Briefings in Bioinformatics, 2024, **25(5)**, bbae377
https://doi.org/10.1093/bib/bbae377
Problem Solving Protocol

DeePhafier: a phage lifestyle classifier using a multilayer self-attention neural network combining protein information

Yan Miao [1], Zhenyuan Sun [1], Chen Lin [2], Haoran Gu [1], Chenjing Ma [1], Yingjian Liang [3], Guohua Wang [1,*]

[1]College of Computer and Control Engineering, Northeast Forestry University, No. 26 Hexing Road, Harbin, 150040, Heilongjiang, China
[2]National Institute for Data Science in Health and Medicine, Xiamen University, No. 4221 Xiangannan Road, Xiamen, 361102, Fujian, China
[3]Key Laboratory of Hepatosplenic Surgery, Ministry of Education, Department of General Surgery, the First Affiliated Hospital of Harbin Medical University, No. 23 Postal Street, Harbin, 150007, Heilongjiang, China

*Corresponding author. E-mail: ghwang@nefu.edu.cn

Abstract

Bacteriophages are the viruses that infect bacterial cells. They are the most diverse biological entities on earth and play important roles in microbiome. According to the phage lifestyle, phages can be divided into the virulent phages and the temperate phages. Classifying virulent and temperate phages is crucial for further understanding of the phage–host interactions. Although there are several methods designed for phage lifestyle classification, they merely either consider sequence features or gene features, leading to low accuracy. A new computational method, DeePhafier, is proposed to improve classification performance on phage lifestyle. Built by several multilayer self-attention neural networks, a global self-attention neural network, and being combined by protein features of the Position Specific Scoring Matrix matrix, DeePhafier improves the classification accuracy and outperforms two benchmark methods. The accuracy of DeePhafier on five-fold cross-validation is as high as 87.54% for sequences with length >2000bp.

Keywords: metagenome; phage lifestyle classification; self-attention network; PSSM matrix

Introduction

Bacteriophages, viruses that infect bacterial cells, are the most common and diverse biological entities in the biosphere [1]. It is estimated that at least 100 million phages exist globally, which is 10 times that of bacteria on average [2]. In microbiome, they play an important role in host death, metabolism, physiology, and evolution by interacting with their host. When a phage infects a bacterial cell, it exists as one of two lifestyles, namely virulent phage or temperate phage. As a virulent phage, its genome is replicated multiple times during infecting, and the newly replicated copies are released into the surrounding environment by lysis, extrusion, or budding. In contrast, when temperate phages infect bacteria, they either integrate their DNA into the bacterial genome or transform their DNA into a loop to form a stable plasmid. As the host bacterium grows and divides, the temperate phage will exist as a prophage in this semi-stable lifestyle. During the subsequent process of bacterial host cell division, the prophage will remain present until appropriate environmental conditions allow the temperate phage to enter a virulent lifestyle and be released from the host bacterium. This shift to a virulent lifestyle is known as induction and is usually caused by host cell damage [3] or environmental stress [4, 5]. Although phages may destroy bacteria, they also benefit bacterial populations in some cases and have a crucial impact on the composition of microbial communities [6]. Thus, the accurate classification of phage lifestyle contributes to the understanding of phage population changes, genomics, and microbiology [7]. It plays key roles in understanding the phage–host interactions and their effects in microbial community regulation. In addition, the accurate identification of virulent phages has significant application values in Phage Therapy and Biocontrol [3, 8].

In recent years, several tools have been proposed to distinguish virulent and temperate phages. Although there are few marker genes, phages may still have high frequencies of functional genes used to determine their lifestyles. For example, Emerson *et al.* [9] found that temperate phages have some functional genes, such as integrases and kinases. Schmidt *et al.* [10] found that leucine substitutions in DNA polymerase genes are closely associated with temperate phages. Based on these functional genes, two protein-based methods were developed. PHACTS [7] utilized all protein sequence information of phage genome and uses Random Forest algorithms to discriminate whether a phage is virulent or temperate at the protein sequence level. It proved that virulent phages typically contain genes associated with phage lysis, nucleotide metabolism, or structural proteins, and temperate phages typically contain genes associated with toxins, excision, integration, lysogenicity, or expression regulation. Mavrich [11] classified phages containing at least one temperate phage Pham as temperate phages. These two methods are both based on protein sequences, classifying phage lifestyles based on their gene

Received: May 20, 2024. **Revised:** July 4, 2024. **Accepted:** July 19, 2024

features. However, they cannot effectively deal with metagenomic data because reconstructing the complete genomes of all organisms in metagenomic data is not currently possible. Furthermore, only a few reads in metagenome may contain functional genes that contribute to classification [12]. PHACTS shows that if a phage genome contains more than 25 proteins, it can be classified with 95% accuracy, while the accuracy is greatly reduced if there are fewer proteins in the phage genome. For example, the accuracy of PHACTS is about 65% if only five proteins are present in a phage sequence, and 50% if there are only two proteins.

To deal with the shortcomings, PhagePred [1] is proposed by extracting features of k-mer frequency and using Markov model to identify phage lifestyles from metagenomes. However, by identifying viral sequences using k-mer based methods [12–14], k-mer frequency feature generates a lot of noise in short sequences [15] and the accuracy reduces a lot when the sequences become shorter.

In order to identify phage lifestyle directly from metagenomic data, DeePhage [16] is proposed by encoding bases as one-hot vectors before building a convolutional neural network (CNN) model for automated feature extraction. Similar to the methods using CNN model to identify viral sequences, these methods only focus on the local information of a sequence and ignore the global information of the whole sequence, which implies the performance still needs to be improved.

To improve the identification accuracy of phage lifestyles, DeePhafier is proposed based on a multilayer self-attention neural network combining protein information. It directly extracts high-level features from a sequence by combining global self-attention and local attention and combines the protein features from genes to improve the identification performance.

Methods

Construction of DeePhafier

The construction of DeePhafier is shown in Fig. 1. It is built by a set of parallel Basic Multi-layer Self-attention Network Models (BMSNMs) and the corresponding protein feature embedding models. The input of BMSNM is a codon sequence with maximum length of 100 codons (detailed in Supplementary S7).

Firstly, a query sequence is randomly divided into k short sequences of 300 bp. Secondly, these short sequences are embedded with protein features and then are input into k BMSNMs. The rest of $(k - m)$ BMSNMs are fed by zero. Thirdly, the output of each BMSNM is sequentially input into a single-layer global self-attention neural network as the order of its position in the original query long sequence. Finally, the co-relationships between each short sequence are learned and then classified through a fully connected layer and a softmax layer. The chosen m and k is detailed in supplementary S2.

BMSNM

When the input sequence is shorter than 300bp, it is equivalent to training only one BMSNM. The structure of BMSNM is shown in Fig. 2. Each layer of the self-attention network in BMSNM is constructed as a local self-attention network. Similar to the local self-attention mechanism of Poolingformer [17], when computing the attention values, a sliding window of length $(2w_1 + 1)$ with stride of 1 is constructed, where every input is only calculated with w_1 inputs to its left and right. For the $i - th$ input with window size of w_1, the sliding window is denoted as $\Psi(i, w_1)$:

$$\Psi(i, w_1) = \{i - w_1, \ldots, i - 1, i, i + 1, \ldots, i+w_1\} \tag{1}$$

The output vector corresponding to the $i - th$ input of the first layer in the self-attention network is

$$\boldsymbol{L}_{1i} = \text{soft}\max\left(\alpha_1 q_{1i}^T \boldsymbol{K}_{\Psi(i,w_1)}\right) \boldsymbol{V}_{\Psi(i,w_1)}^T. \tag{2}$$

The structure of the second layer and third layer in the self-attention network is like that of the first layer, except that the size of the sliding window is set to $(2w_2 + 1)$ and $(2w_3 + 1)$, respectively. The outputs of the second layer and third layer in the self-attention network are

$$\boldsymbol{L}_{2i} = \text{soft}\max\left(\alpha_2 q_{2i}^T \boldsymbol{K}_{\Psi(i,w_2)}\right) \boldsymbol{V}_{\Psi(i,w_2)}^T \tag{3}$$

$$\boldsymbol{L}_{3i} = \text{soft}\max\left(\alpha_3 q_{3i}^T \boldsymbol{K}_{\Psi(i,w_3)}\right) \boldsymbol{V}_{\Psi(i,w_3)}^T. \tag{4}$$

The structure of the fourth layer has a sliding window size of $(2w_4 + 1)$ with a sliding step size ξ. Its output is

$$\boldsymbol{L}_{4i} = \text{soft}\max\left(\alpha_4 q_{4i}^T \boldsymbol{K}_{\Psi(i,w_4)}\right) \boldsymbol{V}_{\Psi(i,w_4)}^T, \tag{5}$$

where $K_{\Psi(i,w_j)}$ and $V_{\Psi(i,w_j)}$ are the key vectors and value vectors, respectively, constructed from the inputs of this layer in the sliding window $\Psi(i, w_j)$ from the $j-th$ layer. α_j is a constant-valued scalar for compression in the $j - th$ layer. q_{ji} is a query vector element constructed in the $j-th$ layer according to the $i-th$ input.

Furthermore, several residual connections are added between each layer to the four-layer self-attention neural network. Specifically, the input of the third layer is the sum of outputs from the second layer L_{2i} and the first layer L_{1i}. The input of the fourth layer is the sum of outputs from the third layer L_{3i} and the first two layers $\boldsymbol{L}_{1i}$, L_{2i}. The dimension of the final output is n/ξ.

For each layer in the self-attention neural network, a pooling operation (with a pooling kernel $\mathcal{K}_j$ and a pooling step λ_j) is introduced to both $K_{\Psi(i,w_j)}$ and $V_{\Psi(i,w_j)}$:

$$\tilde{\boldsymbol{K}}_{\Psi(i,w_j)} = \text{maxpooling}\left(\boldsymbol{K}_{\Psi(i,w_j)}; \kappa_j, \lambda_j\right) \tag{6}$$

$$\tilde{V}_{\Psi(i,w_j)} = \max\text{pooling}\left(\boldsymbol{V}_{\Psi(i,w_j)}; \kappa_j, \lambda_j\right) \tag{7}$$

This pooling operation reduces the number of parameters and the computational complexity, as well as plays a role similar to Dropout [18] to improve the generalization ability [19]. The computational complexity of DeePhafier is calculated in Supplementary S3. To confirm the dimension of the input to the later single-layer global self-attention neural network is not too large, the pooling kernel is $\mathcal{K}_j = 5$ and the pooling stride is $\lambda_j = 3$.

When training the models of DeePhafier with different length sequences, the optimization algorithms were all chosen to be the Adam algorithm with default parameters; batch-size is 64; learning rate is 0.001; epoch is 300.

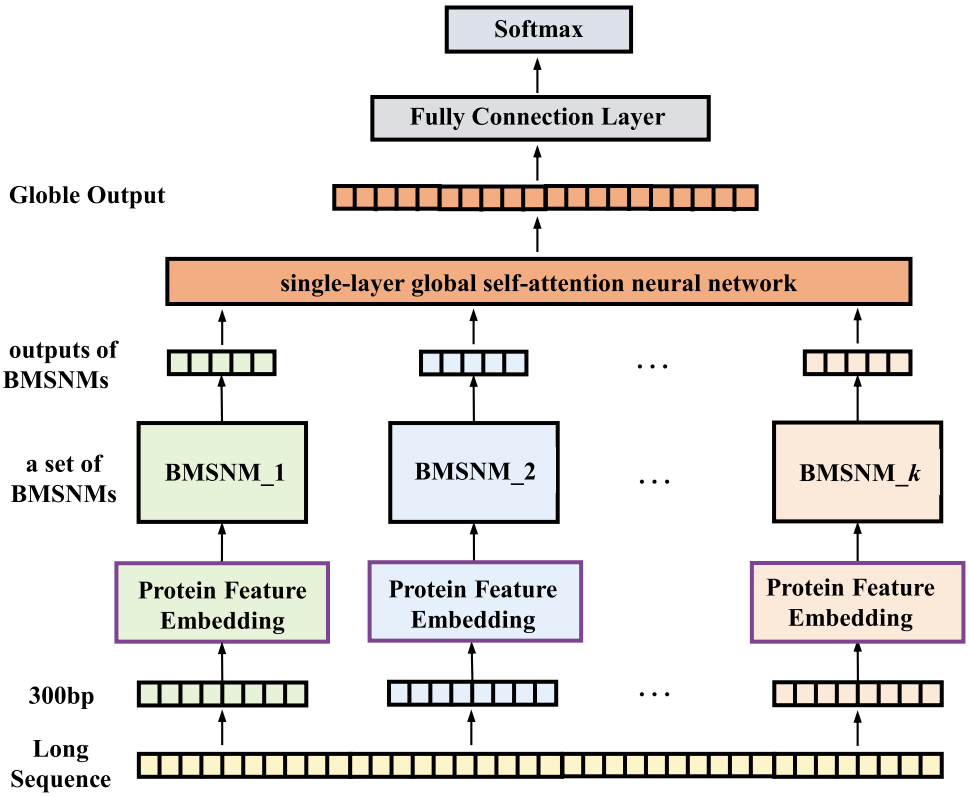


Figure 1. The construction of DeePhafier.

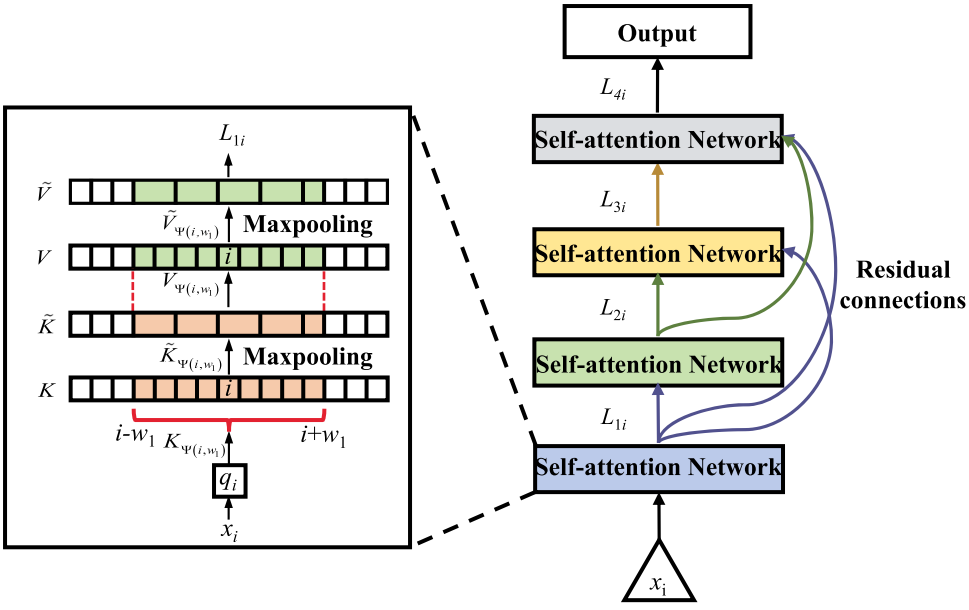


Figure 2. Basic multi-layer self-attention network model.

During the search and comparison process, the maximum number of iterations was set to 3, the E-value for each iteration was set to 0.001, and the rest of the parameters were selected as default.

Protein feature embedding

Position Specific Scoring Matrix (PSSM) is commonly used to extract features of evolutionary information on protein amino acid sequences [20–22], which is widely applied in many fields including protein interaction prediction [23], protein subcellular localization [24], protein secondary structure prediction [25], and so on. All sequences from the training and testing datasets are searched and compared by PSI-BLAST [26] against the homologous sequences from the SWISS-PROT reference database, resulting in homology information that is PSSM. For a protein sequence with the length of L, the PSSM matrix obtained by PSI-BLAST can be expressed as a two-dimensional vector of $L \times 20$. The first dimension represents the position of each amino acid on the protein sequence and the second dimension represents 20 standard amino acids. Each element value in this matrix represents the

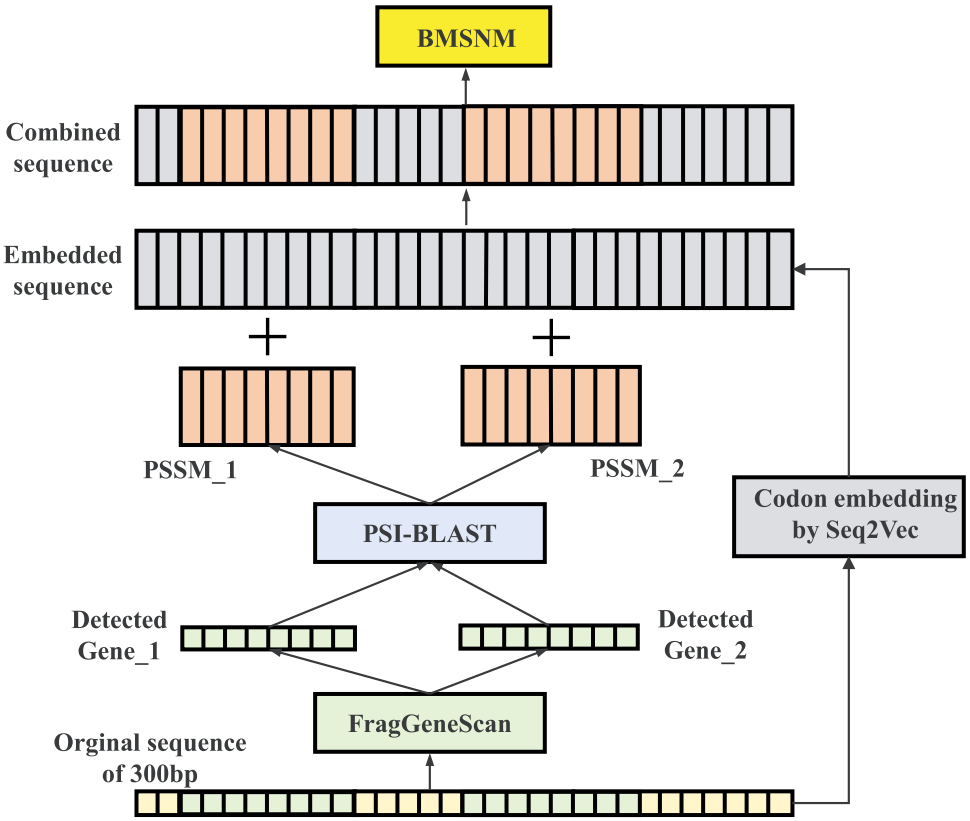


Figure 3. The combination of protein features with codon embedding vectors.

magnitude of the possibility that each amino acid occurs at each position in the sequence (detailed in Supplementary S4).

PHACTS [7] only uses the protein information of the phage genome to distinguish virulent or temperate phages, indicating that protein features extracted from phages play a positive role in classifying phage lifestyles [16]. Therefore, DeePhafier uses FragGeneScan [27] to identify gene sequences as well as their positions from phage sequences. Then these detected gene sequences are searched and compared by PSI-BLAST against the SWISS-PROT reference database to obtain a PSSM matrix. According to the position of genes detected in the phage sequence, the normalized PSSM matrix is added to the original sequence that is firstly embedded by Seq2Vec [12] (shown in Fig. 3). Finally, the codon embedded sequence combined with protein features is fed into a BMSNM for further feature extraction and classification.

Results

Datasets

Phage datasets

Since there is no public dataset with reliable virulence annotation for each phage sequence, contigs extracted from complete phage genomes with accurate annotations are used to train and test DeePhafier. A total of 77 virulent phage genomes and 148 temperate phage genomes with quality annotations [7] used by DeePhage [16] were built as dataset MD (McNair's Dataset). A total of 1299 virulent phage genomes and 535 temperate phage genomes constructed by Song [1] were chosen as dataset SD (Song's Dataset). Phages in MD dataset were manually labeled. Their annotations are extremely reliable. The phage RefSeq genomes on NCBI (https://www.ncbi.nlm.nih.gov/refseq/) were labeled by bioinformatic softwares [11], building the SD dataset, existing some theoretical errors. Therefore, all SD datasets as well as 70% of the MD dataset (containing 54 randomly selected virulent phage genomes and 104 temperate phage genomes) were selected as training data (1353 virulent phage genomes and 639 temperate phage genomes totally). The remaining 30% of the MD dataset (containing the remaining 23 virulent phage genomes and 44 temperate phage genomes) were chosen as testing data. The details can be found in Supplementary S5.1.

Five-fold cross-validation is used to test DeePhafier. During each cross-validation, the training set and validation set were divided according to the complete phage genomes in order to test whether DeePhafier has the ability to identify new phage. Specifically, the 1353 virulent phage genomes and 639 temperate phage genomes from the training data were divided equally into five sets. Each of the first four sets contains 271 virulent phage genomes and 128 temperate phage genomes, and the last set contains 269 virulent phage genomes and 127 temperate phage genomes (shown in Table 1.). A total of 20 000 sequences (containing 10 000 virulent phage sequences and 10 000 temperate phage sequences) with lengths of 100–300bp (GA), 300–500bp (GB), 500–1000bp (GC), 1000–2000bp (GD), and >2000bp (GE) were randomly subsampled from each genome set. For each fold at each length, four sets were selected as the training dataset (80 000 sequences in total) and the remaining one set were selected as the validation dataset (20 000 sequences in total). The 23 virulent phage genomes and 44 temperate phage genomes in the testing data were randomly cut off according to five lengths above (GA, GB, GC, GD, GE) to form the testing dataset (containing 10 000 virulent phage sequences and 10 000 temperate phage sequences).

Real metagenome datasets

A CAMI_high dataset [13], a CAMI Marine dataset [12], and a human gut metagenome dataset [14] were selected as the real metagenome datasets to test the performance of DeePhafier.

Table 1. Composition of the phage dataset

Groups		Group 1	Group 2	Group 3	Group 4	Group 5
Number of virulent phage genomes		271	271	271	271	269
Number of temperate phage genomes		128	128	128	128	127
	GA	10 000	10 000	10 000	10 000	10 000
	GB	10 000	10 000	10 000	10 000	10 000
Number of virulent phage sequences	GC	10 000	10 000	10 000	10 000	10 000
	GD	10 000	10 000	10 000	10 000	10 000
	GE	10 000	10 000	10 000	10 000	10 000
	GA	10 000	10 000	10 000	10 000	10 000
	GB	10 000	10 000	10 000	10 000	10 000
Number of temperate phage sequences	GC	10 000	10 000	10 000	10 000	10 000
	GD	10 000	10 000	10 000	10 000	10 000
	GE	10 000	10 000	10 000	10 000	10 000

Table 2. Five-fold cross-validation results for the three methods

Length	Criteria	First fold	Second fold	Third fold	Fourth fold	Fifth fold
	Accuracy	0.7921	0.7948	0.7883	0.7890	0.7851
	Recall	0.8055	0.8075	0.8037	0.7959	0.7881
<300bp	Precision	0.7845	0.7875	0.7797	0.7851	0.7834
	Specificity	0.7787	0.7821	0.7729	0.7821	0.7821
	F1 SCORE	0.7948	0.7974	0.7915	0.7904	0.7857
	Accuracy	0.8298	0.8326	0.8221	0.8335	0.8236
	Recall	0.8210	0.8271	0.8162	0.8262	0.8173
300–500bp	Precision	0.8357	0.8362	0.8259	0.8384	0.8277
	Specificity	0.8386	0.8380	0.8279	0.8407	0.8299
	F1 SCORE	0.8283	0.8316	0.8210	0.8322	0.8225
	Accuracy	0.8473	0.8478	0.8370	0.8451	0.8306
	Recall	0.8403	0.8492	0.8311	0.8382	0.8286
500–1000bp	Precision	0.8522	0.8467	0.8409	0.8498	0.8318
	Specificity	0.8543	0.8463	0.8428	0.8519	0.8325
	F1 SCORE	0.8462	0.8480	0.8360	0.8440	0.8302
	Accuracy	0.8621	0.8648	0.8566	0.8559	0.8522
	Recall	0.8711	0.8751	0.8625	0.8689	0.8581
1000–2000bp	Precision	0.8557	0.8574	0.8524	0.8469	0.8481
	Specificity	0.8531	0.8544	0.8507	0.8429	0.8463
	F1 SCORE	0.8633	0.8661	0.8574	0.8577	0.8531
	Accuracy	0.8687	0.8750	0.8621	0.8754	0.8593
	Recall	0.8725	0.8782	0.8639	0.8781	0.8596
>2000bp	Precision	0.8659	0.8725	0.8607	0.8734	0.8590
	Specificity	0.8649	0.8717	0.8602	0.8727	0.8589
	F1 SCORE	0.8692	0.8754	0.8623	0.8757	0.8593

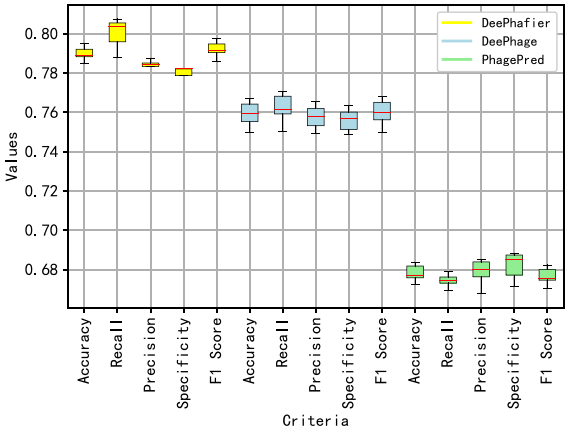
All sequences in each dataset were aligned by BLAST (default parameters) [28] with 1376 virulent phage genomes and 683 temperate phage genomes in the phage genome dataset (MD+SD), respectively. Results with E-value <10-5 is considered as virulent or temperate phages. As a result, 1866 virulent phage sequences and 1463 temperate phage sequences were obtained from the CAMI_high dataset. A total of 8721 virulent phage sequences and 6905 temperate phage sequences were obtained from the CAMI Marine dataset. A total of 775 virulent phage sequences and 940 temperate phage sequences were obtained from the human gut metagenome dataset.

Performance comparison on the Phage datasets

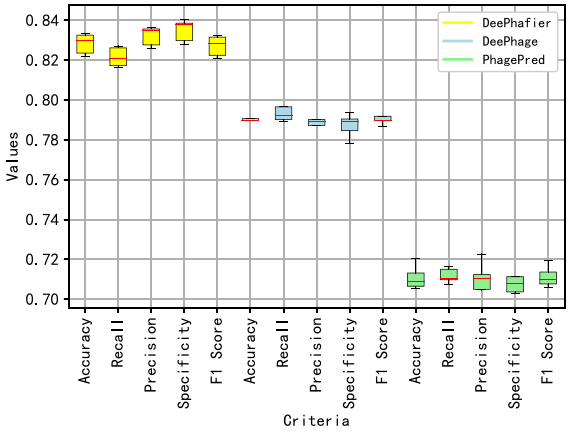
Two tools, DeePhage [16] and PhagePred [1], were chosen as benchmark methods for comparison. When training DeePhafier with different length sequences, the optimization algorithms were all chosen as Adam algorithm with default parameters. The batch size is set as 64, learning rate is set as 0.001, and epoch is 300. Five-fold cross-validation experiments using the datasets in Section Phage datasets were conducted to test the performance of DeePhafier and benchmark methods. Virulent phages were considered as positive samples and temperate phages were considered as negative samples. The simple results were shown in Table 2 (detailed results are shown in Supplementary Table S2). In each fold of cross-validation, DeePhafier outperformed DeePhage and PhagePred on all criteria (detailed in Supplementary S6). Based on the testing results of the five-fold cross-validation, five boxplots of each criterion for sequences of different lengths were plotted in Fig. 4. The performance of PhagePred is much lower than DeePhafier and DeePhage. For short sequences <500bp, the performance of DeePhafier is greater than DeePhage. And for sequences >500bp, the identification results of DeePhafier and DeePhage are similar, but DeePhafier still achieves a better performance than DeePhage.

Performance on the CAMI high dataset

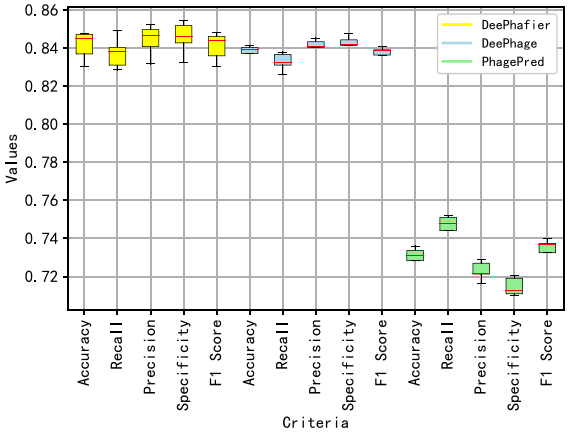
DeePhage and the other two method were used to classify phage lifestyles in the CAMI_high dataset, and the ROC curves were plotted in Fig. 5. Although some parts of the DeePhafier's ROC



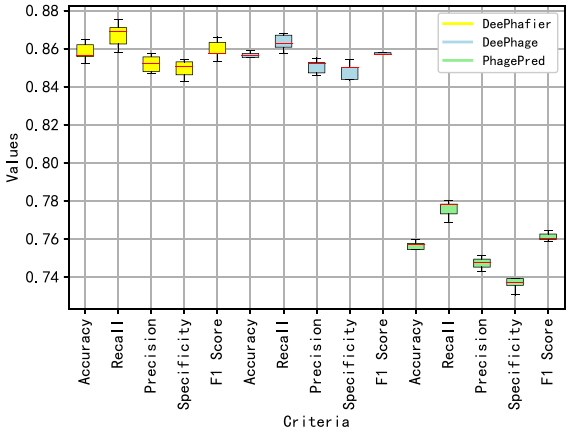
(a) <300bp



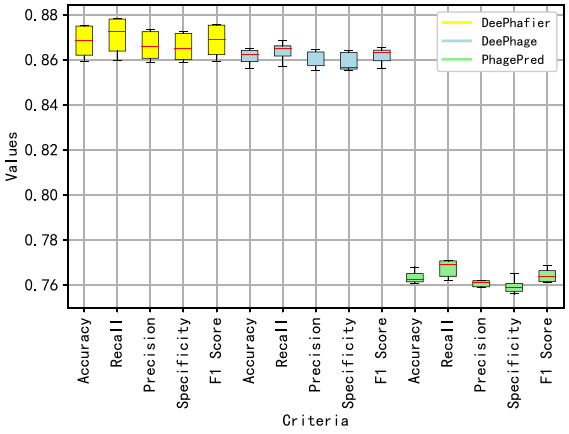
(b) 300-500bp



(c) 500-1,000bp



(d) 1,000-2,000bp



(e) >2,000bp

Figure 4. Boxplots of each criterion for sequences of different lengths.

curve were under that of DeePhage, DeePhafier still achieved the highest AUC value of 0.8110, which is 0.0092 and 0.1316 higher than DeePhage and PhagePred, respectively. The accuracies, recalls, precisions, specificities, and F1 scores of the three methods are shown in Table 3 (bold represents the optimal data). The five criteria of DeePhafier were 0.7918, 0.8038, 0.8210, 0.7765, 0.8123, which were 0.0177, 0.0074, 0.0212, 0.0308, 0.0142 and 0.1183, 0.1334, 0.0949, 0.0991, 0.1152 higher than DeePhage and PhagePred, respectively.

Performance comparison on the CAMI Marine dataset

The ROC curves of the three methods for phage lifestyle classification on the CAMI marine dataset are shown in Fig. 6. When FPR

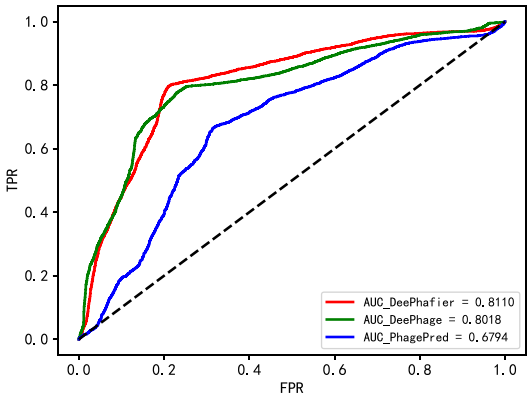


Figure 5. ROC curves and AUC values of the three methods on the CAMI_high dataset.

Table 3. Criteria for phage lifestyle classification on CAMI_high dataset

Criteria	PhagePred	DeePhage	DeePhafier
Accuracy	0.6735	0.7741	**0.7918**
Recall	0.6704	0.7964	**0.8038**
Precision	0.7261	0.7998	**0.8210**
Specificity	0.6774	0.7457	**0.7765**
F1 SCORE	0.6971	0.7981	**0.8123**

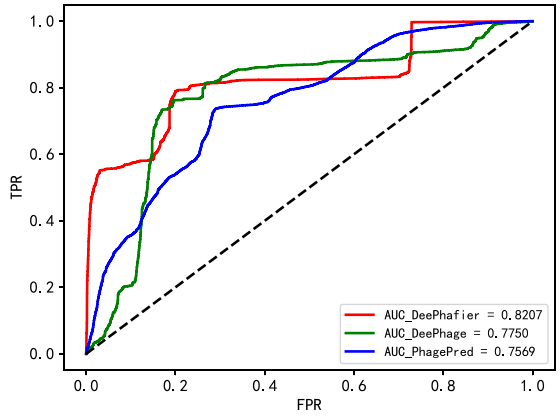


Figure 6. ROC curves and AUC values of classification results on the CAMI Marine dataset.

is around 0.2 and 0.3–0.7, the ROC curve of DeePhafier is below that of DeePhage. Even then, DeePhafier still obtains the highest AUC value of 0.8207, which is 0.0457 and 0.0638 higher than that of DeePhage and PhagePred, respectively. The accuracies, recalls, precisions, specificities, and F1 scores are shown in Table 4 (bold represents the optimal results). The five criteria of DeePhafier are 0.7866, 0.8064, 0.8103, 0.7616, 0.8084, which are 0.0309, 0.0368, 0.0196, 0.0189, 0.0284 and 0.0645, 0.0779, 0.0473, 0.0473, 0.0630 higher than DeePhage and PhagePred, respectively.

Performance comparison on the real human gut metagenome

DeePhafier, DeePhage, and PhagePred are tested by classifying phage lifestyles in the real human gut metagenome. The identification results are shown in Fig. 7. Every part of the DeePhafier's ROC curve is above that of the other two benchmark methods. The AUC value of DeePhafier is 0.7727, which is 0.0511 and 0.1474 higher than DeePhage and PhagePred, respectively. Based on the

Table 4. Five classification criteria of the three methods on the CAMI Marine dataset

Criteria	PhagePred	DeePhage	DeePhafier
Accuracy	0.7221	0.7577	**0.7866**
Recall	0.7285	0.7696	**0.8064**
Precision	0.7630	0.7907	**0.8103**
Specificity	0.7143	0.7427	**0.7616**
F1 SCORE	0.7454	0.7800	**0.8084**

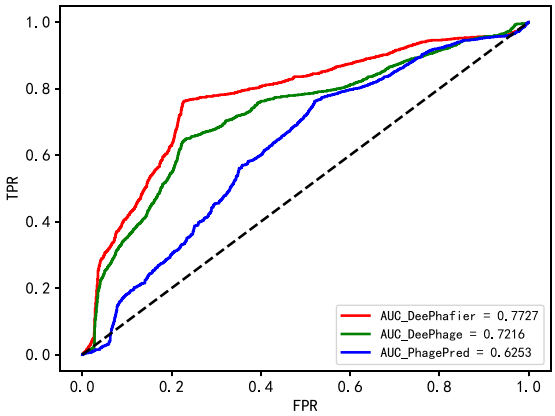


Figure 7. ROC curves and AUC values of classification results on the human gut metagenome.

Table 5. Five classification criteria of the three methods on the human gut metagenome

Criteria	PhagePred	DeePhage	DeePhafier
Accuracy	0.6394	0.7096	**0.7673**
Recall	0.6540	0.6981	**0.7626**
Precision	0.5621	0.6720	**0.7332**
Specificity	0.6287	0.7191	**0.7713**
F1 SCORE	0.6046	0.6848	**0.7476**

classification results of the three methods and the real labels of the testing dataset, their accuracies, recalls, precisions, specificities, and F1 scores are shown in Table 5. DeePhafier achieves the best results among all five criteria, 0.7673 for accuracy, 0.7626 for recall, 0.7332 for precision, 0.7713 for specificity, and 0.7476 for F1 score, respectively, which are 0.0577, 0.0649, 0.0612, 0.0522, 0.0828 and 0.1279, 0.1086, 0.1711, 0.1426, 0.1430 higher than DeePhage and PhagePred, respectively.

Significance test of method performance

In order to prove the significant improvement of DeePhafier over the other two methods more comprehensively, Friedman test [29, 30] and Nemenyi [31] test were utilized as statistical measures, which compared the performance of multiple algorithms on multiple datasets (detailed in Supplementary S8).

A total of 48 criteria, containing AUC values, accuracies, recalls, precisions, specificities, and F1 scores generated from the testing dataset (five-fold cross-validation), the CAMI_high dataset, the CAMI Marine dataset, and the human gut metagenome dataset of the three methods, were ranked. The results of Friedman test and Nemenyi test are shown in Fig. 8 (the dots indicate the average ranking of the three methods and the length of each line represents the CD value). There is an intersection region between DeePhafier and both DeePhage and PhagePred, which indicates

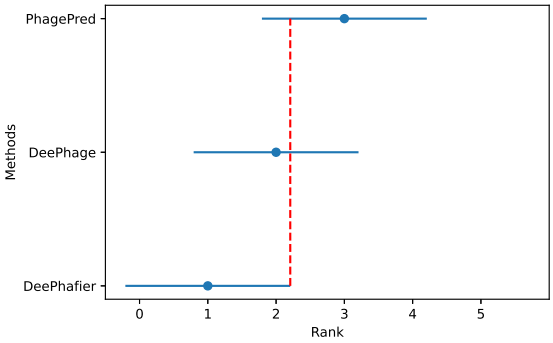


Figure 8. Results of Friedman test and Nemenyi test for three methods.

that there is no significant difference between the performance of DeePhafier and two benchmark methods in classifying phage lifestyles. However, the performance of DeePhafier is still better than two benchmark methods.

Discussion

DeePhafier is proposed to improve the identification accuracy of phage lifestyles. It is built by a multilayer self-attention neural network combining protein information. It works because of three points. Firstly, self-attention neural network has been proved to be effective in learning features from sequences [19, 32, 33], especially being good at dealing with these of several hundreds of words [12, 13]. So, a set of BMSNMs are established to learn features from each part in a sequence. Secondly, the single-layer global self-attention neural network is input by the features generated from these BMSNMs sequentially to further learn higher level features. Moreover, the single-layer global self-attention neural network could construct relationships among short subsequences in BMSNMs without omitting the order of their positions in the original long sequences. Thirdly, according to the positions of genes detected in the phage sequence, the normalized PSSM matrix is added to the codon-embedded sequence, which combining sequence features and protein features.

To prove the effectiveness of several constructions in DeePhafier, we made a comparison between DeePhafier and some variants. The first variant was established by excluding the single-layer global self-attention neural network in DeePhafier, namely Model_1. The second variant was built by abandoning the residual connections in BMSNMs, namely Model_2. The third variant was constructed by removing the pooling operations from BMSNMs, namely Model_3. The codon embeddings of a query sequence were substituted by one-hot vectors (64 dimensions), namely Model_4, where the PSSM matrix was zero-padded to 64 dimensions before being added to the one-hot vectors. Finally, the PSSM matrix was removed before the BMSNMs, namely Model_5. A total of 400 000 sequences from the second fold in the training dataset (80 000 sequences for each length) were used to train these five variant models, and the 10 000 virulent phage sequences and 10 000 temperate phage sequences in the testing dataset were used for testing. The hyperparameters in the training strategy were all the same as DeePhafier (detailed classification results are shown in Supplementary Table S4).

The classification performances of the five variant models are not better than DeePhafier, for all lengths. The single-layer global self-attention neural network in DeePhafier could learn global features [17] from query sequences after several BMSNMs. And the positional encoding mechanism could learn the order of subsequences from BMSNMs. This is why DeePhafier outperforms Model_1 and the gap becomes bigger as the length of query sequence is longer. The residual connections in BMSNM could let the low-level features pass through directly to the output and prevent the whole neural network from overfitting [34]. Thus, the performance of Model_2 drops dramatically because of the residual connections being absent. The operation of maxpooling is mainly designed for generalization and reducing parameters [35]. Without maxpooling, Model_3 performs as well as DeePhafier for short sequences. For long sequences, the performance of Model_3 is a little worse than DeePhafier, which may be caused by bad generalization without maxpooling. Before a query sequence is being input to a deep learning neural network, it is always better if the query sequence is embedded to a vector, which has been proved in Virtifier [12], DETIRE [14], CHEER [36], and so on. When the codon embedding strategy is substituted by one-hot encoding, the performance of Model_4 drops dramatically. Sequence features and protein features describe a DNA sequence in two different ways. Sequence features focus on the composition of the entire sequence and protein features pay more attention on gene expression [7]. Both of the two types of features contribute a lot to phage lifestyle classification. As a result, Model_5 has a rather bad performance without PSSM matrix.

Above all, the designs containing a single-layer global self-attention neural network, residual connections, operation of maxpooling, codon embedding, and PSSM matrix all contribute a lot to the outperformance of DeePhafier. Even then, more complex protein features and multi-head self-attention mechanism may further improve the performance of DeePhafier on classifying phage lifestyles.

Bacteriophages play a very important role in controlling bacterial population size and benefit bacterial populations in some cases. For example, different bacteriophages in soil metagenome can be used to jointly inhibit soil pathogenic bacteria. Thus, the accurate classification of phage lifestyle is the first step for understanding phage–host interactions and their effects in human gut microbiome, which can be used to early diagnosis of colorectal cancer (CRC). Furthermore, the accurate identification of virulent phages contributes to Phage Therapy and Biocontrol. We hope DeePhafier could play an important role in many aspects of virus–host analysis.

Conclusions

Bacteriophages play a very important role in controlling bacterial population size and benefit bacterial populations in some cases. For example, different bacteriophages in soil metagenome can be used to jointly inhibit soil pathogenic bacteria. Thus, the accurate classification of phage lifestyle is the first step for understanding the phage–host interactions and their effects in human gut microbiome, which can be used in the early diagnosis of CRC. Furthermore, the accurate identification of virulent phages contributes to Phage Therapy and Biocontrol. DeePhafier is designed to improve the identification accuracy of phage lifestyles. It is built by several multilayer self-attention neural networks and a global self-attention neural network. Combined by protein features of the PSSM matrix, DeePhafier improves the identification performance and outperforms two benchmark methods. We hope DeePhafier could play an important role in many aspects of phage–host analysis.

Key Points

- Classifying virulent and temperate phages is crucial for further understanding of phage–host interactions. DeePhafier improves the classification accuracy and has played a crucial role in advancing our understanding of phage–host interactions.
- DeePhafier is the first method that learns sequence features by self-attention neural networks for phage lifestyle classification. BMSNM is designed to learn high-level features without too much computational complexity. The single-layer global self-attention neural network can learn the order of subsequences from the original query long sequence.
- Existing methods merely either consider sequence features or gene features, leading to low accuracy. DeePhafier combines the codon embedded sequence and protein features of the PSSM matrix. This enriches the extracted features, and improves the accuracy of DeePhafier in classifying virulent and temperate phages.

Acknowledgments

The authors thank the anonymous reviewers for their valuable suggestions.

Supplementary data

Supplementary data is available at *Briefings in Bioinformatics* online.

Funding

This research was funded by the Fundamental Research Funds for the Central Universities and National Natural Science Foundation of China (NSFC) [62301139 and 62225109].

Data availability

The virulent and temperate phage genomes in Section Phage datasets can be found at https://doi.org/10.1093/gigascience/giab056, https://doi.org/10.1007/s40484-019-0187-4, and NCBI phage RefSeq genomes (https://www.ncbi.nlm.nih.gov/refseq/). The CAMI_high dataset can be found at https://data.cami-challenge.org/camiClient.jar. The CAMI Marine dataset can be found at https://data.cami-challenge.org/participate. The real human gut metagenome dataset can be found from NCBI [SRA052203].

Competing interests

No competing interest is declared.

Author contributions

Y.M. proposed the method and wrote the manuscript. Y.M. and GH.W. conceived the experiments. C.M. pre-processed the sequences from the metagenomes. ZY.S. and Y.M. established the BMSNM model. Y.M., ZY.S., GH.W., and C.L. established the Protein feature embedding model. H.G. and Y.L. helped to do the experiments. C.M., C.L., and Y.M. analyzed the results. All of the authors reviewed the manuscript.

References

1. Ren J, Song K, Deng C. *et al.* Identifying viruses from metagenomic data using deep learning. *Quant Biol* 2020;**8**:64–77. https://doi.org/10.1007/s40484-019-0187-4.
2. Mirzaei MK, Maurice CF. Ménage à trois in the human gut: interactions between host, bacteria and phages. *Nat Rev Microbiol* 2017;**15**:397–408. https://doi.org/10.1038/nrmicro.2017.30.
3. Witkin EM. Ultraviolet mutagenesis and inducible DNA repair in Escherichia coli. *Bacteriol Rev* 1976;**40**:869–907. https://doi.org/10.1128/br.40.4.869-907.1976.
4. Clark DW, Meyer H, Leist C. *et al.* Effects of growth medium on phage production and induction in Escherichia coli K-12 lambda lysogens. *J Biotechnol* 1986;**3**:271–80. https://doi.org/10.1016/0168-1656(86)90009-X.
5. Clarke KJ. Virus particle production in lysogenic bacteria exposed to protozoan grazing. *FEMS Microbiol Lett* 1998;**166**:177–80. https://doi.org/10.1111/j.1574-6968.1998.tb13887.x.
6. Wommack KE, Bhavsar J, Polson SW. *et al.* VIROME: a standard operating procedure for analysis of viral metagenome sequences. *Stand Genomic Sci* 2012;**6**:427–39. https://doi.org/10.4056/sigs.2945050.
7. McNair K, Bailey BA, Edwards RA. PHACTS, a computational approach to classifying the lifestyle of phages. *Bioinformatics* 2012;**28**:614–8. https://doi.org/10.1093/bioinformatics/bts014.
8. Fujiki J, Yoshida S, Nakamura T. *et al.* Novel virulent bacteriophage ΦSG005, which infects *Streptococcus gordonii*, forms a distinct clade among streptococcus viruses. *Viruses* 2021;**13**:1964. https://doi.org/10.3390/v13101964.
9. Emerson JB, Thomas BC, Andrade K. *et al.* Dynamic viral populations in hypersaline systems as revealed by metagenomic assembly. *Appl Environ Microbiol* 2012;**78**:6309–20. https://doi.org/10.1128/AEM.01212-12.
10. Schmidt HF, Sakowski EG, Williamson SJ. *et al.* Shotgun metagenomics indicates novel family a DNA polymerases predominate within marine virioplankton. *ISME J* 2014;**8**:103–14. https://doi.org/10.1038/ismej.2013.124.
11. Mavrich TN, Hatfull GF. Bacteriophage evolution differs by host, lifestyle and genome. *Nat Microbiol* 2017;**2**:1–9.
12. Miao Y, Liu F, Hou T. *et al.* Virtifier: a deep learning-based identifier for viral sequences from metagenomes. *Bioinformatics* 2022;**38**:1216–22. https://doi.org/10.1093/bioinformatics/btab845.
13. Liu F, Miao Y, Liu Y. *et al.* RNN-VirSeeker: a deep learning method for identification of short viral sequences from metagenomes. *IEEE/ACM Trans Comput Biol Bioinform* 2020;**19**:1–9. https://doi.org/10.1109/TCBB.2020.3044575.
14. Miao Y, Bian J, Dong G. *et al.* DETIRE: a hybrid deep learning model for identifying viral sequences from metagenomes. *Front Microbiol* 2023;**14**:1169791. https://doi.org/10.3389/fmicb.2023.1169791.
15. Galiez C, Siebert M, Enault F. *et al.* WIsH: who is the host? Predicting prokaryotic hosts from metagenomic phage contigs. *Bioinformatics* 2017;**33**:3113–4. https://doi.org/10.1093/bioinformatics/btx383.
16. Wu S, Fang Z, Tan J. *et al.* DeePhage: distinguishing virulent and temperate phage-derived sequences in metavirome data with a deep learning approach. *GigaScience* 2021;**10**:giab056. https://doi.org/10.1093/gigascience/giab056.

17. Zhang H, Gong Y, Shen Y. *et al.* Poolingformer: long document modeling with pooling attention. *arXiv e-prints* 2021; arXiv:2105.04371. https://arxiv.org/abs/2105.04371.
18. Alom MZ, Taha TM, Yakopcic C. *et al.* A state-of-the-art survey on deep learning theory and architectures. *electronics* 2019;**8**:292. https://doi.org/10.3390/electronics8030292.
19. Tay Y, Dehghani M, Bahri D. *et al.* Efficient transformers: a survey. *ACM Comput Surv* 2020;**55**:1–28. https://doi.org/10.1145/3530811.
20. Du Z, Huang T, Uversky VN. *et al.* Predicting TF proteins by incorporating evolution information through PSSM. *IEEE/ACM Trans Comput Biol Bioinform* 2022;**20**:1319–26. https://doi.org/10.1109/TCBB.2022.3199758.
21. Zhou S, Zhou Y, Liu T. *et al.* PredLLPS_PSSM: a novel predictor for liquid–liquid protein separation identification based on evolutionary information and a deep neural network. *Brief Bioinform* 2023;**24**:bbad299. https://doi.org/10.1093/bib/bbad299.
22. Guo Y, Wu J, Ma H. *et al.* EPTool: a new enhancing PSSM tool for protein secondary structure prediction. *J Comput Biol* 2021;**28**:362–4. https://doi.org/10.1089/cmb.2020.0417.
23. Murakami Y, Mizuguchi K. Applying the Naïve bayes classifier with kernel density estimation to the prediction of protein–protein interaction sites. *Bioinformatics* 2010;**26**:1841–8. https://doi.org/10.1093/bioinformatics/btq302.
24. Liu B, Zhang D, Xu R. *et al.* Combining evolutionary information extracted from frequency profiles with sequence-based kernels for protein remote homology detection. *Bioinformatics* 2014;**30**:472–9. https://doi.org/10.1093/bioinformatics/btt709.
25. Rashid M, Saha S, Raghava GP. Support vector machine-based method for predicting subcellular localization of mycobacterial proteins using evolutionary information and motifs. *BMC Bioinformatics* 2007;**8**:1–9. https://doi.org/10.1186/1471-2105-8-337.
26. Altschul SF, Madden TL, Schäffer AA. *et al.* Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 1997;**25**:3389–402. https://doi.org/10.1093/nar/25.17.3389.
27. Rho M, Tang H, Ye Y. FragGeneScan: predicting genes in short and error-prone reads. *Nucleic Acids Res* 2010;**38**:e191–1. https://doi.org/10.1093/nar/gkq747.
28. Altschul SF, Gish W, Miller W. *et al.* Basic local alignment search tool. *J Mol Biol* 1990;**215**:403–10. https://doi.org/10.1016/S0022-2836(05)80360-2.
29. Friedman M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J Am Stat Assoc* 1937;**32**:675–701. https://doi.org/10.1080/01621459.1937.10503522.
30. Friedman M. A comparison of alternative tests of significance for the problem of m rankings. *Ann Math Stat* 1940;**11**:86–92. https://doi.org/10.1214/aoms/1177731944.
31. Nemenyi PB. Distribution-free multiple comparisons. *Princeton University*, 1963.
32. Sun S, Liu Y, Li Q. *et al.* Short-term multi-step wind power forecasting based on spatio-temporal correlations and transformer neural networks. *Energ Conver Manage* 2023;**283**:116916. https://doi.org/10.1016/j.enconman.2023.116916.
33. Wang L, Li R, Zhang C. *et al.* UNetFormer: a UNet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery. *ISPRS J Photogramm Remote Sens* 2022;**190**:196–214. https://doi.org/10.1016/j.isprsjprs.2022.06.008.
34. Shafiq M, Gu Z. Deep residual learning for image recognition: a survey. *Appl Sci* 2022;**12**:8972. https://doi.org/10.3390/app12188972.
35. Han Z, Li Z, Yamanashi Y. *et al.* Design of max pooling operation circuit for binarized neural networks using single-flux-quantum circuit. *IEEE Trans Appl Supercond* 2023;**33**:1–5. https://doi.org/10.1109/TASC.2023.3241144.
36. Shang J, Sun Y. CHEER: HierarCHical taxonomic classification for viral mEtagEnomic data via deep leaRning. *Methods* 2021;**189**:95–103. https://doi.org/10.1016/j.ymeth.2020.05.018.