Briefings in Bioinformatics, 23(1), 2022, 1–10

https://doi.org/10.1093/bib/bbab385
Problem Solving Protocol

DeepHost: phage host prediction with convolutional neural network

WANG Ruohan, ZHANG Xianglilan, WANG Jianping and LI Shuai Cheng

Corresponding author: WANG Jianping, Department of Computer science, City University of Hong Kong. Tel.: +852-3442-7737; Fax: +852-3442-0503; E-mail: jianwang@cityu.edu.hk. LI Shuai Cheng, Department of Computer science, City University of Hong Kong. Tel.: +852-3442-9412; Fax: +852-3442-0503; E-mail: shuaicli@cityu.edu.hk

Abstract

Next-generation sequencing expands the known phage genomes rapidly. Unlike culture-based methods, the hosts of phages discovered from next-generation sequencing data remain uncharacterized. The high diversity of the phage genomes makes the host assignment task challenging. To solve the issue, we proposed a phage host prediction tool—DeepHost. To encode the phage genomes into matrices, we design a genome encoding method that applied various spaced *k*-mer pairs to tolerate sequence variations, including insertion, deletions, and mutations. DeepHost applies a convolutional neural network to predict host taxonomies. DeepHost achieves the prediction accuracy of 96.05% at the genus level (72 taxonomies) and 90.78% at the species level (118 taxonomies), which outperforms the existing phage host prediction tools by 10.16–30.48% and achieves comparable results to BLAST. For the genomes without hits in BLAST, DeepHost obtains the accuracy of 38.00% at the genus level and 26.47% at the species level, making it suitable for genomes of less homologous sequences with the existing datasets. DeepHost is alignment-free, and it is faster than BLAST, especially for large datasets. DeepHost is available at https://github.com/deepomicslab/DeepHost.

Key words: phage–host relationship; convolutional neural network; genome encoding

Introduction

Viruses are the most abundant and diverse organisms on Earth [1]. More than 90% of the viruses are bacteriophages (or phages) [2], which infect bacteria or archaea. They bind to special receptors on the host cells' surfaces, inject their genome into the host cells, and replicate themselves from there. Most of the phages have specific hosts [3], while some phages can infect bacteria from multiple species, even multiple genera. [4, 5] They participate in host cell lysis and regulate bacterial diversity. The horizontal gene transfer (HGT) between phages and hosts drives their co-evolution [6]. Therefore, phages not only affect individual bacteria but also significantly affect the ecosystems. Despite the essential role of phages in the microbiome, laboratory culture's difficulty has hindered the comprehensive study of phages [7]. The next-generation sequencing provides culture-free methods to discover new phages, expanding the phage diversity largely [8, 9]. However, unlike the culture-based methods which reveal the host directly, the host of phages detected from next-generation sequencing (NGS) data remains unknown. The high diversity of phage sequences makes the host identification task difficult, limiting the study of the interplay between phages and hosts. With more NGS datasets available, it is urgent to design methods for phage host identification.

Recently, several *in silico* methods are designed for phage host prediction. These computational approaches can be categorized

WANG Ruohan is a Ph.D. candidate in the Department of Computer Science at City University of Hong Kong. She studies algorithms and phage analysis.
ZHANG Xianglilan is an associate professor in State Key Laboratory of Pathogen and Biosecurity, Beijing Institute of Microbiology and Epidemiology. Her research areas include machine learning and data mining in high-throughput sequencing data analysis.
WANG Jianping is a professor in the Department of Computer Science at City University of Hong Kong. Her research areas include networking, cloud computing and autonomous driving.
LI Shuai Cheng is an associate professor in the Department of Computer Science at City University of Hong Kong. His research areas include algorithms, machine learning and omics data analysis.
Submitted: 8 June 2021; **Received (in revised form):** 10 August 2021

into abundance-based methods [10], CRISPRs-based methods [11, 12] and sequence-based methods [13–15]. The abundance-based methods study the abundance profiles of phage and bacteria, then choose bacteria with the most similar abundance patterns as their hosts for phages. The CRISPRs-based methods are based on the fact where bacteria will obtain DNA sequences, referred to as spacer, from phages if the phages attack their cells [16]. These methods attempt to identify the host–phage relationship by detecting the spacer sequences. The sequence-based methods use either the sequence homology between phages and their host [14, 17, 18] or the genetic similarity between the phages and the same host [13, 19]. A review [10] has evaluated these methods on their benchmark dataset, and sequence-based methods achieve better performance. Most of the sequence-based methods calculate the k-mer frequency to evaluate the genome dissimilarity. They applied various k-mer statistics to measure the distance between two sequences, such as Euclidean distance, Manhattan distance and Chebyshev distance [14]. There are also different selection criteria, such as selecting N nearest neighbors or adopting the most common host from the N samples [13]. These hyper-parameters affect the prediction performance substantially [20].

Nevertheless, these sequence-based methods only consider exact matches since they are based on k-mer frequencies. However, viral genome sequences are heterogeneous [21], and they can contain sequence variations, including single nucleotide polymorphisms (SNPs) and insertions or deletions (InDel). The overlook of variations may affect the performance of sequence-based methods. Moreover, some exact match sequences fail to incorporate information for host prediction, such as the conserved sequences around integration sites [22, 23]. Focusing on more unique patterns increases the prediction accuracy. Therefore, we need a more flexible method for phage host prediction.

Deep learning, especially convolutional neural network (CNN) [24], can extract the informative features from the input. However, encoding the genomic sequences into matrices for the input of CNN remains a challenge. The most common approach is one-hot encoding, which encodes A (Adenine) as (1000), C (Cytosine) as (0100), G (Guanine) as (0010) and T (Thymine) as (0001). Consequently, a genomic sequence of length L is encoded into a matrix of shape $L \times 4$. Nonetheless, phage genome sizes ranges from 35 to 100 kb [25], prohibiting the one-hot encoding. An alternative approach is encoding the genomic sequences into vectors with k-mer frequencies. However, the method fails to incorporate distant interaction along the sequences, e.g., co-evolution loci. Hence, we design a new encoding method in this work to address the phage host prediction. In our method, we encode genomes into matrices with various spaced distances. The different spaced distances tolerate mutations and InDels. Then the matrices are combined into a three-dimensional (3D) matrix to feed into CNN.

We implement a phage host prediction tool, DeepHost, with the new encoding method. DeepHost encodes phage genomes into 3D matrices and applies CNN to predict their host taxonomies. DeepHost has the following advantages: (1) DeepHost is alignment-free; that is, it does not need to compare the sequence or k-mers with that of the dataset, reducing time and space required when dealing with large datasets. (2) DeepHost applies spaced k-mers, which tolerates SNPs and InDels. (3) DeepHost utilizes a learning-based model, which gives the classification results directly and relieves us from choosing measures and selection criteria.

Methods

Dataset

First, we constructed a phage genome dataset. We downloaded 3670 phage complete genomes from NCBI (https://www.ncbi.nlm.nih.gov/genomes/) and 2480 phage complete genomes from EMBL (https://www.ebi.ac.uk/genomes/) (January 2021), and identified the host taxonomy for the phages with information from GeneBank files. We also included 3755 phage genomes from PhagesDB [26] and 1551 phage genomes from a phage evolution database [27] (Supplementary Table S1). After removing the redundancy and phages with unidentified hosts, there were 8756 phage genomes left. Next, we classified the phages according to host genus and host species, respectively. For the model's reliability, we only kept the taxonomies with more than five phage genomes. For species taxonomy, the phages with hosts annotated with 'sp.' were removed. Finally, we obtained 8595 phages with 72 host taxonomies at genus level and 7483 phages with 118 host taxonomies at species level. The host distribution is shown in Figure 1. The dataset is available in our repository.

3D matrix construction

For illustration, we introduce the construction process of the matrices with the spaced distance of 0 and 1. Denote a DNA sequence of length l as $N_1N_2N_3N_4...N_l$, where $N_i \in \{A, C, G, T\}, 1 \leq i \leq l$. Taking 2-mers as an example, the spaced distance between N_1N_2 and N_3N_4 is zero, while the spaced distance between N_1N_2 and N_4N_5 is one. Then we can construct two matrices to record the occurrence number of each pair of 2-mers with fixed spaced distances 0 and 1, respectively. The rows and columns of the matrix are all possible 2-mers. For a predefined spaced distance, the matrix's shape should be $4^2 \times 4^2$. However, we notice that the matrix would be extremely sparse with large k values. Therefore, we use two 4×4 matrices for a given spaced distance instead. In the first matrix, A and T are encoded as 0, while C and G are encoded as 1. In the second matrix, A and C are encoded as 0, while G and T are encoded as 1. In this case, we encode A as $[0, 0]$, C as $[1, 0]$, G as $[1, 1]$ and T as $[0, 1]$. We take the sequence in Figure 2 for illustration. For a given spaced distance 0 or 1, we collect all 2-mers pairs and encode them into binary codes. 'GC' and 'CA' are encoded as '11' and '10' in the first matrix, as well as '10' and '00' in the second matrix. Therefore, we add one to the element in the 3rd row and the 2nd column of the first matrix, and the 2nd row and the 0th column of the second matrix. Then we divide the numbers in the two matrices by the length of the genome to transfer k-mer numbers into k-mer frequencies. Finally, we combine the two-dimensional (2D) matrices into a 3D matrix. If we use k-mers with the length of K and consider N possible spaced distance, we will obtain a 3D matrix of shape $2^K \times 2^K \times 2N$ for each genome.

CNN and training process

After encoding the phage genomes into 3D matrices, we applied CNN to extract features from the matrices and classify the genomes. Figure 3 demonstrates the architecture of our proposed CNN. The input of our CNN was the constructed 3D matrix. The $2N$ (N is the number of defined spaced distance) layers of the matrix can be regarded as channels. In the two convolutional layers, we used 100 kernels of size 2×2 to scan each channel of the input with the stride of one to extract k-mer features. The rectified linear unit (ReLU) [28] function $f(x) = \max\{0, x\}$ was

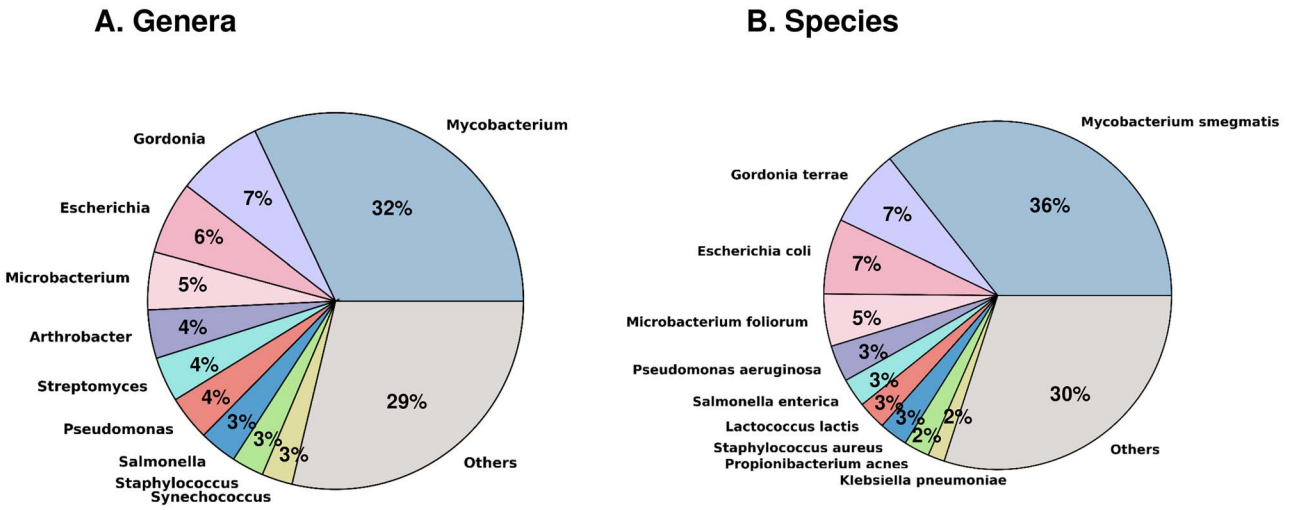

Figure 1. Distributions of host genera and species. The distributions of host genera (A) and host species (B) for our phage dataset.

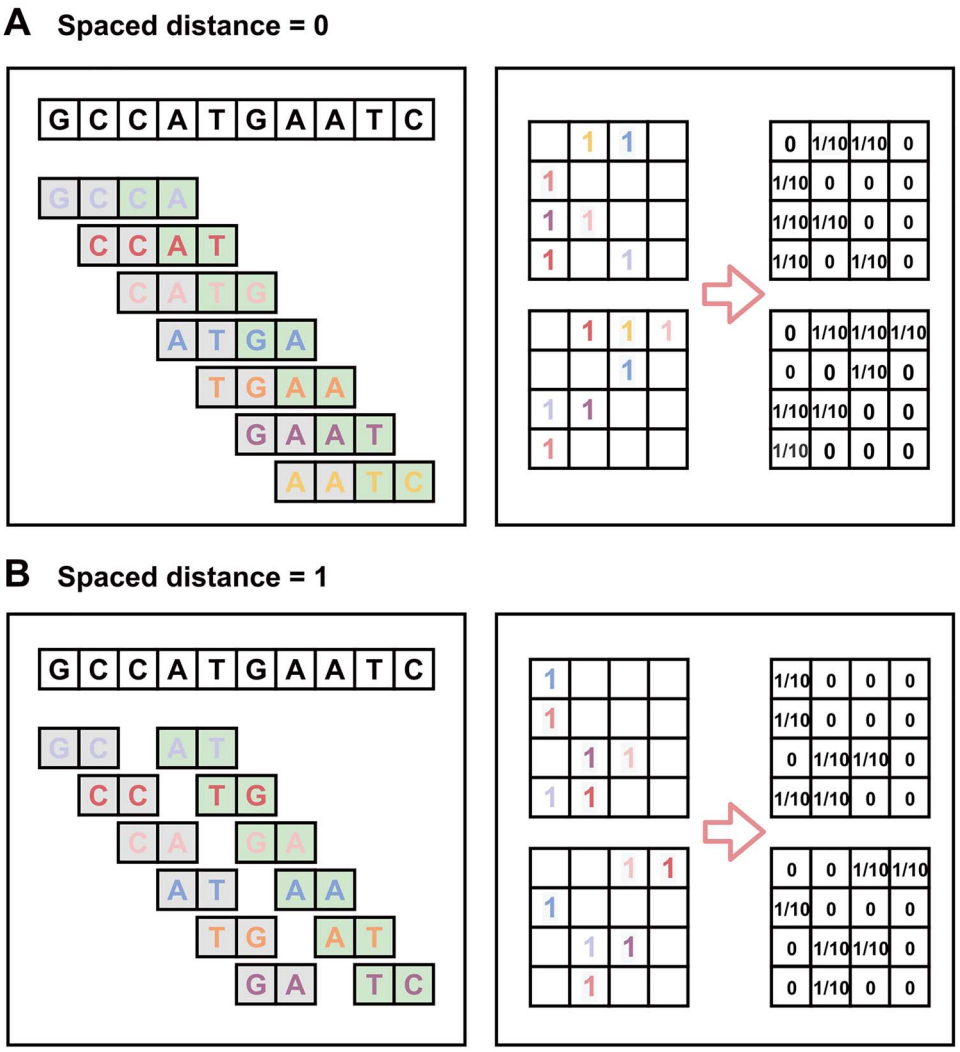

Figure 2. Illustration of the matrix construction process. Given a DNA sequence, all possible 2-mer pairs are collected with spaced distance of 0 (upper) and 1 (lower). For each distance, we construct two matrices.

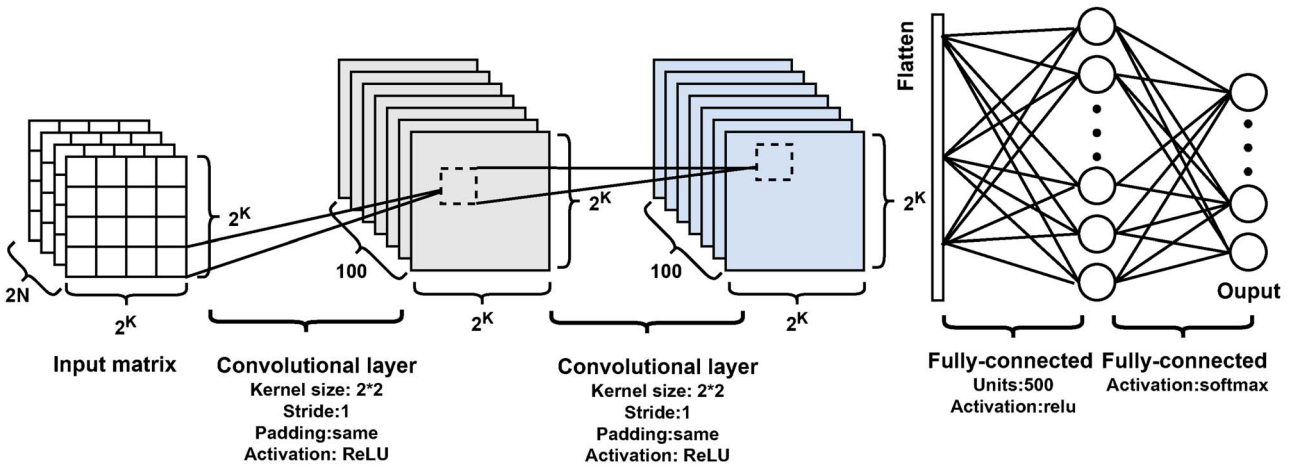

Figure 3. The architecture of our proposed CNN. The input of our CNN is the 3D matrix of shape $2^K * 2^K * 2N$. The first two layers are two 2-D convolutional layers, which consist of 100 kernels with the size of 2 *2. Two fully-connected layers follow, and the softmax activation function is applied for the final result.

applied in the layers. After flattening the output features of the convolutional layers, we applied a fully connected layer with 500 neurons and the ReLU activation function to improve the nonlinear expression ability. Then the output of the fully connected layer was fed into the final prediction layer. The prediction layer applied Softmax activation function [29] to predict the score of each host taxonomy. The taxonomy with the highest score was chosen as the predicted result.

We randomly split our phage dataset into training (80%), validation (10%) and test sets (10%), respectively. In the training process, we applied cross-entropy [30] as the loss function and optimized the loss function with Adam algorithm [31] with learning rate of 10^{-4}. The training process worked through the genome sequences and their reverse complementary sequences in the training set 20 times. More settings on the hyper-parameters can be found in our repository. We trained two CNNs for host genus prediction and host species prediction, respectively. DeepHost also provides an option for users to train the models according to the customized datasets.

Prediction with bacterial genomes

DeepHost outputs the host taxonomy with the highest predicted score. For the phages mined from metagenomics, DeepHost provides a pipeline to combine bacterial genomes for more accurate prediction. First, Kraken2 [32] is applied to bacterial genomes for taxonomic classification and reports all bacterial taxonomies in the metagenome. Then DeepHost calculates the predicted scores for these taxonomies and outputs the one with the highest score. Since the taxonomic classification for bacterial genomes restricts the prediction range, DeepHost is likely to achieve better performance with the information from the metagenomic data.

Performance evaluation metrics

Host genus or species prediction can be regarded as a multiple-classes prediction problem. To evaluate the models based on a balance between recall and precision, we calculated accuracy and F1-score for performance evaluation. For binary labels classification, F1-score is defined as follows:

$$\text{F1-score} = \left(\frac{Recall^{-1} + Precision^{-1}}{2} \right)^{-1} \quad (1)$$

F1-score is inapplicable for multiple-classes problems, so we used Macro F1-score and Weighted F1-score as metrics instead. Macro F1-score gives the mean of F1-score for every class which assumes the same importance for each class. Weighted F1-score gives the weighted mean of F1-score for each class, with large classes given higher weights.

$$\text{Macro F1-score} = \frac{1}{N} \sum_{i=1}^{N} \text{F1-score}_i \quad (2)$$

$$\text{Weighted F1-score} = \frac{n_i}{n_{all}} \sum_{i=1}^{N} \text{F1-score}_i \quad (3)$$

where N is the number of classes, n_i is the number of samples in class i and n_{all} is the number of all samples.

Results

Overview of DeepHost

Figure 4 provides an overview for DeepHost. We collected 11 456 phage genomes from public datasets and extracted the hosts' information. The phage genomes and their host information formed our phage dataset. After removing the redundancy and the phages with unknown hosts, we encoded the remaining 8756 phage genomes into matrices with our genome encoding method. Then, we labeled each genome with host taxonomies at genus level and species level, respectively. Next, we randomly chose 90% of the samples for training and validation. We trained two CNNs for genus classification and species classification. The structures of the two CNNs were the same except for the output layer. We used the remaining 10% of the data to evaluate the prediction performance of DeepHost.

K-mer length and spaced distance selection

First, we tried out different k-mer lengths (k) and numbers of spaced distances (l), then chose the set that yielded the best classification accuracy. For k, we tried five different values 2, 3, 4, 5 and 6. For l, we tried four different sets: 1 ($\{0\}$), 2 ($\{0,1\}$), 3 ($\{0,1,2\}$), 4 ($\{0,1,2,3\}$). We applied DeepHost with different combinations of k and l to the phage host prediction

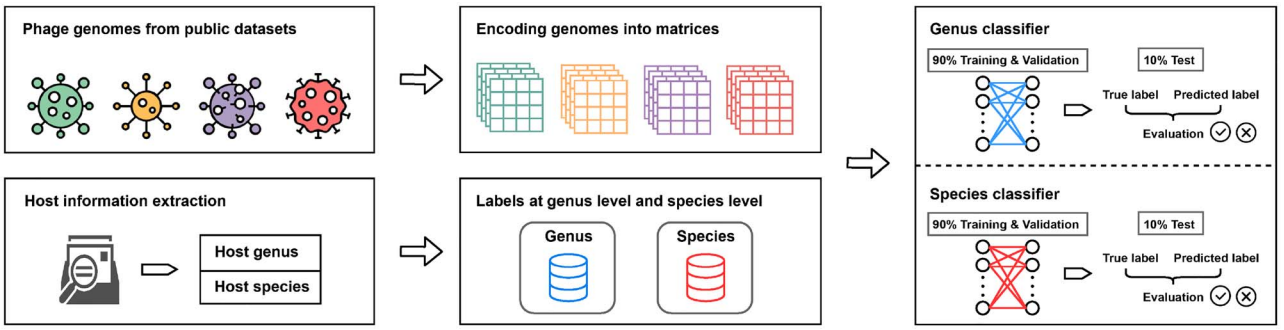

Figure 4. Overview of our phage host prediction method. First, we downloaded phage genomes from public datasets, and extracted host information. Then we encoded genomes into matrices with our encoding method and labeled every genome with host genus and host species, respectively. Finally, we trained two CNNs with 90% of the data to predict phage hosts at genus and species level and used 10% of the data to test the performance of our CNNs.

Table 1. The host genus (upper) and species (lower) prediction accuracy of DeepHost with different k-mer length (k) and spaced distance (d) combinations. The best accuracy values are shown in bold.

Accuracy	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$
$l = 1$ ({0})	88.02%	93.02%	94.30%	94.41%	94.88%
$l = 2$ ({0, 1})	88.72%	93.60%	94.88%	94.88%	95.11%
$l = 3$ ({0, 1, 2})	89.07%	92.79%	94.65%	**96.05%**	**95.81%**
$l = 4$ ({0, 1, 2, 3})	89.77%	93.25%	94.88%	**95.93%**	**95.93%**
Accuracy	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$
$l = 1$ ({0})	79.55%	81.52%	87.03%	88.90%	88.36%
$l = 2$ ({0, 1})	80.35%	83.69%	87.56%	89.17%	89.57%
$l = 3$ ({0, 1, 2})	82.22%	85.16%	88.50%	**90.78%**	**91.04%**
$l = 4$ ({0, 1, 2, 3})	81.01%	85.70%	88.23%	**90.91%**	**90.10%**

problem. The results are shown in Table 1. DeepHost achieved better performance when k-mer length is longer and more spaced distances are considered. However, we noticed that DeepHost with $k = 5$ and $l = 3$ has already achieved the highest accuracy. Longer k and more l failed to improve the performance but increased the memory and runtime. Therefore, we applied DeepHost with $k = 5$ and $l = 3$ in the subsequent experiments.

Comparison between CNN and conventional machine learning models

We compared the CNN model of DeepHost with four conventional machine learning classifiers, including SVM [33], logistic regression [34], decision tree [35] and random forest [36]. We applied our matrix encoding method to generate the input matrices for all the classifiers. Figure 5 shows the comparison of prediction performance. First, we calculated the accuracy, Macro F1-score, and weighted F1-score for the five classifiers (Figure 5(a)). Our CNN achieved better performance than other classifiers by a wide margin. We noticed that every classifier had a lower macro F1-score than weighted F1-score, especially for SVM, logistic regression and decision tree. The reason is that the models have better performance on common classes than rare classes. For example, all of the five classifiers achieved a high F1-score (> 0.9) for the most common species *Mycobacterium smegmatis*. However, for rare species such as *Streptomyces venezuelae*, only CNN obtained F1-scores higher than 0.9 (Supplementary Table S2 and S3). Next, we showed the weighted average receiver operating characteristics (ROC) curve for the five classifiers (Figure 5(b)). CNN had the largest area under the receiver operating characteristics curve (AUC) value. Finally, we calculated F1-scores of all the classes for the five classifiers (Figure 5(c)). Typically, CNN obtained the best F1-score for most classes. We noted that it was hard for models to predict the samples from rare classes, and CNN achieved the most solid performance. We also tried other CNN models and CNNs with more layers, and it turned out that the two-layer CNN had the best performance (Supplementary Table S4 and S5). There is a correlation of GC content and genome size between phage genomes and their host genomes [37]. We incorporated these characteristics in our model, but the accuracy was not improved, indicating the encoding method of DeepHost has a good performance on capturing these characteristics from genomes (Supplementary Figure S1).

Host prediction performance evaluation

To evaluate the host prediction performance of DeepHost, we compared the prediction accuracy of DeepHost with k-mer frequency-based method, BLAST-based method, and two state-of-the-art phage host prediction tools, HostPhinder [13], and VirHostMatcher [14]. HostPhinder calculates the similarity between query phages and the phages in their database and reports the host of the most similar phage. VirHostMatcher calculates the genetic similarity between the query phages and the bacteria, then chooses the bacteria with the highest similarity score as the prediction. For HostPhinder and VirHostMatcher, we applied the hyper-parameters recommended in the papers. For k-mer frequency-based method, we used the k-mer frequency vectors as features and applied CNN to make predictions. For the BLAST-based method, we used the training and validation set to create the local nucleotide database and applied BLAST to aligning the genomes in the test set with the database. The host

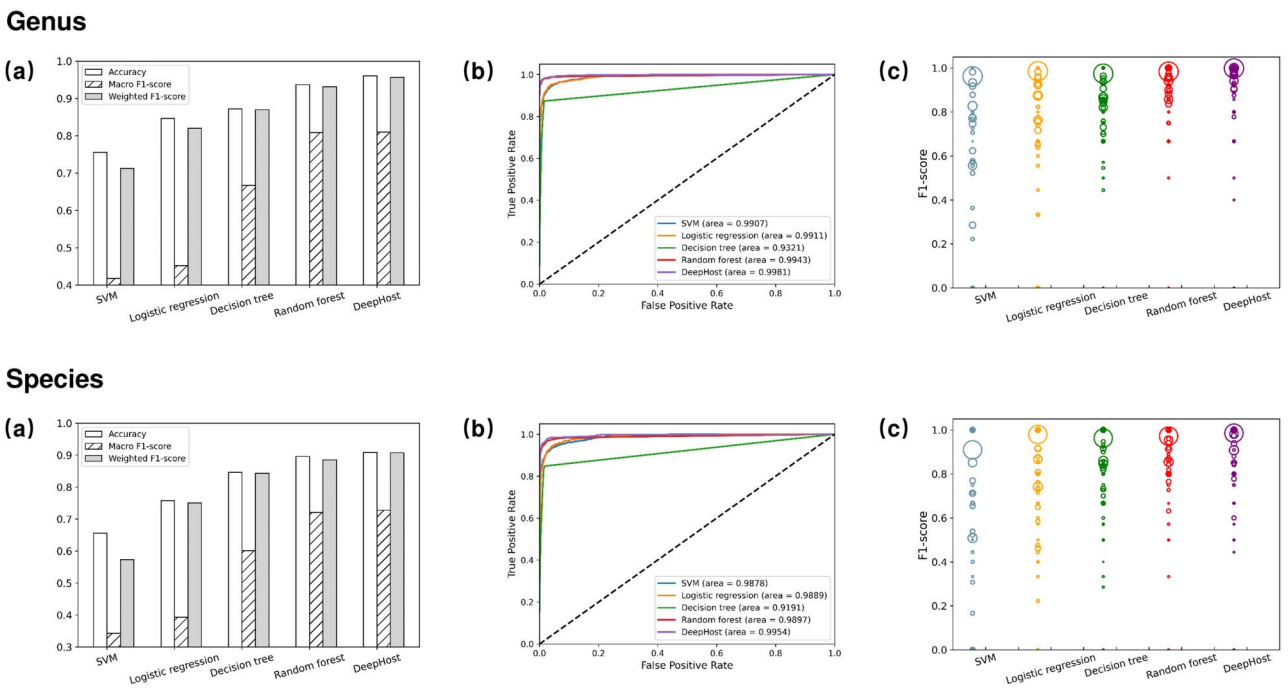

Figure 5. Comparison of DeepHost with conventional machine learning classifiers for genus and species prediction performance. (a) The comparison of accuracy, macro F1-score, and weighted F1-score for the five classifiers. (b) The weighted average ROC curve of the five classifiers. The AUC values are shown in the legend. (c) The F1-score of each class for the five classifiers. Each circle represents a class, with the circle size suggesting the class size.

taxonomy of the hit with the lowest e-value was returned as the predicted taxonomy.

Compared with HostPhinder and *k*-mer frequency-based method, DeepHost and BLAST achieved better performance at both genus and species levels (Figure 6 A). For VirHostMatcher, we applied the host dataset provided in their repository, including 71 host genomes from 40 genus taxonomies and 43 species taxonomies. We kept the phages in this host range for testing (577 phages for genus prediction and 420 phages for species prediction). Here, DeepHost can utilize the bacterial information for prediction (see Methods). The bacterial information helped DeepHost achieve better performance, and the observed higher accuracy of DeepHost was significant compared to VirHostMatcher (Figure 6 B). In Supplementary Figure S2, we showed that the host restriction method is better than directly including bacterial genomes in the CNN model.

Prediction on the genomes with no-hit in BLAST

In the host prediction performance evaluation experiment, we found that around 1% of the genome sequences in the test set have not hit with BLAST. For the genomes that failed to make predictions with BLAST, assigning a random taxonomy for them cannot achieve a good performance since there are 72 genus taxonomies and 118 species taxonomies in our dataset. Assigning the phages with the most common taxonomy also fails since the phages with no-hit in BLAST often come from rare taxonomy. To evaluate the performance of DeepHost on these genomes, we extracted all phage genomes with not hit in BLAST from the dataset and used DeepHost to predict their host. DeepHost obtained the accuracy of 38.00% for host genus prediction and 26.47% for host species prediction (Table 2). Figure 7 shows the confusion matrix of DeepHost for the samples with no hit in BLAST.

DeepHost prediction decomposition

To explore the patterns recognized by CNN, we applied DeepLIFT [38] to decompose the predictions of DeepHost. DeepLIFT uses a backpropagation algorithm to assign contribution scores for every neuron in each layer of the neural network. For illustration, we randomly chose two genomes from *Mycobacterium* and *Staphylococcus* in the test set, then computed contribution scores for the first layer of the input (Figure 8). A positive score means a positive influence on the right prediction and vice versa. We chose the three largest scores and found their corresponding binarized 10-mers. Next, we calculated the frequencies of the three binarized 10-mers in the genomes of training set from the five most common genera, *Mycobacterium*, *Escherichia*, *Pseudomonas*, *Salmonella* and *Staphylococcus*. For the genome from *Mycobacterium*, the three binarized 10-mers have the highest frequencies in *Mycobacterium*, and the same goes for the genome from *Staphylococcus*. We also found that the binarized 10-mer '0000000000' had the highest frequencies for genomes from *Escherichia*, *Pseudomonas*, *Salmonella* and *Staphylococcus*, but did not have high contribution scores (Figure 9). This was because the 10-mer had high frequencies in more than one genus. Therefore, DeepHost extracted specific k-mers to make predictions.

Computational efficiency evaluation

We evaluated the computational efficiency of DeepHost with the host genus prediction problem as the benchmark task. First, we checked the influence of parameters on runtime. The runtime of DeepHost stayed stable with the different lengths of *k*-mer and increased linearly with the number of spaced distances (Figure 10 A and B) increased. Since the best classification performance was achieved with $k = 5$ and $l = 3$, we used this parameter setting to compare BLAST and DeepHost's computational efficiency. To compare the runtime on large datasets, we expanded

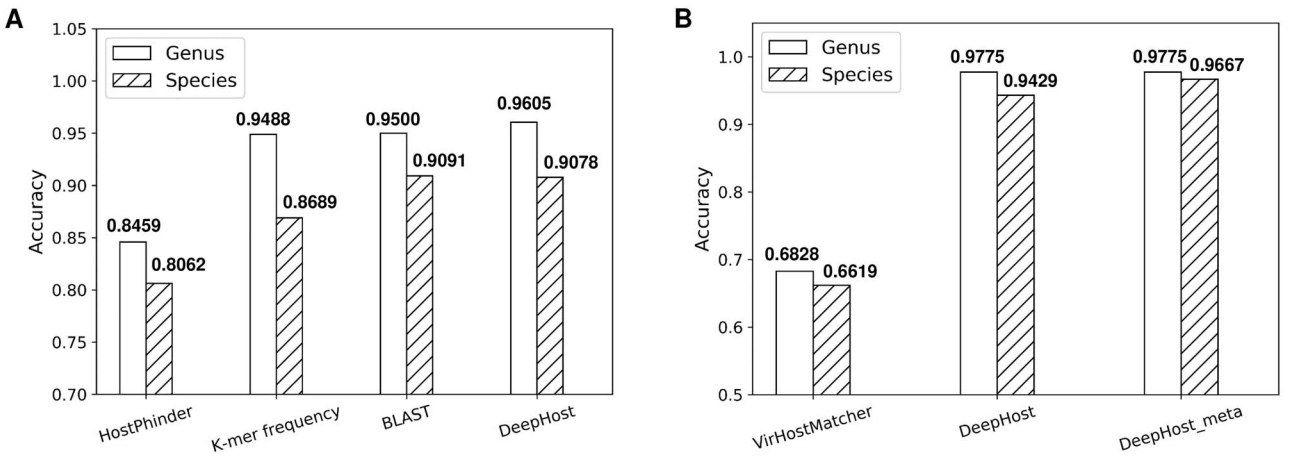

Figure 6. Comparison of phage host prediction accuracy. A. For HostPhinder, k-mer frequency-based method, BLAST-based method, and DeepHost, we compare their host prediction accuracy at genus level and species level. B. For VirHostMatcher, DeepHost, and DeepHost with bacterial genomes from metagenomic analysis, we compare their host prediction accuracy at genus level and species level.

Table 2. Prediction accuracy for the samples with no hit in BLAST.

Methods	Genus	Species
Assigned with the most common class	0%	0%
Assigned with a random class	1.39%	0.85%
DeepHost	**38.00%**	**26.47%**

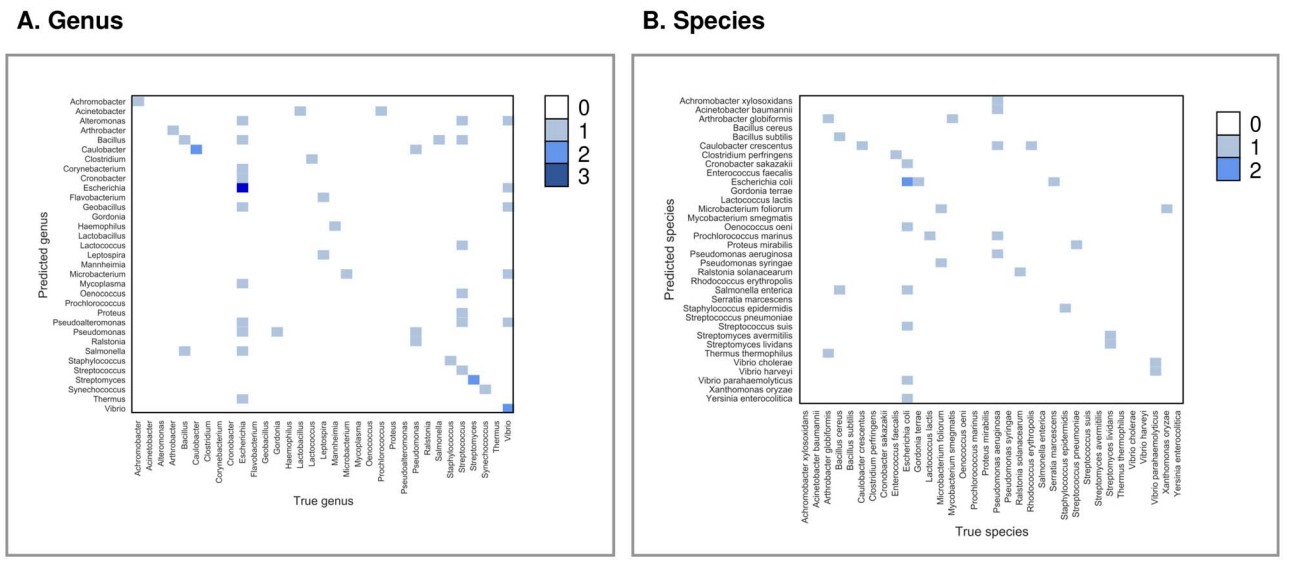

Figure 7. The confusion matrix of DeepHost for the samples with no hit in BLAST. For the phage genomes with no hit when BLAST against all other phage genomes, we construct the confusion matrix to show the genus (A) and species (B) prediction performance of DeepHost. The rows represent the predicted taxonomies, while the columns represent the actual taxonomies. The colors represent the numbers of the corresponding cases.

the training sets one time, two times, four times, eight times and 16 times, and used the training sets to create the BLAST database and train DeepHost. Then we compared the runtime for the test process (Figure 10 C). As we expected, the runtime of DeepHost did not increase with the training data size, while the runtime of BLAST increased linearly with database size. DeepHost achieved better computational efficiency, especially for large datasets.

Discussion

We have proposed DeepHost, a new phage host prediction tool with CNN. Since phage genomes have varying lengths and contain frequent SNPs and InDels, it is challenging to encode the genomes into matrices. To solve this gap, we have designed a genome encoding method to encode genomes of various lengths

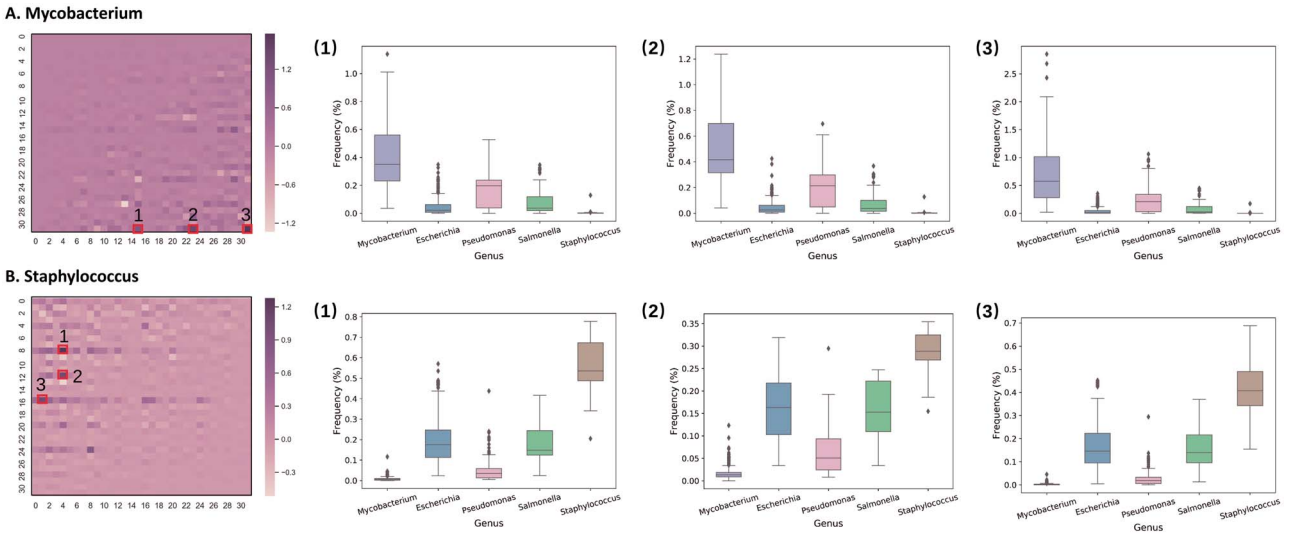

Figure 8. K-mer analysis for CNN prediction decomposition. For a genome from *Mycobacterium* (A) and a genome from *Staphylococcus* (B), the heatmaps show the scores on the input matrix, and the locations of the three largest scores are framed. The three box plots show the frequencies of the corresponding binarized k-mers in the five most common genera of the training set.

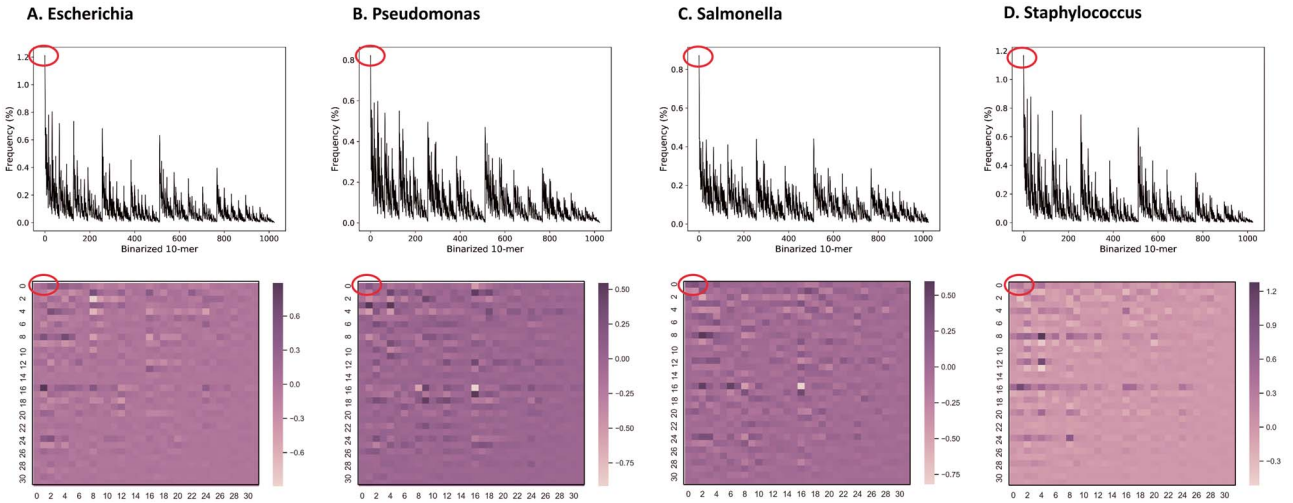

Figure 9. The frequencies of binarized k-mers and the influence scores of binarized k-mers on the prediction. For four genomes from A (*Escherichia*), B (*Pseudomonas*), C (*Salmonella*), and D (*Staphylococcus*), the binarized k-mer frequencies are plotted (upper), and the binarized k-mers with the highest frequency (0000000000) are circled. The heatmaps (lower) show the influence scores of binarized k-mers given by DeepLIFT. The scores of 0000000000 are circled.

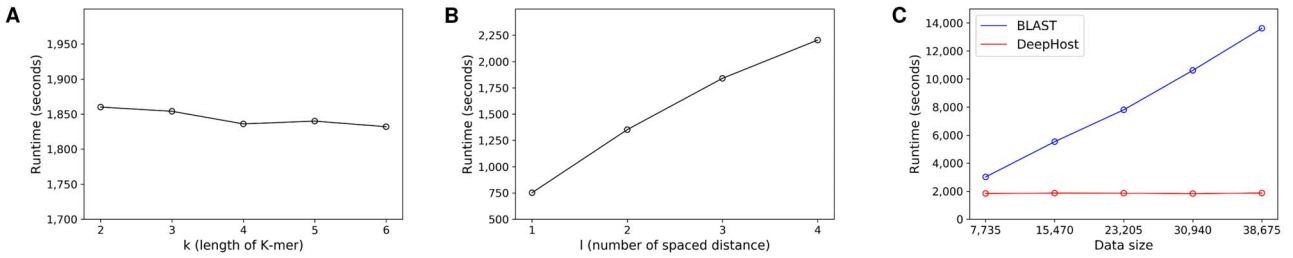

Figure 10. The runtime of DeepHost for benchmark task. A. The runtime of DeepHost with different k. B. The runtime of DeepHost with different l. C. The runtime of BLAST and DeepHost for different data sizes.

into 3D matrices of the same shape. A 3D matrix consist of 2D matrices with different spaced distances. For the 2D matrix with the spaced distance of 0, each value in the matrix represents the frequency of the corresponding binarized $(2 \times k)$-mers, which provides information for CNN to extract genome features. The 2D matrices with other spaced distances are designed to handle SNPs and InDels. For example, when an insertion occurs between two k-mers M_1 and M_2, the spaced distance between M_1 and M_2 will change from zero to one. If we only use a matrix with the spaced distance of zero, the information of pattern M_1M_2

would lose, but if we use multiple matrices, the information will be catered on the matrix with the spaced distance of one. Our experiment proves that the accuracy is higher for the matrices with more layers. In this paper, we have shown that our encoding method achieves better performance than k-mer frequency-based method. The major difference between the two methods is that our encoding method can tolerate SNPs and Indels. BLAST also allows mismatches in the alignment, so it also achieved good performance.

DeepHost applies CNN as the classifier. First, CNN can deal with the input of multiple channels. Like RGB channels that comprise images, our encoding method generates multiple channels for various spaced distances to represent genomes. CNN uses convolutional kernels to scan each channel individually and adds them up, thus combining the information from all channels to make the final predictions. Additionally, CNN is suitable for extracting patterns from the input matrices. We have shown that CNN makes predictions with the information from the k-mers that have high frequencies in only one genus or species. CNN recognizes these k-mers and used the patterns for predictions. VirHostMatcher and HostPhinder also apply k-mer frequency vectors to represent genomes, but their prediction accuracy is much worse than k-mer frequency-based method and DeepHost. VirHostMatcher and HostPhinder give equal importance to all the k-mers, and the k-mers without significant signals may lead to incorrect predictions.

The phage sequences mined from metagenomic data have less similarity with the sequences in existing datasets [10]. BLAST and other alignment search tools may fail to find homologous sequences in this case. Most sequence-based prediction tools use exact match k-mers, requiring high sequence similarity between query phage and phages in the dataset. Our experiments show that DeepHost achieves satisfying performance on the sequences that cannot be covered by BLAST. The reason is that the encoding method of DeepHost allows variants, and CNN is good at extracting important features.

Recently, a complete temperate phage database mined from bacterial NGS data has revealed that a very small part of phages possess multiple hosts in species and genus level [5]. After removing the phages with uncertain host and hosts out of range, we harvested 16 phages genomes and their host information (Supplementary Table S6). All of the 16 phages have cross-species hosts, and 15 of them have cross-genera hosts. We apply DeepHost to the genomes and analyze the probability of each host taxonomy. The results are given in Supplementary Figure S3. The predicted probabilities of all the hosts are ranked top among the 118 species. In our future work, we plan to collect more phage sequences with multiple hosts and train an individual model for multiple host analysis.

In our experiments, BLAST has a comparable performance with DeepHost. However, DeepHost achieves a better computational efficiency, especially for large datasets. It is crucial since the number of metagenomic datasets is increasing rapidly nowadays. For DeepHost, we use the dataset to train a CNN model, and we only need the model to make predictions for the query phages. Consequently, the dataset size does not affect the runtime for the prediction process but only changes our CNN model's weight values. The high computational efficiency makes DeepHost a reliable tool to characterize the host for phages and expand our knowledge of phage–host relationships from metagenomic datasets.

In this paper, we design the genome encoding method for phage host taxonomy prediction. Genome sequences often have various lengths and contain SNPs and InDels. One-hot encoding and k-mer frequency vectors are not suitable for them. Our encoding method has the ability to deal with this kind of sequence. Our encoding method can also be applied to other genome analysis tasks, such as functional classification and homology detection. Moreover, The encoding method is not limited to genome sequences. It can be adopted to other DNA sequences, mRNA sequences and protein sequences for various classification or regression problems.

Key Points

- We propose a phage host prediction tool DeepHost which applied CNN. To encode phage genomes with various sequence lengths and frequent variations into matrices, we design a genome encoding method that tolerates SNPs and InDels.
- We show that the CNN model of DeepHost extracted unique k-mer patterns from the input matrices to make predictions instead of using frequent k-mers.
- DeepHost achieves better prediction accuracy than other host prediction tools, and it also performs well on the sequences which have less homology in the datasets. DeepHost achieves high computational efficiency compared with alignment-based methods.
- The genome encoding method of DeepHost can be adopted to other sequence analysis problems, not limited to genome sequences.

Funding

This work was supported by Strategy Research Grant (7005215).

Acknowledgments

We would like to thank the Department of Computer Science, City University of Hong Kong for providing NVIDIA Tesla K80 graphics card for us to do the experiments.

References

1. Güemes AGC, Youle M, Cantú VA, *et al*. Viruses as winners in the game of life. *Annual Review of Virology* 2016;**3**:197–214.
2. Yutin N, Makarova KS, Gussow AB, *et al*. Discovery of an expansive bacteriophage family that includes the most abundant viruses from the human gut. *Nat Microbiol* 2018;**3**(1):38–46.
3. Holmfeldt K, Middelboe M, Nybroe O, *et al*. Large variabilities in host strain susceptibility and phage host range govern interactions between lytic marine phages and their flavobacterium hosts. *Appl Environ Microbiol* 2007;**73**(21):6730–9.
4. Ross A, Ward S, Hyman P. More is better: selecting for broad host range bacteriophages. *Front Microbiol* 2016;**7**:1352.
5. Zhang X, Wang R, Xie X, *et al*. Mining bacterial ngs data vastly expands the complete genomes of temperate phages-bioRxiv. 2021; 2021.1507.2021.452192.
6. Maciejewska B, Olszak T, Drulis-Kawa Z. Applications of bacteriophages versus phage enzymes to combat and cure bacterial infections: an ambitious and also a realistic application? *Appl Microbiol Biotechnol* 2018;**102**(6):2563–81.

7. Kortright KE, Chan BK, Koff JL, *et al*. Phage therapy: a renewed approach to combat antibiotic-resistant bacteria. *Cell Host Microbe* 2019;**25**(2):219–32.
8. Reyes A, Haynes M, Hanson N, *et al*. Viruses in the faecal microbiota of monozygotic twins and their mothers. *Nature* 2010;**466**(7304):334–8.
9. Camarillo-Guerrero LF, Almeida A, Rangel-Pineros G, *et al*. Massive expansion of human gut bacteriophage diversity. *Cell* 2021;**184**(4):1098–109.
10. Edwards RA, McNair K, Faust K, *et al*. Computational approaches to predict bacteriophage–host relationships. *FEMS Microbiol Rev* 2016;**40**(2):258–72.
11. Stern A, Mick E, Tirosh I, *et al*. Crispr targeting reveals a reservoir of common phages associated with the human gut microbiome. *Genome Res* 2012;**22**(10):1985–94.
12. Minot S, Bryson A, Chehoud C, *et al*. Rapid evolution of the human gut virome. *Proc Natl Acad Sci* 2013;**110**(30): 12450–5.
13. Villarroel J, Kleinheinz KA, Jurtz VI, *et al*. Hostphinder: a phage host prediction tool. *Viruses* 2016;**8**(5):116.
14. Ahlgren NA, Ren J, Lu YY, *et al*. Alignment-free oligonucleotide frequency dissimilarity measure improves prediction of hosts from metagenomically-derived viral sequences. *Nucleic Acids Res* 2017;**45**(1):39–53.
15. Galiez C, Siebert M, Enault F, *et al*. Wish: who is the host? predicting prokaryotic hosts from metagenomic phage contigs. *Bioinformatics* 2017;**33**(19):3113–4.
16. Horvath P, Barrangou R. Crispr/cas, the immune system of bacteria and archaea. *Science* 2010;**327**(5962):167–70.
17. Pride DT, Wassenaar TM, Ghose C, *et al*. Evidence of host-virus co-evolution in tetranucleotide usage patterns of bacteriophages and eukaryotic viruses. *BMC Genomics* 2006;**7**(1):1–13.
18. Roux S, Hallam SJ, Woyke T, *et al*. Viral dark matter and virus–host interactions resolved from publicly available microbial genomes. *Elife* 2015;**4**:e08490.
19. Ogilvie LA, Bowler LD, Caplin J, *et al*. Genome signature-based dissection of human gut metagenomes to extract subliminal viral sequences. *Nat Commun* 2013;**4**(1):1–16.
20. Zielezinski A, Girgis HZ, Bernard G, *et al*. Benchmarking of alignment-free sequence comparison methods. *Genome Biol* 2019;**20**(1):1–18.
21. Huang E-S, Huong S-M. Gary E Tegtmeier, and Charles Alford. Cytomegalovirus: genetic variation of viral genomes. *Ann N Y Acad Sci* 1980;**354**:332–46.
22. Gregory MA, Till R, Smith MCM. Integration site for streptomyces phage bt1 and development of site-specific integrating vectors. *J Bacteriol* 2003;**185**(17):5320–3.
23. Groth AC, Calos MP. Phage integrases: biology and applications. *J Mol Biol* 2004;**335**(3):667–78.
24. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 2012;**25**:1097–105.
25. Ha AD, Denver DR. Comparative genomic analysis of 130 bacteriophages infecting bacteria in the genus pseudomonas. *Front Microbiol* 2018;**9**:1456.
26. Russell DA, Hatfull GF. Phagesdb: the actinobacteriophage database. *Bioinformatics* 2017;**33**(5):784–6.
27. Mavrich TN, Hatfull GF. Bacteriophage evolution differs by host, lifestyle and genome. *Nat Microbiol* 2017;**2**(9):1–9.
28. Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks. In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. Fort Lauderdale, FL, USA: JMLR Workshop and Conference Proceedings, 2011, 315–23.
29. Anzai Y. *Pattern recognition and machine learning*. Amsterdam, The Netherlands: Elsevier, 2012.
30. Rubinstein RY, Kroese DP. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. New York, NY, USA: Springer Science & Business Media, 2013.
31. Kingma DP, Ba J. *Adam: A method for stochastic optimization*. San Diego, CA, USA, arXiv preprint arXiv:1412.6980, 2014.
32. Wood DE, Lu J, Langmead B. Improved metagenomic analysis with kraken 2. *Genome Biol* 2019;**20**(1):1–13.
33. Cortes C, Vapnik V. Support-vector networks. *Machine learning* 1995;**20**(3):273–97.
34. Hosmer DW, Jr, Lemeshow S, Sturdivant RX. *Applied logistic regression*, Vol. **398**. Amherst: John Wiley & Sons, 2013.
35. Breiman L. *Classification and regression trees*. Wadsworth: Routledge, 2017.
36. Breiman L. Random forests. *Machine learning* 2001;**45**(1):5–32.
37. Almpanis A, Swain M, Gatherer D, *et al*. Correlation between bacterial g+ c content, genome size and the g+ c content of associated plasmids and bacteriophages. *Microbial genomics* 2018;**4**(4):e000168.
38. Shrikumar A, Greenside P, Kundaje A. Learning important features through propagating activation differences. In: *International Conference on Machine Learning*. Sydney, Australia: PMLR, 2017, 3145–53.