies on biological issues without sufficient and readily accessible samples.

Acknowledgements

We would like to express our deep appreciation toward Prof. Chunhua Shen, Mr Shenqin Jiang, Mr Anh-Dzung Doan, Mr Haokui Zhang, Mr Haiming Xu, Dr Yuankai Qi for their inspirational discussion. This work was implemented when the first author was visiting the University of Adelaide.

Funding

This work was supported by the National Natural Science Foundation of China [31970635 to J.Y.]. The first author was awarded a scholarship under the State Scholarship Fund from the China Scholarship Council.

Conflict of Interest: none declared.

References

Agarap,A.F. (2018) Deep learning using rectified linear units (ReLU). *arXiv Preprint*, arXiv:1803.08375.

Bileschi,M.L. *et al.* (2019) Using deep learning to annotate the protein universe. *bioRxiv*, 626507.

Billington,S. *et al.* (2002) Identification and role in virulence of putative iron acquisition genes from *Corynebacterium pseudotuberculosis*. *FEMS Microbiol. Lett.*, **208**, 41–45.

Boeckmann,B. (2003) The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res.*, **31**, 365–370.

Chen,L. *et al.* (2016) VFDB 2016: hierarchical and refined dataset for big data analysis—10 years on. *Nucleic Acids Res.*, **44**, D694–697.

Chung,J. *et al.* (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv Preprint*, arXiv:1412.3555.

Cordes,F.S. *et al.* (2003) Helical structure of the needle of the type III secretion system of *Shigella flexneri*. *J. Biol. Chem.*, **278**, 17103–17107.

Cover,T.M. and Thomas,J.A. (2006) Elements of Information Theory. 2nd edn. John Wiley & Sons, Hoboken, NJ.

Crooks,G.E. (2004) WebLogo: a sequence logo generator. *Genome Res.*, **14**, 1188–1190.

Cui,W. *et al.* (2013) Computationally identifying virulence factors based on KEGG pathways. *Mol. BioSystems*, **9**, 1447–1452.

El-Gebali,S. *et al.* (2019) The Pfam protein families database in 2019. *Nucleic Acids Res.*, **47**, D427–D432.

Fiannaca,A. *et al.* (2018) Deep learning models for bacteria taxonomic classification of metagenomic data. *BMC Bioinformatics*, **19**, 198.

Filloux,A. and Whitfield,C. (2016) Editorial: the many wonders of the bacterial cell surface. *FEMS Microbiol. Rev.*, **40**, 161–163.

Fu,L. *et al.* (2012) CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, **28**, 3150–3152.

Garg,A. and Gupta,D. (2008) VirulentPred: a SVM based prediction method for virulent proteins in bacterial pathogens. *BMC Bioinformatics*, **9**, 62.

Goldberg,Y. and Levy,O. (2014) Word2vec explained: deriving Mikolov *et al.*'s negative-sampling word-embedding method. *arXiv Preprint*, arXiv: 1402.3722.

Gupta,A. *et al.* (2014) MP3: a software tool for the prediction of pathogenic proteins in genomic and metagenomic data. *PLoS One*, **9**, e93907.

Hamid,M.-N. and Friedberg,I. (2019) Identifying antimicrobial peptides using word embedding with deep recurrent neural networks. *Bioinformatics*, **35**, 2009–2016.

King,G. and Zeng,L. (2001) Logistic regression in rare events data. *Polit. Anal.*, **9**, 137–163.

Kingma,D.P. and Ba,J. (2014) Adam: a method for stochastic optimization. *arXiv Preprint*, arXiv:1412.6980.

LeCun,Y. *et al.* (2015) Deep learning. *Nature*, **521**, 436–444.

Levy,O. and Goldberg,Y. (2014) Neural word embedding as implicit matrix factorization. *Adv. Neural Inf. Process. Syst.*, **27**, 2177–2185.

Liu,B. *et al.* (2019) VFDB 2019: a comparative pathogenomic platform with an interactive web interface. *Nucleic Acids Res.*, **47**, D687–D692.

Manuel Martinez-Garcia,P. *et al.* (2015) T346Hunter: a novel web-based tool for the prediction of type III, type IV and type VI secretion systems in bacterial genomes. *PLoS One*, **10**, e0119317.

Min,S. *et al.* (2017) Deep learning in bioinformatics. *Brief. Bioinform.*, **18**, 851–869.

Poplin,R. *et al.* (2018) A universal SNP and small-indel variant caller using deep neural networks. *Nat. Biotechnol.*, **36**, 983–987.

Pundhir,S. and Kumar,A. (2011) SSPred: a prediction server based on SVM for the identification and classification of proteins involved in bacterial secretion systems. *Bioinformation*, **6**, 380–382.

Sachdeva,G. *et al.* (2005) SPAAN: a software program for prediction of adhesins and adhesin-like proteins using neural networks. *Bioinformatics*, **21**, 483–491.

Seo,S. *et al.* (2018) DeepFam: deep learning based alignment-free method for protein family modeling and prediction. *Bioinformatics*, **34**, i254–i262.

Srivastava,N. *et al.* (2014) Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, **15**, 1929–1958.

Su,M.-G. *et al.* (2014) Incorporating amino acids composition and functional domains for identifying bacterial toxin proteins. *BioMed Res. Int.*, **2014**, 1–7.

Tatusov,R. (2000) The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic Acids Res.*, **28**, 33–36.

Umarov,R. *et al.* (2019) Promoter analysis and prediction in the human genome using sequence-based deep learning models. *Bioinformatics*, **35**, 2730–2737.

van der Maaten,L. and Hinton,G. (2008) Visualizing data using t-SNE. *J. Mach. Learn. Res.*, **9**, 2579–2605.

van Oosten,M. *et al.* (2015) Targeted imaging of bacterial infections: advances, hurdles and hopes. *FEMS Microbiol. Rev.*, **39**, 892–916.

Veltri,D. *et al.* (2018) Deep learning improves antimicrobial peptide recognition. *Bioinformatics*, **34**, 2740–2747.

Vinatzer,B. *et al.* (2005) Bioinformatics correctly identifies many type III secretion substrates in the plant pathogen *Pseudomonas syringae* and the biocontrol isolate P-fluorescens SBW25. *Mol. Plant-Microbe Interact.*, **18**, 877–888.

Wang,J. and Wang,L. (2019) Deep learning of the back-splicing code for circular RNA formation. *Bioinformatics*, **35**, 5235–5242.

Wang,J. *et al.* (2018) Bastion6: a bioinformatics approach for accurate prediction of type VI secreted effectors. *Bioinformatics*, **34**, 2546–2555.

Wang,Y. *et al.* (2014) Prediction of bacterial type IV secreted effectors by C-terminal features. *BMC Genomics*, **15**, 50.

Weiss,K. *et al.* (2016) A survey of transfer learning. *J. Big Data*, **3**, 9.

Wold,S. *et al.* (1987) Principal component analysis. *Chemometr. Intell. Lab. Syst.*, **2**, 37–52.

Yu,L. *et al*. (2010) SecretP: identifying bacterial secreted proteins by fusing Chou's pseudo-amino acid composition. *J. Theor. Biol*., **267**, 1–6.

Zeng,C. and Zou,L. (2019) An account of *in silico* identification tools of secreted effector proteins in bacteria and future challenges. *Brief. Bioinform*., **20**, 110–129.

Zhang,X. *et al*. (2015) Character-level convolutional networks for text classification. *Adv. Neural Inf. Process. Syst*., **28**, 649–657.

Zhuang,Z. *et al*. (2019) A simple convolutional neural network for prediction of enhancer–promoter interactions with DNA sequence data. *Bioinformatics*, **35**, 2899–2906.

Zou,L. *et al*. (2013) Accurate prediction of bacterial type IV secreted effectors using amino acid composition and PSSM profiles. *Bioinformatics*, **29**, 3135–3142.